

# Использование и настройка vRealize Automation Cloud Assembly

Декабрь 2022 г.  
vRealize Automation 8.7

Актуальная техническая документация доступна на веб-сайте VMware:

<https://docs.vmware.com/ru/>

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

**VMware Россия**  
Россия, 125284, г. Москва  
ул. Беговая, д.3, стр.1  
Бизнес-центр "NORDSTAR TOWER" 30й этаж  
Телефон: +7 495 212 29 00  
[www.vmware.com/ru](http://www.vmware.com/ru)

© 2022 VMware, Inc. Все права защищены. [Информация об авторских правах и товарных знаках.](#)

# Содержание

## 1 Что такое Cloud Assembly 8

Функции Cloud Assembly 9

## 2 Учебники 12

Развертывание виртуальной машины 14

Настройка и тестирование инфраструктуры и развертываний vSphere 20

Настройка и предоставление производственной рабочей нагрузки 38

Использование тегов для управления ресурсами vSphere 45

Добавление облачного шаблона в каталог Service Broker с помощью настраиваемой формы запроса 55

Внедрение ресурсов vSphere и управление ими 67

Инфраструктура и развертывания с несколькими облачными средами 76

Часть 1. Настройка примера инфраструктуры 77

Часть 2. Создание примера проекта 83

Часть 3. Проектирование и развертывание примера облачного шаблона 84

Настройка VMware Cloud on AWS 102

Настройка базового рабочего процесса VMware Cloud on AWS 102

Настройка изолированной сети в VMware Cloud on AWS 117

Настройка интеграции внешнего IPAM Infoblox 122

Добавление обязательных расширяемых атрибутов в приложении Infoblox перед развертыванием пакета загрузки 124

Загрузка и развертывание пакета внешнего поставщика IPAM 125

Создание запущенной среды для точки интеграции IPAM 127

Добавление точки интеграции внешнего поставщика IPAM Infoblox 128

Настройка сети и профиля сети для использования внешнего поставщика IPAM в существующей сети 131

Определение и развертывание облачного шаблона, в котором используется назначение диапазона внешнего поставщика IPAM 134

Использование свойств, связанных с Infoblox, для интеграций IPAM в облачных шаблонах 137

Управление сбором данных сети с помощью фильтров Infoblox 141

## 3 Настройка Cloud Assembly для организации 143

Что такое роли пользователей vRealize Automation 143

Роли пользователей организаций и служб 145

Настраиваемые роли пользователей 170

Примеры использования. Роли пользователей как средство управления доступом 175

Встроенная роль администратора инфраструктуры 196

Добавление учетных записей облачной службы 198

Учетные данные, необходимые для работы с облачными учетными записями 198

Создание облачной учетной записи Microsoft Azure	216
Создание облачной учетной записи Amazon Web Services	220
Создание облачной учетной записи Google Cloud Platform	222
Создание облачной учетной записи vCenter	224
Создание облачной учетной записи NSX-V	225
Создание облачной учетной записи NSX-T	227
Создание облачной учетной записи VMware Cloud on AWS	231
Создание облачной учетной записи VMware Cloud Foundation	232
Создание облачной учетной записи VMware Cloud Director в vRealize Automation	234
Интеграция с другими приложениями	240
Использование интеграции GitLab и GitHub	240
Настройка интеграции внешнего поставщика IPAM	247
Обновление до новой версии пакета интеграции внешнего поставщика IPAM	248
Настройка интеграции с приложением My VMware в службе Cloud Assembly	250
Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly	250
Работа с Kubernetes в Cloud Assembly	255
Что такое управление конфигурацией в Cloud Assembly	283
Создание интеграции SaltStack Config	298
Создание интеграции Active Directory в Cloud Assembly	303
Настройка интеграции VMware SDDC Manager	306
Интеграция с vRealize Operations Manager	307
Что такое планы внедрения	324
Внедрение выбранных компьютеров в качестве отдельного развертывания	326
Расширенная настройка	329
Настройка прокси-сервера Интернета	329
Что можно делать с сопоставлением NSX-T с несколькими системами vCenter	332
Что произойдет, если удалить связь облачной учетной записи NSX	333
Использование IPAM SDK для создания внешнего пакета интеграции IPAM для определенного поставщика	334
Использование vRealize Automation с решением Azure VMware Solution	335
Использование vRealize Automation с Google Cloud VMware Engine	335
Использование vRealize Automation с решением Oracle Cloud VMware Solution	336
Использование vRealize Automation с VMware Cloud on Dell EMC	337
<b>4 Создание инфраструктуры ресурсов</b>	<b>338</b>
Добавление облачных зон	338
Дополнительные сведения об облачных зонах	339
Как добавлять сопоставления конфигураций ресурсов	342
Дополнительные сведения о сопоставлении конфигураций ресурсов	343
Как добавлять сопоставления образов	343
Дополнительные сведения о сопоставлении образов	344
Как добавлять профили сети	350



Дополнительные сведения о профилях сетей	350
Использование параметров сети	357
Использование параметров группы безопасности	362
Использование параметров подсистемы балансировки нагрузки	363
Настройка поддержки сети по требованию в профиле сети для интеграции внешнего поставщика IPAM	365
Настройка поддержки существующей сети в профиле сети для интеграции внешнего поставщика IPAM	368
Как добавлять профили хранилища	368
Дополнительные сведения о профилях хранилища	368
Использование карт ценообразования	372
Создание карт ценообразования для vSphere и VMC	374
Как использовать теги	379
Создание политики расстановки тегов	381
Использование тегов возможностей в Cloud Assembly	383
Использование тегов ограничений в службе Cloud Assembly	385
Стандартные теги	387
Как Cloud Assembly обрабатывает теги	388
Настройка простой структуры расстановки тегов	388
Как работать с ресурсами	390
Вычислительные ресурсы	390
Сетевые ресурсы	390
Ресурсы безопасности	394
Ресурсы хранилища	396
Дополнительные сведения о ресурсах	396
Настройка ресурсов арендатора от нескольких поставщиков с помощью vRealize Automation	419
Создание виртуальной частной зоны для vRealize Automation	419
Управление конфигурацией виртуальной частной зоны для арендаторов vRealize Automation	423
Создание глобального сопоставления образов и конфигураций ресурсов для арендаторов vRealize Automation	425
Настройка сопоставлений образов и конфигураций ресурсов для конкретного арендатора в vRealize Automation	428
Создание подписок с поддержкой расширяемости для поставщиков или арендаторов	429
Работа с устаревшими виртуальными частными зонами в новых версиях vRealize Automation	431

## 5 Добавление проектов и управление ими 433

Как добавить проект для группы разработчиков	433
Дополнительные сведения о проектах	436
Использование тегов проекта и настраиваемых свойств	436
Использование политик размещения уровня проекта	438
Что такое затраты на проект	443
Какова роль проектов во время развертывания	443

## 6 Проектирование развертываний 445

- Начало работы с проектами 447
- Помощь с завершением создания кода 450
- Привязки и зависимости 452
- Управление версиями шаблонов 454
- Входные данные пользователей в запросах 456
  - Действия vRealize Orchestrator в качестве входных данных 462
- Группы свойств 466
  - Группы входных свойств 467
  - Группы свойств констант 478
  - Дополнительные сведения о группах свойств 481
- Флаги ресурса для запросов 482
- Выражения 485
  - Синтаксис выражений 489
- Секретные свойства 495
- Удаленный доступ 496
- Размещение диска SCSI 500
- Инициализация компьютера 503
  - Спецификации настройки vSphere 504
  - Команды настройки 504
  - Статические IP-адреса vSphere 507
  - Отложенное развертывание 512
  - Гостевая настройка Windows 513
- Кластеры компьютеров и дисков 517
- Настраиваемое именование развернутых ресурсов 520
- Ресурс SaltStack Config 523
- Конфигурации Terraform 529
  - Подготовка среды выполнения Terraform 529
  - Подготовка для конфигураций Terraform 537
  - Проектирование конфигураций Terraform 539
  - Дополнительные сведения о конфигурациях Terraform 544
- Настраиваемые типов ресурсов 547
  - Создание облачного шаблона, обеспечивающего добавление пользователей в Active Directory 551
  - Создание облачного шаблона, который включает в себя SSH 556
- Подготовка к регулярному обслуживанию 560
  - Использование входных данных облачного шаблона для обновлений при регулярном обслуживании 561
  - Создание действия с ресурсом для виртуальной машины vMotion 562
- Другие примеры кода 572
  - Проверяемый облачный шаблон 572
  - Примеры ресурсов vSphere 579

Количество ядер на сокет и количество ЦП	583
Сети, ресурсы безопасности и подсистемы балансировки нагрузки	584
Облачный шаблон с возможностями Puppet и доступом по имени пользователя и паролю	613
Схема свойств ресурса	623
Специальные свойства	623
Другие способы создания шаблонов	623
Увеличение и автоматизация жизненных циклов приложений	624
Подписки на действия с поддержкой расширяемости	624
Подписки на рабочие процессы с поддержкой расширяемости	653
Дополнительные сведения о подписках с поддержкой расширяемости	660
<b>7 Управление развертываниями и ресурсами</b>	<b>674</b>
Управление развертываниями	674
Мониторинг развертываний	678
Что можно сделать в случае неудачного развертывания Cloud Assembly	679
Управление жизненным циклом завершенного развертывания	682
Какие действия можно выполнять в развертываниях	686
Управление ресурсами	700
Работа с отдельными ресурсами	704
Работа с обнаруженными компьютерами	706

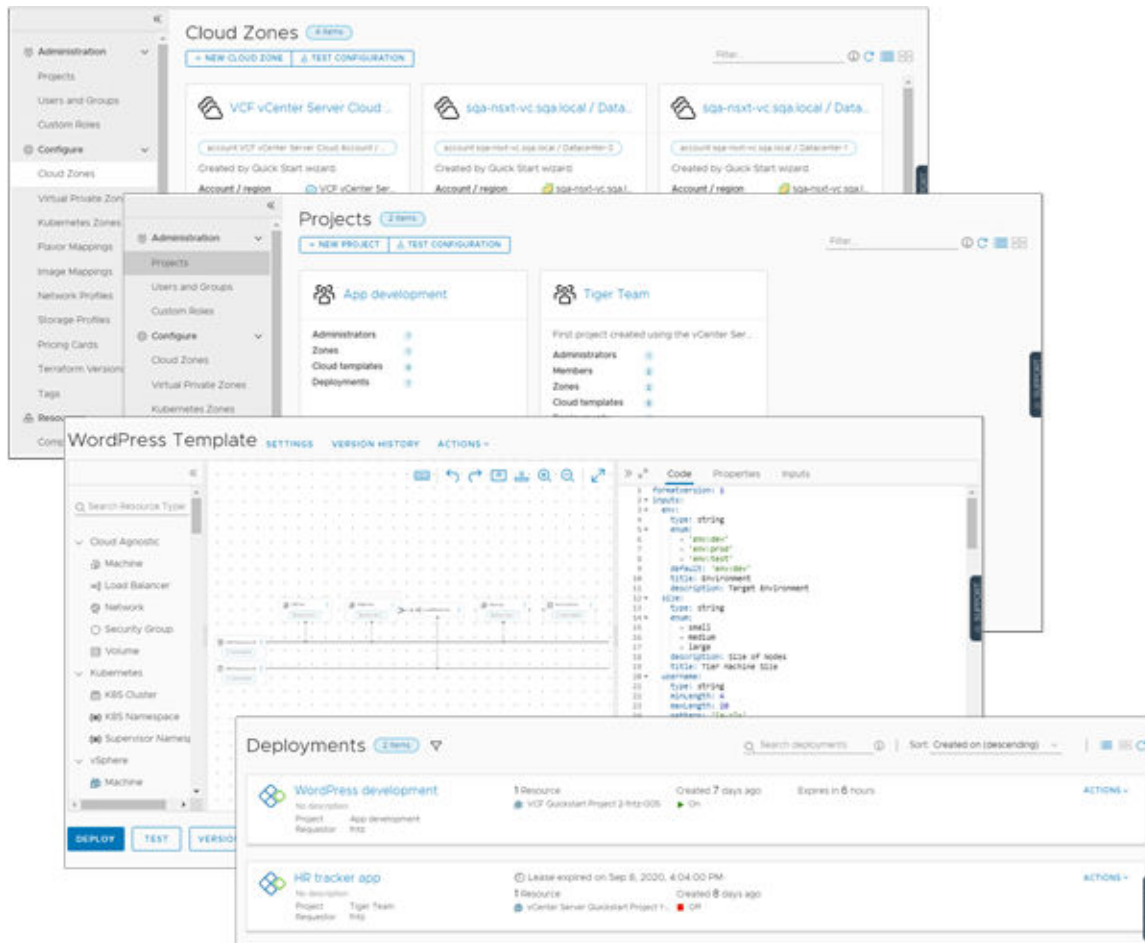
# Что такое Cloud Assembly

# 1

vRealize Automation Cloud Assembly можно использовать для подключения к поставщикам общедоступных и частных облачных сред с целью развертывания компьютеров, приложений и служб, создаваемых для этих сред. Рабочие группы разрабатывают облачные шаблоны как код в среде, которая поддерживает циклический рабочий процесс — от разработки и тестирования до выпуска в производственную среду. На этапе подготовки можно выполнять развертывание в различных облачных средах. Служба — это управляемая платформа VMware, предоставляющая ПО и сети по модели «как услуга».

В обзоре Cloud Assembly рассмотрены следующие основные функции.

- На вкладке «Ресурсы» отображается текущее состояние подготовленных, обнаруженных, внедренных и других ресурсов. Здесь можно получить доступ к сведениям о ресурсах и действиям по регулярному обслуживанию, используемым для управления ресурсами.
- Вкладка «Проектирование» является основным разделом для разработки. С помощью холста и редактора YAML можно разрабатывать и развертывать компьютеры и приложения.
- На вкладке «Инфраструктура» можно добавлять и упорядочивать ресурсы поставщика облачных служб и пользователей. На этой вкладке также представлены сведения о развернутых облачных шаблонах.
- На вкладке «Расширяемость» можно расширять и автоматизировать жизненный цикл приложения. Можно подписаться на события, которые используются для запуска действий с поддержкой расширяемости или рабочих процессов vRealize Orchestrator.
- На вкладке «Оповещения» содержатся уведомления о емкости, производительности и доступности ресурсов инфраструктуры. Чтобы видеть и использовать оповещения, необходимо настроить интеграцию с vRealize Operations Manager.
- На вкладке «Управление арендаторами» отображаются различные арендаторы, настроенные поставщиком услуг. Здесь можно выделять виртуальные частные зоны или отменять их выделение.



В эту главу входят следующие разделы:

- [Функции Cloud Assembly](#)

## Функции Cloud Assembly

Cloud Assembly — это служба разработки и развертывания облачных шаблонов. Администраторы и рабочие группы могут использовать эту службу для развертывания компьютеров, приложений и служб в рамках ресурсов поставщика облачных услуг.

Администратор Cloud Assembly, называемый, как правило, администратором облачных систем, может настраивать инфраструктуру подготовки ресурсов и создавать проекты, объединяющие пользователей и ресурсы.

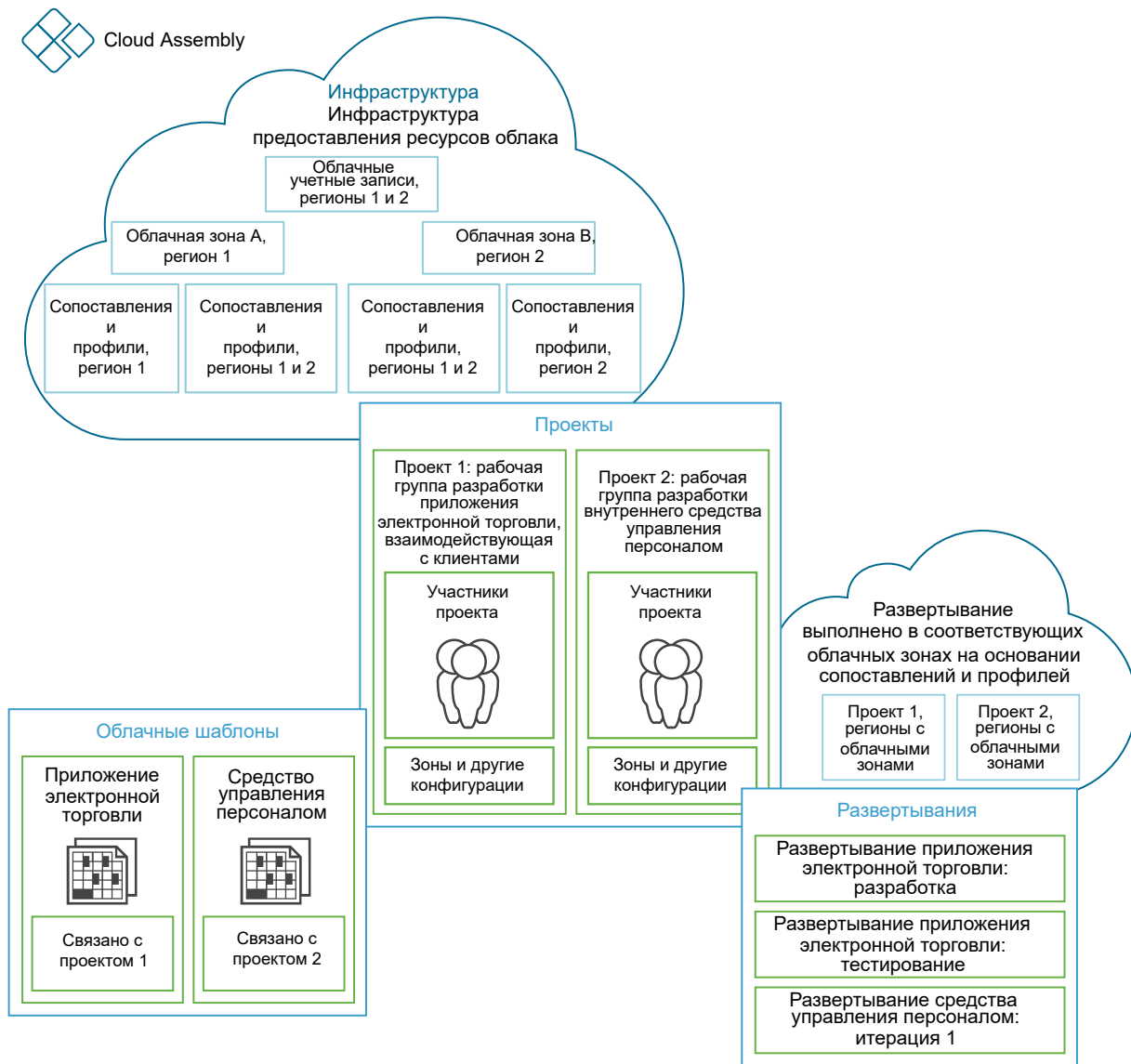
- Добавьте учетные записи поставщиков облачных услуг. См. раздел [Добавление учетных записей облачной службы в Cloud Assembly](#).
- Определите, какие области или хранилища данных будут теми облачными зонами, в которых разработчики смогут выполнять развертывания. См. раздел [Дополнительные сведения об облачных зонах Cloud Assembly](#).
- Создайте политики, определяющие облачные зоны. См. раздел [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).

- Создайте проекты, в которых будут объединены разработчики и облачные зоны. См. раздел [Использование тегов проекта и настраиваемых свойств в Cloud Assembly](#).

Пользователи с ролью разработчика облачных шаблонов могут быть участниками одного проекта или нескольких. Они создают и развертывают шаблоны в облачных зонах, связанных с соответствующими проектами.

- Разработайте облачные шаблоны для проектов с помощью холстов проекта. См. раздел [Начало работы с проектами Cloud Assembly](#).
- Выполните развертывание облачных шаблонов в облачных зонах проектов в соответствии с политиками и ограничениями.
- Воспользуйтесь возможностями управления развертываниями, в том числе удаления неиспользуемых приложений. См. раздел [Управление развертываниями Cloud Assembly](#).

Вас приветствует Cloud Assembly. Пример того, как определить инфраструктуру, а затем создать и развернуть облачный шаблон, см. в разделе [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).



# Учебники по Cloud Assembly

# 2

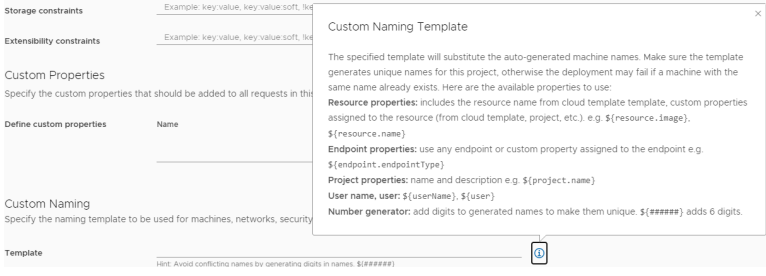
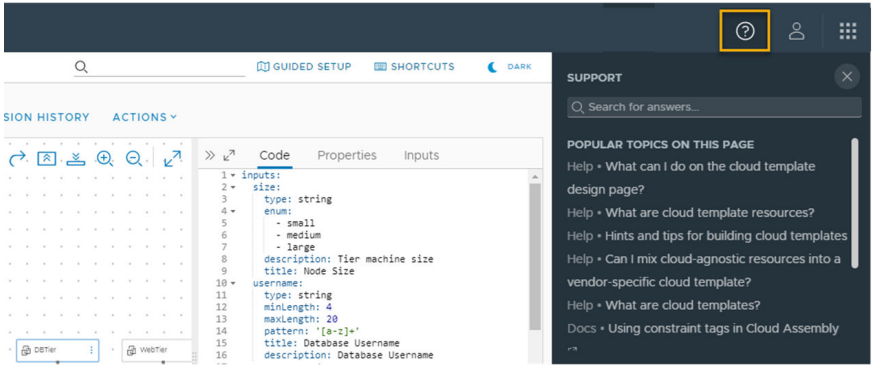
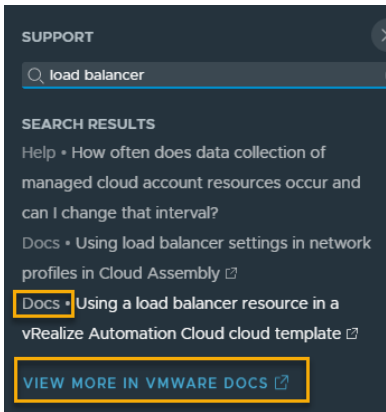
В учебниках демонстрируется выполнение типовых задач, которые позволят вам получить необходимые навыки для работы с Cloud Assembly.

В начале учебника присутствует напоминание о том, что в этом руководстве содержится информация, дополняющая учебники. Предусмотрены ссылки на соответствующие разделы.

## Доступ к поддержке пользователей

Также важно, что в рамках всего приложения предоставляется помощь для пользователей. Справочные инструменты помогают понять функции и получить информацию о том, как заполнять текстовые поля. Внешняя документация содержит более подробные сведения, примеры кода и сценарии использования.



Тип поддержки	Доступ к поддержке	Пример
Справка по полям	Щелкните значок <b>Сведения</b> (i) рядом с полем.	
Контекстная справка на панели поддержки	Щелкните значок «Справка» (?) рядом с именем и организацией.	
Доступ к внешней документации	Щелкните заголовок статьи, помеченной как <b>Документы</b> , или нажмите кнопку <b>Узнать больше в VMware Docs</b> .	

В эту главу входят следующие разделы:

- Учебник. Развертывание виртуальной машины в Cloud Assembly
- Учебник «Настройка и тестирование инфраструктуры и развертываний vSphere в Cloud Assembly»
- Учебник «Настройка Cloud Assembly для предоставления производственной рабочей нагрузки»
- Учебник «Использование тегов в Cloud Assembly для управления ресурсами vSphere»
- Учебник «Добавление облачного шаблона Cloud Assembly в каталог Service Broker с помощью настраиваемой формы запроса»
- Учебник. Внедрение ресурсов vSphere в vRealize Automation и управление ими
- Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»

- Учебник «Настройка VMware Cloud on AWS для vRealize Automation»
- Учебник «Настройка интеграции определенного внешнего поставщика IPAM для vRealize Automation»

## Учебник. Развертывание виртуальной машины в Cloud Assembly

Администратор Cloud Assembly может развернуть простую виртуальную машину, для которой не требуется знать, как создать облачный шаблон. В этом учебнике для новых пользователей Cloud Assembly описывается процесс подготовки и создание виртуальной машины, а также показано, где управлять развернутой машиной.

Этот способ позволяет быстро развернуть машину на основе шаблонов образов, конфигураций размеров, хранилища и сетей, определенных поставщиком облачных служб. Это быстрая проверка облачной учетной записи и проектов.

Виртуальную машину можно создать для любого из следующих поставщиков облачных служб.

- Amazon Web Services
- Google Cloud Platform
- Microsoft Azure
- vCenter Server
- VMware Cloud on AWS

В этом учебнике для примера используется Google Cloud Platform.

### Перед началом работы

- Вам должна быть назначена роль администратора Cloud Assembly. См. раздел [Роли пользователей организаций и служб в vRealize Automation](#). Без этой роли вы даже не увидите возможность создания новой ВМ.

### Шаг 1. Добавление облачной учетной записи

Облачные учетные записи обеспечивают учетные данные, используемые Cloud Assembly для подключения к поставщику облачных служб.

1. Выберите **Инфраструктура > Подключения > Учетные записи облачной службы**.
2. Щелкните **Добавить облачную учетную запись** и выберите тип учетной записи.

Доступ к сведениям о конфигурации можно получить с помощью следующих ссылок.

- [Создание облачной учетной записи Amazon Web Services в vRealize Automation](#)
- [Создание облачной учетной записи Google Cloud Platform в vRealize Automation](#)
- [Создание облачной учетной записи Microsoft Azure в vRealize Automation](#)

- [Создание облачной учетной записи vCenter в vRealize Automation](#)
- [Создание облачной учетной записи VMware Cloud on AWS в vRealize Automation](#)

После добавления облачной учетной записи Cloud Assembly собирает информацию о ресурсах из целевой учетной записи поставщика облачных служб. Позже вы будете использовать ее для развертывания виртуальной машины.

## Шаг 2. Создание проекта

Проект связывает пользователей и облачные зоны облачной учетной записи.

В этом учебнике имя проекта — **Create VM Project**. Этот проект является демонстрационным и включает в себя облачные зоны для всех поддерживаемых платформ.

1. Выберите **Инфраструктура > Администрирование > Проекты**.
2. Щелкните **Создать проект**.
3. Введите имя.

В этом учебнике используется имя **Create VM Project**.

4. Чтобы другие пользователи могли использовать этот проект, перейдите на вкладку **Пользователи** и добавьте пользователей в проект.
5. Перейдите на вкладку **Подготовка** и щелкните **Добавить зону**, чтобы добавить по крайней мере одну облачную зону для облачных учетных записей, в которые выполняется развертывание.

Помните, что этот проект является демонстрационным и включает в себя облачные зоны для всех поддерживаемых облачных платформ.

**Create VM Project** DELETE

Summary Users **Provisioning** Kubernetes Provisioning Integrations

**Zones**  
Specify the zones that can be used when users provision deployments in this project. ⓘ

+ ADD ZONE × REMOVE

<input type="checkbox"/>	Name	Status	Description	Priority	Instances	Memory Limit (MB)	CPU Limit	Storage Limit (GB)	Capability Tags
<input type="checkbox"/>	dsadsa-vsphere / SDDC-Datacenter	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
<input type="checkbox"/>	yingzhi-GCP / us-east1	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
<input type="checkbox"/>	AWS / af-south-1	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
<input type="checkbox"/>	vc65 / Datacenter	--		0	Unlimited	Unlimited	Unlimited	Unlimited	
<input type="checkbox"/>	Azure Test / West US	--		0	Unlimited	Unlimited	Unlimited	Unlimited	

1 - 5 of 5 zones

6. Щелкните **Создать**.

## Шаг 3. Создание и развертывание виртуальной машины

1. Выберите **Ресурсы > Ресурсы > Виртуальные машины**, а затем щелкните **Создать ВМ**.
2. Настройте необходимые параметры на общей странице мастера и нажмите кнопку **Далее**.

В этом учебнике в качестве облачной учетной записи, в которой будет развернута виртуальная машина, используется Google Cloud Platform.

Помните, что эти значения приведены только для примера. Вам необходимо указать значения для вашей среды.

Таблица 2-1. Примеры значений для первой страницы мастера

Настройка	Пример значения
Имя	Google Cloud Create VM
Проект	Create VM Project
Облачная зона	yingzhi-GCP/us-east1

- Выберите образ и конфигурацию, которые будут использованы для создания виртуальной машины.

Доступные значения собираются из целевой облачной зоны. Образ — это операционная система, конфигурация — заданные параметры размера. Для некоторых типов целевых поставщиков необходимо указать ЦП и память. Для этого целевого поставщика необходимо выбрать из заданных параметров.

- Нажмите кнопку **Далее**.

Чтобы только развернуть машину, нажмите кнопку **Создать**. Чтобы следовать учебнику, щелкните **Далее** и добавьте необязательное хранилище и сеть для этой виртуальной машины.

- Чтобы добавить новый диск, щелкните **Добавить жесткий диск**, введите имя и размер.

- Нажмите кнопку **Далее**.
- Чтобы добавить сетевой адаптер, щелкните **Добавить сетевой адаптер**.
- Выберите его в результатах поиска.

- Щелкните **Создать**.

Откроется страница «Развертывания», где можно отслеживать ход развертывания.

## Шаг 4. Управление новой виртуальной машиной как развертыванием

После успешного завершения процесса развертывания можно начать управлять развертыванием.

Дополнительные сведения об управлении развертываниями см. в разделе [Управление развертываниями Cloud Assembly](#).

Список всех возможных действий по регулярному обслуживанию для всех типов ресурсов см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

- Выберите **Ресурсы > Развертывания** и найдите виртуальную машину.

В этом учебнике имя развертывания — Google Cloud Create VM.

- Чтобы выполнить разрешенное действие уровня развертывания из этого представления, щелкните вертикальное многоточие и выберите действие.

Deployments 20 items of 100 🔍

	Name	Address	Owner	Project	Status	Expires on	Price
>	gcp_811d09ff-efe1-4da4-a949-5be98ab62c...		@vmware.com	Create VM Project		Never	
>	Google Cloud Create VM_6f6d0315-ddc8-4...		@vmware.com	Create VM Project		Never	
>			@vmware.com	cmbu-08-project		Never	
>	le-f792-43d5-885d-2b45e...		@vmware.com	Create VM Project		Never	
>	-South		@vmware.com	Sales		Never	
>			@vmware.com	Sales		Never	

Actions menu for the selected item:

- Change Owner
- Change Project
- Delete
- Edit Deployment
- Edit Tags
- Power Off
- Power On

3. Чтобы получить дополнительные сведения о развертывании, включая топологию, щелкните имя развертывания.

Обратите внимание, что у этого развертывания простая топология. Для более сложных развертываний также представлена полная топология, которая может включать в себя компьютеры, подсистемы балансировки нагрузки, сетевые подключения и другие компоненты.

Можно также просмотреть журнал развертывания, который представляет собой журнал всех действий с компонентами развертывания, и выполнить разрешенные действия на уровне компьютера.

Google Cloud Create VM\_6f6d0315-ddc8-4f5d... Create Successful ACTIONS | 🔍

No description

Owner: cnugent@vmware.com  
Requestor:   
Project: Create VM Project

Expires on: Never  
Last updated: Dec 3, 2021, 2:55:10 PM  
Created on: Dec 3, 2021, 2:52:57 PM

HIDE SUMMARY

Topology History

Search resources

Google Cloud ...

Google Cloud Create VM\_6f6d0315-ddc8-4f5d... ACTIONS

General

Resource name: mcm-20211203215331-000020  
Account / Region: ylnazhi-GCP/us-east1  
Status: On  
Address: 34.74.168.22  
Compute host: us-east1-b

Storage

Name	Capacity (GB)	Type	Encrypted
create-vm-new-disk-1-524598563851	4	HDD	true
mcm-20211203215331-000020	10	HDD	true

Actions menu for the selected item:

- Add Disk
- Create Snapshot
- Delete
- Power Off
- Resize
- Resize Boot Disk

## Шаг 5. Управление новой виртуальной машиной как ресурсом

Виртуальной машиной можно управлять не только как развертыванием, но и вместе с другими ресурсами. Ресурсы могут включать в себя развернутые, обнаруженные и внедренные виртуальные машины, тома хранилища, а также ресурсы сети и безопасности.

Обнаруженные ресурсы — это ресурсы, собранные из облачного экземпляра. Обнаруженными ресурсами можно управлять с помощью ограниченного набора действий по регулярному обслуживанию, таких как включение и выключение. Дополнительные сведения о работе с обнаруженными ресурсами см. в разделе [Работа с обнаруженными ресурсами в Cloud Assembly](#).

Внедренные ресурсы — это обнаруженные ресурсы, переведенные под полное управление. Ими можно управлять с помощью более надежных действий по регулярному обслуживанию. Дополнительные сведения о внедрении обнаруженных ресурсов см. в разделе [Что такое планы внедрения в Cloud Assembly](#).

Поскольку вы работаете с этой развернутой машиной, к ней применимы дополнительные действия по регулярному обслуживанию. Доступность действий зависит от состояния машины и имеющихся у вас разрешений на выполнение действий по регулярному обслуживанию.

1. Выберите **Ресурсы > Ресурсы > Виртуальные машины**.
2. Найдите машину.

Name	Deployment	VM State	Account / Region	Address	Project	Origin	Tags
vm-administrator-VLDX...		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
vm-administrator-N6CE...		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...	192.167.211.142		Discovered	-
mcm-20211203215331-0...	Google Cloud Create VM_6f...	On	yingzhi-GCP / us-east1	34.74.168.22	Create VM Proj...	Deployed	-

3. Чтобы выполнить разрешенное действие уровня машины из этого представления, щелкните вертикальное многоточие и выберите действие.

Name	Deployment	VM State	Account / Region	Address	Project	Origin	Tags
vm-administrator-VLDX...		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
vm-administrator-N6CE...		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...	192.167.211.142		Discovered	-
mcm-20211203215331-0...	Google Cloud Create VM_6f...	On	yingzhi-GCP / us-east1	34.74.168.22	Create VM Proj...	Deployed	-
		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-

4. Чтобы просмотреть сведения о ресурсе машины, щелкните имя машины.

Полезные сведения в этом примере включают хранилище, сеть и настраиваемые свойства.

The screenshot displays the 'Virtual Machines' section of the vRealize Automation Cloud Assembly interface. On the left, a list of VMs is shown, including 'vm-administrator-VLDX...', 'vm-administrator-N6CE...', and 'mcm-20211203215331-0...'. The right pane shows the details for the selected VM 'mcm-20211203215331-000020'.

**VM State**: On

**Address**: 34.74.168.22

**Account / region**: yingzhi-GCP / us-east1

**Origin**: Deployed

**Deployment**: Google Cloud Create VM\_6f6d0315-ddc8-4f5d-9e1e-563c149a836d

**Tags**

**Volumes**

Name	Capacity	Type
create-vm-new-disk-1-524598563851	4 GB	HDD
mcm-20211203215331-000020	10 GB	HDD

**Networks**

Name	Address	Assignment Type
default	10.142.0.56	dynamic

**Custom Properties**

Name	Value
resourceId	3b43b1a6-105c-4d68-8562-f84d545d07a0
zone_overlapping_migrated	true
project	d952119a-7354-4dc2-afd5-718755917230
zone	us-east1-b
environmentName	Google Cloud Platform
providerId	1393403671676923083
id	/resources/compute/3b43b1a6-105c-4d68-8562-f84d545d07a0

## Учебник «Настройка и тестирование инфраструктуры и развертываний vSphere в Cloud Assembly»

Если вы впервые обращаетесь к vRealize Automation или вам нужно освежить свои знания, этот учебник поможет выполнить настройку Cloud Assembly. В процессе обучения вы добавите конечные точки облачной учетной записи vSphere, определите инфраструктуру, добавите пользователей в проекты, а затем спроектируете и развернете рабочую нагрузку с помощью VMware Cloud Templates на основе типов ресурсов vSphere.

Несмотря на то что этот учебник охватывает только начальный этап, он позволяет освоить необходимые навыки для дальнейшей автоматизации самообслуживания и итерационной разработки, которые применимы во многих общедоступных и частных облаках. Этот учебник содержит информацию для решений VMware vCenter Server и NSX-T. На основе полученных знаний вы сможете добавлять другие типы облачных учетных записей и создавать более сложные облачные шаблоны.

По мере обучения предлагаются примеры данных. Замените эти примеры значениями, используемыми в вашей среде.

Все действия, описанные в этом учебнике, выполняются в Cloud Assembly.

Этот процесс настройки составляет основу разработки в Cloud Assembly. По мере создания инфраструктуры и приобретения опыта разработки облачных шаблонов вы сможете повторять и расширять этот рабочий процесс.

## Начало работы

- Вам должна быть назначена роль администратора Cloud Assembly. См. раздел [Роли пользователей организаций и служб в vRealize Automation](#).



- Если вы не использовали мастер быстрого запуска VMware vCenter Server или VMware Cloud Foundation в консоли vRealize Automation, можно сделать это сейчас.

Эти рабочие процессы на основе мастера включают в себя большинство настроек, встречающихся в этом учебнике.

Этот учебник дает возможность получить практический опыт, который расширит ваше представление о настройке рабочей инфраструктуры и развертывании рабочих нагрузок.

См. раздел [Настройка Cloud Assembly](#) в руководстве *Начало работы*.

- Если вы еще не выполнили пошаговую настройку, которая описана в Cloud Assembly, это можно сделать сейчас. Пошаговая настройка охватывает большинство процедур, рассмотренных в этом учебнике. Чтобы открыть инструкции пошаговой настройки, щелкните **Пошаговая настройка** в правой части панели вкладок.
- Для выполнения задачи у вас должны быть учетные данные vCenter Server и NSX. Дополнительные сведения о разрешениях учетной записи см. в разделе [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#). Если планируется добавить в проекты дополнительных пользователей, убедитесь, что они зарегистрированы в службе Cloud Assembly.

## Шаг 1. Добавление облачных учетных записей vCenter Server и NSX

Облачные учетные записи предоставляют учетные данные, которые vRealize Automation использует для подключения к vCenter Server и связанному серверу NSX.

### 1. Добавление облачной учетной записи vCenter Server.

Облачная учетная запись vCenter Server предоставляет учетные данные vCenter, которые Cloud Assembly использует для обнаружения ресурсов и развертывания облачных шаблонов.

Дополнительные сведения об облачных учетных записях vCenter Server см. в разделе [Создание облачной учетной записи vCenter в vRealize Automation](#).

- а) Выберите **Инфраструктура > Подключения > Учетные записи облачной службы**.
- б) Щелкните **Добавить облачную учетную запись** и выберите **vCenter**.
- в) Введите значения.

**New Cloud Account**

Name \* vCenter Server Account

Description

vCenter Server Credentials

vCenter IP address / FQDN \* sc2vc05.cmbu.local ⓘ

Username \* mgmt@cmbu.local

Password \* .....

VALIDATE ✓ Credentials validated successfully. ✕

Configuration

Allow provisioning to these datacenters \* ☒ wld01-DC

☒ Create a cloud zone for the selected datacenters

NSX cloud account 🔍 Search for cloud accounts

Capabilities

Capability tags Enter capability tags ⓘ

**ADD** **CANCEL**

Помните, что все значения приведены только для примера. Необходимо указать значения для вашей среды.

Настройка	Пример значения
Имя	Учетная запись vCenter Server
IP-адрес или полное доменное имя vCenter	your-dev-vcenter.company.com
Имя пользователя и пароль	vCenterCredentials@yourCompany.com

- г) Чтобы проверить учетные данные, щелкните **Проверить**.
  - д) Чтобы **разрешить предоставление в этих ЦОД**, выберите один центр обработки данных или несколько.
  - е) Пропустите облачную учетную запись NSX. Она будет настроена позже путем привязки учетной записи vCenter Server к облачной учетной записи NSX.
  - ж) Нажмите кнопку **Добавить**.
2. Добавление связанной облачной учетной записи NSX.

Облачная учетная запись NSX-T предоставляет учетные данные NSX-T, которые Cloud Assembly использует для обнаружения сетевых ресурсов и развертывания сетей с облачными шаблонами.

Дополнительные сведения об облачных учетных записях NSX-T см. в разделе [Создание облачной учетной записи vCenter в vRealize Automation](#).

- а) Выберите **Инфраструктура > Подключения > Учетные записи облачной службы**.
- б) Щелкните **Добавить облачную учетную запись** и выберите NSX-T или NSX-V. В этом учебнике используется **NSX-T**.
- в) Введите значения.

**New Cloud Account**

Name \* NSX-T Account

Description

NSX-T Credentials

NSX-T IP address / FQDN \* sc2vc05-vip-nsx-mgmt.cmbu.local ⓘ

Username \* mgmt@cmbu.local

Password \* .....

NSX mode Policy ⓘ

VALIDATE ✔ Credentials validated successfully. X

Associations

vCenter cloud accounts + ADD X REMOVE

<input type="checkbox"/>	Name	Status	Identifier	Type
<input type="checkbox"/>	vCenter Server Account	✔ OK	sc2vc05.cmbu.local	vCenter

1 - 1 of 1 cloud accounts

Capabilities

Capability tags Enter capability tags ⓘ

ADD CANCEL

Эти значения используются только для примера. Необходимо указать значения для вашей среды.

Настройка	Пример значения
Имя	Учетная запись NSX-T
IP-адрес или полное доменное имя vCenter	your-dev-NSX-vcenter.company.com
Имя пользователя и пароль	NSXCredentials@yourCompany.com
Режим NSX	<p>Не знаете, что выбрать?</p> <p>Используйте встроенную справку по продукту. Щелкните значок информации справа от поля. Обратите внимание, что в справке по полям содержатся сведения, которые помогают настроить параметры.</p> <p>В этом примере выберите <b>Политика</b>.</p>

- г) Чтобы проверить учетные данные, щелкните **Проверить**.

- д) Чтобы привязать облачную учетную запись vCenter, созданную на предыдущем шаге, щелкните **Добавить** а затем выберите **Учетная запись vCenter**.

Привязка облачной учетной записи vCenter обеспечивает безопасность сети.

- е) На странице облачной учетной записи NSX щелкните **Добавить**.

## Шаг 2. Определение вычислительных ресурсов облачной зоны

Облачные зоны — это группы вычислительных ресурсов в учетной записи или регионе, которые предоставляются участникам проектов. Участники проекта развертывают облачные шаблоны с помощью ресурсов в назначенных им облачных зонах. Если требуется более детальное управление развертыванием облачных шаблонов проектов, можно создать несколько облачных зон с разными вычислительными ресурсами.

С помощью учетных записей или регионов облачные поставщики связывают ресурсы с изолированными регионами или хранилищами данных. Учетная запись указывает тип облачной учетной записи, а регион определяет регион или хранилище данных. vCenter Server использует хранилища данных, а предоставляемые ресурсы — это выбранные кластеры и пулы ресурсов.

В рамках данного курса необходимо убедиться, что облачные зоны включают в себя ресурсы, которые обеспечивают достижение целей группы разработки проекта и при этом соответствуют требованиям к бюджету и управлению.

Дополнительные сведения об облачных зонах см. в разделе [Дополнительные сведения об облачных зонах Cloud Assembly](#).

1. Выберите **Инфраструктура > Настроить > Облачные зоны**.
2. Щелкните облачную зону, добавленную для экземпляра vCenter Server, и введите значения.

vCenter Account Cloud Zone
DELETE

Summary
Compute
Projects

A cloud zone defines a set of compute resources that can be used for provisioning.

Account / region \*
vCenter Account / wld01-DC

Name \*
vCenter Account Cloud Zone

Description

Placement policy \*
DEFAULT

Folder
Select folder

Capabilities

Capability tags are effectively applied to all compute resources in this cloud zone, but only in the context of this cloud zone.

Capability tags
Enter capability tags

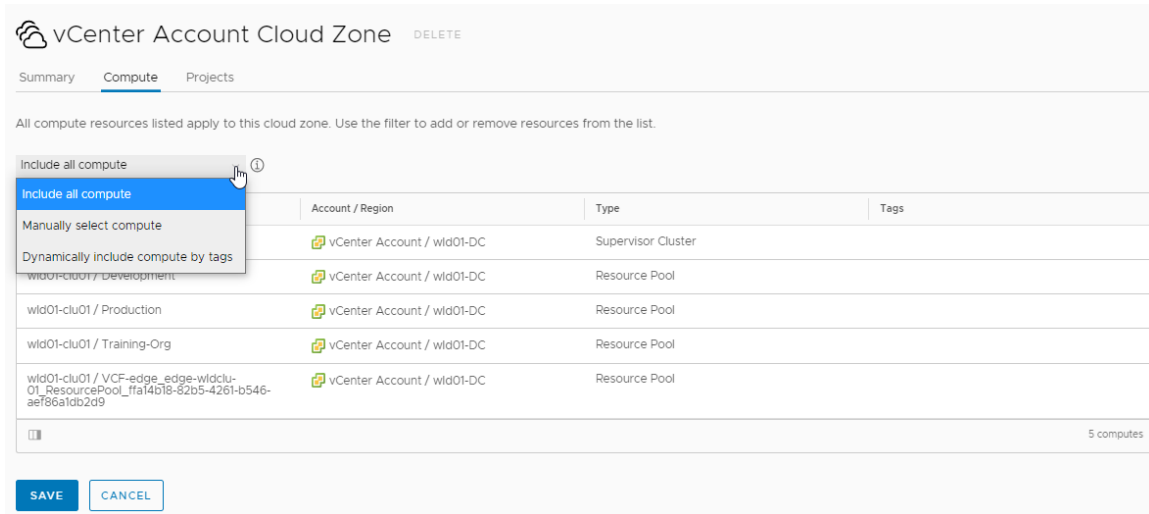
SAVE
CANCEL

Настройка	Пример значения
Учетная запись/область	Учетная запись vCenter/название центра обработки данных
Имя	vCenter Server Cloud Zone Заданное значение нельзя изменить. Если необходимо настроить другой центр обработки данных для другого экземпляра vCenter Server, необходимо создать новую облачную зону, в которой можно выбрать учетную запись или регион.
Описание	Все вычислительные ресурсы vCenter Server для разработки.
Политика	По умолчанию Не забывайте обращаться к справке, если у вас возникли затруднения при заполнении поля.

Помните, что все значения являются примерными. Необходимо использовать параметры зоны для вашей среды.

- Перейдите на вкладку **Вычислительные ресурсы** и убедитесь, что указаны все вычислительные ресурсы.

Если один из них нужно исключить, перейдите в раздел **Выбрать вычислительные ресурсы вручную** и добавьте только те из них, которые требуется включить в облачную зону.



4. Нажмите **Сохранить**.
5. Этот процесс можно повторить для любых дополнительных облачных зон, но при этом необходимо, чтобы имена зон были уникальными.

### Шаг 3. Настройка ресурсов, доступных для учетной записи или региона

Вы добавили учетную запись или регион в облачную зону. Теперь необходимо определить для облачной учетной записи возможные размеры компьютеров (сопоставления конфигураций ресурсов), сопоставления образов, профили сетей и профили хранилища. При разворачивании облачного шаблона оценивается соответствие определениям сопоставлений и профилей. При этом гарантируется, что рабочая нагрузка содержит соответствующий объем ресурсов (сопоставление конфигураций) компьютера, образ, сети и хранилище.

1. Настройка сопоставления конфигураций для учетной записи или регионов.

Определение конфигурации сравнимо с подбором размера одежды. В зависимости от конфигурации облачного шаблона применяемое сопоставление конфигураций определяет объем ресурсов ЦП и памяти.

Дополнительные сведения о сопоставлении конфигураций см. в разделе [Дополнительные сведения о сопоставлении конфигураций ресурсов в vRealize Automation](#).

- а) Выберите **Инфраструктура > Настроить > Сопоставления конфигураций ресурсов**.
- б) Щелкните **Новое сопоставление конфигураций** и введите значения, определяющие небольшие, средние и крупные компьютеры.

Помните, что это примерное значение. Необходимо выбрать соответствующую учетную запись или регионы и определить размеры.

small DELETE

Allows you to define flavors by name in a cloud-agnostic way. ⓘ

Flavor name \* small

Configuration \*

Account / Region	Value
vCenter Account / wld01-DC	2

1 GB

Настройка	Пример значения
Имя конфигурации ресурсов	small
Учетная запись/область	Учетная запись vCenter/ЦОД
Количество ЦП	2
Объем памяти	1 ГБ

в) Щелкните **Создать**.

г) Чтобы создать дополнительные размеры, настройте сопоставления средних и крупных конфигураций для учетной записи или региона.

Настройка	Пример значения
Имя конфигурации ресурсов	средний
Учетная запись/область	vCenter Account/Datacenter
Количество ЦП	4
Объем памяти	2 ГБ
Имя конфигурации ресурсов	масштабный
Учетная запись/область	vCenter Account/Datacenter
Количество ЦП	8
Объем памяти	4 ГБ

## 2. Настройка сопоставления образов для учетной записи или регионов.

Образы — это операционная система для компьютеров в облачном шаблоне. При работе с образами vCenter Server необходимо выбрать шаблоны vCenter.

Дополнительные сведения о сопоставлении образов см. в разделе [Дополнительные сведения о сопоставлении образов в vRealize Automation](#).

а) Выберите **Инфраструктура > Настроить > Сопоставления образов**.

б) Щелкните **Новое сопоставление образов** и найдите образы для учетной записи или региона.

Помните, что это примерное значение. Необходимо выбрать образы, которые были обнаружены в вашей учетной записи или регионе.

Настройка	Пример значения
Имя образа	centos
Учетная запись/область	Учетная запись vCenter
Изображение	centos7

- в) Щелкните **Создать**.
  - г) Повторите процедуру, чтобы создать дополнительные сопоставления образов. Например, сопоставление ubuntu для учетной записи или региона.
3. Настройка профилей сетей.

Профили сетей определяют сети и их параметры, доступные для учетной записи или региона. Профили должны поддерживать целевые среды развертывания.

Эта задача предоставляет минимальные сведения о конфигурации для успешного выполнения. Чтобы получить дополнительные сведения о профилях сетей, см. [Дополнительные сведения о профилях сетей в vRealize Automation](#).

- а) Выберите **Инфраструктура > Настроить > Профили сетей**.
- б) Щелкните **Создать профиль сети** и создайте профиль для учетной записи или региона vCenter Account/Datacenter.



**Network Profile** [DELETE](#)

Summary **Networks** Network Policies Load Balancers Security Groups

A network profile defines a group of networks and network settings used when machines are provisioned.

**Account / region** vCenter Account / wld01-DC

**Name \*** Network Profile

**Description** Networks for development teams.

**Capabilities**  
Capability tags listed here are matched to constraint tags in the cloud template.

**Capability tags** Enter capability tags

Настройка	Пример значения
Учетная запись/область	vCenter Account/Datacenter
Имя	Network Profile
Описание	Сети для команд разработки.

в) Перейдите на вкладку **Сети** и нажмите кнопку **Добавить сеть**.

**Network Profile** [DELETE](#)

Summary **Networks** Network Policies Load Balancers Security Groups

Networks listed here are used when provisioning to existing, on-demand, or public networks.

[+ ADD NETWORK](#) [TAGS](#) [MANAGE IP RANGES](#) [REMOVE](#)

<input type="checkbox"/>	Name	Account / Region	Zone	Network Domain	CIDR	Support Public IP	Default for Zone	Origin	Tags
<input type="checkbox"/>	DevProject-004	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.1.64/27	--	--	Deployed	
<input type="checkbox"/>	External-mcm13/3520-150877845350	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	172.16.1.64/28	--	--	Discovered	
<input type="checkbox"/>	seg-domain-c8e2a5389de-2772-43f5-9eaa-eddc05e35996-vmware-system-nsx-0	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	10.244.0.0/28	--	--	Discovered	<a href="#">external_id.8...</a> <a href="#">ncp/project_u...</a> <a href="#">ncp/cluster.d...</a> <a href="#">ncp/version.1...</a> <a href="#">ncp/project.v...</a>

1 - 3 of 3 networks

г) Выберите сети NSX, которые должны быть доступны для группы разработчиков приложений.

В этом примере используется сеть NSX-T под названием DevProject-004.

д) Перейдите на вкладку **Политики сети** и создайте политику.

**New Network Profile**

Summary Networks **Network Policies** Load Balancers Security Groups

Use these settings when creating outbound, private and routed networks. ⓘ

Isolation policy: None ⓘ

Network Resources  
Provide on-demand network resources.

Tier-0 logical router: Q TO ⓘ

Edge cluster: edge-widclu-01 ⓘ

**CREATE** **CANCEL**

Настройка	Пример значения
Политика изоляции	Нет
Логический маршрутизатор уровня 0	Tier-0-router
Пограничный кластер	EdgeCluster

е) Щелкните **Создать**.

#### 4. Настройка профилей хранилища.

Профили хранилища определяют диски для учетной записи или региона. Профили должны поддерживать целевые среды развертывания.

Дополнительные сведения о профилях хранилища см. в разделе [Дополнительные сведения о профилях хранилища в vRealize Automation](#).

а) Выберите **Инфраструктура > Настроить > Профиль хранилища**.

б) Щелкните **Создать профиль хранилища** и создайте профиль для учетной записи или региона vCenter Account/Datacenter.

Если значения не указаны в таблице, оставьте значения по умолчанию.

**Storage Profile**

Account / region: vCenter Account / wld01-DC

Name: Storage Profile

Description:

Disk type: ☒ Standard disk ☐ First class disk (FCD) ⓘ

Storage policy: Datastore default ⓘ

Datastore / cluster: wld01-sc2vc05-wld01-clu01-vsan01 ⓘ

Provisioning type: Unspecified ⓘ

Shares: Unspecified ⓘ

Limit IOPS: ⓘ

Disk mode: Dependent ⓘ

☐ Supports encryption ⓘ

☒ Preferred storage for this region ⓘ

Capability tags: Enter capability tags ⓘ

**SAVE** **CANCEL**

Настройка	Пример значения
Учетная запись/область	vCenter Account/Datacenter
Имя	Профиль хранилища
Хранилище данных или кластер	Выбрано хранилище данных достаточной емкости, которое доступно всем узлам.
Предпочтительный объем хранилища для этой области	Установите этот флажок.

в) Щелкните **Создать**.

## Шаг 4. Создание проекта

На этом этапе вы определяете параметры, обеспечивающие выполнение целей проекта.

- Каким пользователям нужен доступ к вычислительным ресурсам, чтобы создавать и развертывать облачный шаблон приложения? Дополнительные сведения о возможностях различных ролей проекта см. в разделе [Роли пользователей организаций и служб в vRealize Automation](#).
- Будут ли участники проекта создавать приложения, предназначенные для производственной среды? Какие ресурсы необходимы?
- Какие облачные зоны им потребуются? Какие приоритеты и ограничения должны быть установлены для каждой зоны проекта?

В этом учебнике подобраны задания по поддержке группы разработки, которая создает и развивает корпоративное приложение.

Эта задача предоставляет минимальные сведения о конфигурации для успешного выполнения. Чтобы получить дополнительные сведения о проектах, см. [Дополнительные сведения о проектах Cloud Assembly](#).

1. Выберите **Инфраструктура > Администрирование > Проекты**.
2. Щелкните **Создать проект** и введите имя **Development Project**.
3. Перейдите на вкладку **Пользователи** и щелкните **Добавить пользователей**.

В этот момент добавлять пользователей не требуется. Но если нужно, чтобы с облачными шаблонами работали другие пользователи, они должны быть участниками проекта.

4. Введите адреса электронной почты, чтобы добавить пользователей в качестве участников или администраторов проекта, в зависимости от того, какие у них должны быть разрешения.

5. Щелкните **Предоставление**, а затем **Добавить зоны > Облачная зона**.
6. Добавьте облачные зоны, в которых пользователи могут выполнять развертывание.

Кроме того, можно установить ограничения на ресурсы для облачной зоны в проекте. В будущем можно установить различные ограничения для других проектов.

Параметр облачной зоны проекта	Пример значения
Облачная зона	vCenter Account Cloud Zone
Приоритет подготовки	1
Ограничение по экземплярам	5

7. Добавление в проект дополнительных облачных зон.
8. Щелкните **Создать**.

9. Чтобы убедиться, что проект добавлен в облачную зону, выберите **Инфраструктура > Настроить > Облачные зоны** и откройте карточку облачной зоны vCenter Account Zone, чтобы просмотреть содержимое вкладки **Проекты**. Должен отобразиться проект Development Project.

## Шаг 5. Проектирование и развертывание базового облачного шаблона

На этом этапе вы будете разрабатывать и развертывать облачный шаблон, чтобы убедиться, что инфраструктура правильно настроена для его поддержки. Позднее при создании приложения в соответствии с требованиями проекта вы сможете дополнить шаблон.

Лучшим способом создания облачного шаблона является последовательное добавление компонентов с проверочным развертыванием после каждого изменения. Процесс, описываемый в учебнике, начинается с простого компьютера, в который постепенно добавляются дополнительные ресурсы.

В примерах используется редактор кода YAML. Он позволяет представить фрагменты кода в удобной форме. Если вы предпочитаете пользовательский интерфейс на основе диалоговых окон, нажмите **Входные данные**.

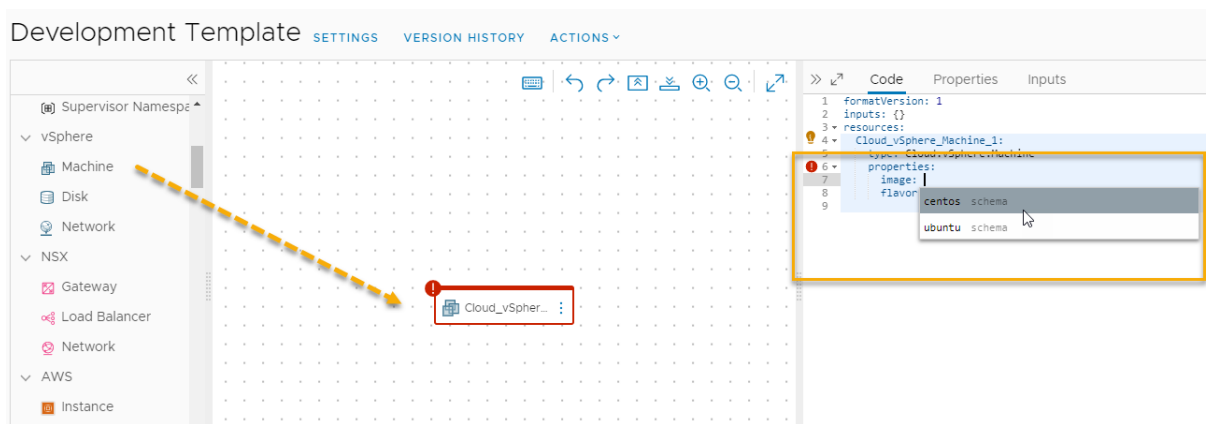
Возможности облачных шаблонов гораздо шире тех, что представлены в этом учебнике. Чтобы получить дополнительные сведения, см. [Глава 6 Проектирование развертываний Cloud Assembly](#).

В этом учебнике используются типы ресурсов vSphere и NSX. Эти типы ресурсов можно развертывать только в конечных точках облачной учетной записи vCenter Server. Кроме того, для создания облачных шаблонов, которые можно развертывать в любой конечной точке, можно использовать типы ресурсов, независимых от облачной среды. Пример настройки инфраструктуры и проектирования шаблона для любой конечной точки см. в разделе [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).



Ссылку на видео, где показаны основные этапы данной процедуры, см. в разделе [Проектирование и развертывание базового облачного шаблона](#).

1. Выберите **Проектирование > Облачные шаблоны**.
2. Выберите **Создать из > Пустой холст**.
3. В поле **Имя** введите **Development Template**, выберите **Проект Development Project** и нажмите **Создать**.
4. Добавьте компьютер vSphere на холст проекта, проверьте его и разверните.



- а) Перетащите **компьютер vSphere** с панели типов ресурсов на холст.

Обратите внимание, что на панели **Код** отображается YAML для компьютера; для образа и предварительно указанных свойств ЦП и памяти значения не заданы. Этот шаблон можно настроить таким образом, чтобы он позволял гибко изменять объем ресурсов.

- б) Чтобы выбрать значение образа, поместите указатель между одинарными кавычками для `image` и выберите **centos** из списка настроенных образов.

Помните, что это примерное значение. Если вы не настроили образ `centos`, выберите тот образ, который вы настроили.

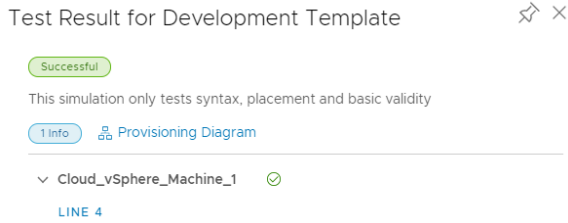
- в) Создайте строку под свойством образа и введите или выберите `flavor`, а затем выберите `small` в списке.
- г) Удалите `cpuCount` и `totalMemory`.

YAML должен выглядеть так, как показано в этом примере.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
```

- д) Щелкните **Тест**.

Тест позволяет проверить синтаксис и размещение облачного шаблона. Успешная проверка не гарантирует отсутствие ошибок при развертывании шаблона.



В случае неудачной проверки щелкните **Схема предоставления** и найдите точки сбоя. Дополнительные сведения об использовании схем для устранения неисправностей см. в разделе [Тестирование базового облачного шаблона](#).

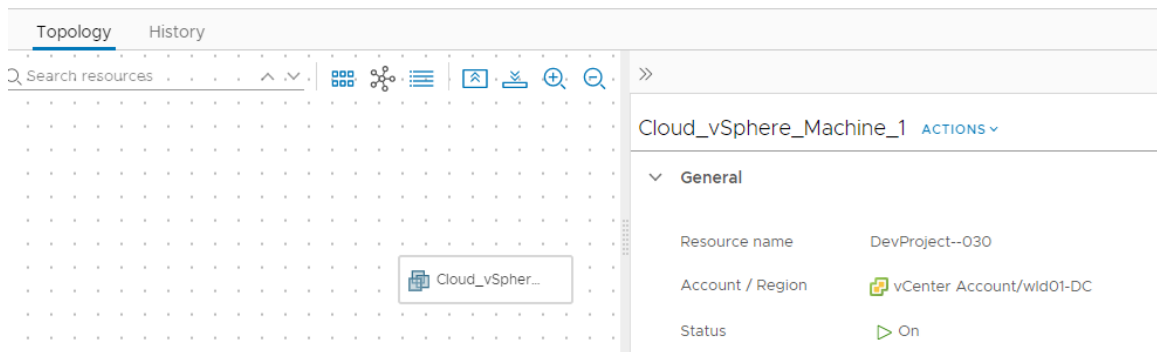
е) Щелкните **Развернуть**.

ж) Укажите **имя развертывания (DevTemplate — компьютер)** и нажмите **Развернуть**.

Ход развертывания можно отслеживать на странице «Сведения о развертывании DevTemplate» или на странице «Развертывания». Выберите **Ресурсы > Развертывания**.

В случае сбоя при развертывании можно устранить проблему и исправить шаблон. См. раздел [Что можно сделать в случае неудачного развертывания Cloud Assembly](#).

Успешное развертывание выглядит, как пример на странице «Развертывания».



5. Настройте управления версиями шаблона и добавьте сеть.

Управление версиями облачного шаблона требуется для помещения шаблона в каталог Service Broker; также при разработке полезно иметь работоспособную версию для отмены изменений.

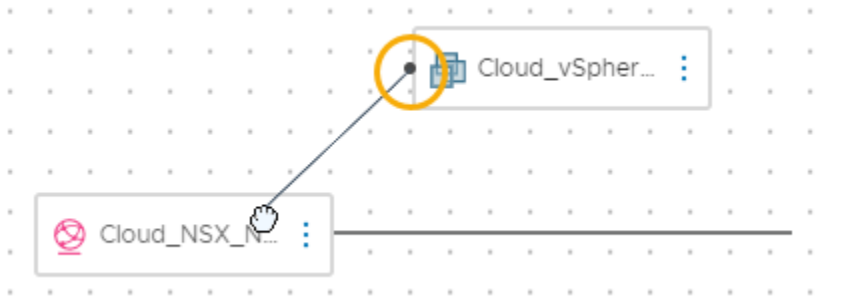
а) Откройте шаблон на холсте проекта.

б) Щелкните **Версия**, введите **Описание** аналогично **простому развертываемому компьютеру** и нажмите **Создать**.

в) С панели типов ресурсов перетащите на холст тип ресурса **Сеть NSX**.

г) Подключение компьютера к сети.

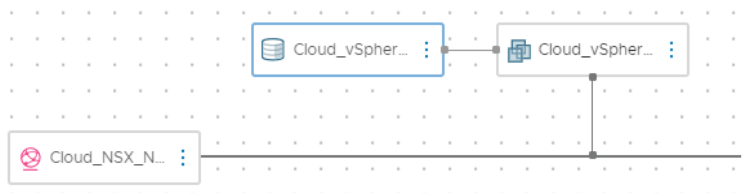
Щелкните маленький кружок на компоненте компьютера и перетащите подключение к сети.



Обратите внимание, что теперь YAML выглядит, как в этом примере.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
      attachedDisks: []
  Cloud_NSX_Network_1:
    type: Cloud.NSX.Network
    properties:
      networkType: existing
```

- д) Щелкните **Тест**, чтобы проверить шаблон.
  - е) Щелкните **Развернуть**.
  - ж) Введите имя **DevTemplate – компьютер – сеть** и нажмите **Развернуть**.
  - з) Отслеживайте ход выполнения и оцените успешность развертывания.
6. Настройте управления версиями шаблона и добавьте диск с данными.
- а) Откройте шаблон на холсте проекта.
  - б) Настройте управления версиями шаблона.
- Введите в качестве описания **Компьютер с существующей сетью**.
- в) С панели типов ресурсов перетащите на холст тип ресурса **Диск vSphere**.
  - г) Подключите диск к компьютеру.





Обратите внимание, что теперь YAML выглядит, как в этом примере.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
  Cloud_NSX_Network_1:
    type: Cloud.NSX.Network
    properties:
      networkType: existing
```

- д) Проверка шаблона.
- е) Разверните шаблон с именем **DevTemplate — компьютер — сеть — хранилище**.
- ж) Отслеживайте ход выполнения и оцените успешность развертывания.
- з) Настройте управления версиями шаблона.

Введите в качестве описания **Компьютер с существующей сетью и диском хранилища**.

Эта окончательная версия обеспечивает добавление рабочего шаблона в каталог служб.

## Результаты обучения

Вы выполнили процедуру настройки Cloud Assembly в качестве рабочей системы. Теперь вы знакомы со следующими понятиями.

- Облачные учетные записи — это учетные данные, связывающие Cloud Assembly с конечными точками поставщика облачных служб.
- Облачные зоны — это выбранные вычислительные ресурсы в учетной записи или регионах, которые затем выделяются разным проектам в зависимости от их потребностей и ваших задач по управлению затратами.
- Ресурсы инфраструктуры — это определения ресурсов, связанных с учетной записью или регионами, которые используются в облачных шаблонах.
- Проекты — это способ предоставления пользователям доступа к облачным зонам в зависимости от целей разработки приложений в проекте.

- Облачные шаблоны — это определения рабочих нагрузок приложений, которые итерационно разрабатываются и развертываются в проекте.

Этот учебник закладывает фундамент разработки с использованием Cloud Assembly. Изученный процесс можно использовать для создания инфраструктуры и совершенствования навыков разработки облачных шаблонов.

## Учебник «Настройка Cloud Assembly для предоставления производственной рабочей нагрузки»

Администратору облачных систем требуется автоматизировать процесс развертывания для проекта, чтобы решение Cloud Assembly выполняло всю необходимую работу, когда разработчики облачных шаблонов создают и развертывают шаблоны. Например, рабочие нагрузки развертываются с определенным настраиваемым шаблоном именования компьютеров, компьютеры добавляются в определенное организационное подразделение Active Directory, при этом используются конкретные записи DNS и диапазоны IP-адресов.

Автоматизация процесса развертывания проекта упрощает управление несколькими проектами в различных центрах обработки данных и облачных средах.

Не требуется выполнять все указанные здесь задачи. Любые из них можно совмещать в зависимости от целей управления.

### Перед началом работы

Для выполнения заданий из этого учебника требуется настроенная инфраструктура и развернутый облачный шаблон с компьютером и сетью. Убедитесь, что в вашей системе уже настроены указанные ниже параметры.

- Вы успешно выполнили все шаги, указанные в учебнике по инфраструктуре. См. раздел [Учебник «Настройка и тестирование инфраструктуры и развертываний vSphere в Cloud Assembly»](#).
- Вам назначена роль администратора Cloud Assembly. См. раздел [Роли пользователей организаций и служб в vRealize Automation](#).

### Настройка имен компьютеров

В этом задании необходимо настроить формат имени компьютеров, развертываемых для проекта разработки. Имена должны включать значение свойства costCenter в проекте, тип ресурса, выбираемый во время развертывания, и уникальный номер. Например, DevProject-centos-021.

Этот пример можно изменить в соответствии с требованиями к именованию в вашей организации.

Дополнительные сведения о проектах см. в разделе [Глава 5 Добавление проектов Cloud Assembly и управление ими](#).



Ссылку на видео, где приведен пример настраиваемого именования, см. в разделе [Создание настраиваемого шаблона именования для развертываний](#).

1. Выберите **Инфраструктура > Проекты**.
2. Выберите существующий проект или создайте новый.  
В этом учебнике используется имя проекта Development Project.
3. Щелкните **Создать**.
4. На странице «Проекты» щелкните имя проекта на плитке, чтобы настроить его.
5. Перейдите на вкладку **Пользователи** и добавьте пользователей, которые являются участниками этого проекта.
6. Перейдите на вкладку **Подготовка**.
  - а) В разделе «Зоны» щелкните **Добавить зону** и добавьте возможные облачные зоны, где должны развертываться рабочие нагрузки для этого проекта.
  - б) В разделе «Настраиваемые свойства» добавьте настраиваемое свойство с именем **costCenter** и значением **DevProject**.

- в) В разделе «Настраиваемое именование» добавьте следующий шаблон именования.

```
${resource.costCenter}-${resource.installedOS}-${###}
```

`${resource.installedOS}` зависит от операционной системы, выбранной при развертывании облачного шаблона.

7. Нажмите **Сохранить**.
8. Обновите облачный шаблон, используя входное значение для типа операционной системы.

Входные значения — это способ непосредственной настройки формы запроса на развертывание, который упрощает процесс разработки. Создавая входные значения, можно использовать один облачный шаблон для развертывания рабочих нагрузок с разными конфигурациями. Например, с разными размерами или операционными системами.

В этом примере используется шаблон Development Template из предыдущего учебника. См. раздел [Шаг 5. Проектирование и развертывание базового облачного шаблона](#).

- а) Выберите **Проектирование** и откройте шаблон Development Template.

б) На панели «Код» обновите YAML, внося следующие изменения.

- В разделе `Inputs` добавьте `installedOS`.

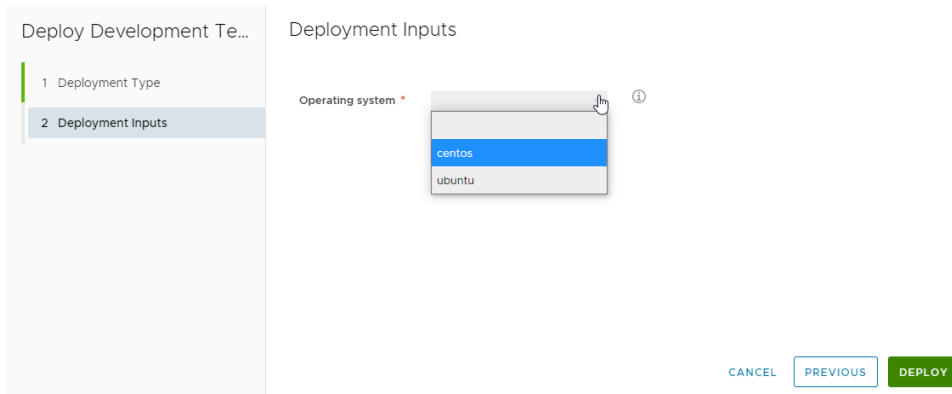
На следующем этапе можно увидеть, что для указания образа также необходимо ввести значение `installedOS`. При добавлении строк в раздел `enum` значения (в этом примере это `centos` и `ubuntu`) должны совпадать с именами образов, которые были заданы в меню **Инфраструктура > Настройка > Сопоставление образов**. Например, если имя сопоставления образа — `CentOS`, а не `centos`, то в разделе «Входные данные» необходимо указать `CentOS`.

```
inputs:
  installedOS:
    type: string
    title: OS Type
    description: Select the operating system.
    enum:
      - centos
      - ubuntu
```

- В разделе `Cloud_vSphere_Machine_1` замените `image` на входной параметр `installedOS` (`{input.installedOS}`) и добавьте настраиваемое свойство `installedOS` с тем же входным параметром.

```
resources:
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: ${input.installedOS}
      installedOS: ${input.installedOS}
      flavor: small
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
  Cloud_NSX_Network_1:
    type: Cloud.NSX.Network
    properties:
      networkType: existing
```

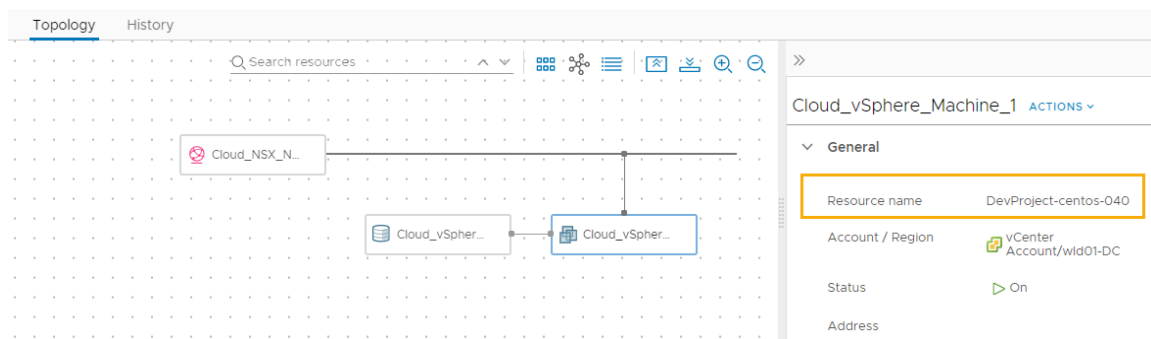
- в) Щелкните **Развернуть** и введите имя **Тест развертывания с настраиваемым именем**.
- г) Нажмите кнопку **Далее**.
- д) В этом же раскрывающемся меню выберите операционную систему **centos**.



е) Щелкните **Развернуть**.

9. Отслеживайте ход выполнения и оцените успешность развертывания.

Имя компьютера в этом примере — DevProject-centos-026. Напоминаем, что этот пример основан на учебнике, упомянутом в начале этого задания.



## Создание записей компьютеров в Active Directory

При подготовке рабочей нагрузки можно создавать записи компьютеров в Active Directory. Настроив в Cloud Assembly автоматическое выполнение этой задачи при развертывании проекта, администраторы облачных систем могут существенно облегчить свою работу.

1. Добавление интеграции Active Directory.

а) Выберите **Инфраструктура > Подключения > Интеграции**.

Эти шаги охватывают базовые параметры настройки Active Directory, рассматриваемые в этом учебнике по записям компьютеров в Active Directory. Дополнительные сведения об интеграции Active Directory см. в разделе [Создание интеграции Active Directory в Cloud Assembly](#).

б) Щелкните **Добавить интеграцию** и выберите **Active Directory**.

- в) Введите имя, которое используется для этой интеграции.
- г) Введите **узел или IP-адрес LDAP** и соответствующие учетные данные.
- д) Введите **базовое DN**.

В этом учебнике в качестве примера используется **ou=AppDev,dc=cmbu,dc=local**. AppDev — это родительский объект подразделения компьютера, которое должно быть добавлено в проект.

- е) Нажмите кнопку **Добавить**.

## 2. Добавление проекта в интеграцию.

## 3. В интеграции Active Directory перейдите на вкладку **Проекты** и щелкните **Добавить проект**.

Add Projects

Select a project and the OU it will be mapped to by adding its relative DN. The effective DN is created by appending the RDN to the integration base DN (**dc=cmbu,dc=local**).

- а) Выберите проект разработки приложения (App Development).
- б) Введите относительные DN. Например, **OU=AppDev-Computers**.

- в) Оставьте выключенными переключатели «Переопределяет» и «Игнорирует».

Эта процедура предназначена для автоматизации процесса для проекта. Она не служит для настройки параметров в шаблонах.

- г) Нажмите кнопку **Добавить**.

4. Чтобы сохранить изменения в интеграции, нажмите кнопку **Сохранить**.
5. Разверните облачный шаблон для проекта и убедитесь, что компьютер добавлен в нужное организационное подразделение Active Directory.

## Настройка сетевых DNS-серверов и диапазона внутренних IP-адресов

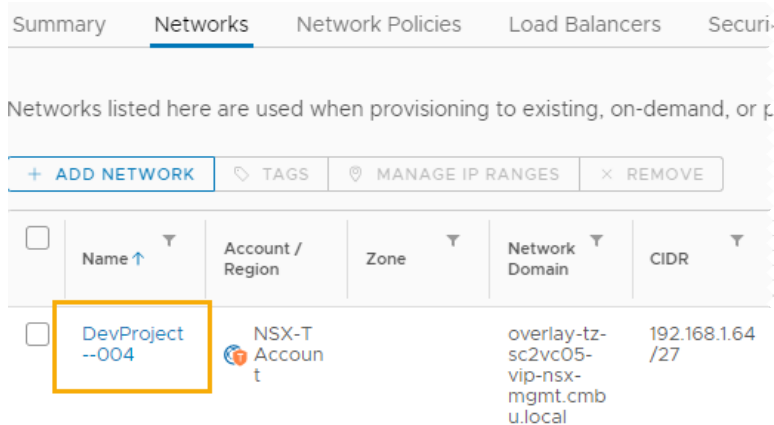
Добавьте или обновите профиль сети, чтобы включить в него DNS-серверы и диапазоны внутренних IP-адресов.

Облачная учетная запись для vSphere, NSX-V или NSX-T должна быть уже создана. См. [Учебник «Настройка и тестирование инфраструктуры и развертываний vSphere в Cloud Assembly»](#) или [Добавление учетных записей облачной службы в Cloud Assembly](#).

1. Выберите **Инфраструктура > Настроить > Профили сети**.
2. Выберите существующий профиль или создайте новый.
3. На вкладке **Сводка** выберите **Учетная запись/регион** и введите имя.

В этом учебнике в качестве имени профиля сети используется Network Profile.

4. Добавьте сети.
  - а) Перейдите на вкладку **Сети**.
  - б) Щелкните **Добавить сеть**.
  - в) Добавьте одну сеть NSX или vSphere или несколько.
  - г) Нажмите кнопку **Добавить**.
5. Настройте DNS-серверы.
  - а) В списке сетей на вкладке **Сети** щелкните имя сети.



- б) Введите IP-адреса DNS-сервера, который должен использоваться для этой сети.

DevProject--004

DNS servers

192.168.1.22  
192.168.1.23

DNS search domains

company.local

DNS Servers

Use a comma separated list or new lines.

- в) Нажмите **Сохранить**.

6. Задайте диапазон IP-адресов сети.

- а) Установите флажок рядом с именем в списке сетей.

Network Profile **DELETE**

Summary **Networks** Network Policies Load Balancers Security Groups

Networks listed here are used when provisioning to existing, on-demand, or public networks. ⓘ

+ ADD NETWORK TAGS **MANAGE IP RANGES** REMOVE

<input type="checkbox"/>	Name	Account / Region	Zone	Network Domain	CIDR	Su Pu
<input type="checkbox"/>	External-mcm1343745-148168716643	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	172.16.12.64/28	
<input type="checkbox"/>	NSX-mcm1376447-151082888186	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.100.32/28	
<input checked="" type="checkbox"/>	NSX-mcm39835-146434698964	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.1.0/27	

1

- б) Нажмите **Управление диапазонами IP-адресов**.

- в) В диалоговом окне «Управление диапазонами IP-адресов» щелкните **Новый диапазон IP-адресов**.



## New IP Range

Network *	NSX-mcm1376447-151082888186
Source	<input checked="" type="radio"/> Internal <input type="radio"/> External
Name *	DevProject Range
Description	<div></div>
CIDR	192.168.100.32/28
Start IP address *	192.168.100.34
End IP address *	192.168.100.46

г) Введите имя.

Например, **DevProject Range**.

д) Чтобы определить диапазон, введите **начальный IP-адрес** и **конечный IP-адрес**.

е) Нажмите кнопку **Добавить**.

ж) Добавьте дополнительные диапазоны или нажмите кнопку **Заккрыть**.

- Добавьте облачную зону, содержащую учетную запись или регион, связанные с сетью и настроенные для проекта разработки.
- Разверните облачный шаблон для проекта и проверьте предоставление компьютера в указанном диапазоне IP-адресов.

## Учебник «Использование тегов в Cloud Assembly для управления ресурсами vSphere»

Теги — это эффективные метаданные, которые можно связать с ресурсами и добавлять в шаблоны. Теги можно использовать в различных сценариях управления, в том числе, для размещения рабочей нагрузки и разметки ресурсов.

### Краткое введение в теги

В этом разделе представлены простые сведения о тегах, которые применяются к указанным шагам. Более подробные сведения о тегах см. в [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#).

- Теги возможностей и ограничений

Теги можно использовать для управления развертываниями на основе возможностей ресурсов. Например, администратору облака может быть необходимо развертывать итерационно разрабатываемые облачные шаблоны в специальном пуле ресурсов для разработки, а шаблоны, предназначенные для производственной среды, — в другом пуле ресурсов.

- Чтобы определить возможности ресурсов, в них добавляются теги возможностей.
- Теги ограничений применяются в облачных шаблонах и определяют ресурсы, которые должны использоваться развернутыми ресурсами.
- Теги-метки

Для управления ресурсами можно добавлять теги в виде меток или описаний объектов. Для управления нужно использовать результаты поиска наилучших ресурсов, которые могут быть разными для похожих объектов, добавление пользовательских аннотаций к объектам, предоставление сведений сторонним системам, создание критериев членства в группах безопасности, обеспечение согласованности между связанными доменами программно-определяемого ЦОД.

## Перед началом работы

- Ознакомьтесь с ресурсами и облачным шаблоном, которые определены в [Учебник «Настройка и тестирование инфраструктуры и развертываний vSphere в Cloud Assembly»](#). Здесь используются примеры значений из учебника.

## Использование тегов для управления размещением рабочей нагрузки

В этом простом примере используются теги среды разработки и производственной среды для демонстрации способа применения тегов возможностей и ограничений. Сначала необходимо добавить теги возможностей в вычислительные ресурсы пула ресурсов vCenter Server, а затем добавить теги в облачный шаблон.

В примере облачного шаблона показан способ использования входных данных, с помощью которого пользователь, выполняющий развертывание, может указать, где следует выполнять развертывание: в среде разработки или в пуле ресурсов производственной среды.

Пример использования одинаковых тегов для определения размещения в мультиоблачной среде см. в [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).

1. Добавьте теги возможностей в пулы ресурсов.
  - а) Выберите пункт **Инфраструктура > Ресурсы > Вычислительные ресурсы**.

- б) Откройте облачную зону и нажмите **Вычислительные ресурсы**.

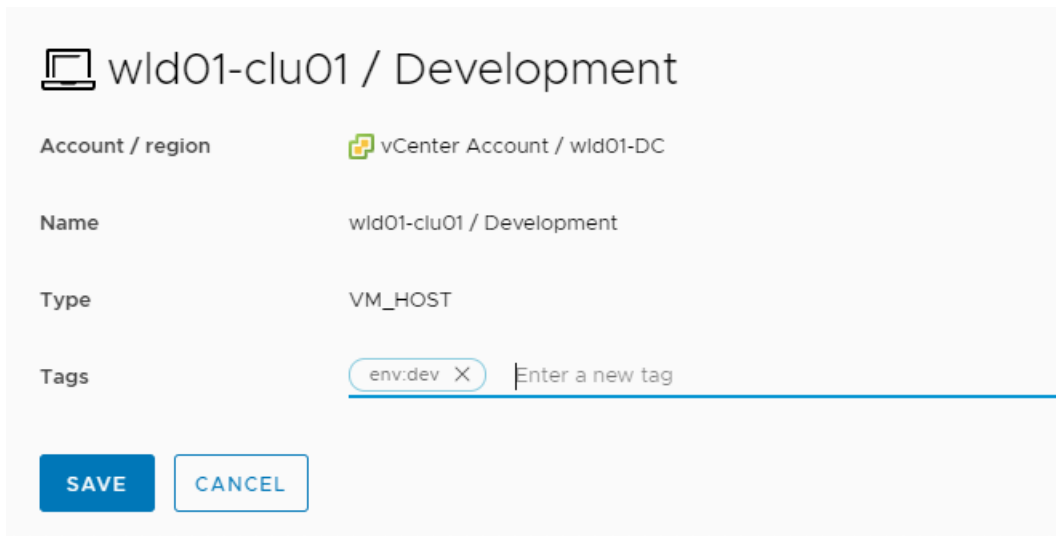


- в) Найдите и щелкните пул ресурсов, в котором необходимо развернуть рабочие нагрузки, связанные с разработкой.

В этом учебнике используются следующие значения. Помните, что все значения приведены только для примера. Необходимо указать значения для вашей среды.

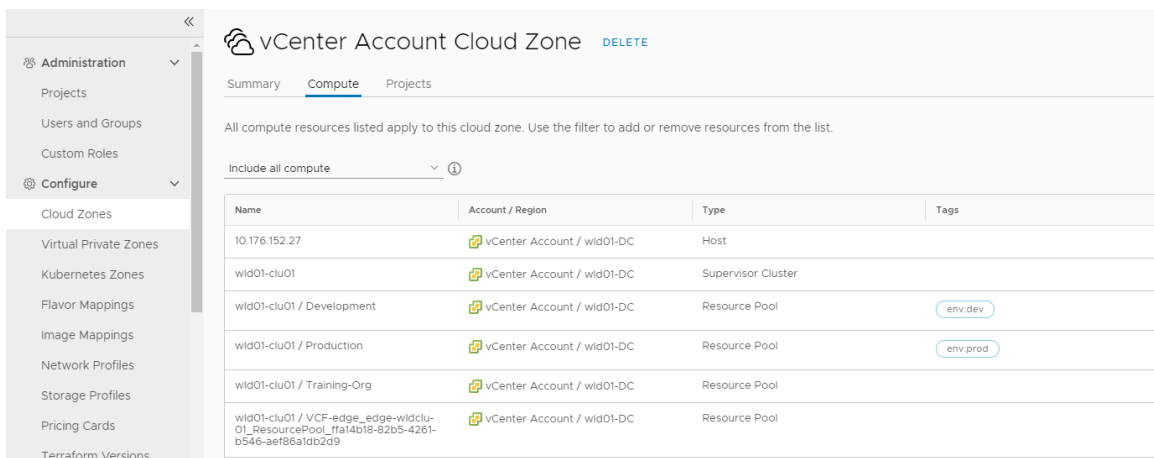
Пример пула ресурсов	Пример тега
wid01-clu01/Development	env:dev
wid01-clu01/Production	env:prod

- г) Добавьте тег **env:dev** и нажмите кнопку **Сохранить..**



- д) Повторите процесс для пула ресурсов, в котором необходимо развернуть производственные рабочие нагрузки, и добавьте тег **env:prod**.
2. Убедитесь, что теги возможностей добавлены в пулы ресурсов в облачной зоне.
- а) Выберите **Инфраструктура > Настроить > Облачные зоны**.
- б) Откройте облачную зону, связанную с проектом, и нажмите **Вычислительные ресурсы**.

В этом примере облачной зоной является облачная зона учетной записи vCenter, а теги добавлены в два пула ресурсов (wid01-clu01/Development и wid01-clu01/Production).



### 3. Добавьте теги ограничений в облачный шаблон.

Теги ограничений используются для указания места развертывания шаблона.

а) Выберите **Проектирование > Облачные шаблоны**, а затем откройте шаблон.

В этом учебнике используется шаблон с именем Development Project.

б) Ознакомьтесь с кодом YAML для шаблона на панели «Код».

Этот код YAML является отправной точкой для данного учебника.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: medium
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 5
  Cloud_NSX_Network_1:
    type: Cloud.NSX.Network
    properties:
      networkType: existing
```

в) Добавьте тег ограничения к ресурсу Cloud\_vSphere\_Machine\_1, используя \${input.placement} в качестве переменной.

```
resources:
  Cloud_vSphere_Machine_1:
```

```

type: Cloud.vSphere.Machine
properties:
  image: centos
  flavor: medium
  constraints:
    - tag: '${input.placement}'
  networks:
    - network: '${resource.Cloud_NSX_Network_1.id}'
  attachedDisks:
    - source: '${resource.Cloud_vSphere_Disk_1.id}'

```

- г) Определите переменную размещения в разделе «Входные данные».

```

inputs:
  placement:
    type: string
    enum:
      - env:dev
      - env:prod
    default: env:dev
    title: Select Placement for Deployment
    description: Target Environment

```

- д) Убедитесь, что итоговый код YAML похож на следующий пример.

```

formatVersion: 1
inputs:
  placement:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
    default: 'env:dev'
    title: Select Placement for Deployment
    description: Target Environment
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
      constraints:
        - tag: '${input.placement}'
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 5

```

```
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing
```

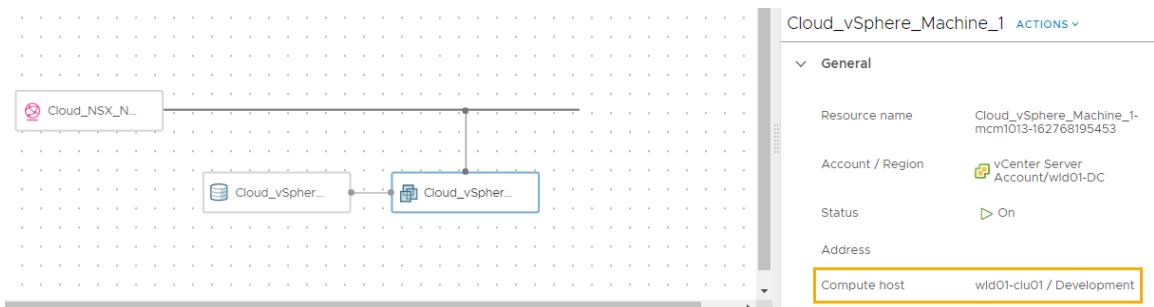
- е) Чтобы посмотреть, как переменная тега работает с доступными ресурсами, нажмите **Тестировать**, а затем выберите **env:dev**.



Повторите тест, используя **env:prod**. После успешного выполнения обоих тестов разверните шаблон и убедитесь, что он работает.

4. Разверните шаблон, чтобы протестировать размещение рабочей нагрузки.
  - а) В конструкторе облачных шаблонов щелкните **Развернуть**.
  - б) Введите **Deployment Tag Dev** в поле **Имя развертывания** и нажмите **Далее**.
  - в) Выберите **env:dev** в раскрывающемся меню **Выбор размещения для развертывания** и нажмите **Развернуть**.
5. Убедитесь, что ресурсы шаблона развернуты в выбранном пуле ресурсов.
  - а) Выберите **Ресурсы > Развертывания** и найдите развертывание Deployment Tag Dev.
  - б) Откройте сведения о развертывании и нажмите **Топология**.
  - в) Щелкните компьютер vSphere и разверните сведения о компьютере на правой панели.
  - г) В разделе **Общие** найдите **Вычислительный узел** и убедитесь, что значение соответствует пулу ресурсов, который соответствует тегу env:dev.

В этом примере используется значение wid01-clu01 / Development. Оно показывает, что рабочая нагрузка была развернута в нужном пуле ресурсов на основе выбранного тега ограничений.



- д) Повторите процесс развертывания. В этот раз выберите **env:prod**.

## Добавление тегов в качестве меток, которые можно использовать в vCenter Server и NSX-T

Теги можно добавлять в развертывания, которые затем можно использовать для управления ресурсами.

В этом примере добавляются теги для идентификации компьютера MySQL и сети. Кроме того, добавляется тег для идентификации веб-сети. С учетом того, как теги работают в существующих сетях по сравнению с сетями по требованию, можно выбрать один из двух вариантов.

- В случае использования профиля существующей сети, который применялся в предыдущем разделе, тег NGINX:web не добавляется к существующим объектам в NSX-T. Поэтому операции проверки в отношении этого тега в NSX-T можно игнорировать.
- В случае создания профиля сети по требованию можно обновить сеть в коде YAML, чтобы использовать маршрутизируемую сеть или сеть по требованию. В этом примере используется сеть по требованию, поэтому показан тег NGINX:web для нового объекта в NSX-T.

Следующий код YAML взят из предыдущего примера, за исключением того, что в нем используется тип маршрутизируемой сети по требованию. Код содержит теги ограничений.

В этом учебнике используются следующие значения. Помните, что все значения приведены только для примера. Необходимо указать значения для вашей среды.

```
formatVersion: 1
inputs:
  placement:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
    default: 'env:dev'
    title: Select Placement for Deployment
    description: Target Environment
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
      constraints:
        - tag: '${input.placement}'
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 5
  Cloud_NSX_Network_1:
    type: Cloud.NSX.Network
    properties:
      networkType: routed
      constraints:
```

```
- tag: 'net:od'
```

1. Выберите **Проектирование > Облачные шаблоны**, а затем откройте шаблон.
2. В свойствах Cloud\_vSphere\_Machine\_ добавьте следующий тег.

```
tags:
- key: db
  value: mysql
```

3. Добавьте теги сетевого адаптера вычислительной машины.

```
tags:
- key: db
  value: mysql
```

4. Добавьте теги логического коммутатора или сегмента NSX.

```
tags:
- key: NGINX
  value: web
```

5. Убедитесь, что код YAML похож на код в следующем примере.

```
formatVersion: 1
inputs:
  placement:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
    default: 'env:dev'
    title: Select Placement for Deployment
    description: Target Environment
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: small
      constraints:
        - tag: '${input.placement}'
      tags:
        - key: db
          value: mysql
    networks:
      - network: '${resource.Cloud_NSX_Network_1.id}'
        tags:
```



```

- key: db
  value: mysql
attachedDisks:
- source: '${resource.Cloud_vSphere_Disk_1.id}'
Cloud_vSphere_Disk_1:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 5
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: routed
    constraints:
      - tag: 'net:od'
tags:
- key: NGINX
  value: web

```

## 6. Разверните шаблон.

В этом примере используется шаблон с именем **Development template w tags**.

## 7. Чтобы проверить теги в развертывании, откройте развертывание и перейдите на вкладку **Топология**.

а) Щелкните компьютер в топологии.

б) Разверните раздел **Общие** данного компьютера и найдите метку «Теги».

Тег имеет значение db:mysql.

в) Разверните раздел **Сеть** и найдите столбец «Теги» для сети.

Тег имеет значение db:mysql.

**Development template w tags** Create Successful ACTIONS | C

No description

Owner	fritz	Expires on	Never
Requestor	fritz	Last updated	Mar 8, 2021, 4:31:01 PM
Project	Development Project	Created on	Mar 8, 2021, 4:09:14 PM
Cloud Template	Development Template		

[HIDE SUMMARY](#)

---

**Topology** History

Search resources

Cloud\_NSX\_N...

Cloud\_vSphere...

Cloud\_vSphere...

Cloud\_vSphere\_Machine\_1 ACTIONS

**General**

Resource name Cloud\_vSphere\_Machine\_1-mcm1019-163638575175

Account / Region vCenter Server Account/wld01-DC

Status On

Address

Compute host wld01-clu01 / Development

**Tags** db:mysql

**Storage**

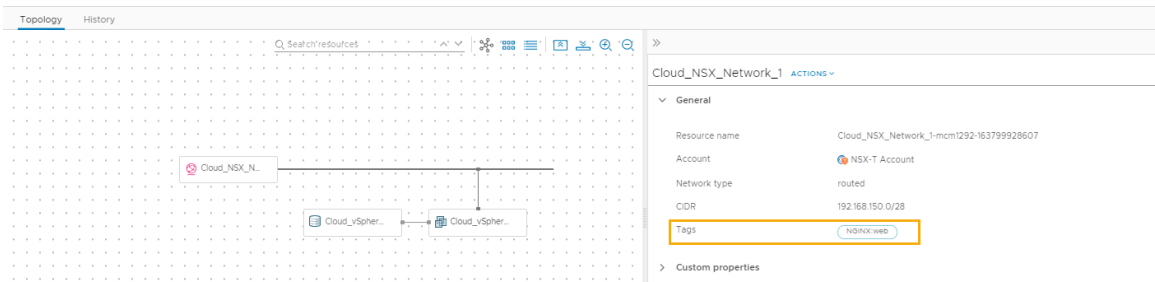
**Network**

Index	Name	Address	Assignment Type	Security Groups	Tags
0	DevProject-004		dynamic		db:mysql

**Custom properties**

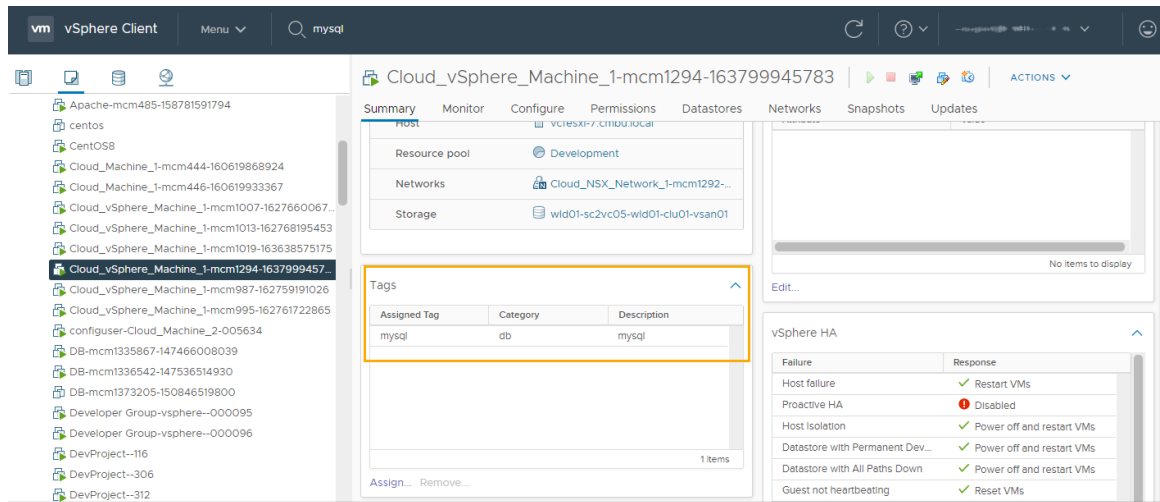
- г) Щелкните сеть в топологии и разверните раздел **Общие**, чтобы найти метку «Тег».

Тег имеет значение NGINX:web.



8. Чтобы проверить теги в vCenter Server, войдите в экземпляр vCenter Server, где развернута эта рабочая нагрузка.

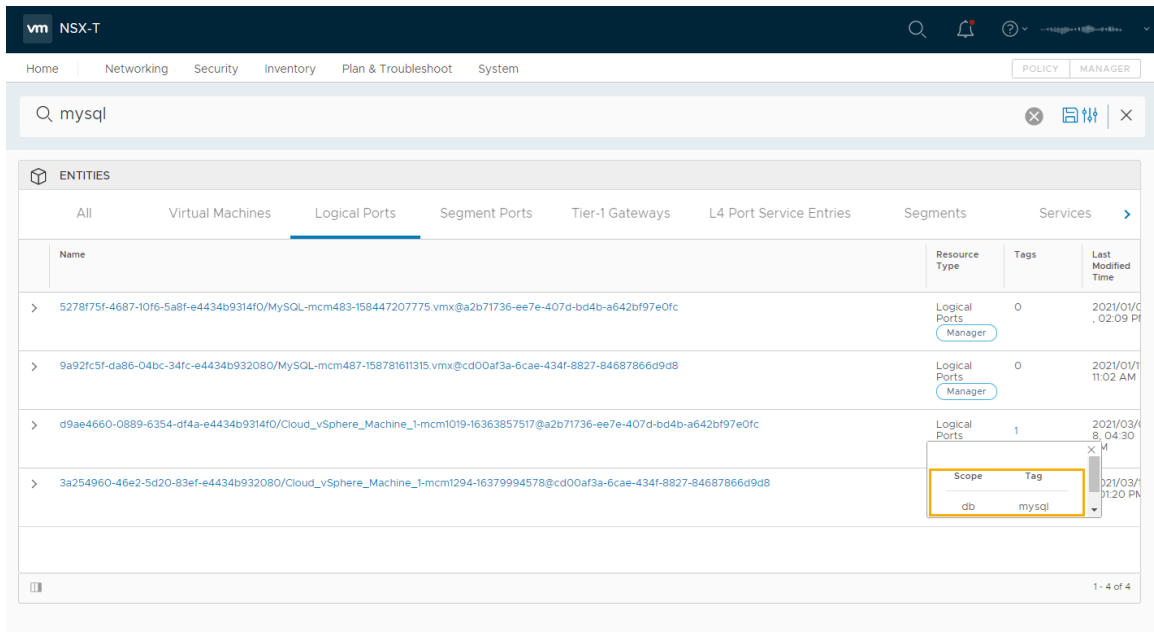
- а) Найдите виртуальную машину и панель «Теги».



9. Чтобы проверить теги в NSX-T, войдите в экземпляр NSX-T, где настроена эта сеть.

- а) Нажмите **Политика** в правом верхнем углу.
- б) Чтобы найти тег db:mysql, связанный с сетевым адаптером, найдите **mysql**.
- в) Нажмите **Логические порты** и найдите развернутый компьютер vSphere.
- г) Щелкните число в столбце «Теги».

Поля «Область» и «Тег» содержат, соответственно, значения db и mysql.

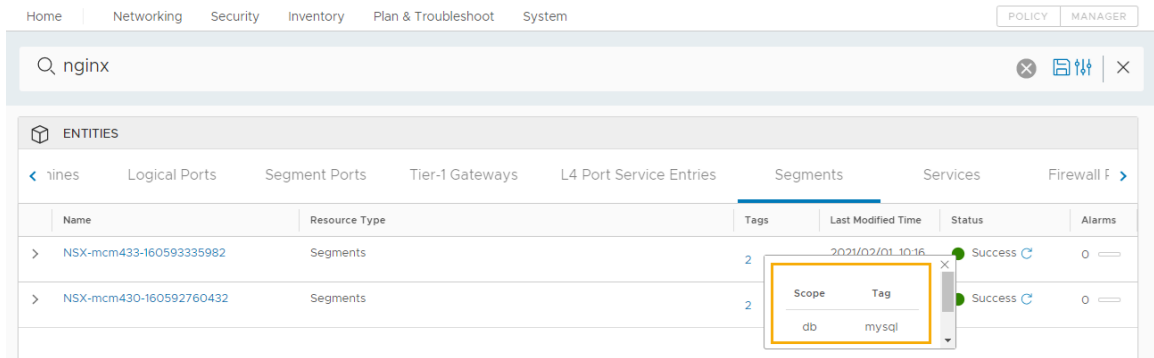


д) Чтобы найти тег NGINX:web, связанный с сегментом, найдите сеть.

В данном примере имя сети — **Cloud\_NSX\_Network\_1-mcm1292-163799928607**.

е) Найдите строку «Сегменты» и щелкните число в столбце «Теги».

Поля «Область» и «Тег» содержат, соответственно, значения NGINX и web.



## Учебник «Добавление облачного шаблона Cloud Assembly в каталог Service Broker с помощью настраиваемой формы запроса»

Во время итеративной разработки облачного шаблона или после получения финальной версии шаблона можно открыть доступ к шаблону в каталоге самообслуживания Service Broker. Для удобства пользователей можно создать настраиваемую форму запроса. Настраиваемая форма более функциональна, чем простой ввод данных шаблона.

## Начало работы

- Убедитесь, что инфраструктура поддерживает ваш шаблон. Если это не так, для начала изучите [Учебник «Настройка и тестирование инфраструктуры и развертываний vSphere в Cloud Assembly»](#) и другие учебники.
- Убедитесь, что некоторые пулы ресурсов помечены как `env:dev` и `env:prod`. Дополнительные сведения см. в разделе [Учебник «Использование тегов в Cloud Assembly для управления ресурсами vSphere»](#).
- Убедитесь в том, что у вас есть развертываемый облачный шаблон, аналогичный приведенному ниже. В этом учебнике используется следующий шаблон.

```
formatVersion: 1
inputs:
  installedOS:
    type: string
    title: Operating System
    description: Select the operating system.
    enum:
      - centos
      - ubuntu
  placement:
    type: string
    enum:
      - 'env:dev'
      - 'env:prod'
    default: 'env:dev'
    title: Select Placement for Deployment
    description: Target Environment
resources:
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: '${input.installedOS}'
      installedOS: '${input.installedOS}'
      flavor: small
      constraints:
        - tag: '${input.placement}'
      tags:
        - key: db
          value: mysql
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
          tags:
            - key: db
              value: mysql
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
  Cloud_NSX_Network_1:
```

```

type: Cloud.NSX.Network
properties:
  networkType: existing
tags:
  - key: NGINX
    value: web

```

## Шаг 1. Добавление входных данных в облачный шаблон

К существующим данным о типе ОС в ходе этого процесса добавляется информация о размере и обновляются сведения о размещении. В Service Broker можно настроить следующие три поля формы запроса.

1. В Cloud Assembly выберите **Проект > Облачный шаблон**, создайте шаблон или откройте приведенный выше.

Образец шаблона используется для иллюстрации различных параметров и включает в себя образцы значений. Адаптируйте его для своей среды.

2. Добавьте переменную размера и задайте размеры в разделе «Входные данные».

- а) В разделе Cloud\_vSphere\_Machine\_1 добавьте переменную в свойство flavor.

```
flavor: '${input.size}'
```

- б) В разделе «Входные данные» добавьте пользовательский размер для имени, чтобы пользователь мог выбрать размер развертывания. Это действие сравнимо с подбором размера одежды. Размер задается для облачных зон.

```

size:
  type: string
  title: Deployment size
  description: Select the the deployment t-shirt size.
  enum:
    - small
    - medium
    - large

```

3. Измените данные о размещении с тега на описательный термин.

Эти теги ограничений будут сопоставлены с тегами возможностей, добавленными в [Учебник «Использование тегов в Cloud Assembly для управления ресурсами vSphere»](#).

- а) В разделе «Входные данные» добавьте пользовательский элемент **размещение**, чтобы пользователь мог выбрать среду разработки или производственную среду в качестве области размещения для развертывания.

В этом примере используется атрибут `oneOf`, который позволяет при отправке строк, требуемых для развертывания, использовать метку на естественном языке. Например, теги `env:dev` и `env:prod`.

```
placement:
  type: string
  oneOf:
    - title: Development
      const: 'env:dev'
    - title: Production
      const: 'env:prod'
  default: 'env:dev'
  title: Select Deployment Placement
  description: Target Environment
```

4. Убедитесь, что код YAML полностью выглядит так, как в следующем примере.

```
formatVersion: 1
inputs:
  installedOS:
    type: string
    title: Operating system
    description: Select the operating system.
    enum:
      - centos
      - ubuntu
  placement:
    type: string
    oneOf:
      - title: Development
        const: 'env:dev'
      - title: Production
        const: 'env:prod'
    default: 'env:dev'
    title: Select Deployment Placement
    description: Target Environment
  size:
    type: string
    title: Deployment size
    description: Select the the deployment t-shirt size.
    enum:
      - small
      - medium
      - large
resources:
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: '${input.installedOS}'
```

```

installedOS: '${input.installedOS}'
flavor: '${input.size}'
constraints:
  - tag: '${input.placement}'
tags:
  - key: db
    value: mysql
networks:
  - network: '${resource.Cloud_NSX_Network_1.id}'
    tags:
      - key: db
        value: mysql
attachedDisks:
  - source: '${resource.Cloud_vSphere_Disk_1.id}'
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing
  tags:
    - key: NGINX
      value: web

```

- Щелкните **Развернуть** и убедитесь, что вторая страница запроса выглядит, как в следующем примере. После развертывания проверьте, находится ли оно в выбранной среде разработки пула производственных ресурсов.

## Шаг 2. Создание версии и публикация облачного шаблона

Теперь у вас есть шаблон для развертывания, и вы можете открыть к нему доступ в каталоге Service Broker и позволить другим пользователям развертывать его. Чтобы облачный шаблон можно было найти и добавить в каталог, необходимо опубликовать его. Для этого нужно создать новую версию, сделать моментальный снимок шаблона, а затем опубликовать его.

- Выберите **Проект > Облачный шаблон** и откройте шаблон на холсте проекта.
- Щелкните **Версия** и введите описание.

Creating Version ×

Version \*

7

Last Version: 6

Description

Placement inputs added and tested.

Change Log

Release

☒ Release this version to the catalog

This cloud template is restricted to this project in the catalog. Edit shareability in cloud template level settings.

CANCEL

CREATE

3. Поставьте флажок **Опубликовать** и нажмите **Создать**.

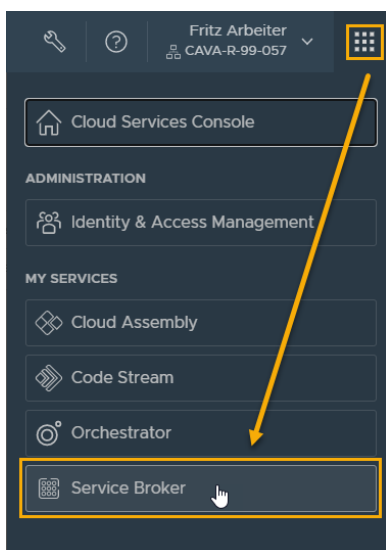
Публикация облачного шаблона не предполагает его автоматическое добавление в Service Broker, но позволяет найти его и добавить в каталог.

### Шаг 3. Добавление облачного шаблона в каталог Service Broker

Каталог «Service Broker» предназначен для предоставления облачных шаблонов другим пользователям в организации, которым не нужно уметь создавать шаблоны. Они используют каталог, чтобы их развертывать.

Прежде чем добавить шаблон в качестве элемента каталога, его необходимо импортировать в Service Broker. Импортируются только опубликованные облачные шаблоны.

1. Чтобы открыть Service Broker из Cloud Assembly, перейдите в меню приложения в правом верхнем углу.



2. Нажмите **Service Broker**.



3. Импортируйте облачный шаблон.

- а) В Service Broker выберите **Содержимое и политики > Источники содержимого**.
- б) Щелкните **Создать**, а затем выберите **VMware Cloud Templates**.
- в) Введите **Имя**.

В этом учебнике введите **Cloud Assembly DevProject**.

- г) В поле **Проект** выберите **Проект разработки**, созданный в Cloud Assembly.
- д) Щелкните **Проверить**.

Система должна сообщить, что найден по крайней мере один элемент.

- е) После проверки щелкните **Создать и импортировать**.

Cloud Assembly DevProject добавляется в список в качестве источника содержимого.

4. Откройте доступ к шаблону в каталоге.

- а) Выберите **Содержимое и политики > Общий доступ к содержимому**.
- б) В поле **Проект** выберите **Проект разработки**.
- в) Нажмите **Добавить элементы**, а затем выберите ???
- г) В диалоговом окне **Общий доступ к элементам** выберите **Cloud Assembly DevProject** и нажмите **Сохранить**.

5. Чтобы убедиться, что шаблон разработки добавлен в каталог, щелкните **Каталог**.

6. Щелкните **Запрос** на карточке шаблона разработки.

Здесь должны отображаться входные данные из облачного шаблона. Далее требуется настроить форму запроса.

**New Request**

Development Template Version 8

Project \* Development Project

Deployment Name \*

Operating system \*

Select Deployment Placement Development

Deployment size \*

## Шаг 4. Создание настраиваемой формы для шаблона

Эта настраиваемая форма нужна, чтобы пользователь мог выбрать операционную систему и размещение на основе тегов env:dev или env:prod. Теги env:dev позволяют пользователю выбрать малый или средний размер; большой размер недоступен. Если пользователь использует env:prod, большой размер выбрать также нельзя. Он скрыт от пользователя, но включается в запрос.

1. Для создания настраиваемой формы в Service Broker выберите **Содержимое и политики > Содержимое**.
2. Щелкните вертикальное многоточие слева от шаблона разработки, а затем **Настройка**.
3. Настройте входной параметр.
  - а) Выбирая поля на холсте, настройте свойства так, как указано в следующей таблице.

Название поля холста	Внешний вид	Значения	Ограничения
Операционная система	<p>Метка и тип</p> <ul style="list-style-type: none"> <li>■ Метка = операционная система</li> </ul>	<p>Параметры значений</p> <ul style="list-style-type: none"> <li>■ Параметры значений = константа</li> <li>■ Источник значения = centos   CentOS, ubuntu   Ubuntu</li> </ul> <p>В этом примере используются значения изменения всех названий операционных систем нижнего уровня на предпочтительное название ОС.</p>	
Выберите размещение развертывания		<p>Параметры значений</p> <ul style="list-style-type: none"> <li>■ Параметры значений = константа</li> <li>■ Источник значения = env:dev   Development, env:prod   Production</li> </ul>	
Размер развертывания	<p>Видимость</p> <ul style="list-style-type: none"> <li>■ Источник значения = условное значение</li> <li>■ Заданное значение = Да, если значение «Выбрать размещение развертывания» равно env:dev</li> </ul>	<p>Значение по умолчанию</p> <ul style="list-style-type: none"> <li>■ Источник значения = условное значение</li> <li>■ Заданное значение = большой, если значение «Выбрать размещение» равно env:prod</li> </ul> <p>Параметры значений</p> <ul style="list-style-type: none"> <li>■ Параметры значений = константа</li> <li>■ Источник значения = small   Small, medium   Medium</li> </ul> <p>Обратите внимание, что источник значений не содержит значение «Большой». Оно отсутствует, так как доступно только для параметра «Производство», где является единственным значением. Значение «Большой» включается в запрос развертывания автоматически.</p>	

б) Чтобы включить форму в каталоге, щелкните **Включить**.

в) Нажмите **Сохранить**.

4. Чтобы результаты были верны, хотя бы при отправке запросов для малой разработки и производства, протестируйте форму в каталоге.

При проверке результатов используйте следующие примеры.

- а) Проверьте форму запроса малой разработки, введя имя, в данном случае **Test small**, и выбрав параметры CentOS, «Разработка» и «Малая».

- б) Чтобы проверить малое развертывание разработки, выберите **Ресурсы > Развертывания** и нажмите развертывание **Test small**.

- в) На вкладке «Топология» щелкните **Cloud\_vSphere\_Machine**, а затем найдите раздел «Настраиваемые свойства» на правой панели.

Просмотрите значения **cpuCount = 2** и **flavor = small**.

- г) Проверьте форму запроса производства, введя имя, в данном случае **Test large**, и выбрав параметры CentOS и «Производство».

Помните, что вы настроили форму так, чтобы не отображать и не запрашивать пользователя выбрать размер.

**New Request**

Development Template Version 3

Project \* Development Project

Deployment Name \* Test large

Operating System \* CentOS

Select Deployment Placement Production

- д) Чтобы проверить развертывание в производственной области, выберите **Ресурсы > Развертывания** и нажмите развертывание Test large.
- е) На вкладке «Топология» щелкните Cloud\_vSphere\_Machine, а затем найдите раздел «Настраиваемые свойства» на правой панели.

Просмотрите значения cpuCount = 8 и flavor = large.

**test large** Create Successful ACTIONS |

No description

Owner: fritz  
Requestor: fritz  
Project: Development Project  
Cloud Template: Development Template, version: 6

Expires on: Never  
Last updated: May 21, 2021, 5:14:56 PM  
Created on: May 21, 2021, 4:53:05 PM

**Topology** History

Search resources

Cloud\_NSX\_N...

Cloud\_vSphere...

Cloud\_vSphere...

costCenter: DevProject

cpuCount: 8

datastoreName: wld01-sc2vc05-wld01-clu...

endpointId: d827e01c-df9e-4c80-9f1d

flavor: large

image: centos

imageId: centos7

## Шаг 5. Управление версиями облачного шаблона в каталоге

В большинстве случаев в каталоге Service Broker требуется иметь только последние облачные шаблоны. Следующая процедура поддерживает итеративный процесс разработки: вы выпускаете версию шаблона и добавляете ее в каталог. Затем вы улучшаете шаблон и хотите заменить текущую версию на новую.

На шаге 2 создается версия и выпускается шаблон. Теперь вы знакомы с процессом. На шаге 3 шаблон добавляется в каталог. Эта процедура объединяет два шага, так как вы выполняете итеративную разработку и обновляете каталог, используя последнюю версию.

В каталоге могут быть доступны несколько версий.

1. В Cloud Assembly создайте версию шаблона, которую необходимо сделать доступной в каталоге.

- а) Выберите **Проект > Облачный шаблон** и откройте шаблон на холсте проекта.
- б) Нажмите **Журнал версий**.
- в) Найдите версию, которую нужно добавить в каталог, и нажмите **Версия**.
- г) Введите **описание**, установите флажок **Версия** и нажмите **Создать**.

На этом этапе можно сохранить старую версию в каталоге. Если нужно иметь несколько версий, пропустите следующий шаг («Отмена публикации версии»).

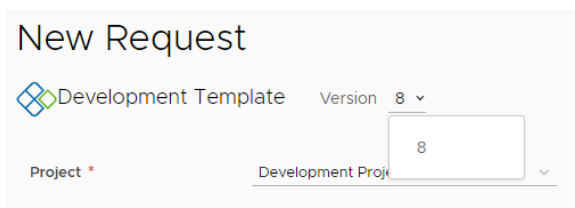
- д) Чтобы сделать доступной только одну версию шаблона в каталоге, проверьте журнал версий и нажмите **Отменить публикацию** для каждой версии, которая в каталоге не нужна.
2. Чтобы обновить каталог Service Broker, используя последнюю версию, и заменить любую старую версию, необходимо получить новую версию.

- а) В Service Broker выберите **Содержимое и политики > Источники содержимого**.
- б) Щелкните источник содержимого Cloud Assembly DevProject, который используется в этом учебнике.
- в) Щелкните **Проверить**.

Должно появиться сообщение о том, что элемент найден.

- г) Нажмите **Сохранить и импортировать**.
3. Убедитесь, что в каталоге отображаются нужные версии, или версии отсутствуют.
- а) В Service Broker нажмите **Каталог**.
  - б) Найдите нужный элемент каталога и нажмите **Запрос**.
  - в) В верхней части формы запроса нажмите **Версия** и проверьте наличие версии или версий.

На следующем снимке экрана отображается 8.



# Учебник. Внедрение ресурсов vSphere в vRealize Automation и управление ими

Администратор облачных систем, который недавно добавил новую облачную учетную запись, может осуществлять управление некоторыми из рабочих нагрузок vCenter Server с помощью Cloud Assembly и Service Broker. В этом учебнике рассматривается процесс внедрения и настройка некоторых параметров управления для существующих рабочих нагрузок vSphere.

Примеры задач управления включают добавление ресурсов в проект, создание и применение политики утверждения в Service Broker, выполнение нескольких действий по регулярному обслуживанию ресурсов для демонстрации инструментов управления жизненным циклом и запуска политики утверждения.

Этот учебник рассчитан на относительно новых пользователей Cloud Assembly, которые, тем не менее, уже настроили новую облачную учетную запись vSphere. После добавления облачной учетной записи Cloud Assembly обнаруживает текущие неуправляемые ресурсы в экземпляре vSphere.

## Начало работы

- Добавьте новую учетную запись vCenter Server. Дополнительные инструкции см. в разделе [Создание облачной учетной записи vCenter в vRealize Automation](#).
- Убедитесь, что вашей учетной записи пользователя назначены по меньшей мере роль администратора Cloud Assembly и роль администратора Service Broker. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Чтобы надлежащим образом протестировать политику утверждения с позиции одного из пользователей, убедитесь в том, что у вас есть учетная запись пользователя, для которой назначены только следующие роли пользователя. В этом учебнике имя пользователя — Sylvia.
  - Участник организации
  - Пользователь Cloud Assembly
  - Пользователь Service Broker

Дополнительные сведения о ролях см. в разделе [Что такое роли пользователей vRealize Automation](#).

## Шаг 1. Убедитесь, что Cloud Assembly обнаружены ресурсы.

Когда вы добавляете учетную запись vCenter Server, Cloud Assembly обнаруживает ресурсы в экземпляре vCenter Server. Вы можете убедиться в том, что компьютеры, которыми необходимо управлять, доступны для внедрения.

1. В Cloud Assembly выберите **Ресурсы > Ресурсы > Виртуальные машины**.
2. В таблице просмотрите параметры **Источник** и **Учетная запись/регион**.

Если для источника указано значение «обнаружен», то компьютер обнаружен в экземпляре vSphere, а не развернут vRealize Automation или уже внедрен.

В этом примере учетная запись/регион — это vCenter Account / wld01-DC.

Virtual Machines ▼ Q Search resources ?

[+ NEW VM](#)

Name	Status	Account / Region	Address	Project	Owner	Creation Time	Origin	Tags
DevProject-116	▶ On	vCenter Account / wld01-DC	N/A			Jul 26, 2021, 2:29:15 PM	Discover d	
DevProject-centos-010	▶ On	vCenter Account / wld01-DC	N/A	Onboarding Project	fritz	Jul 26, 2021, 2:29:18 PM	Deployed	db:mysql
DevProject-centos-012	▶ On	vCenter Account / wld01-DC	N/A			Jul 26, 2021, 2:29:18 PM	Discover d	
DevProject-centos-013	▶ On	vCenter Account / wld01-DC	N/A			Jul 26, 2021, 2:29:15 PM	Discover d	db:mysql
DevProject-centos-016	▶ On	vCenter Account / wld01-DC	N/A	Onboarding Project	sylvia	Jul 26, 2021, 2:29:15 PM	Deployed	db:mysql

## Шаг 2. Создайте целевой проект

Создайте проект, которому можно назначить внедренные компьютеры. Для управления ресурсами они должны быть частью проекта, который включает в себя исходную облачную зону, где эти ресурсы были изначально развернуты.

Чтобы выполнить тестирование согласно этому учебнику, требуется еще один пользователь, который не является администратором. На этом шаге вы как администратор должны добавить пользователя Sylvia в качестве участника проекта.

Дополнительные сведения о проектах см. в разделе [Глава 5 Добавление проектов Cloud Assembly и управление ими](#).

1. В Cloud Assembly выберите **Инфраструктура > Проекты > .**

2. На странице «Проекты» щелкните **Новый проект**.

3. Введите **Имя** проекта.

В этом учебнике используется имя проекта **Onboarding Project**.

4. Перейдите на вкладку **Пользователи**.

а) Выберите **Добавить пользователей** и добавьте по крайней мере одного пользователя в качестве участника проекта.

В этом учебнике будет добавлен пользователь Sylvia.

б) Нажмите кнопку **Добавить**.

5. Выберите **Подготовка**.

а) Выберите **Добавить зону**.

б) Выберите **Облачная зона**.

в) Выберите учетную запись/регион, которые были определены на шаге 1.

В этом учебнике это vCenter Account/wld01-DC.



**New Project**

Summary Users **Provisioning** Kubernetes Provisioning

Zones

Specify the zones that can be used when users provision deployments in this project. ⓘ

[+ ADD ZONE](#) [X REMOVE](#)

<input type="checkbox"/>	Name	Status	Description	Priority	Instances	Memory Limit (MB)	CPU Limit	Storage Limit (GB)	Capability Tags
<input type="checkbox"/>	vCenter Account / wld01-DC	--		0	Unlimited	Unlimited	Unlimited	Unlimited	

1 - 1 of 1 zones

Specify the placement policy that will be applied when selecting a cloud zone for provisioning.

Placement policy **DEFAULT** ⓘ

г) Нажмите кнопку **Добавить**.

6. Щелкните **Создать**.

### Шаг 3. Создайте и запустите план внедрения

Администратор облачных систем может внедрять обнаруженные компьютеры из экземпляра vSphere, что позволяет управлять ресурсами с помощью действий по регулярному обслуживанию.

Дополнительные сведения о планах внедрения см. в разделе [Что такое планы внедрения в Cloud Assembly](#).

1. В Cloud Assembly выберите **Инфраструктура > Внедрение**, а затем **Новый план внедрения**.
2. Введите сведения о внедрении.

Настройка	Пример значения
Имя плана	wld01-DC Onboarding Plan
Учетная запись облачной службы	Учетная запись vCenter
Проект по умолчанию	Onboarding Project

3. Щелкните **Создать**.
4. Добавьте компьютеры, которые необходимо внедрить.

Не запускайте план внедрения, пока не будут выполнены все описанные ниже шаги.

- а) Щелкните **Компьютеры** и выберите **Добавить компьютеры**.
- б) Выберите компьютеры, которые необходимо включить в план, а затем нажмите **ОК**.

В этом учебнике выбрано только два компьютера.

- в) В диалоговом окне «Создание развертываний» выберите **Создать развертывания плана для каждого компьютера**, а затем нажмите **Создать**.

Выберите этот параметр, если необходимо сделать компьютеры отдельными развертываниями, чтобы ими можно было управлять как отдельными ресурсами.

- г) Выбранные компьютеры будут добавлены в список.

**wld01-DC Onboarding Plan**

Summary **Machines** Deployments

Machines listed here are onboarded when the plan runs.

[ADD MACHINES](#) [REMOVE](#) [Filter...](#) [i](#) [C](#)

<input type="checkbox"/>	Name	Status	Power	Address	Deployment	Custom properties	Tags
<input type="checkbox"/> >	DevProject-centos-013	<a href="#">Pending</a>	<a href="#">On</a>		Deployment-5e3ac...	Inherited	<a href="#">db.mysql</a>
<input type="checkbox"/> >	DevProject-centos-204	<a href="#">Pending</a>	<a href="#">On</a>		Deployment-50507...	Inherited	<a href="#">db.mysql</a>

## 5. Переименуйте развертывания.

- Щелкните **Развертывания** на странице внедрения.
- Чтобы изменить сгенерированное имя развертывания, выберите развертывание и нажмите **Переименовать**.
- Введите новое имя и нажмите **Сохранить**.

Например, Onboarded machine 1.

- Повторите при необходимости.

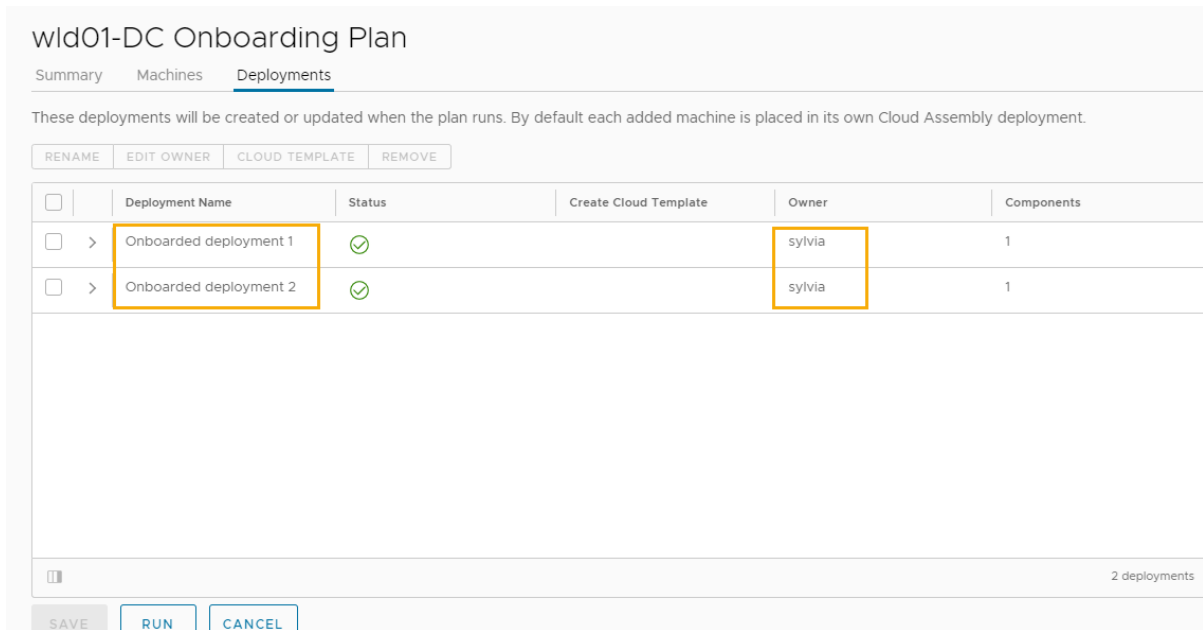
## 6. Назначьте владельца развертываниям.

Если вы не назначите владельца, то сами станете владельцем. Владелец должен быть участником целевого проекта.

В этом учебнике все развертывания назначаются одному и тому же владельцу. При необходимости развертывания можно назначать разным владельцам.

- Выберите все развертывания и нажмите **Изменить владельца**.
- Выберите владельца и нажмите **Сохранить**.

Просмотрите изменения имени и владельца развертывания в таблице.

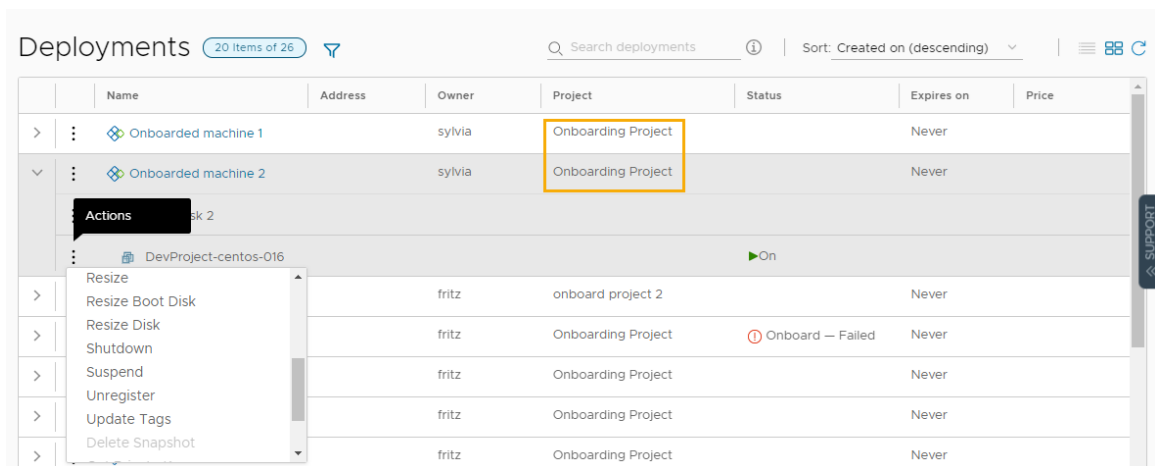


## 7. Щелкните Запустить.

После запуска плана внедрения изменение имени и назначение владельцев невозможно. При добавлении дополнительных компьютеров в план можно изменить имя и владельца.

8. Просмотрите ресурсы, которые были внедрены в качестве развертываний.

- Выберите **Ресурсы > Развертывания**.
- Поиск развертывания можно выполнить по его имени, проекту или владельцу.



Теперь, после внедрения компьютеров в vRealize Automation, ими можно управлять.

## Шаг 4. Измените размер развертывания

Выполните этот шаг в качестве администратора облачных систем и ознакомьтесь с принципами действий по регулярному обслуживанию. Изменения, которые можно внести в развертывания, называются действиями по регулярному обслуживанию. Использование действий по регулярному обслуживанию является первым шагом в управлении ресурсами.

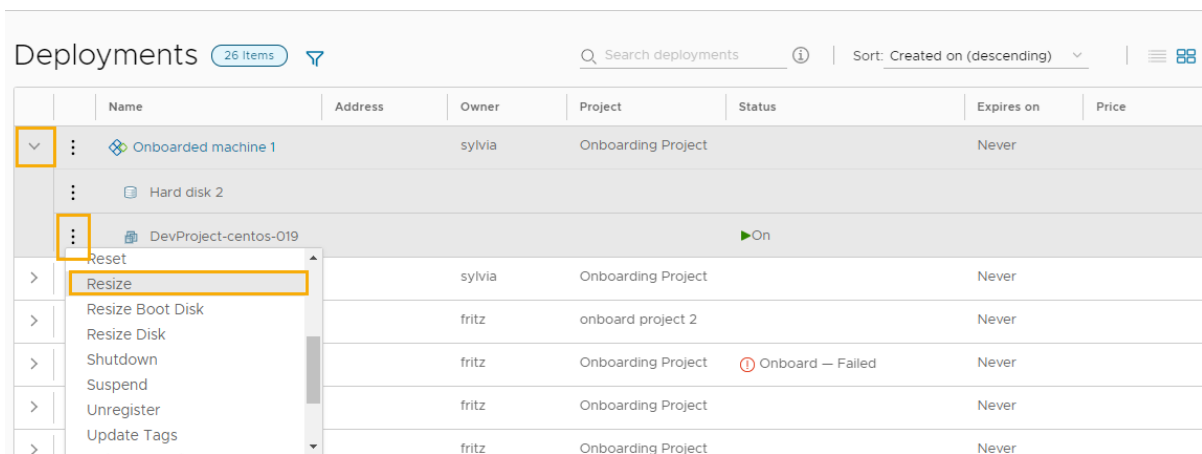
В этом учебнике предполагается, что количество ЦП на компьютере слишком велико и следует его сократить. Процедура предполагает выполнение действия изменения размера на включенном компьютере vSphere. Кроме того, предполагается отсутствие политик регулярного обслуживания, согласно которым пользователю запрещено выполнять это действие.

Доступные действия зависят от типа и состояния ресурса, а также политик регулярного обслуживания. Дополнительные сведения о действиях по регулярному обслуживанию см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

1. В Cloud Assembly выберите **Ресурсы > Развертывания**, а затем найдите внедренные развертывания.

Можно использовать параметры поиска или фильтрации.

2. Разверните развертывание, используя стрелку слева, а затем щелкните вертикальное многоточие на имени компьютера и выберите **Изменить размер**.

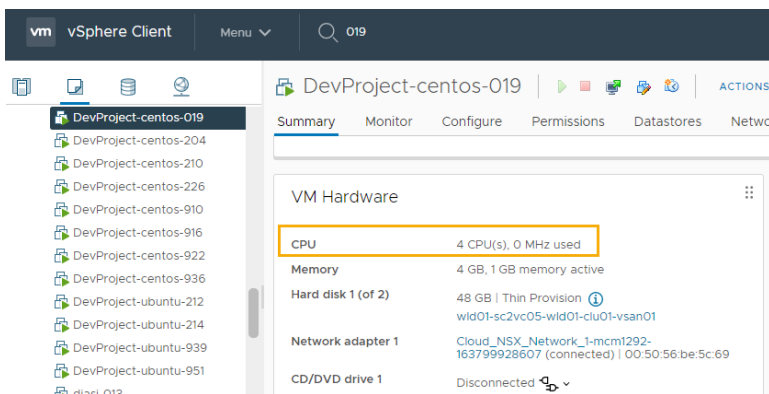


3. В диалоговом окне **Изменить размер** уменьшите количество ЦП до **4** и нажмите **Отправить**.

Указанное значение является примером. Измените количество ЦП так, как требуется в вашей среде.

Это действие выполняется на компьютере.

4. Чтобы убедиться, что количество ЦП изменено, откройте развертывание и проверьте настраиваемое свойство `cpuCount` для компьютера.
5. Кроме того, количество ЦП можно проверить в vCenter Server.



## Шаг 5. Примените политики утверждения

Администратор облачных систем может применить управление в vRealize Automation, чтобы ограничить действия пользователей или разрешить выполнение этих действий только после их утверждения. В этом учебнике показано, как применять политики утверждения к действию изменения размера, чтобы пользователи не могли перенастроить компьютер, что может привести к серьезным последствиям, без утверждения этого действия вами или другим администратором.

Политики создаются в Service Broker. Однако они применяются к соответствующим запросам в Cloud Assembly и Service Broker.

Утверждающий должен обработать запрос на утверждение в Service Broker.

1. В Service Broker выберите **Содержимое и политики > Политики > Определения**, а затем щелкните **Создать политику**.
2. Щелкните **Политика утверждения**.
3. Настройте политику утверждения.

**Resize Approval Policy** DELETE

Approval policies control who must agree to a deployment or day 2 action before the request is provisioned. ⓘ

Type: Approval

Name \*: Resize Approval Policy

Description:

Scope \*:   
☐ Organization / Multiple Projects   
Apply the policy to all or a selection of projects in this organization. To target multiple projects, select project based criteria. ⓘ   
☒ Project   
Apply the policy to a single project in this organization. ⓘ   
 Onboarding Project

Criteria:   ⓘ

Approval type \*: ☒ User based ☐ Role based ⓘ

Approver mode \*: ☒ Any ☐ All ⓘ

Approvers \*:   ⓘ

<input type="checkbox"/>	Name	Email	Type
<input type="checkbox"/>	Fritz Arbeiter	fritz	User
<input type="checkbox"/>	1 user		

Auto expiry decision \*: Reject ⓘ

Auto expiry trigger \*: 1 days ⓘ

Actions \*:   ⓘ

<input type="checkbox"/>	Actions
<input type="checkbox"/>	Cloud.vSphere.Machine.Resize

В следующей таблице приведены примеры значений, используемых при создании политики.

Настройка	Пример значения
Имя	Политика утверждения изменения размера
Scope	Выберите <b>Проект</b> , а затем <b>Onboarding Project</b> . Политика утверждения запускается, когда пользователь, который является участником проекта, выполняет изменение размера в рамках действий по регулярному обслуживанию.
Тип утверждения	На основе пользователей Это значение позволяет указать утверждающих.
Режим утверждающего	Любой При наличии нескольких утверждающих запрос на утверждение может быть обработан минимум одним утверждающим.
Утверждающие	Добавьте себя в качестве утверждающего.
Решение об автоматическом истечении срока действия	Отклонить Отклонение непросмотренного запроса снижает риск того, что компьютер станет непригодным к использованию или будет иметь слишком много ресурсов.
Триггер автоматического истечения срока действия	1
Действия	Выберите действие изменения размера, которое запускает политику утверждения. 1 Введите <b>machine.resize</b> в поле поиска. 2 Щелкните <b>Выбрать несколько</b> в раскрывающемся списке результатов поиска. 3 Выберите <b>Cloud.vSphere.Machine.Resize</b> .  В этом учебнике, подготовленном на основе vSphere, необходимо выбрать действие vSphere.Machine. Если необходимо, чтобы политика действий применялась к другим типам ресурсов, можно добавить другие действия Machine.Resize.

## Шаг 6. Сделайте запрос на изменение размера от имени пользователя

На этом шаге вы должны выполнить вход в Service Broker в качестве участника организации и пользователя Service Broker и сделать запрос на изменение размера в рамках действий по регулярному обслуживанию. Этот запрос обеспечивает создание запроса на утверждение. Пользователь может выполнить те же шаги в Cloud Assembly.

На следующем шаге необходимо будет выполнить вход в качестве пользователя, который назначен утверждающим на шаге 5, и утвердить запрос.

1. Выполните вход в Service Broker в качестве пользователя.

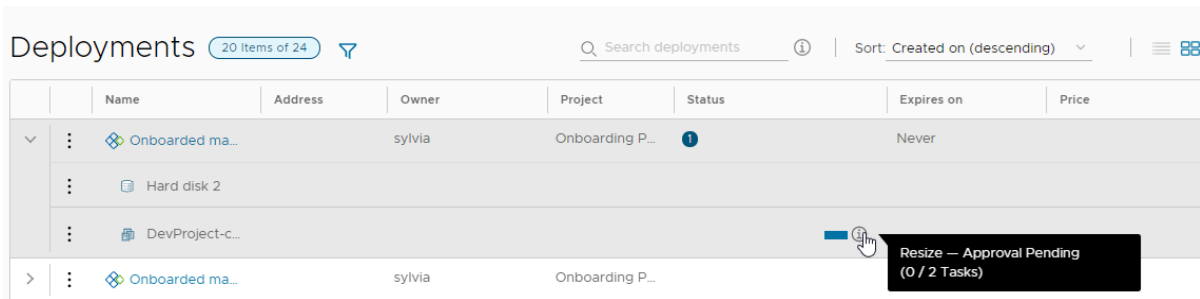
В этом учебнике пользователь — Sylvia.

2. Выберите **Ресурсы > Развертывания** и найдите Onboarded machine 1.

Это развертывание, для которого на шаге 4 было выполнено действие по изменению размера: количество ЦП на компьютере было изменено с 8 до 4. Если вы использовали другое значение, измените количество ЦП в соответствии с потребностями тестирования.

3. Запустите действие **Изменить размер** на компьютере, увеличив количество ЦП до 6.
4. Обратите внимание, что запрос ожидает утверждения.

Чтобы увидеть состояние ожидания, наведите указатель мыши на значок информации в таблице или откройте развертывание и просмотрите вкладку **История**.



	Name	Address	Owner	Project	Status	Expires on	Price
▼	Onboarded ma...		sylvia	Onboarding P...	1	Never	
	Hard disk 2						
	DevProject-c...						
>	Onboarded ma...		sylvia	Onboarding P...			

5. Пользователь Sylvia не может применить запрашиваемое изменение, пока оно не будет утверждено.
6. Выполните выход из Service Broker в качестве пользователя.

На шаге 7 вы выполните вход в качестве назначенного утверждающего и обработаете запрос.

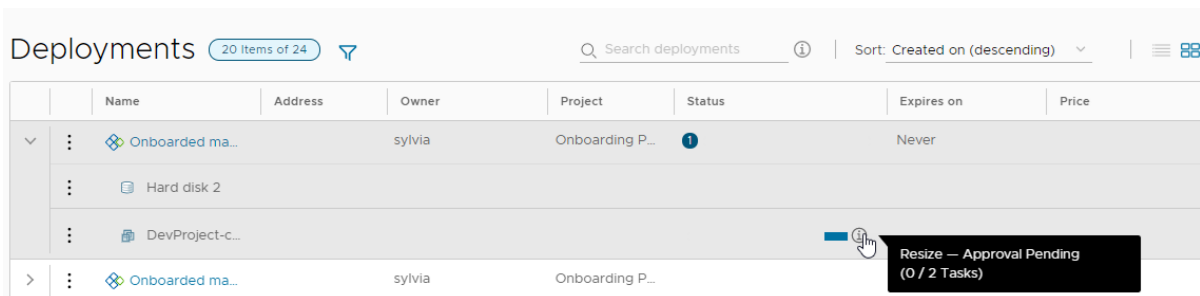
## Шаг 7. Обработайте запрос на утверждение

Если запрос требует утверждения, утверждающий получает электронное сообщение. В этом учебнике не требуется ждать сообщения. Вместо этого можно сразу перейти к обработке запросов на утверждение на вкладке Service Broker Утверждения.

1. Выполните вход в Service Broker в качестве пользователя, назначенного утверждающим на шаге 5.
- В этом учебнике утверждающим является пользователь Fritz.

2. Выберите **Ресурсы > Развертывания** и найдите Onboarded machine 1.

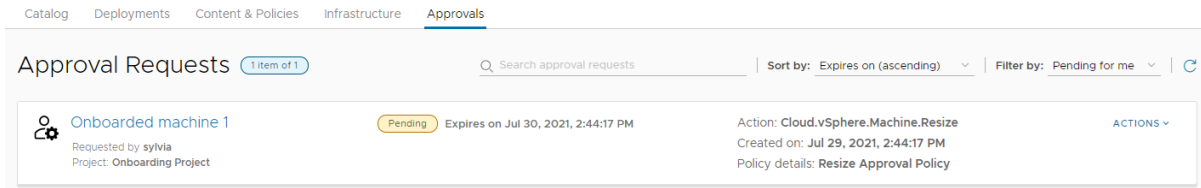
В таблице отображается такое же состояние, как отображалось для пользователя Sylvia.



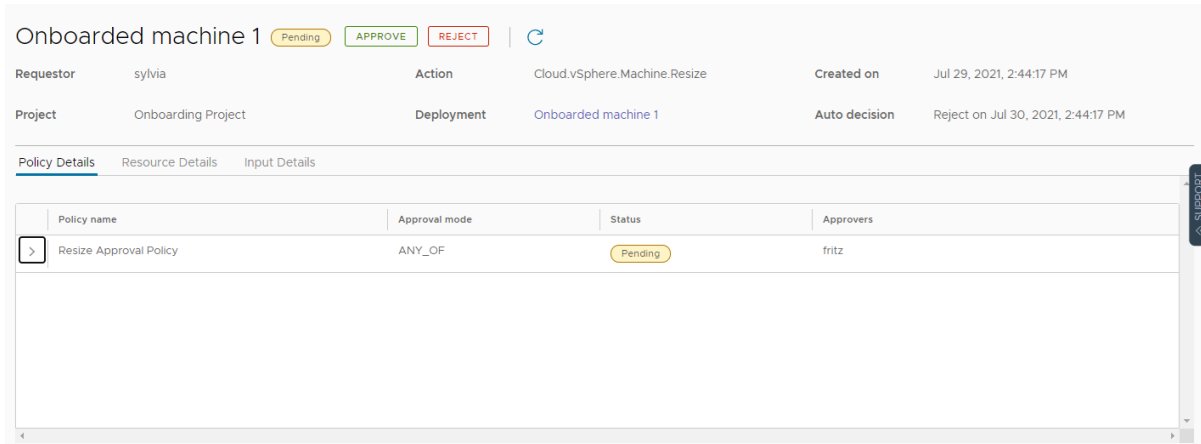
	Name	Address	Owner	Project	Status	Expires on	Price
▼	Onboarded ma...		sylvia	Onboarding P...	1	Never	
	Hard disk 2						
	DevProject-c...						
>	Onboarded ma...		sylvia	Onboarding P...			

3. Перейдите на вкладку **Утверждения**.

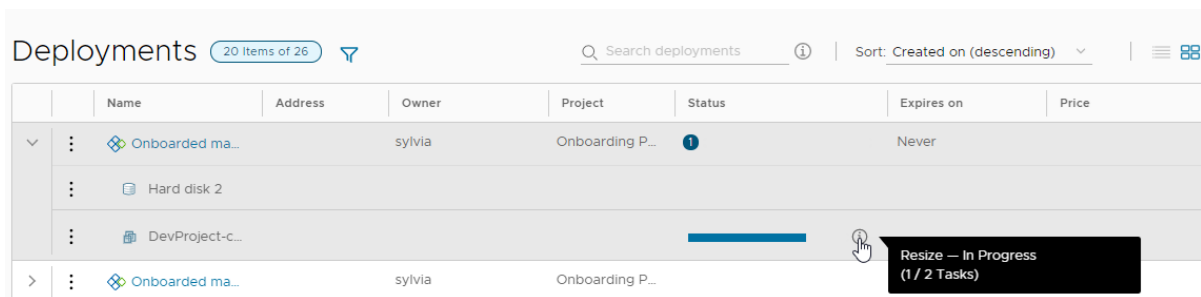
Обратите внимание, что имеется ожидающий запрос на утверждение.



4. Для просмотра сведений о запросе щелкните имя развертывания.



5. Щелкните **Утвердить**, при необходимости введите комментарий и щелкните **Утвердить**.
6. Вернитесь на страницу **Развертывания**, и вы увидите, что действие пользователя Sylvia по изменению размера теперь в процессе выполнения.



7. После выполнения действия изменения размера можно проверить количество ЦП в сведениях о развертывании и в vSphere Client.

В этом учебнике был рассмотрен процесс внедрения компьютеров в vRealize Automation, что позволяет управлять жизненным циклом ресурса.

## Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»

В этом учебнике по Cloud Assembly показано, как можно выполнять развертывание в схеме с несколькими облачными средами. Один и тот же облачный шаблон развертывается с несколькими поставщиками (в данном случае — AWS и Microsoft Azure).



В качестве объекта используется сайт WordPress. Ознакомьтесь с последовательностью настройки, чтобы понять весь процесс создания.

Обратите внимание, что приведенные далее имена и значения используются только для примера. Их нельзя перенести в вашу среду один к одному.

Используйте собственные варианты значений в зависимости от потребностей облачной инфраструктуры и развертывания.

## Часть 1. Настройка примера инфраструктуры Cloud Assembly

Сначала настройте ресурсы, в которых позже технические специалисты Cloud Assembly смогут разрабатывать, тестировать приложение и размещать его в производственной среде.

Инфраструктура включает в себя облачные целевые объекты, а также определения для доступных компьютеров, сетей и хранилища, которые будут нужны сайту WordPress.

### Необходимые условия

Войдите в Cloud Assembly как администратор Cloud Assembly.

#### 1. Добавление облачных учетных записей

На этом этапе администратор облачных систем добавляет две учетные записи облачной службы. Пример проекта предполагает выполнение разработки и тестирования в AWS и выпуск в производственную среду в Azure.

1. Выберите **Инфраструктура > Подключения > Облачные учетные записи**.
2. Щелкните **Добавить облачную учетную запись**, выберите Amazon Web Services и введите значения.

Настройка	Пример значения
Идентификатор ключа доступа	R5SDR3PXVV2ZW8B7YNSM
Секретный ключ доступа	SZXAINXU4UHNAQ1E156S
Имя	OurCo-AWS
Описание	WordPress

Помните, что все значения являются примерными. Содержимое вашей учетной записи будет разным.

3. Чтобы проверить учетные данные, щелкните **Проверить**.
4. Нажмите кнопку **Добавить**.
5. Измените добавленную учетную запись (**Конфигурация**) и разрешите подготовку для областей "us-east-1" и "us-west-2".
6. Щелкните **Добавить облачную учетную запись**, выберите Microsoft Azure и введите значения.

Настройка	Пример значения
Идентификатор подписки	ef2avpf-dfdv-zxlugui1i-g4h0-i8ep2jwp4c9arbfе
Идентификатор арендатора	dso9wv3-4zgc-5nrcy5h3m-4skf-nnovp40wfxsro22r
Идентификатор клиентского приложения	bg224oq-3ptp-mbhi6aa05-q511-uflyjr2sttyik6bs
Секретный ключ клиентского приложения	7uqxi57-0wtn-kymgf9wcj-t2l7-e52e4nu5fig4pmd
Имя	OurCo-Azure
Описание	WordPress

- Чтобы проверить учетные данные, щелкните **Проверить**.
- Нажмите кнопку **Добавить**.
- Измените добавленную учетную запись (**Конфигурация**) и разрешите подготовку для области "East US".

## 2. Добавление облачных зон

На этом этапе в примере администратор облачных систем добавляет три облачные зоны, по одной зоне для разработки, тестирования и производства.

- Выберите **Инфраструктура > Настроить > Облачные зоны**.
- Щелкните **Создать облачную зону** и введите значения для среды разработки.

Параметр облачной зоны	Пример значения
Учетная запись/область	OurCo-AWS/us-east-1
Имя	OurCo-AWS-US-East
Описание	WordPress
Политика размещения	По умолчанию
Теги возможностей	env:dev

Помните, что все значения являются примерными. Настройки зоны будут разными.

- Щелкните **Вычислительные ресурсы** и убедитесь, что они содержат ожидаемые зоны.
- Щелкните **Создать**.
- Повторите процесс два раза, используя значения для тестовой и производственной сред.

Параметр облачной зоны	Пример значения
Учетная запись/область	OurCo-AWS/us-west-2
Имя	OurCo-AWS-US-West
Описание	WordPress

Параметр облачной зоны	Пример значения
Политика размещения	По умолчанию
Теги возможностей	env:test

Параметр облачной зоны	Пример значения
Учетная запись/область	OurCo-Azure/East US
Имя	OurCo-Azure-East-US
Описание	WordPress
Политика размещения	По умолчанию
Теги возможностей	env:prod

### 3. Добавление сопоставлений конфигураций ресурсов

На этом этапе в примере администратор облачных систем добавляет сопоставления конфигураций ресурсов, чтобы учитывать потребности в ресурсах, которые могут варьироваться в зависимости от развертывания.

Сопоставление конфигураций ресурсов учитывает развертывания компьютеров разного размера и неформально называется «определение размера футболки».

1. Выберите **Инфраструктура > Настроить > Сопоставления конфигураций ресурсов**. Для каждой облачной зоны необходимо разрешить малые, средние и крупные конфигурации ресурсов.
2. Щелкните **Создать сопоставление конфигураций ресурсов** и введите значения для облачной среды разработки.

Настройка	Пример значения
Имя конфигурации ресурсов	small
Учетная запись/область	OurCo-AWS/us-east-1
Значение	t2.micro
Учетная запись/область	OurCo-AWS/us-west-2
Значение	t2.micro
Учетная запись/область	OurCo-Azure/East US
Значение	Standard_A0

Помните, что все значения являются примерными. Конфигурации ресурсов будут разными.

3. Щелкните **Создать**.
4. Повторите процесс два раза, используя значения для средней и крупной конфигураций ресурсов.

Настройка	Пример значения
Имя конфигурации ресурсов	средний
Учетная запись/область	OurCo-AWS/us-east-1
Значение	t2.medium
Учетная запись/область	OurCo-AWS/us-west-2
Значение	t2.medium
Учетная запись/область	OurCo-Azure/East US
Значение	Standard_A3

Настройка	Пример значения
Имя конфигурации ресурсов	масштабный
Учетная запись/область	OurCo-AWS/us-east-1
Значение	t2.large
Учетная запись/область	OurCo-AWS/us-west-2
Значение	t2.large
Учетная запись/область	OurCo-Azure/East US
Значение	Standard_A7

#### 4. Добавление сопоставлений образов

На этом этапе в примере администратор облачных систем добавляет сопоставление образов для Ubuntu, узел для сервера WordPress и сервер базы данных MySQL.

Спланируйте операционную систему путем добавления сопоставлений образов. Для каждой облачной зоны требуется сопоставление образов Ubuntu.

1. Выберите **Инфраструктура > Настроить > Сопоставления образов**.
2. Щелкните **Создать сопоставление образов** и введите значения для серверов Ubuntu.

Настройка	Пример значения
Имя образа	ubuntu
Учетная запись/область	OurCo-AWS/us-east-1
Значение	ubuntu-16.04-server-cloudimg-amd64
Учетная запись/область	OurCo-AWS/us-west-2
Значение	ubuntu-16.04-server-cloudimg-amd64
Учетная запись/область	OurCo-Azure/East US
Значение	azul-zulu-ubuntu-1604-923eng

Помните, что все значения являются примерными. Образы будут разными.

3. Щелкните **Создать**.

## 5. Добавление профилей сети

На этом этапе в примере администратор облачных систем добавляет профиль сети в каждую облачную зону.

В каждом профиле администратор добавляет сеть для компьютеров WordPress и вторую сеть, которая будет находиться на другой стороне потенциальной подсистемы балансировки нагрузки. Через вторую сеть, возможно, будут подключаться пользователи.

1. Выберите **Инфраструктура > Настроить > Профили сети**.
2. Щелкните **Создать профиль сети** и введите значения для облачной зоны разработки.

Параметр профиля сети	Пример значения
Учетная запись/область	OurCo-AWS/us-east-1
Имя	devnets
Описание	WordPress

3. Щелкните **Сети**, а затем **Добавить сеть**.
4. Выберите wpnet, appnet-public и щелкните **Добавить**.

Помните, что все значения являются примерными. Имена сетей будут разными.

5. Щелкните **Создать**.

В этом примере Wordpress не нужно задавать параметры политики сети или безопасности сети.

6. Повторите процесс два раза, чтобы создать профиль сети для примеров тестовых и производственных облачных зон Wordpress. В каждом случае добавьте сети wpnet и appnet-public.

Параметр профиля сети	Пример значения
Учетная запись/область	OurCo-AWS/us-west-2
Имя	testnets
Описание	WordPress

Параметр профиля сети	Значение
Учетная запись/область	OurCo-Azure/East US
Имя	prodnets
Описание	WordPress

## 6. Добавление профилей хранилища

На этом этапе в примере администратор облачных систем добавляет профиль хранилища в каждую облачную зону.

Администратор размещает быстродействующее хранилище в производственной зоне, а стандартное хранилище — в среде разработки и тестирования.

1. Выберите **Инфраструктура > Настроить > Профили хранилища**.
2. Щелкните **Создать профиль хранилища** и введите значения для облачной зоны разработки.

После выбора учетной записи/области появляются дополнительные поля.

Параметр профиля хранилища	Пример значения
Учетная запись/область	OurCo-AWS/us-east-1
Имя	OurCo-AWS-US-East-Disk
Описание	WordPress
Тип устройства	EBS
Тип тома	SSD-диск общего назначения
Теги возможностей	storage:general

Помните, что все значения являются примерными.

3. Щелкните **Создать**.
4. Повторите процесс, чтобы создать профиль для облачной зоны тестирования.

Параметр профиля хранилища	Пример значения
Учетная запись/область	OurCo-AWS/us-west-2
Имя	OurCo-AWS-US-West-Disk
Описание	WordPress
Тип устройства	EBS
Тип тома	SSD-диск общего назначения
Теги возможностей	storage:general

5. Повторите процесс, чтобы создать профиль для производственной облачной зоны, которая имеет другие параметры, так как это зона Azure.

Параметр профиля хранилища	Пример значения
Учетная запись/область	OurCo-Azure/East US
Имя	OurCo-Azure-East-US-Disk
Описание	WordPress
Тип хранилища	Управляемые диски
Тип диска	Premium LRS

Параметр профиля хранилища	Пример значения
Кэширование диска с ОС	Только для чтения
Кэширование диска с данными	Только для чтения
Теги возможностей	storage:fast

## Следующие шаги

Создайте проект для идентификации пользователей и определения параметров подготовки. См. раздел [Часть 2. Создание примера проекта Cloud Assembly](#).

## Часть 2. Создание примера проекта Cloud Assembly

В образце проекта Cloud Assembly назначаются пользователи, которые могут выполнять подготовку, и настраиваются ограничения действий при подготовке.

В проектах определяются параметры пользователей и подготовки.

- Пользователи и уровень разрешений для их ролей
- Приоритет для развертываний в ходе их подготовки для облачной зоны
- Максимальное количество экземпляров развертывания для каждой облачной зоны

### Процедура

1. Выберите **Инфраструктура > Администрирование > Проекты**.
2. Щелкните **Создать проект** и введите имя WordPress.
3. Щелкните **Пользователи**, а затем **Добавить пользователей**.
4. Добавьте адреса электронной почты и роли для пользователей.

Для успешного добавления пользователя администратору VMware Cloud Services должен быть разрешен доступ к Cloud Assembly для данного пользователя.

Помните, что используемые здесь адреса являются примерными.

- chris.ladd@ourco.com, участник
- kerry.mott@ourco.com, участник
- pat.tubb@ourco.com, администратор

5. Щелкните **Подготовка**, а затем **Добавить облачную зону**.

6. Добавьте облачные зоны, в которых пользователи могут выполнять развертывание.

Параметр облачной зоны проекта	Пример значения
Облачная зона	OurCo-AWS-US-East
Приоритет подготовки	1
Макс. количество экземпляров	5
Облачная зона	OurCo-AWS-US-West
Приоритет подготовки	1
Макс. количество экземпляров	5
Облачная зона	OurCo-Azure-East-US
Приоритет подготовки	0
Макс. количество экземпляров	1

7. Щелкните **Создать**.
8. Выберите **Инфраструктура > Настроить > Облачные зоны** и откройте ранее созданную зону.
9. Щелкните **Проекты** и убедитесь, что для проекта WordPress разрешено выполнение подготовки в зоне.
10. Проверьте другие созданные зоны.

#### Следующие шаги

Создайте базовый облачный шаблон.

## Часть 3. Проектирование и развертывание примера шаблона Cloud Assembly

Затем нужно определить пример приложения (сайт WordPress) в форме универсального облачного шаблона. Шаблон можно развертывать в разных облачных средах без изменения его проекта.

Пример включает в себя сервер приложений WordPress, сервер базы данных MySQL и ресурсы поддержки. Шаблон изначально содержит всего несколько ресурсов и впоследствии расширяется благодаря их изменению и добавлению дополнительных ресурсов.

Это значения из [Часть 1. Настройка примера инфраструктуры Cloud Assembly](#). Инфраструктура была настроена администратором облачных систем:

- Две учетные записи облачной службы: AWS и Azure.
- Три среды облачных зон:
  - Разработка—OurCo-AWS-US-East
  - Тестирование—OurCo-AWS-US-West
  - Производство—OurCo-Azure-East-US
- Сопоставления конфигураций ресурсов с малыми, средними и крупными вычислительными ресурсами для каждой зоны.
- Сопоставления образов для Ubuntu, настроенные в каждой зоне.



- Профили сети с внутренними и внешними подсетями для каждой зоны.
- Хранилище, в котором необходимо провести развертывание, общее хранилище для зоны разработки и тестирования, а также быстрое хранилище для производственной зоны.
- Пример проекта включает в себя все три среды облачных зон и пользователей, которые могут создавать проекты.

#### Необходимые условия

Для выполнения этой процедуры сначала ознакомьтесь со значениями своей собственной инфраструктуры. В этом примере AWS применяется для разработки и тестирования, а Azure — для производственных процессов. При создании собственного облачного шаблона введите собственные значения (обычно они предоставляются администратором облачных систем).

#### Процедура

##### 1. Создание базового облачного шаблона

В этом примере проекта Cloud Assembly работа начинается с облачного шаблона, который содержит только минимальные ресурсы WordPress, например только один сервер приложений.

##### 2. Тестирование базового облачного шаблона

Во время проектирования часто можно создать простой облачный шаблон, а затем развертывать и тестировать его по мере его расширения. В этом примере показаны некоторые выполняемые тесты, встроенные в Cloud Assembly.

##### 3. Расширение облачного шаблона

После создания и тестирования базового шаблона Cloud Assembly примера приложения его можно расширить до многоуровневого приложения, которое может быть развернуто в среде разработки, тестирования и, наконец, в производственной среде.

#### Создание базового облачного шаблона

В этом примере проекта Cloud Assembly работа начинается с облачного шаблона, который содержит только минимальные ресурсы WordPress, например только один сервер приложений.

Cloud Assembly — это средство, работающее по модели "инфраструктура как код". Чтобы начать работу, перетащите ресурсы на холст проекта. Затем введите сведения с помощью редактора кода, расположенного справа от холста.

Редактор кода позволяет непосредственно вводить, вырезать и вставлять код. Если код редактировать неудобно, можно выбрать ресурс на холсте, перейти на вкладку **Свойства** редактора кода и ввести там значения. Введенные значения отображаются в коде так, как если бы они были введены напрямую.

#### Процедура

1. Выберите **Проектирование > Облачные шаблоны** и нажмите **Создать из > Пустой холст**.
2. Назовите облачный шаблон **Wordpress-BP**.
3. Выберите проект **WordPress** и нажмите **Создать**.

- Из ресурсов слева на странице проектирования облачных шаблонов перетащите на холст два независимых от облачной среды компьютера.

Компьютеры служат в качестве сервера приложения WordPress и сервера базы данных MySQL (DBTier).

- В правой части измените код YAML компьютера: добавьте имена, образы, конфигурации ресурсов и теги ограничений.

```
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: ubuntu
      flavor: small
      constraints:
        - tag: env:dev
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu
      flavor: small
      constraints:
        - tag: env:dev
```

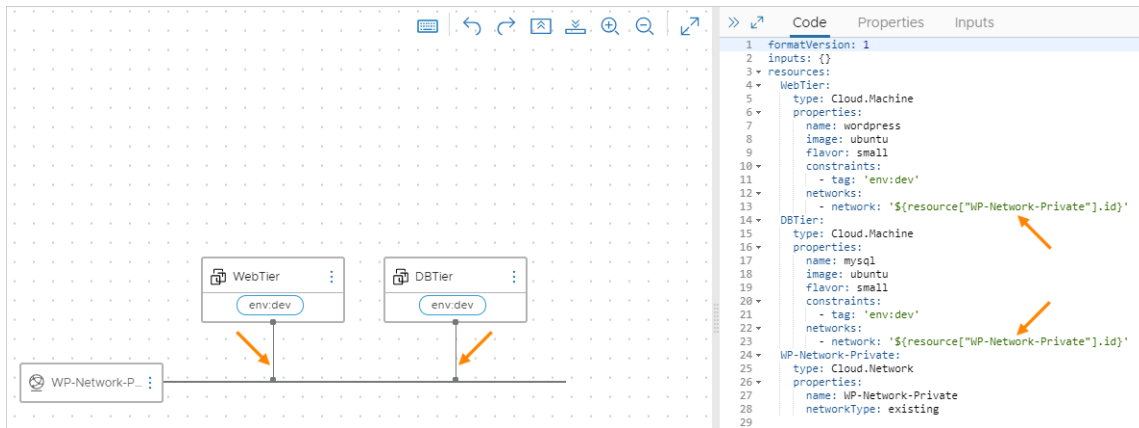
- Перетащите независимую от облачной среды сеть на холст и измените ее код.

```
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
```

- Подключите компьютеры к сети:

На холсте наведите указатель мыши на блок сети, щелкните и удерживайте пузырек, где линия касается блока, перетащите указатель в блок компьютера и отпустите.

При создании линий подключения обратите внимание, что код сети автоматически добавляется к компьютерам в редакторе.



## 8. Добавьте запрос на ввод данных пользователем.

В некоторых местах в образце инфраструктуры было настроено несколько вариантов. Например:

- Среды облачных зон для разработки, тестирования и производства
- Сопоставления конфигураций ресурсов для малых, средних и крупных компьютеров

Нужный параметр можно задать непосредственно в облачном шаблоне, но лучше дать возможность пользователю настроить параметр во время развертывания шаблона. Добавление запроса на ввод данных пользователем позволяет создать один шаблон, который можно развернуть по-разному, а не создавать большое количество однозначно запрограммированных шаблонов.

- a) Создайте в коде раздел `inputs`, чтобы пользователи могли выбрать размер компьютера и целевую среду во время развертывания. Укажите выбираемые значения:

```
inputs:
  env:
    type: string
    enum:
      - env:dev
      - env:prod
      - env:test
    default: env:dev
    title: Environment
    description: Target Environment
  size:
    type: string
    enum:
      - small
      - medium
      - large
    description: Size of Nodes
    title: Tier Machine Size
```

- б) В разделе `resources` кода добавьте код `${input.input-name}` для запроса на выбор значений пользователем:

```
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: ubuntu
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
  WP-Network-Private:
```

```
type: Cloud.Network
properties:
  name: WP-Network-Private
  networkType: existing
```

9. Наконец, оптимизируйте код `WebTier` и `DBTier`, используя следующие примеры. Для кода `WP-Network-Private` дополнительные изменения не требуются.

Обратите внимание, что расширенные возможности включают в себя доступ с вводом данных для входа к серверу базы данных и сценариям инициализации `cloudConfig` на этапе развертывания.

Компонент	Пример
Дополнительные входные данные DBTier	<pre> username:   type: string   minLength: 4   maxLength: 20   pattern: '[a-z]+'   title: Database Username   description: Database Username userpassword:   type: string   pattern: '[a-z0-9A-Z@#]+\$'   encrypted: true   title: Database Password   description: Database Password </pre>
Ресурс DBTier	<pre> DBTier:   type: Cloud.Machine   properties:     name: mysql     image: ubuntu     flavor: '\${input.size}'     constraints:       - tag: '\${input.env}'     networks:       - network: '\${resource["WP-Network-Private"].id}'         assignPublicIpAddress: true     remoteAccess:       authentication: usernamePassword       username: '\${input.username}'       password: '\${input.userpassword}'     cloudConfig:         #cloud-config       repo_update: true       repo_upgrade: all       packages:         - mysql-server       runcmd:         - sed -e '/bind-address/ s/^#*\/#/' -i /etc/mysql/mysql.conf.d/ mysql.cnf         - service mysql restart         - mysql -e "CREATE USER 'root'@'%' IDENTIFIED BY 'mysqlpassword';"         - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';"         - mysql -e "FLUSH PRIVILEGES;"     attachedDisks: [] </pre>
Ресурс WebTier	<pre> WebTier:   type: Cloud.Machine   properties:     name: wordpress     image: ubuntu     flavor: '\${input.size}'     constraints:       - tag: '\${input.env}'     networks:       - network: '\${resource["WP-Network-Private"].id}'         assignPublicIpAddress: true     cloudConfig:   </pre>

Компонент	Пример
	<pre> #cloud-config repo_update: true repo_upgrade: all packages: - apache2 - php - php-mysql - libapache2-mod-php - mysql-client - gcc - make - autoconf - libc-dev - pkg-config - libmcrypt-dev - php-pear - php-dev runcmd: - mkdir -p /var/www/html/mywordpresssite &amp;&amp; cd /var/www/html &amp;&amp; wget https://wordpress.org/latest.tar.gz &amp;&amp; tar -xzf /var/www/html/ latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1 - i=0; while [ \$i -le 10 ]; do mysql --connect-timeout=3 -h \$ {DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" &amp;&amp; break    sleep 15; i=\$((i+1)); done - mysql -u root -pmysqlpassword -h \${DBTier.networks[0].address} -e "create database wordpress_blog;" - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/ html/mywordpresssite/wp-config.php - pecl channel-update pecl.php.net - pecl update-channels - pecl install mcrypt - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME', 'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed -i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER', 'root' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed -i -e s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD', 'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed -i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '\${DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp- config.php - sed -i '950i extension=mcrypt.so' /etc/php/7.4/apache2/php.ini - service apache2 reload </pre>

### Пример. Пример кода готового базового облачного шаблона

```

formatVersion: 1
inputs:
  env:
    type: string
    enum:
      - env:dev
      - env:prod
      - env:test
    default: env:dev
    title: Environment
    description: Target Environment
  size:
    type: string

```

```

enum:
  - small
  - medium
  - large
description: Size of Nodes
title: Tier Machine Size
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username
userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#]+$'
  encrypted: true
  title: Database Password
  description: Database Password
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: ubuntu
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all
      packages:
        - apache2
        - php
        - php-mysql
        - libapache2-mod-php
        - mysql-client
        - gcc
        - make
        - autoconf
        - libc-dev
        - pkg-config
        - libmcrypt-dev
        - php-pear
        - php-dev
      runcmd:
        - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget
        https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
        mywordpresssite --strip-components 1
        - i=0; while [ $i -le 10 ]; do mysql --connect-timeout=3 -h $
        {DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;

```



```

i=$((i+1)); done
    - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/mywordpresssite/
wp-config.php
    - pecl channel-update pecl.php.net
    - pecl update-channels
    - pecl install mcrypt
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER',
'root' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i
-e s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD',
'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '$
${DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-config.php
    - sed -i '950i extension=mcrypt.so' /etc/php/7.4/apache2/php.ini
    - service apache2 reload
DBTier:
  type: Cloud.Machine
  properties:
    name: mysql
    image: ubuntu
    flavor: '${input.size}'
    constraints:
      - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    remoteAccess:
      authentication: usernamePassword
      username: '${input.username}'
      password: '${input.userpassword}'
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all
      packages:
        - mysql-server
      runcmd:
        - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
        - service mysql restart
        - mysql -e "CREATE USER 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
        - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%;'"
        - mysql -e "FLUSH PRIVILEGES;"
    attachedDisks: []
  WP-Network-Private:
    type: Cloud.Network
    properties:
      name: WP-Network-Private
      networkType: existing

```

## Следующие шаги

Для тестирования облачного шаблона проверьте синтаксис и разверните его.

## Тестирование базового облачного шаблона

Во время проектирования часто можно создать простой облачный шаблон, а затем развертывать и тестировать его по мере его расширения. В этом примере показаны некоторые выполняемые тесты, встроенные в Cloud Assembly.

Чтобы убедиться, что развертывание работает надлежащим образом, облачный шаблон можно протестировать и развернуть несколько раз. Постепенно добавляйте новые ресурсы, а затем снова выполняйте тестирование и развертывание.

### Необходимые условия

Создайте базовый облачный шаблон. См. раздел [Создание базового облачного шаблона](#).

### Процедура

1. Щелкните **Облачные шаблоны** и откройте облачный шаблон WordPress-BP.  
Базовый облачный шаблон появляется на холсте проекта и в редакторе кода.
2. Чтобы проверить синтаксис шаблона, размещение и основной период действия, щелкните **Тест** в левом нижнем углу.
3. Введите входные значения и щелкните **Тест**.

Testing Basic

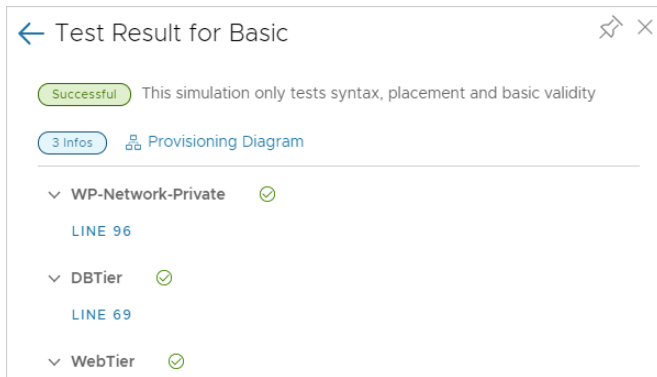
Environment  ⓘ

Tier Machine Size  ⓘ

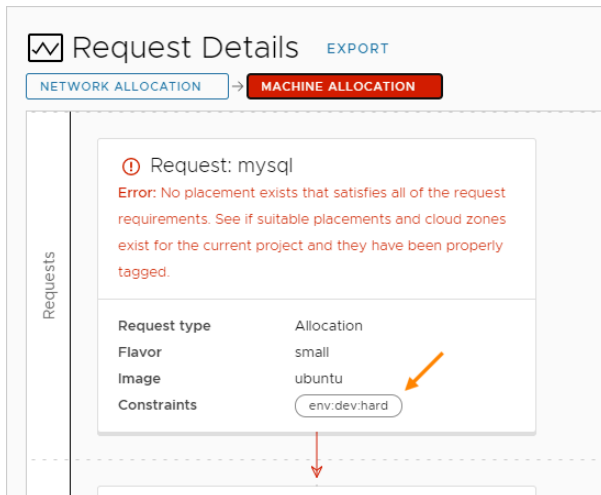
Database Username

Database Password

Тест — это только моделирование. На самом деле виртуальные машины или другие ресурсы не развертываются.







Тест содержит ссылку на **схему подготовки**, на которой можно проверить смоделированный процесс развертывания и увидеть все, что произошло. В ходе моделирования могут возникать проблемы, например отсутствие каких бы то ни было заданных возможностей ресурсов, соответствующих жестким ограничениям в облачном шаблоне. В приведенном ниже примере ошибки облачная зона тега возможности `env:dev` не найдена в заданной инфраструктуре.



Успешное завершение моделирования не гарантирует, что шаблон будет развернут без ошибок.

4. После моделирования шаблона щелкните **Развернуть** в левом нижнем углу.
5. Выберите **Создать новое развертывание**.
6. Назначьте развертыванию имя **WordPress for OurCo** и нажмите **Далее**.
7. Введите входные значения и щелкните **Развернуть**.
8. Чтобы убедиться, что шаблон развернут успешно, выберите **Ресурсы > Развертывания**.

В случае сбоя развертывания щелкните его имя и перейдите на вкладку **Журнал**, чтобы ознакомиться с сообщениями, которые могут помочь в ходе устранения проблем.

Timestamp	Status	Resource type	Resource name
Sep 8, 2020, 1...	CREATE_IN_PROGRESS	 Cloud.Machine	WebTier
Sep 8, 2020, 1...	CREATE_FINISHED	 Cloud.Machine	DBTier
Sep 8, 2020, 1...	CREATE_IN_PROGRESS	 Cloud.Machine	DBTier
Sep 8, 2020, 1...	CREATE_FINISHED	 Cloud.Network	WP-Network-Private
Sep 8, 2020, 1...	CREATE_IN_PROGRESS	 Cloud.Network	WP-Network-Private

В некоторых записях журнала в правой части может быть ссылка **Схема предоставления**. Данная схема похожа на смоделированную схему, где можно проверить блок-схему для точек принятия решения Cloud Assembly в процессе предоставления.

Чтобы отобразить другие блок-схемы, выберите пункты **Инфраструктура > Действия > Запросы**.

9. Чтобы убедиться, что приложение работает, откройте начальную страницу WordPress в браузере.

а) Подождите, пока серверы WordPress не будут полностью созданы и инициализированы.

В зависимости от среды, инициализация может выполняться в течение 30 минут или более.

б) Чтобы найти полное доменное имя или IP-адрес сайта, выберите **Ресурсы > Развертывания > Топология**.

в) На холсте щелкните WebTier и найдите IP-адрес на панели справа.

г) Введите IP-адрес в рамках полного URL-адреса на начальной странице WordPress.

В этом примере используется следующий полный URL-адрес:

`http://{IP-адрес}/mywordpresssite`

или

`http://{IP-адрес}/mywordpresssite/wp-admin/install.php`

10. После проверки WordPress в браузере, если над приложением необходимо еще поработать, внесите изменения в шаблон и повторно выполните развертывание с помощью параметра **Обновить существующее развертывание**.

11. Предусмотрите возможность создания версий облачного шаблона. Если изменение приводит к сбою развертывания, можно вернуться к работающей версии.

а) На странице проектирования облачных шаблонов щелкните **Версия**.

б) На странице «Создание версии» введите **WP-1.0**.

Не вводите пробелы в именах версий.

в) Щелкните **Создать**.

Чтобы просмотреть версию или вернуться к ней на странице «Проектирование» перейдите на вкладку **Журнал версий**.

12. Теперь можно выполнить базовое развертывание. Попробуйте осуществить первое улучшение во время развертывания, увеличив ресурсы ЦП и памяти на серверах приложения и базы данных.

Обновите до размера «средний узел» для них обоих. Используя один и тот же шаблон, выберите **средний** во время развертывания, повторите развертывание и снова проверьте приложение.

#### Следующие шаги

Расширьте облачный шаблон в приложении, пригодном для производства, путем добавления в него дополнительных ресурсов.

### Расширение облачного шаблона

После создания и тестирования базового шаблона Cloud Assembly примера приложения его можно расширить до многоуровневого приложения, которое может быть развернуто в среде разработки, тестирования и, наконец, в производственной среде.

Чтобы расширить облачный шаблон, добавьте следующие усовершенствования.

- Возможность кластеризации серверов приложений для увеличения емкости
- Общедоступная сеть и подсистема балансировки нагрузки перед серверами приложений
- Сервер резервного копирования с хранилищем архива

#### Необходимые условия

Создайте базовый облачный шаблон и проверьте его. См. разделы [Создание базового облачного шаблона](#) и [Тестирование базового облачного шаблона](#).

#### Процедура

1. Щелкните **Облачные шаблоны** и откройте облачный шаблон WordPress-BP.

На холсте проекта и в редакторе кода появляется базовый шаблон.

2. Внесите дополнения и изменения, используя пример кода и рисунок.

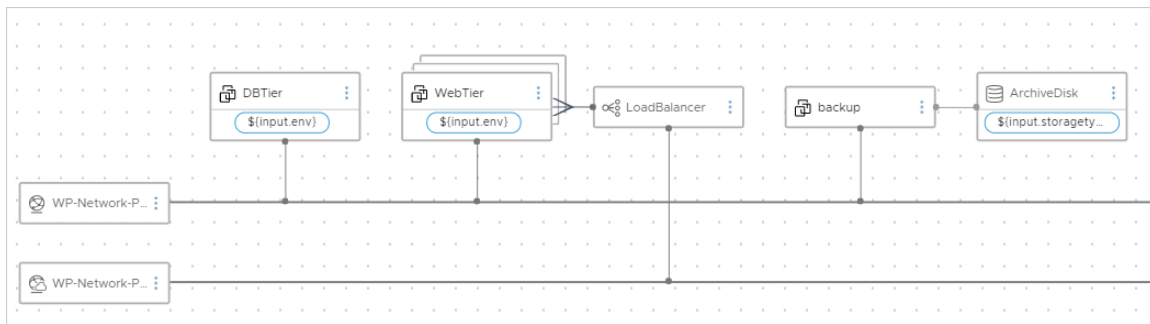
Для перетаскивания новых ресурсов на холст, например подсистемы балансировки нагрузки, можно использовать графический пользовательский интерфейс, а завершить настройку можно в редакторе кода.

- а) Добавьте запрос на ввод данных `count`, чтобы сделать сервер приложения WordPress в кластере.
- б) Добавьте независимую от облачной среды подсистему балансировки нагрузки.
- в) Подключите подсистему балансировки нагрузки к кластеру серверов приложений WordPress.
- г) Добавьте независимый от облачной среды компьютер резервного копирования.
- д) Подключите компьютер резервного копирования к частной или внутренней сети.
- е) Добавьте независимую от облачной среды общедоступную или внешнюю сеть.
- ж) Подключите подсистему балансировки нагрузки к общедоступной сети.

- з) Добавьте том независимого от облачной среды хранилища, чтобы использовать его в качестве диска с архивом.
  - и) Подключите диск с архивом к компьютеру резервного копирования.
  - к) Добавьте запрос на ввод данных по скорости работы диска с архивом.
3. Выполните развертывание, тестирование и внесите изменения точно так же, как это было сделано для базового облачного шаблона.

Можно обновить существующие развертывания или даже развернуть новые экземпляры, чтобы было можно сравнивать развертывания.

Целью является достижение надежно функционирующего воспроизводимого шаблона, который можно использовать для производственных развертываний.



#### Пример. Пример кода расширенного облачного шаблона

```
formatVersion: 1
inputs:
  env:
    type: string
    enum:
      - env:dev
      - env:prod
      - env:test
    default: env:dev
    title: Environment
    description: Target Environment
  size:
    type: string
    enum:
      - small
      - medium
      - large
    description: Size of Nodes
    title: Tier Machine Size
  username:
    type: string
    minLength: 4
```

```

    maxLength: 20
    pattern: '[a-z]+'
    title: Database Username
    description: Database Username
  userpassword:
    type: string
    pattern: '[a-z0-9A-Z@#\$]+'
    encrypted: true
    title: Database Password
    description: Database Password
  count:
    type: integer
    default: 2
    maximum: 5
    minimum: 2
    title: WordPress Cluster Size
    description: WordPress Cluster Size (Number of Nodes)
  storagetype:
    type: string
    enum:
      - storage:general
      - storage:fast
    description: Archive Storage Disk Type
    title: Archive Disk Type
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: ubuntu
      flavor: '${input.size}'
      count: '${input.count}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
          assignPublicIpAddress: true
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all
      packages:
        - apache2
        - php
        - php-mysql
        - libapache2-mod-php
        - mysql-client
        - gcc
        - make
        - autoconf
        - libc-dev
        - pkg-config
        - libmcrypt-dev
        - php-pear
        - php-dev

```

```

    runcmd:
      - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget
https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
      - i=0; while [ $i -le 10 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
      - mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
      - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/mywordpresssite/
wp-config.php
      - pecl channel-update pecl.php.net
      - pecl update-channels
      - pecl install mcrypt
      - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER',
'root' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i
-e s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD',
'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '$
{DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-config.php
      - sed -i '950i extension=mcrypt.so' /etc/php/7.4/apache2/php.ini
      - service apache2 reload
DBTier:
  type: Cloud.Machine
  properties:
    name: mysql
    image: ubuntu
    flavor: '${input.size}'
    constraints:
      - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    remoteAccess:
      authentication: usernamePassword
      username: '${input.username}'
      password: '${input.userpassword}'
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all
      packages:
        - mysql-server
      runcmd:
        - sed -e '/bind-address/ s/^#*\/#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
        - service mysql restart
        - mysql -e "CREATE USER 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
        - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';"
        - mysql -e "FLUSH PRIVILEGES;"
    attachedDisks: []
  LoadBalancer:
    type: Cloud.LoadBalancer
    properties:

```



```

name: myapp-lb
network: '${resource["WP-Network-Public"].id}'
instances:
  - '${WebTier.id}'
routes:
  - protocol: HTTP
    port: '80'
    instanceProtocol: HTTP
    instancePort: '80'
    healthCheckConfiguration:
      protocol: HTTP
      port: '80'
      urlPath: /mywordpresssite/wp-admin/install.php
      intervalSeconds: 6
      timeoutSeconds: 5
      unhealthyThreshold: 2
      healthyThreshold: 2
    internetFacing: true
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
WP-Network-Public:
  type: Cloud.Network
  properties:
    name: WP-Network-Public
    networkType: public
backup:
  type: Cloud.Machine
  properties:
    name: backup
    flavor: '${input.size}'
    image: ubuntu
    networks:
      - network: '${resource["WP-Network-Private"].id}'
    attachedDisks:
      - source: '${resource.ArchiveDisk.id}'
ArchiveDisk:
  type: Cloud.Volume
  properties:
    name: ArchiveDisk
    capacityGb: 5
    constraints:
      - tag: '${input.storagetype}'

```

### Следующие шаги

Определите собственную инфраструктуру и создайте собственные облачные шаблоны.

См. разделы [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#) и [Глава 6 Проектирование развертываний Cloud Assembly](#).

## Учебник «Настройка VMware Cloud on AWS для vRealize Automation»

В данном учебнике по vRealize Automation показан процесс определения инфраструктуры ресурсов и настройки параметров облачного шаблона для развертывания в среде VMware Cloud on AWS.

Данная процедура предполагает, что администратор облака уже настроил центр обработки данных программно-определяемого ЦОД VMware Cloud on AWS организации, как это описано в разделе *Развертывание программно-определяемого ЦОД и управление им* в [документации по началу работы с VMware Cloud on AWS](#).

Ознакомьтесь с последовательностью настройки, чтобы освоить процесс конфигурации среды для VMware Cloud on AWS. Обратите внимание, что используемые здесь значения приведены только в качестве примера. Подумайте о том, где можно внести необходимые изменения и как экстраполировать примеры значений на актуальные параметры, чтобы облачная инфраструктура и развертывание соответствовали вашим требованиям.



Дополнительные сведения см. в видео [Настройка VMware Cloud on AWS для Cloud Assembly](#).

### Процедура

#### 1. Настройка базового рабочего процесса VMware Cloud on AWS в vRealize Automation

В этом примере использования рассматривается процесс определения инфраструктуры ресурсов и параметров соответствующего облачного шаблона для развертывания в среде VMware Cloud on AWS.

#### 2. Настройка изолированной сети в рамках рабочего процесса VMware Cloud on AWS в службе vRealize Automation

В этой процедуре изолированная сеть добавляется в развертывание VMware Cloud on AWS в службе vRealize Automation.

## Настройка базового рабочего процесса VMware Cloud on AWS в vRealize Automation

В этом примере использования рассматривается процесс определения инфраструктуры ресурсов и параметров соответствующего облачного шаблона для развертывания в среде VMware Cloud on AWS.

В этой процедуре выполняется настройка инфраструктуры, которая поддерживает развертывание облачных шаблонов на основе ресурсов в существующей среде VMware Cloud on AWS.

### Необходимые условия

- Прежде чем создать и настроить облачную учетную запись VMware Cloud on AWS в службе Cloud Assembly, необходимо убедиться, что вы являетесь участником организации в существующей среде программно-определяемого ЦОД VMware Cloud on AWS. Сведения о настройке службы VMware Cloud on AWS см. в [документации по VMware Cloud on AWS](#).

- Чтобы обеспечить необходимое подключение между существующим программно-определяемым ЦОД VMware Cloud on AWS в vCenter и облачной учетной записью VMware Cloud on AWS в Cloud Assembly, необходимо установить сетевое подключение и добавить правила брандмауэра с помощью VPN или аналогичного сетевого решения. См. раздел [Подготовка программно-определяемого ЦОД VMware Cloud on AWS к подключению с помощью облачных учетных записей VMware Cloud on AWS в vRealize Automation](#).

## Процедура

1. [Подготовка программно-определяемого ЦОД VMware Cloud on AWS к подключению с помощью облачных учетных записей VMware Cloud on AWS в vRealize Automation](#)

При использовании облачных учетных записей VMware Cloud on AWS в среде vRealize Automation необходимо создать сетевое подключение и настроить правила для обмена данными между программно-определяемым ЦОД в vCenter и облачными учетными записями VMware Cloud on AWS в vRealize Automation.

2. [Создание облачной учетной записи VMware Cloud on AWS в службе vRealize Automation в рамках образца рабочего процесса](#)

На данном шаге выполняется создание облачной учетной записи VMware Cloud on AWS в службе vRealize Automation.

3. [Создание облачной зоны для развертываний VMware Cloud on AWS в службе vRealize Automation](#)

На данном шаге выполняется создание облачной зоны, определяющей вычислительный ресурс, к которому пользователь CloudAdmin может получать доступ при работе с VMware Cloud on AWS в службе vRealize Automation.

4. [Настройка профилей сети и хранилища для развертываний VMware Cloud on AWS в службе vRealize Automation](#)

На данном этапе выполняется настройка профиля сети и профиля хранилища для указания ресурсов, доступных пользователю VMware Cloud on AWS CloudAdmin в службе vRealize Automation.

5. [Создание проекта для поддержки развертываний VMware Cloud on AWS в службе vRealize Automation](#)

На данном шаге выполняется определение проекта vRealize Automation, который может использоваться для управления доступом к ресурсам со стороны развертываний VMware Cloud on AWS.

6. [Определение ресурса компьютера vCenter в проекте облачного шаблона для поддержки развертывания VMware Cloud on AWS в службе vRealize Automation](#)

На данном шаге ресурс компьютера vCenter необходимо перетащить на холст проекта и добавить настройки для развертывания VMware Cloud on AWS в vRealize Automation.

## Подготовка программно-определяемого ЦОД VMware Cloud on AWS к подключению с помощью облачных учетных записей VMware Cloud on AWS в vRealize Automation

При использовании облачных учетных записей VMware Cloud on AWS в среде vRealize Automation необходимо создать сетевое подключение и настроить правила для обмена данными между программно-определяемым ЦОД в vCenter и облачными учетными записями VMware Cloud on AWS в vRealize Automation.

Настройте необходимые подключения и правила для обмена данными с программно-определяемым ЦОД.

Чтобы обеспечивать необходимое подключение между существующим программно-определяемым ЦОД узла VMware Cloud on AWS в vCenter и облачной учетной записью VMware Cloud on AWS в vRealize Automation, необходимо установить сетевое подключение между этими двумя элементами с помощью VPN или аналогичной сети.

1. Настройте VPN-подключение через общедоступный Интернет или AWS Direct Connect.

Дополнительные сведения о настройке VPN-подключения к локальному центру обработки данных, а также о настройке AWS Direct Connect для VMware Cloud on AWS см. в документе *Сети и система безопасности в VMware Cloud on AWS* в [документации по VMware Cloud on AWS](#).

2. Убедитесь, что полное доменное имя (FQDN) vCenter Server может быть определено по частному IP-адресу в сети управления.

Дополнительные сведения о настройке адреса преобразования полного доменного имени в vCenter Server см. в документе *Сети и система безопасности в VMware Cloud on AWS* в [документации по VMware Cloud on AWS](#).

3. Настройте необходимые правила брандмауэра.

Чтобы обеспечить обмен данными, необходимо настроить правила брандмауэра для шлюза управления в консоли VMware Cloud on AWS в программно-определяемом ЦОД. Правила должны быть указаны в разделе правил брандмауэра для **шлюза управления**. Создайте следующие правила брандмауэра, используя параметры на вкладке **Сетевые подключения и безопасность** консоли программно-определяемого ЦОД.

- Разрешите входящий сетевой трафик в ESXi для служб HTTPS (TCP 443) только на обнаруженный IP-адрес устройства или сервера vRealize Automation или на виртуальный IP-адрес подсистемы балансировки нагрузки vRealize Automation.
- Разрешите входящий сетевой трафик в vCenter для служб ICMP (все ICMP), единого входа (TCP 7444) и HTTPS (TCP 443) только на обнаруженный IP-адрес устройства или сервера vRealize Automation или на виртуальный IP-адрес подсистемы балансировки нагрузки vRealize Automation.
- Разрешите входящий сетевой трафик в NSX-T Manager для служб HTTPS (TCP 443) только на обнаруженный IP-адрес устройства или сервера vRealize Automation или на виртуальный IP-адрес подсистемы балансировки нагрузки vRealize Automation.

Обязательные правила брандмауэра приведены в таблице ниже.

Таблица 2-2. Обязательные правила брандмауэра для шлюза управления

Имя	Источник	Назначение	Служба
vCenter	Блок CIDR локального центра обработки данных	vCenter	Any (весь трафик)
Проверка связи для vCenter	Любой	vCenter	ICMP (все ICMP)
NSX Manager	Блок CIDR локального центра обработки данных	NSX Manager	Any (весь трафик)
Проверка связи между локальной системой и ESXi	Блок CIDR локального центра обработки данных	Только управление ESXi	ICMP (все ICMP)
Локальная система — консоль удаленного доступа ESXi и предоставление	Блок CIDR локального центра обработки данных	Только управление ESXi	TCP 902
Локальная система — ВМ программно-определяемого ЦОД	Блок CIDR локального центра обработки данных	Блок CIDR логической сети программно-определяемого ЦОД	Any (весь трафик)
ВМ программно-определяемого ЦОД — локальная система	Блок CIDR логической сети программно-определяемого ЦОД	Блок CIDR локального центра обработки данных	Any (весь трафик)

Дополнительные сведения см. в разделе *Сети и система безопасности в VMware Cloud on AWS* и в *руководстве по эксплуатации VMware Cloud on AWS* в [документации по VMware Cloud on AWS](#).

После настройки обязательных правил доступа к шлюзу и правил брандмауэра можно продолжить процесс создания облачной учетной записи VMware Cloud on AWS. См. раздел [Создание облачной учетной записи VMware Cloud on AWS в службе vRealize Automation](#) в рамках образца рабочего процесса.

## Создание облачной учетной записи VMware Cloud on AWS в службе vRealize Automation в рамках образца рабочего процесса

На данном шаге выполняется создание облачной учетной записи VMware Cloud on AWS в службе vRealize Automation.

Сопутствующие сведения см. в [документации по VMware Cloud on AWS](#).

### Необходимые условия

- Эта процедура предполагает, что у вас есть необходимые учетные данные администратора, в том числе учетные данные CloudAdmin VMware Cloud on AWS для целевого программно-определяемого ЦОД в vCenter, и что на порте 443 включен доступ HTTPS. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- В этой процедуре предполагается, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Чтобы обеспечивать необходимое подключение между существующим программно-определяемым ЦОД узла VMware Cloud on AWS в vCenter и облачной учетной записью VMware Cloud

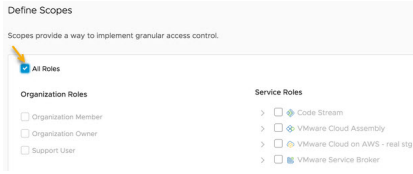
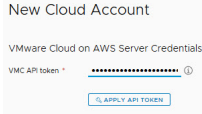
on AWS в vRealize Automation, необходимо установить сетевое подключение и настроить правила брандмауэра с помощью VPN или аналогичного сетевого решения. См. раздел [Подготовка программно-определяемого ЦОД VMware Cloud on AWS к подключению с помощью облачных учетных записей VMware Cloud on AWS в vRealize Automation](#). Если используется внешний прокси-сервер HTTP, для него необходимо настроить протокол IPv4.

- Если у вас нет внешнего доступа к Интернету, настройте прокси-сервер Интернета. См. раздел [Настройка прокси-сервера Интернета для vRealize Automation](#).

#### Процедура

1. Выберите **Инфраструктура > Подключения > Учетные записи облачной службы**.
2. Щелкните **Добавить облачную учетную запись**, выберите VMware Cloud on AWS и введите значения.

Пример значений и сопутствующие сведения приведены в таблице ниже.

Настройка	Пример значения и инструкция	Описание
Маркер API-интерфейса VMC	<ol style="list-style-type: none"> <li>Щелкните значок справки (?) в конце строки <b>маркера API-интерфейса VMC</b>, а затем щелкните ссылку на <b>страницу маркеров API-интерфейса</b> в текстовом поле справки, чтобы открыть вкладку <b>Маркеры API</b> на странице <b>Моя учетная запись</b> организации.</li> <li>Нажмите кнопку <b>Создать маркер</b>, чтобы отобразить параметры <b>создания нового маркера API</b>.</li> <li>Введите имя нового маркера, например <b>myinitials_mytoken</b>.</li> <li>Установите для параметра <b>Маркер TTL</b> значение <b>Без ограничения срока действия</b>.</li> </ol> <p>В случае создания маркера, для которого задан срок действия, по истечении срока действия данного маркера операции VMware Cloud on AWS из vRealize Automation не будут выполняться до тех пор, пока вы не укажете новый маркер в облачной учетной записи.</p> <ol style="list-style-type: none"> <li>В разделе <b>Определение охвата</b> выберите <b>Все роли</b>.</li> </ol>  <ol style="list-style-type: none"> <li>Нажмите кнопку <b>Создать</b>.</li> <li>На странице созданного маркера нажмите кнопку <b>Копировать</b>, а затем <b>Продолжить</b>.</li> <li>Вернитесь на страницу <b>Новая облачная учетная запись</b>, вставьте скопированный маркер в строку <b>Маркер API-интерфейса VMC</b> и нажмите кнопку <b>Применить маркер API</b>.</li> </ol> 	<p>Можно создать новый маркер или использовать существующий маркер, назначенный организации (см. страницу <b>Маркеры API-интерфейса</b> по ссылке).</p> <p>Обязательные роли, которые необходимо задать для маркера API в разделе <b>Определение охвата</b>:</p> <ul style="list-style-type: none"> <li>■ <b>Роли в организации</b> <ul style="list-style-type: none"> <li>■ Участник организации</li> <li>■ Владелец организации</li> </ul> </li> <li>■ <b>Роли служб (Service Roles) — VMware Cloud on AWS</b> <ul style="list-style-type: none"> <li>■ Администратор</li> <li>■ Администратор NSX Cloud</li> <li>■ Аудитор NSX Cloud</li> </ul> </li> </ul> <p><b>Примечание</b> Скопируйте, загрузите или распечатайте сгенерированный маркер. После того, как вы покинете эту страницу, извлечь его будет невозможно.</p> <p>Примените созданный или предоставленный маркер для подключения к доступной среде программно-определяемого ЦОД в рамках подписки VMware Cloud on AWS организации и для заполнения списка имен программно-определяемого ЦОД.</p> <p>Если службы vRealize Automation и VMware Cloud on AWS находятся в разных организациях, необходимо переключиться на организацию VMware Cloud on AWS, а затем создать маркер.</p> <p>Дополнительные сведения о маркерах API-интерфейса см. в разделе <a href="#">Создание маркеров API</a>.</p>
Имя программно-определяемого ЦОД	В рамках данного примера выберите <b>Datacenter:Datacenter-abz</b> .	Выберите программно-определяемый ЦОД из списка доступных в рамках подписки на VMware Cloud on AWS. Список программно-определяемых ЦОД составлен на основе маркера API-интерфейса VMware Cloud on AWS.

Настройка	Пример значения и инструкция	Описание
	Допустимое имя программно-определяемого ЦОД автоматически заполняет записи полного доменного имени vCenter и NSX-T. Если облачный прокси-сервер уже развернут в программно-определяемом ЦОД, значение облачного прокси-сервера также автоматически заполняется.	NSX-V Программно-определяемые ЦОД vRealize Automation не поддерживаются и не отображаются в списке
IP-адрес или полное доменное имя vCenter	Адрес автоматически заполняется в соответствии с выбранным программно-определяемым ЦОД.	Введите IP-адрес или полное доменное имя vCenter Server в указанном программно-определяемом ЦОД. IP-адрес по умолчанию задается как частный IP-адрес. В зависимости от типа сетевого подключения, используемого для доступа к программно-определяемому ЦОД, адрес по умолчанию может отличаться от IP-адреса сервера NSX Manager в указанном программно-определяемом ЦОД.
IP-адрес или полное доменное имя диспетчера NSX	Адрес автоматически заполняется в соответствии с выбранным программно-определяемым ЦОД.	Введите IP-адрес или полное доменное имя диспетчера NSX в указанном программно-определяемом ЦОД. IP-адрес по умолчанию задается как частный IP-адрес. В зависимости от типа сетевого подключения, используемого для доступа к программно-определяемому ЦОД, адрес по умолчанию может отличаться от IP-адреса сервера NSX Manager в указанном программно-определяемом ЦОД. Облачные учетные записи VMware Cloud on AWS поддерживают NSX-T.
Имя пользователя и пароль vCenter	Имя пользователя автоматически заполняется с использованием значения cloudadmin@vmc.local.	Введите имя пользователя vCenter для указанного программно-определяемого ЦОД, если оно отличается от значения по умолчанию. Указанному пользователю требуются учетные данные администратора облака. Пользователю не требуются учетные данные CloudGlobalAdmin. Введите пароль пользователя.
Проверка	Щелкните <b>Проверить</b> . Если вы получите Error updating endpoint <Имя>: Endpoint already exists, облачная учетная запись уже связана с этим программно-определяемым ЦОД.	Действие проверки подтверждает наличие прав доступа к указанному экземпляру vCenter и позволяет убедиться в том, что vCenter работает.
Имя и описание	Введите <b>OurCo-VMC</b> в качестве имени облачной учетной записи. Введите <b>Образец развертывания для VMC</b> в качестве описания облачной учетной записи.	



Настройка	Пример значения и инструкция	Описание
Разрешить предоставление ресурсов для этих центров обработки данных	Эта информация доступна только для чтения.	Вывод списка доступных центров обработки данных в указанной среде программно-определяемого ЦОД VMware Cloud on AWS.
Создать облачную зону	Снимите этот флажок. В рамках данного примера облачная зона будет создана на более позднем этапе рабочего процесса.	См. раздел <a href="#">Дополнительные сведения об облачных зонах Cloud Assembly</a> .
Теги возможностей	Оставьте это поле пустым. В этом рабочем процессе теги возможностей не используются.	Используйте теги в соответствии с корпоративной политикой расстановки тегов. См. разделы <a href="#">Как использовать теги для управления ресурсами и развертываниями Cloud Assembly</a> и <a href="#">Создание политики расстановки тегов</a> .

Как и для виртуальных машин, развертываемых в vSphere, теги компьютера можно настроить для виртуальной машины, развертываемой в VMware Cloud on AWS. Тег компьютера также можно обновить после первоначального развертывания. Эти теги компьютера позволяют vRealize Automation динамически назначать виртуальную машину соответствующей группе безопасности NSX-T во время развертывания. Дополнительные сведения см. в разделе [Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation](#).

### 3. Нажмите кнопку **Добавить**.

#### Результаты

Выполняется сбор данных по таким ресурсам, как машины и тома, из центра обработки данных программно-определяемого ЦОД VMware Cloud on AWS, и эти ресурсы отображаются в разделе **Ресурсы** вкладки vRealize Automation **Инфраструктура**.

#### Следующие шаги

[Создание облачной зоны для развертываний VMware Cloud on AWS в службе vRealize Automation.](#)

## Создание облачной зоны для развертываний VMware Cloud on AWS в службе vRealize Automation

На данном шаге выполняется создание облачной зоны, определяющей вычислительный ресурс, к которому пользователь CloudAdmin может получать доступ при работе с VMware Cloud on AWS в службе vRealize Automation.

В среде VMware Cloud on AWS используются два набора учетных данных главного администратора: CloudGlobalAdmin и CloudAdmin. Служба Cloud Assembly по умолчанию поддерживает работу с пользователем CloudAdmin. Выполняйте развертывания в ресурсы, доступные пользователю CloudAdmin для VMware Cloud on AWS. Выполнять развертывание в ресурсы, для доступа к которым требуются учетные данные CloudGlobalAdmin для VMware Cloud on AWS, не следует.

Облачные зоны позволяют определить вычислительные ресурсы, в которых облачный шаблон проекта будет выполнять развертывание компьютеров, сетей и хранилищ. См. раздел [Дополнительные сведения об облачных зонах Cloud Assembly](#).

Если не указано иное, значения шагов, вводимые в данной процедуре, применимы только в рамках данного примера рабочего процесса.

#### Необходимые условия

- Завершите процедуру [Создание облачной учетной записи VMware Cloud on AWS](#) в службе vRealize Automation в рамках образца рабочего процесса.
- В этой процедуре предполагается наличие необходимых учетных данных администратора, в том числе учетных данных администратора облака VMware Cloud on AWS для целевого программно-определяемого ЦОД в службе vCenter. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- В этой процедуре предполагается, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).

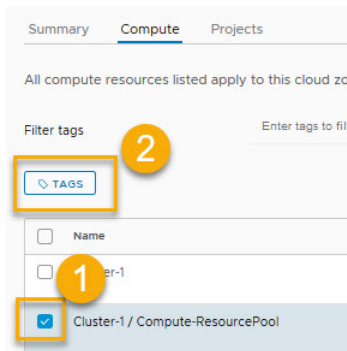
#### Процедура

1. Выберите **Инфраструктура > Настроить > Облачные зоны**.
2. Нажмите кнопку **Новая облачная зона** и введите значения для среды VMware Cloud on AWS.

Настройка	Пример значения
Учетная запись/область	OurCo-VMC или Datacenter:Datacenter-abz Это облачная учетная запись и связанная область, определенные на предыдущем шаге ( <a href="#">Создание облачной учетной записи VMware Cloud on AWS</a> в службе vRealize Automation в рамках образца рабочего процесса).
Имя	VMC_cloud_zone-1
Описание	Только ресурсы VMware Cloud on AWS
Политика размещения	По умолчанию
Теги возможностей	Оставьте это поле пустым. В этом рабочем процессе теги возможностей не используются.

3. Откройте вкладку **Вычисление**.
4. Как показано в области 1 ниже, найдите и выберите вычислительный ресурс, доступный пользователю CloudAdmin. Для этого примера используйте ресурс Cluster 1/ Compute-ResourcePool.

Cluster 1/ Compute-ResourcePool — это вычислительный ресурс по умолчанию для VMware Cloud on AWS.

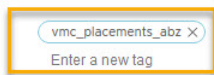


5. Как показано в области 2 выше, добавьте имя тега `vmc_placements_abz`.

Tags

1 object(s) selected

Add tags



Remove tags

no tags ⓘ

6. Отфильтруйте вычислительные ресурсы, которые используются в этой облачной зоне, указав `vmc_placements_abz` в разделе **Фильтр по тегам**.

7. Нажмите **Сохранить**.

<input type="checkbox"/>	Name	Account / region	Type	Tags
<input type="checkbox"/>	Cluster-1		Cluster	
<input checked="" type="checkbox"/>	Cluster-1 / Compute-ResourcePool	OurCo-VMC / SDDC_test1_abz	ResourcePool	vmc_placements_abz
<input type="checkbox"/>	Cluster-1 / Mgmt-ResourcePool		ResourcePool	

В данном примере пользователю CloudAdmin доступен только вычислительный ресурс с именем Cluster 1/ Compute-ResourcePool.

Следующие шаги

Настройка профилей сети и хранилища для развертываний VMware Cloud on AWS в службе vRealize Automation.

## Настройка профилей сети и хранилища для развертываний VMware Cloud on AWS в службе vRealize Automation

На данном этапе выполняется настройка профиля сети и профиля хранилища для указания ресурсов, доступных пользователю VMware Cloud on AWS CloudAdmin в службе vRealize Automation.

Хотя такие параметры, как образ и значение версии, также должны быть заданы, они не приобретают никаких уникальных особенностей в контексте учетных данных пользователя VMware Cloud on AWS. В этом примере при определении облачного шаблона будет использоваться значение конфигурации ресурсов `small` и значение образа `ubuntu-16`.

Общие сведения о сопоставлениях и профилях см. в разделе [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).

Если не указано иное, значения шагов, вводимые в данной процедуре, применимы только в рамках данного примера рабочего процесса.

#### Необходимые условия

- Создать облачную зону. См. раздел [Создание облачной зоны для развертываний VMware Cloud on AWS в службе vRealize Automation](#).
- В этой процедуре предполагается наличие необходимых учетных данных администратора, в том числе учетных данных администратора облака VMware Cloud on AWS для целевого программно-определяемого ЦОД в службе vCenter. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- В этой процедуре предполагается, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).

## Процедура

## 1. Определите профиль сети для развертываний VMware Cloud on AWS.

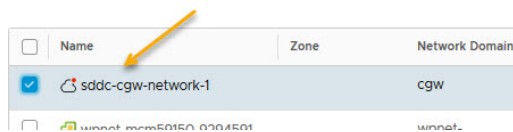
- а) Выберите пункт **Инфраструктура > Настроить > Профили сетей** и нажмите кнопку **Новый профиль сети**.

Настройка	Пример значения
Учетная запись/область	OurCo-VMC или Datacenter:Datacenter-abz
Имя	vmc-network1
Описание	Содержит сети, доступные администраторам облачных шаблонов с учетными данными роли CloudAdmin VMware Cloud on AWS.

**Примечание** Выберите облачную учетную запись VMware Cloud on AWS и соответствующий центр обработки данных программно-определяемого ЦОД, который был создан в разделе [Создание облачной учетной записи VMware Cloud on AWS](#) в службе vRealize Automation в рамках образца рабочего процесса.

- б) Перейдите на вкладку **Сеть** и нажмите кнопку **Добавить сеть**.
- в) Выберите сеть, в которой пользователь VMware Cloud on AWS с учетными данными CloudAdmin можно выполнить развертывание, например `sddc-cgw-network-1`.

Add Network



<input type="checkbox"/>	Name	Zone	Network Domain
<input checked="" type="checkbox"/>	sddc-cgw-network-1		cgw
<input type="checkbox"/>	winnet mcm50150 0204501		winnet-

## 2. Сохраните профиль сети.

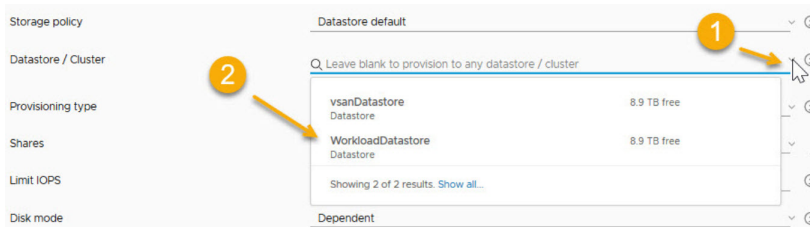
### 3. Определите профиль хранилища для развертываний VMware Cloud on AWS.

Настройте профиль хранилища, целевым объектом которого является хранилище данных или кластер, к которому у пользователя CloudAdmin есть доступ.

- а) Выберите пункт **Инфраструктура > Настроить > Профили хранилищ** и нажмите кнопку **Новый профиль хранилища**.

Настройка	Пример значения
Учетная запись/область	OurCo-VMC или Datacenter:Datacenter-abz  Выберите облачную учетную запись VMware Cloud on AWS и соответствующий центр обработки данных программно-определяемого ЦОД, который был создан в разделе <a href="#">Создание облачной учетной записи VMware Cloud on AWS</a> в службе vRealize Automation в рамках образца рабочего процесса.
Имя	vmc-storage1
Описание	Содержит кластер хранилищ данных, в которых администраторы облачных шаблонов с учетными данными CloudAdmin VMware Cloud on AWS могут выполнять развертывание.

- б) В раскрывающемся меню **Хранилище данных или кластер** выберите хранилище данных **WorkloadDatastore**.



В политике хранилища для VMware Cloud on AWS в службе Cloud Assembly должно использоваться хранилище данных **WorkloadDatastore**, чтобы обеспечить поддержку развертывания VMware Cloud on AWS.

### 4. Сохраните профиль хранилища.

#### Следующие шаги

[Создание проекта для поддержки развертываний VMware Cloud on AWS в службе vRealize Automation.](#)

### Создание проекта для поддержки развертываний VMware Cloud on AWS в службе vRealize Automation

На данном шаге выполняется определение проекта vRealize Automation, который может использоваться для управления доступом к ресурсам со стороны развертываний VMware Cloud on AWS.

Сведения о проектах см. в разделе [Какова роль проектов Cloud Assembly во время развертывания](#).

Если не указано иное, значения шагов, вводимые в данной процедуре, применимы только в рамках данного примера рабочего процесса.

## Необходимые условия

- Завершите процедуру [Настройка профилей сети и хранилища для развертываний VMware Cloud on AWS в службе vRealize Automation](#).
- В этой процедуре предполагается наличие необходимых учетных данных администратора, в том числе учетных данных администратора облака VMware Cloud on AWS для целевого программно-определяемого ЦОД в службе vCenter. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- В этой процедуре предполагается, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).

## Процедура

1. Выберите **Инфраструктура > Администрирование > Проекты**.
2. Нажмите кнопку **Новый проект** и введите имя проекта VMC\_proj-1\_abz.
3. Нажмите кнопку **Пользователи** и выберите **Добавить пользователей**.

Пользователям требуются учетные данные CloudAdmin для доступа к подписке на VMware Cloud on AWS их организации.

- chris.gray@ourco.com, администратор
- kerry.white@ourco.com, участник

4. Щелкните **Предоставление**, а затем **Добавить облачную зону**.
5. Добавьте облачную зону, настроенную на предыдущем шаге.

Настройка	Пример значения
Облачная зона	VMC_cloud_zone-1 Эта облачная зона была создана на предыдущем шаге (Создание облачной зоны для развертываний VMware Cloud on AWS в службе vRealize Automation).
Приоритет подготовки	1
Макс. количество экземпляров	3

6. В рамках данного примера прочие параметры можно игнорировать.

## Следующие шаги

Создайте облачный шаблон для развертывания в среде VMware Cloud on AWS. См. раздел [Определение ресурса компьютера vCenter в проекте облачного шаблона для поддержки развертывания VMware Cloud on AWS в службе vRealize Automation](#).

## Определение ресурса компьютера vCenter в проекте облачного шаблона для поддержки развертывания VMware Cloud on AWS в службе vRealize Automation

На данном шаге ресурс компьютера vCenter необходимо перетащить на холст проекта и добавить настройки для развертывания VMware Cloud on AWS в vRealize Automation.

Создайте проект облачного шаблона для развертывания в доступных ресурсах VMware Cloud on AWS.

Если не указано иное, значения шагов, вводимые в данной процедуре, применимы только в рамках данного примера рабочего процесса.

### Необходимые условия

- Для выполнения этой процедуры у вас должны быть учетные данные разработчика облачных шаблонов. См. раздел [Что такое роли пользователей vRealize Automation](#).
- В этой процедуре предполагается, что у вас есть учетные данные VMware Cloud on AWS CloudAdmin для целевого программно-определяемого ЦОД в среде vCenter (Datacenter:Datacenter-abz). См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Настройте проект и инфраструктуру ресурсов, как описано в предыдущих разделах.

### Процедура

1. Перейдите на вкладку **Проект** и нажмите **Создать**.

Настройка	Пример значения
Имя	vmc-bp_abz
Описание	1
Проект	VMC_proj-1_abz Это созданный ранее проект, который поддерживает облачную зону, которая также была создана ранее. Теперь проект связан с облачной зоной, которая в свою очередь связана с облачной учетной записью или областью VMware Cloud on AWS, созданной ранее.

2. Переместите ресурс компьютера vSphere на холст.
3. Отредактируйте следующий (выделенный полужирным шрифтом) код ресурса облачного шаблона в ресурсе компьютера.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
```



```
image: ubuntu-1604
cpuCount: 1
totalMemoryMB: 1024
folderName: Workloads
```

Параметр `image` может иметь любое значение, отвечающее потребностям развертывания.

Обязательно добавьте инструкцию `folderName: Workloads` в код проекта облачного шаблона, чтобы обеспечить поддержку развертывания VMware Cloud on AWS. Настройка `folderName: Workloads` поддерживает использование учетных данных CloudAdmin в среде программно-определяемого ЦОД VMware Cloud on AWS и является обязательной.

Примечание. Хотя параметр `folderName: Workloads`, показанный в приведенном выше примере кода, является обязательным, его можно добавить непосредственно в код проекта облачного шаблона, как показано выше, или в связанную облачную зону или проект. Если параметр задан в нескольких таких местах, то приоритет будет следующим.

- Параметр проекта переопределяет параметр облачного шаблона и параметр облачной зоны.
- Параметр облачного шаблона переопределяет параметр облачной зоны.

Примечание. При необходимости можно заменить настройки `cpuCount` и `totalMemoryMB` записью `flavor` (размеры), как показано ниже:

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: ubuntu-1604
      flavor: small
      folderName: Workloads
```

Если для параметра `folder` облачной зоны задано значение **Workloads**, то задавать свойство `folderName` в облачном шаблоне требуется только в том случае, если нужно переопределить значение параметра `folder` облачной зоны.

### Следующие шаги

Добавьте изоляцию сети для расширения этого базового рабочего процесса VMware Cloud on AWS.

См. раздел [Настройка изолированной сети в рамках рабочего процесса VMware Cloud on AWS в службе vRealize Automation](#).

## Настройка изолированной сети в рамках рабочего процесса VMware Cloud on AWS в службе vRealize Automation

В этой процедуре изолированная сеть добавляется в развертывание VMware Cloud on AWS в службе vRealize Automation.

При определении облачной учетной записи VMware Cloud on AWS доступны параметры NSX-T, настроенные в службе VMware Cloud on AWS. Дополнительные сведения о настройке параметров NSX-T в службе VMware Cloud on AWS см. в документации по продукту VMware Cloud on AWS <https://docs.vmware.com/ru/VMware-Cloud-on-AWS/index.html>.

Служба vRealize Automation поддерживает использование VMware Cloud on AWS с NSX-T. При этом она не поддерживает использование VMware Cloud on AWS с NSX-V.

Служба vRealize Automation поддерживает изоляцию сети в развертываниях VMware Cloud on AWS. При этом она не поддерживает другие методы работы с сетью применительно к VMware Cloud on AWS.

В данной расширенной версии базового рабочего процесса VMware Cloud on AWS описываются следующие методы создания изолированной сети для использования в облачном шаблоне.

- Настройка изоляции по требованию на основе сети.
- Настройка изоляции по требованию на основе группы безопасности.

#### Необходимые условия

Эта процедура представляет собой расширение базового рабочего процесса VMware Cloud on AWS. В ней используются те же облачная учетная запись и область, облачная зона, проект и профиль сети, которые были настроены в рамках рабочего процесса [Учебник «Настройка VMware Cloud on AWS для vRealize Automation»](#).

#### Процедура

##### 1. Определение изолированной сети для развертывания VMware Cloud on AWS в службе vRealize Automation

Можно настроить изоляцию сети для развертывания VMware Cloud on AWS, используя одну из следующих процедур:

##### 2. Определение компонента сети в облачном шаблоне для поддержки изоляции сети в среде VMware Cloud on AWS в службе vRealize Automation

На данном шаге компонент сетевой машины перетаскивается на холст облачного шаблона vRealize Automation и добавляются настройки для развертывания изолированной сети в целевой среде VMware Cloud on AWS.

#### Определение изолированной сети для развертывания VMware Cloud on AWS в службе vRealize Automation

Можно настроить изоляцию сети для развертывания VMware Cloud on AWS, используя одну из следующих процедур:

- [Настройка изоляции по требованию на основе сети в службе vRealize Automation](#)
- [Настройка изоляции по требованию на основе группы безопасности в службе vRealize Automation](#)

#### Настройка изоляции по требованию на основе сети в службе vRealize Automation

Можно настроить изоляцию сети в соответствии с потребностями развертывания VMware Cloud on AWS путем задания и использования настроек группы безопасности по требованию в профиле сети.

Изолированную сеть можно задать с помощью группы безопасности или с помощью настроек сети по требованию. В этом примере настройка изоляции сети выполняется путем указания настроек сети по требованию в профиле сети. Позже к этой сети в облачном шаблоне можно получать доступ, чтобы затем использовать данный шаблон в развертывании VMware Cloud on AWS.

Если не указано иное, значения шагов, вводимые в данной процедуре, применимы только в рамках данного примера рабочего процесса.

#### Необходимые условия

- Завершите рабочий процесс [Настройка базового рабочего процесса VMware Cloud on AWS в vRealize Automation](#).
- См. раздел [Настройка изолированной сети в рамках рабочего процесса VMware Cloud on AWS в службе vRealize Automation](#).
- В этой процедуре предполагается наличие необходимых учетных данных администратора, в том числе учетных данных администратора облака VMware Cloud on AWS для целевого программно-определяемого ЦОД в службе vCenter. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- В этой процедуре предполагается, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).

#### Процедура

1. Откройте профиль сети, который использовался в базовом рабочем процессе VMware Cloud on AWS, например `vmc-network1`. См. раздел [Настройка профилей сети и хранилища для развертываний VMware Cloud on AWS в службе vRealize Automation](#).
2. Выбрать настройки на вкладке **Сети** не требуется.
3. Откройте вкладку **Политики сети**.
4. Нажмите кнопку **Создать сеть по требованию** и выберите домен сети `sgw` по умолчанию. Укажите соответствующее значение CIDR и размер подсети.
5. Нажмите **Сохранить**.

Когда используется этот профиль сети, машины развертываются в сети в домене по умолчанию. Эта сеть изолирована от остальных сетей посредством частного или исходящего сетевого доступа.

#### Следующие шаги

Настройте сетевой компонент в облачном шаблоне. См. раздел [Определение компонента сети в облачном шаблоне для поддержки изоляции сети в среде VMware Cloud on AWS в службе vRealize Automation](#).

#### Настройка изоляции по требованию на основе группы безопасности в службе vRealize Automation

Можно настроить изоляцию сети в соответствии с потребностями развертывания VMware Cloud on AWS путем создания и использования группы безопасности по требованию в профиле сети.

Изолированную сеть можно задать с помощью группы безопасности или с помощью настроек сети по требованию. В этом примере настройка изоляции сети выполняется путем указания группы безопасности по требованию в профиле сети. Позже эту сеть можно задать в облачном шаблоне, чтобы затем использовать данный шаблон в развертывании VMware Cloud on AWS.

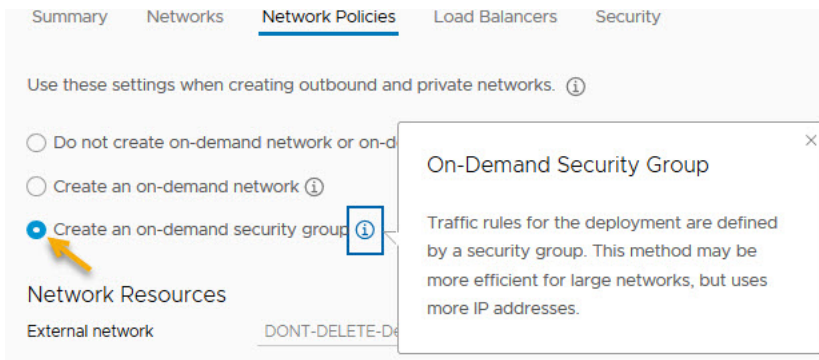
Если не указано иное, значения шагов, вводимые в данной процедуре, применимы только в рамках данного примера рабочего процесса.

#### Необходимые условия

- Завершите рабочий процесс [Настройка базового рабочего процесса VMware Cloud on AWS в vRealize Automation](#).
- См. раздел [Настройка изолированной сети в рамках рабочего процесса VMware Cloud on AWS в службе vRealize Automation](#).
- В этой процедуре предполагается наличие необходимых учетных данных администратора, в том числе учетных данных администратора облака VMware Cloud on AWS для целевого программно-определяемого ЦОД в службе vCenter. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- В этой процедуре предполагается, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).

#### Процедура

1. Откройте профиль сети, который использовался в базовом рабочем процессе VMware Cloud on AWS, например `vmc-network1`. См. раздел [Настройка профилей сети и хранилища для развертываний VMware Cloud on AWS в службе vRealize Automation](#).
2. Выберите существующую сеть, которая использовалась в базовом рабочем процессе VMware Cloud on AWS, например `sddc-cgw-network-1`. См. раздел [Настройка профилей сети и хранилища для развертываний VMware Cloud on AWS в службе vRealize Automation](#).
3. Откройте вкладку **Политики сети**.
4. Выберите параметр **Создать группу безопасности по требованию**.



## 5. Нажмите **Сохранить**.

Когда используется этот профиль сети, машины развертываются в выбранной сети и изолируются с использованием новой политики группы безопасности. Новая политика безопасности разрешает как частный, так и исходящий доступ к сети.

### Следующие шаги

Настройте сетевой компонент в облачном шаблоне. См. раздел [Определение компонента сети в облачном шаблоне для поддержки изоляции сети в среде VMware Cloud on AWS в службе vRealize Automation](#).

## Определение компонента сети в облачном шаблоне для поддержки изоляции сети в среде VMware Cloud on AWS в службе vRealize Automation

На данном шаге компонент сетевой машины перетаскивается на холст облачного шаблона vRealize Automation и добавляются настройки для развертывания изолированной сети в целевой среде VMware Cloud on AWS.

Добавьте параметры изоляции сети в созданный ранее облачный шаблон. Этот облачный шаблон уже связан с проектом и облачной зоной, которые поддерживают развертывание в среде VMware Cloud on AWS, а также с профилем сети и сетью, настроенными для режима изоляции.

Если не указано иное, значения шагов, вводимые в данной процедуре, применимы только в рамках данного примера рабочего процесса.

### Необходимые условия

- Завершите процедуру [Настройка изоляции по требованию на основе группы безопасности в службе vRealize Automation](#) или [Настройка изоляции по требованию на основе сети в службе vRealize Automation](#).
- Для выполнения этой процедуры у вас должны быть учетные данные разработчика облачных шаблонов. См. раздел [Что такое роли пользователей vRealize Automation](#).
- В этой процедуре предполагается, что у вас есть учетные данные пользователя VMware Cloud on AWS с правами администратора облака (CloudAdmin) для целевого программно-определяемого ЦОД в среде vCenter. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).

### Процедура

1. Откройте облачный шаблон, созданный в рамках предыдущего рабочего процесса. См. раздел [Определение ресурса компьютера vCenter в проекте облачного шаблона для поддержки развертывания VMware Cloud on AWS в службе vRealize Automation](#).
2. Перетащите на холст нужный компонент сети из области слева на странице проектирования облачных шаблонов.

3. Измените код YAML компонента сети и укажите в нем тип сети `private` или `outbound`, как показано жирным шрифтом.

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: private
```

ИЛИ

```
resources: Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: vmc_isolated
    networkType: outbound
```

#### Следующие шаги

Теперь можно приступить к разворачиванию облачного шаблона либо закрыть его.

## Учебник «Настройка интеграции определенного внешнего поставщика IPAM для vRealize Automation»

Для управления назначением IP-адресов для разворачиваний облачных шаблонов можно использовать внешнего поставщика IPAM. В данном учебнике описывается настройка интеграции внешнего поставщика IPAM Infoblox в vRealize Automation.

В этой процедуре для создания точки интеграции IPAM для конкретного поставщика используется существующий пакет поставщика IPAM, пакет Infoblox и существующая запущенная среда. Настраивается существующая сеть и создается профиль сети для поддержки выделения IP-адресов внешним поставщиком IPAM. Наконец, создается облачный шаблон, сопоставляемый с сетью и профилем сети, и в сети разворачиваются компьютеры с использованием значений IP-адресов, полученных от внешнего поставщика IPAM.

В качестве ссылки приводятся сведения о получении и настройке пакета поставщика IPAM, а также о настройке запущенной среды, которая получает доступ к облачному прокси-серверу расширяемости для поддержки интеграции поставщика IPAM.

Значения в этом примере рабочего процесса являются примерами значений. Их нельзя перенести в вашу среду одно к одному. Подумайте о том, где вы можете внести изменения в соответствии с потребностями вашей организации.



Ссылку на аналогичный сценарий vRealize Automation, который иллюстрирует процесс интеграции Infoblox IPAM в форме видео, см. в разделе [Интеграция подключаемого модуля Infoblox IPAM с vRealize Automation/vRealize Automation Cloud](#).

## Процедура

### 1. Добавление обязательных расширяемых атрибутов в приложении Infoblox для интеграции с vRealize Automation

Перед загрузкой и развертыванием пакета поставщика Infoblox (infoblox.zip) для интеграции с vRealize Automation с веб-сайта Infoblox или из магазина VMware необходимо добавить обязательные атрибуты расширяемости в Infoblox.

### 2. Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation

Чтобы определить точку интеграции внешнего IPAM в vRealize Automation, необходимо иметь настроенный пакет поставщика IPAM.

### 3. Создание запущенной среды для точки интеграции IPAM в vRealize Automation

Прежде чем определить внешнюю точку интеграции IPAM в vRealize Automation, необходимо создать или получить доступ к существующей запущенной среде, которая будет использоваться в качестве посредника между поставщиком IPAM и vRealize Automation. Обычно запущенная среда является облачной учетной записью Amazon Web Services или Microsoft Azure или локальной точкой интеграции расширяемости на основе действий, которая связана с облачным прокси-сервером расширяемости.

### 4. Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation

vRealize Automation поддерживает интеграцию с внешними поставщиками IPAM. В этом примере в качестве внешнего поставщика IPAM используется Infoblox.

### 5. Настройка сети и профиля сети для использования внешнего поставщика IPAM в существующей сети vRealize Automation

Можно определить существующую сеть для использования значений IP-адресов, которые предоставляются внешним поставщиком IPAM и управляются в его среде, а не в vRealize Automation.

### 6. Определение и развертывание облачного шаблона, в котором используется назначение диапазона внешнего поставщика IPAM в vRealize Automation

Для получения назначений IP-адресов от внешнего поставщика IPAM и управления ими можно задать облачный шаблон. В этом примере в качестве внешнего поставщика IPAM используется Infoblox.

### 7. Использование свойств и расширяемых атрибутов, связанных с Infoblox, для интеграций IPAM в облачных шаблонах vRealize Automation

Для проектов vRealize Automation, содержащих интеграции внешнего IPAM Infoblox, можно использовать свойства, связанные с Infoblox.

## 8. Управление сбором данных сети с помощью фильтров Infoblox в vRealize Automation

В Infoblox сбор данных можно ограничить только теми сетями, которые необходимы для работы vRealize Automation. Это позволяет уменьшить объем передаваемых данных и повысить производительность системы.

## Добавление обязательных расширяемых атрибутов в приложении Infoblox для интеграции с vRealize Automation

Перед загрузкой и развертыванием пакета поставщика Infoblox (`infoblox.zip`) для интеграции с vRealize Automation с веб-сайта Infoblox или из магазина VMware необходимо добавить обязательные атрибуты расширяемости в Infoblox.

Эта процедура применима при создании точки интеграции внешнего IPAM Infoblox с Cloud Assembly.

Перед загрузкой `infoblox.zip` необходимо войти в учетную запись Infoblox, используя данные администратора учетной записи организации, и предварительно создать следующие расширяемые атрибуты Infoblox

- VMware NIC index
- VMware resource ID

### Необходимые условия

- Убедитесь, что у вас есть учетная запись с [Infoblox](#), а также корректные учетные данные доступа для учетной записи Infoblox организации.
- Убедитесь, что поддерживается версия Infoblox WAPI. Интеграция IPAM с Infoblox зависит от Infoblox WAPI версии 2.7. Поддерживаются устройства Infoblox, которые работают с WAPI 2.7.
- См. раздел [Использование свойств и расширяемых атрибутов, связанных с Infoblox, для интеграций IPAM в облачных шаблонах vRealize Automation](#).

### Процедура

1. Войдите в учетную запись Infoblox с помощью учетных данных администратора.

Это те же имя пользователя и пароль администратора, которые указываются при создании точки интеграции внешнего IPAM в Cloud Assembly в меню **Инфраструктура > Подключения > Интеграции > .**

2. Для создания следующих обязательных расширяемых атрибутов в приложении Infoblox используйте процедуру, описанную в документации Infoblox.

- VMware NIC index — тип «Целое число»
- VMware resource ID — тип «Строка»

Процедура описана в разделе *Добавление расширяемых атрибутов* главы [О расширяемых атрибутах](#) в документации Infoblox. См. также [Управление расширяемыми атрибутами](#).



## Следующие шаги

После добавления обязательных атрибутов можно возобновить процесс загрузки и развертывания пакета Infoblox, как описано в [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

## Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation

Чтобы определить точку интеграции внешнего IPAM в vRealize Automation, необходимо иметь настроенный пакет поставщика IPAM.

Пакет интеграции конкретного поставщика можно загрузить на сайте поставщика IPAM или на [VMware Marketplace](#).

---

**Примечание** В этом примере используется предоставленный корпорацией VMware пакет Infoblox (Infoblox.zip), который можно загрузить из [VMware Marketplace](#).

- **Подключаемый модуль Infoblox версии 1.4** — совместим с vRealize Automation 8.3–8.7 и предоставляет все функциональные возможности предыдущих версий. В этой версии можно использовать одно и то же имя узла с разным суффиксом DNS для двух сетевых адаптеров. Дополнительные сведения см. в информации о версии подключаемого модуля.
- **Подключаемый модуль Infoblox версии 1.3** — совместим с vRealize Automation 8.3.x и предоставляет дополнительные фильтры для сбора данных сети. См. раздел [Управление сбором данных сети с помощью фильтров Infoblox в vRealize Automation](#). При использовании vRealize Automation 8.3.x можно применить подключаемый модуль Infoblox 1.4 для получения дополнительных возможностей.

Подключаемый модуль [Infoblox 1.3](#) можно использовать с vRealize Automation 8.1 или 8.2, но только в определенных ситуациях и с осторожностью. См. статью базы знаний [Совместимость Infoblox 1.3 с vRealize Automation 8.x \(82142\)](#).

- **Подключаемый модуль vRA Cloud Infoblox версии 1.2** — совместим с vRealize Automation 8.1.x и 8.2.x
- **Подключаемый модуль vRA Cloud Infoblox версии 1.1** — совместим с vRealize Automation 8.1.x
- **Подключаемый модуль vRA Cloud Infoblox версии 1.0** — совместим с vRealize Automation 8.0.1.x с интернет-подключением к глобальной сети или без него.
- **Подключаемый модуль vRA Cloud Infoblox версии 0.4** — совместим с vRealize Automation 8.0.0.x и 8.0.1.x с интернет-подключением к глобальной сети.

Для интеграции IPAM с Infoblox настроена зависимость от Infoblox WAPI версии 2.7. Поддерживаются все устройства Infoblox, которые работают с WAPI 2.7.

---

Сведения о том, как создать пакет интеграции IPAM для других поставщиков IPAM, если его нет в [VMware Marketplace](#), см. в разделе [Использование IPAM SDK для создания внешнего пакета интеграции IPAM для определенного поставщика для vRealize Automation](#).

Пакет поставщика IPAM содержит сценарии, упакованные вместе с метаданными и другими конфигурациями. Сценарии содержат исходный код, используемый для операций, которые vRealize Automation выполняет совместно с внешним поставщиком IPAM. Примеры операций включают в себя `Allocate an IP address for a virtual machine`, `Fetch a list of IP ranges from the provider` и `Update the MAC address of a host record in the provider`.

#### Необходимые условия

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что у вас есть учетная запись с внешним поставщиком IPAM, например [Infoblox](#) или [Bluecat](#), и корректные учетные данные для доступа к учетной записи организации с помощью поставщика IPAM.
- Если в качестве внешнего поставщика IPAM используется Infoblox, перед продолжением убедитесь, что в учетную запись Infoblox добавлены обязательные расширяемые атрибуты. См. раздел [Добавление обязательных расширяемых атрибутов в приложении Infoblox для интеграции с vRealize Automation](#).

---

**Примечание** Существует проблема с цепочкой сертификатов, связанная с тем, как элемент Python в подключаемом модуле Infoblox обрабатывает квитирование SSL. Дополнительные сведения о проблеме и действия по ее устранению см. в статье базы знаний [Подключаемый модуль vRA Cloud Infoblox вызывает ошибку цепочки сертификатов в процессе проверки подлинности \(88057\)](#).

---

#### Процедура

1. Перейдите на страницу загрузки для подключаемого модуля Infoblox. Ссылки на определенную версию подключаемого модуля Infoblox см. выше.  
 Параметры подключаемых модулей Infoblox, доступных в [VMware Marketplace](#), см. выше.
2. Войдите в систему и загрузите пакет подключаемого модуля.
3. Если это еще не сделано, добавьте необходимые расширяемые атрибуты в Infoblox. См. раздел [Добавление обязательных расширяемых атрибутов в приложении Infoblox для интеграции с vRealize Automation](#).

#### Результаты

Теперь пакет доступен для развертывания. Выберите **Интеграции > Добавить интеграцию > IPAM > Управление поставщиками > Импорт пакета**, как описано в [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#).

## Создание запущенной среды для точки интеграции IPAM в vRealize Automation

Прежде чем определить внешнюю точку интеграции IPAM в vRealize Automation, необходимо создать или получить доступ к существующей запущенной среде, которая будет использоваться в качестве посредника между поставщиком IPAM и vRealize Automation. Обычно запущенная среда является облачной учетной записью Amazon Web Services или Microsoft Azure или локальной точкой интеграции расширяемости на основе действий, которая связана с облачным прокси-сервером расширяемости.

Для интеграции внешнего IPAM требуется запущенная среда. При определении точки интеграции IPAM необходимо создать подключение между Cloud Assembly и поставщиком IPAM, указав доступную запущенную среду.

При интеграции IPAM в запущенной среде используется набор загруженных сценариев для конкретного поставщика или подключаемых модулей, который можно упростить с помощью поставщика функций как услуги (FaaS), например Amazon Web Services Lambda, функций Microsoft Azure или локальной встроенной точки интеграции расширяемости на основе действий (ABX). Запущенная среда используется для подключения к внешнему поставщику IPAM, например Infoblox.

---

**Примечание** Для точки интеграции Infoblox IPAM требуется локальная встроенная точка интеграции расширяемости на основе действий (ABX).

---

Каждый тип запущенной среды имеет свои преимущества и недостатки.

- Точка интеграции расширяемости на основе действий (ABX)
  - Бесплатная, нет дополнительных затрат на использование поставщика.
  - Может подключаться к устройствам поставщика IPAM, находящимся в локальном центре обработки данных за NAT/брандмауэром, который не является общедоступным, например Infoblox.
  - Обладает немного меньшей производительностью, чем у коммерческой среды.
- Amazon Web Services
  - Имеет связанные затраты на подключение/использование FaaS поставщика.
  - Не может быть подключена к устройствам поставщика IPAM, находящимся в локальном центре обработки данных за NAT/брандмауэром, который не является общедоступным.
  - Высокая производительность и высокая надежность.
- Microsoft Azure
  - Имеет связанные затраты на подключение/использование FaaS поставщика.
  - Не может быть подключена к устройствам поставщика IPAM, находящимся в локальном центре обработки данных за NAT/брандмауэром, который не является общедоступным.
  - Высокая производительность и высокая надежность.

## Необходимые условия

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что у вас есть учетная запись с внешним поставщиком IPAM, например [Infoblox](#) или [Bluecat](#), и корректные учетные данные для доступа к учетной записи организации с помощью поставщика IPAM.
- Убедитесь, что у вас есть доступ к развернутому пакету интеграции для поставщика IPAM, например [Infoblox](#) или [BlueCat](#). Этот пакет сначала получают путем загрузки ZIP-файла с сайта поставщика IPAM или с [Vmware Marketplace](#), а затем развертывают в Cloud Assembly.

Дополнительные сведения о том, как развернуть ZIP-файл пакета поставщика и сделать его доступным в виде значения **Поставщик** на странице «Интеграция IPAM», см. в разделе [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

## Процедура

1. Чтобы создать локальное действие расширяемости на основе FaaS, которое будет использоваться в качестве запущенной среды интеграции IPAM, выберите **Расширяемость > Библиотека > Действия**.
2. Щелкните **Создать действие**, введите имя действия и его описание, а затем укажите проект.
3. В раскрывающемся меню **Поставщик FaaS** выберите **Локально**.
4. Заполните форму, чтобы определить действие расширяемости.

Дополнительные сведения о создании действий расширяемости см. в разделе [Увеличение и автоматизация жизненных циклов приложений с помощью действий расширяемости](#).



Дополнительные сведения о запущенной среде см. в видео в блоге [Интеграция подключаемого модуля IPAM](#), начиная приблизительно с 24-й минуты.

## Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation

vRealize Automation поддерживает интеграцию с внешними поставщиками IPAM. В этом примере в качестве внешнего поставщика IPAM используется Infoblox.

Чтобы получить и настроить IP-адреса и связанные характеристики сети для развертываний облачных шаблонов, можно использовать точку интеграции IPAM для конкретного поставщика.

В этом примере создается точка интеграции внешнего поставщика IPAM для поддержки доступа к учетной записи организации с помощью поставщика. В данном примере рабочего процесса поставщиком IPAM является Infoblox, а пакет интеграции для конкретного поставщика уже существует. Хотя эти инструкции относятся к интеграции Infoblox, их можно использовать в качестве ссылки при интеграции другого внешнего поставщика IPAM.

Пакет интеграции конкретного поставщика можно получить на сайте поставщика IPAM или на [VMware Marketplace](#).

В этом примере используется предоставленный корпорацией VMware пакет Infoblox (Infoblox.zip), который можно загрузить из [VMware Marketplace](#). Сведения о последних версиях подключаемого модуля Infoblox, доступных в [VMware Marketplace](#), см. в [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

#### Необходимые условия

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что у вас есть учетная запись с внешним поставщиком IPAM, а также корректные учетные данные для доступа к учетной записи организации с помощью поставщика IPAM.
- Убедитесь, что у вас есть доступ к развернутому пакету интеграции для поставщика IPAM. Развернутый пакет изначально получается путем загрузки ZIP-файла с веб-сайта поставщика IPAM или из магазина VMware Solutions Exchange, а затем развертывается в vRealize Automation.

Дополнительные сведения о том, как загрузить и развернуть ZIP-файл пакета поставщика и сделать его доступным в виде значения **Поставщик** на странице «Интеграция IPAM», см. в разделе [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

- Убедитесь, что у вас есть доступ к настроенной и запущенной среде для поставщика IPAM. Как правило, запущенная среда является локальной встроенной точкой интеграции средств расширяемости на основе действий (ABX).

Сведения о характеристиках запущенной среды см. в разделе [Создание запущенной среды для точки интеграции IPAM в vRealize Automation](#).

- Включите необходимые расширяемые атрибуты в приложении Infoblox. См. раздел [Добавление обязательных расширяемых атрибутов в приложении Infoblox для интеграции с vRealize Automation](#).
- Если у вас нет внешнего доступа к Интернету, можно настроить прокси-сервер Интернета. См. раздел [Настройка прокси-сервера Интернета для vRealize Automation](#).
- Убедитесь, что у вас есть необходимые учетные данные для доступа и использования продукта IPAM Infoblox. Например, перейдите на вкладку «Администрирование» в устройстве Infoblox и настройте записи для администратора, группы и ролей. Вы должны быть членом группы, у которой есть разрешения на уровне администратора или привилегированного пользователя, или настраиваемой группы с разрешениями DHCP, DNS, IPAM и Grid. Такие настройки предоставляют доступ ко всем функциям в подключаемом модуле Infoblox, что позволяет создать интеграцию IPAM Infoblox и дает дизайнерам возможность использовать эту интеграцию IPAM в облачных шаблонах и развертываниях. Дополнительные сведения о разрешениях пользователей см. в документации по продукту Infoblox.

#### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.

## 2. Щелкните **IPAM**.

3. В раскрывающемся меню **Поставщик** выберите настроенный пакет поставщика IPAM в списке, например *Infoblox\_hrg*.

Если список пуст, нажмите **Импортировать пакет поставщика**, найдите существующий ZIP-файл пакета поставщика и выберите его. Если у вас нет ZIP-файла пакета поставщика, его можно получить с веб-сайта поставщика IPAM или на [VMware Marketplace](#).

Дополнительные сведения о том, как развернуть ZIP-файл пакета поставщика в vCenter и сделать его доступным в виде значения **Поставщик** на странице «Интеграция», см. в разделе [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

Сведения о том, как обновить существующую интеграцию IPAM для использования более новой версии пакета интеграции IPAM поставщика, см. в разделе [Обновление до новой версии пакета интеграции внешнего поставщика IPAM в vRealize Automation](#).

4. Введите имя пользователя и пароль администратора для своей учетной записи с внешним поставщиком IPAM, а также заполните все остальные обязательные поля (при наличии), например имя узла поставщика.

В этом примере описывается получение имени узла поставщика Infoblox IPAM с помощью следующей процедуры.

- а) На отдельной вкладке браузера войдите в учетную запись своего поставщика IPAM, используя учетные данные администратора Infoblox.
  - б) Скопируйте свой URL-адрес имени узла.
  - в) Вставьте URL-адрес имени узла в поле **Имя узла** на странице «Интеграция IPAM».
5. В раскрывающемся списке **Запущенная среда** выберите существующую локальную точку интеграции расширяемости на основе действий, например *Infoblox\_abx\_intg*.

Запущенная среда поддерживает связь между vRealize Automation и внешним поставщиком IPAM.

---

**Примечание** Если в качестве запущенной среды интеграции используется облачная учетная запись Amazon Web Services или Microsoft Azure, убедитесь, что устройство поставщика IPAM доступно из Интернета и не находится под защитой NAT или брандмауэра и что оно имеет общедоступное разрешимое DNS-имя. Если поставщик IPAM недоступен, функции Amazon Web Services Lambda или Microsoft Azure не смогут подключиться к нему и интеграция завершится ошибкой. Дополнительные сведения см. в разделе [Создание запущенной среды для точки интеграции IPAM в vRealize Automation](#).

---

Платформа IPAM поддерживает только локальную встроенную запущенную среду расширяемости на основе действий (ABX).

---

**Примечание** Для точки интеграции Infoblox IPAM требуется локальная встроенная точка интеграции расширяемости на основе действий (ABX).

---

Настроенная облачная учетная запись или точка интеграции обеспечивает связь между vRealize Automation и поставщиком IPAM (в данном примере — Infoblox) с помощью связанного облачного прокси-сервера расширяемости. Можно выбрать уже созданного поставщика или создать его.

Сведения о создании запущенной среды см. в разделе [Создание запущенной среды для точки интеграции IPAM в vRealize Automation](#).

## 6. Щелкните **Проверить**.

Так как в этом примере для запущенной среды используется локальная интеграция расширяемости на основе действий, можно просмотреть действие проверки.

- а) Перейдите на вкладку **Расширяемость**.
- б) Выберите **Действия > Выполнение действий**, а затем **Все циклы** или **Циклы интеграции** в фильтре, чтобы убедиться, что действие проверки конечной точки инициировано и выполняется.

## 7. При появлении запроса на доверие самозаверяющему сертификату от поставщика IPAM нажмите кнопку **Принять**.

После принятия самозаверяющего сертификата действие проверки может быть продолжено.

## 8. Введите **имя** для этой точки интеграции IPAM, например *Infoblox\_Integration*, а также **описание**, например *Infoblox IPAM с интеграцией ABX для группы HRG*.

## 9. Нажмите **Добавить**, чтобы сохранить новую внешнюю точку интеграции IPAM.

Выполняется имитация действия сбора данных. Данные о сетях и диапазонах IP-адресов собираются из поставщика IPAM. Для просмотра действия сбора данных выполните следующее.

- а) Перейдите на вкладку **Расширяемость**.
- б) Выберите **Действия > Циклы выполнения** и убедитесь, что действие сбора данных инициировано и запущено. Можно открыть действие «Цикл выполнения» и просмотреть его.

## Результаты

Интеграция конкретного внешнего поставщика IPAM теперь может использоваться с сетями и профилями сетей.

## Настройка сети и профиля сети для использования внешнего поставщика IPAM в существующей сети vRealize Automation

Можно определить существующую сеть для использования значений IP-адресов, которые предоставляются внешним поставщиком IPAM и управляются в его среде, а не в vRealize Automation.

Можно определить сеть для доступа к существующим параметрам IP, которые определены в учетной записи внешнего поставщика IPAM вашей организации. Эта операция распространяется на интеграцию поставщика Infoblox, которая была создана на предыдущем шаге.



В этом примере выполняется настройка профиля сети с существующими сетями, данные о которых были получены из vCenter. Затем для этих сетей выполняется настройка для получения сведений о IP-адресах от внешнего поставщика IPAM (в данном случае Infoblox). Виртуальные машины, которые предоставляются из vRealize Automation и могут быть сопоставлены с этим профилем сети, получают IP-адреса и другие параметры, связанные с протоколом TCP/IP, от внешнего поставщика IPAM.

Дополнительные сведения о сетях см. в разделе [Сетевые ресурсы в vRealize Automation](#).

Дополнительные сведения о профилях сетей см. в разделах [Как добавлять профили сети в vRealize Automation](#) и [Дополнительные сведения о профилях сетей в vRealize Automation](#).

Дополнительные сведения см. в разделе [Настройка поддержки сети по требованию в профиле сети для интеграции внешнего поставщика IPAM в vRealize Automation](#).

#### Необходимые условия

Данная последовательность шагов показана в контексте рабочего процесса интеграции поставщика IPAM. См. раздел [Учебник «Настройка интеграции определенного внешнего поставщика IPAM для vRealize Automation»](#).

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что у вас есть учетная запись с внешним поставщиком IPAM, например [Infoblox](#) или [Bluecat](#), и корректные учетные данные для доступа к учетной записи организации с помощью поставщика IPAM. В данном примере рабочего процесса поставщиком IPAM является Infoblox.
- Убедитесь, что у вас есть точка интеграции IPAM для поставщика IPAM. См. раздел [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#).

#### Процедура

1. Чтобы настроить сеть, выберите **Инфраструктура > Ресурсы > Сети**.
2. На вкладке **Сети** выберите существующую сеть для использования с точкой интеграции поставщика IPAM. В данном примере имя сети — *net.23.117-only-IPAM*.

Данные о сетях в списке были собраны системой vRealize Automation из vCenter в организации.

3. Чтобы получить значения от внешнего поставщика IPAM, убедитесь, что все параметры сети, за исключением **Учетная запись/регион**, **Имя** и **Сетевой домен**, будут пустыми, включая следующие.
  - Домен (см. примечание на шаге 8)
  - CIDR
  - Шлюз по умолчанию
  - Серверы DNS
  - Домены поиска DNS
4. Перейдите на вкладку **Диапазоны IP-адресов** и нажмите **Добавить диапазон IP-адресов IPAM**.



5. В меню **Сеть** выберите только что настроенную сеть, например *net.23.117-only-IPAM*.
6. В меню **Поставщик** выберите точку интеграции IPAM *Infoblox\_Integration*, созданную ранее в рабочем процессе.
7. В раскрывающемся меню **Адресное пространство** выберите одно из представлений сети.

Адресное пространство в Infoblox называется представлением сети.

Представления сети поступают из учетной записи поставщика IPAM. В этом примере используются только что настроенная подсеть сети, например *net.23.117-only-IPAM*, точка интеграции *Infoblox\_Integration*, созданная ранее в рабочем процессе, и адресное пространство с именем *по умолчанию*.

Указанные значения адресного пространства поступают от внешнего поставщика IPAM.

8. В списке отображаемых сетей, доступных для выбранного адресного пространства, выберите одну или несколько сетей, например 10.23.117.0/24.

В этом примере столбцы **Домены** и **Серверы DNS** для выбранной сети содержат значения из Infoblox.

---

**Примечание** Если выбрать сеть на шаге 3, когда был указан домен для vRealize Automation, а затем выбрать сеть в адресном пространстве внешнего поставщика IPAM, которое содержит значение домена, то значение домена в сети внешнего поставщика IPAM будет иметь более высокий приоритет, чем значение домена, указанного в vRealize Automation. Если для параметра «Диапазон IP-адресов IPAM» не задано значение домена, указанное в Cloud Assembly или во внешнем поставщике IPAM, как описано выше, процесс предоставления завершается сбоем.

---

Для Infoblox можно использовать свойство схемы элементов `Infoblox.IPAM.Network.dnsSuffix` на уровне компьютера, чтобы перезаписать значение домена. Дополнительные сведения см. в разделе [Использование свойств и расширяемых атрибутов, связанных с Infoblox, для интеграций IPAM в облачных шаблонах vRealize Automation](#).

9. Нажмите **Добавить**, чтобы сохранить диапазон IP-адресов IPAM для сети.

Диапазон отображается в таблице **Диапазоны IP-адресов**.

10. Перейдите на вкладку **Диапазоны IP-адресов**.

После подготовки компьютера с помощью нового диапазона адресов от внешнего поставщика IPAM в таблице **IP-адреса** будет отображаться новая запись.

11. Чтобы настроить профиль сети для использования сети, выберите **Инфраструктура > Настроить > Профили сети**.

12. Назначьте имя профилю сети, например *Infoblox-NP*, и добавьте следующие параметры.

- Вкладка «Сводка»
  - Укажите облачную учетную запись или регион vSphere.
  - Добавьте тег возможности для профиля сети, например с именем *infoblox\_abx*.

Запишите тег возможности, так как его также необходимо использовать как тег ограничения облачного шаблона, чтобы создать связь предоставляемых элементов в облачном шаблоне.

- Вкладка «Сети»

- Добавьте созданную ранее сеть, например *net.23.117-only-IPAM*.

13. Нажмите кнопку **Сохранить**, чтобы сохранить профиль сети с этими настройками.

#### Результаты

Теперь параметры сети и профиля сети настроены для типа существующей сети, который должен использоваться с интеграцией IPAM Infoblox в облачном шаблоне.

## Определение и развертывание облачного шаблона, в котором используется назначение диапазона внешнего поставщика IPAM в vRealize Automation

Для получения назначений IP-адресов от внешнего поставщика IPAM и управления ими можно задать облачный шаблон. В этом примере в качестве внешнего поставщика IPAM используется Infoblox.

На заключительном этапе рабочего процесса интеграции внешнего поставщика IPAM необходимо задать и развернуть облачный шаблон, который подключает ранее определенную сеть и профиль сети к учетной записи Infoblox данной организации. Это позволяет получать IP-адреса для развернутых виртуальных машин от внешнего поставщика IPAM, а не из vRealize Automation, а также настраивать эти адреса.

Этот рабочий процесс использует Infoblox в качестве внешнего поставщика IPAM. На некоторых этапах образцы значений уникальны для Infoblox, хотя задача заключается в том, чтобы эту процедуру можно было применить к другим внешним интеграциям IPAM.



Соответствующие сведения приводятся в блоге Infoblox [Автоматизация IPAM и DNS для виртуальных машин с помощью VMware vRealize Automation и Infoblox DDI](#).

После развертывания облачного шаблона и запуска виртуальной машины IP-адрес, который используется для каждой виртуальной машины в развертывании, отображается в виде записи сети на странице **Ресурсы > Сети**, в виде новой записи узла в сети поставщика IPAM (в учетной записи поставщика IPAM), а также в записи vSphere Web Client для каждой развернутой VM в экземпляре vCenter узла.

#### Необходимые условия

Такая последовательность шагов показана в контексте рабочего процесса интеграции внешнего поставщика IPAM. См. раздел [Учебник «Настройка интеграции определенного внешнего поставщика IPAM для vRealize Automation»](#).

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что у вас есть учетная запись с внешним поставщиком IPAM, например Infoblox или BlueCat, и корректные учетные данные для доступа к учетной записи организации с помощью поставщика IPAM.

- Убедитесь, что у вас есть доступ администратора к учетной записи узла и вы удовлетворяете всем требованиям к ролям, необходимым для отображения записей о состоянии в записи веб-клиента vSphere для развернутых виртуальных машин в vCenter узла.
- Убедитесь, что у вас есть точка интеграции IPAM для внешнего поставщика IPAM. См. раздел [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#).
- Убедитесь, что настроены сеть vRealize Automation и профиль сети, поддерживающие интеграцию внешнего поставщика IPAM для предполагаемой точки интеграции. См. раздел [Настройка сети и профиля сети для использования внешнего поставщика IPAM в существующей сети vRealize Automation](#).
- Убедитесь, что проект и облачная зона помечены тегами для сопоставления с тегами в точке интеграции IPAM, а также в сети или профиле сети. При необходимости настройте проект для поддержки именования настраиваемых ресурсов.

Дополнительные сведения о функциях проекта и облачной зоны, а также о функциях других элементов инфраструктуры в облачном шаблоне см. в разделе [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).

Дополнительные сведения о тегировании см. в разделе [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#).

Сведения о пользовательском именовании виртуальных машин с помощью настроек в проекте см. в разделе [Настраиваемое именование развернутых ресурсов в Cloud Assembly](#).

## Процедура

1. Выберите **Облачные шаблоны > Создать**, введите указанную ниже информацию на странице **Создать облачный шаблон**, а затем нажмите **Создать**.
  - **Имя** = ipam-bpa
  - **Описание** — облачный шаблон, который использует интеграцию IPAM Infoblox.
  - **Проект** = 123VC
2. В данном примере добавьте на холст облачного шаблона независимые от облачной среды компонент компьютера и компонент сети и соедините их.
3. Измените код облачного шаблона, добавив тег ограничения в компонент сети, который соответствует тегу возможностей, добавленному в профиль сети. В данном примере тег имеет значение *infoblox\_abx*.
4. Измените код облачного шаблона, указав тип назначения сети — *статический*.

При использовании внешнего поставщика IPAM необходимо указать параметр `assignment: static`.

В этом примере указанный IP-адрес 10.23.117.4 доступен в адресном пространстве внешнего IPAM, выбранном для сети в связанном профиле сети. Хотя параметр `assignment: static` является обязательным, `address: значение` не требуется. Выбор внешнего IP-адреса можно начинать с определенного значения, но это не обязательно. Если `address: значение` не указано, внешний поставщик IPAM выбирает следующий доступный адрес во внешней сети IPAM.

5. Проверьте, соответствует ли код облачного шаблона следующему примеру.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
      name: ipam
      constraints:
        - tag: infoblox_abx
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          assignment: static
          address: 10.23.117.4
          name: '${resource.Cloud_Network_1.name}'
```

Примеры свойств Infoblox, доступных для определения параметров DNS и DHCP в облачных шаблонах, см. в разделе [Использование свойств и расширяемых атрибутов, связанных с Infoblox, для интеграций IPAM в облачных шаблонах vRealize Automation](#).

6. Щелкните **Развернуть** на странице облачного шаблона, введите имя развертывания *Infoblox-1* и нажмите **Развернуть** на странице **Тип развертывания**.

7. В ходе развертывания облачного шаблона перейдите на вкладку **Расширяемость** и выберите **Действия > Выполнение действий**, чтобы посмотреть выполнение действия расширяемости *Infoblox\_AllocateIP\_n*.

После того как действие расширяемости завершено и компьютер подготовлен, действие *Infoblox\_Update\_n* передает MAC-адрес в Infoblox.

8. Чтобы увидеть новую запись узла для адреса IPAM в связанной сети 10.23.117.0/24, можно войти в систему и открыть учетную запись Infoblox. Кроме того, можно перейти на вкладку DNS в Infoblox и посмотреть новую запись узла DNS.

9. Чтобы убедиться, что выполняется подготовка виртуальной машины, войдите в vCenter и vSphere Web Client узла, чтобы найти подготовленный компьютер, и просмотрите DNS-имя и IP-адрес.

После запуска подготовленной виртуальной машины MAC-адрес передается в Infoblox с помощью действия расширяемости *Infoblox\_AllocateIP*.

10. Чтобы просмотреть новую запись сети в vRealize Automation, выберите **Инфраструктура > Ресурсы > Сети**, а затем перейдите на вкладку **IP-адреса**.

11. Если удалить развертывание, адрес IPAM виртуальных машин в развертывании будет освобожден, а IP-адреса снова будут доступны внешнему поставщику IPAM для других процессов выделения ресурсов. Для этого события в vRealize Automation действием расширяемости является *Infoblox\_Deallocate*.

## Использование свойств и расширяемых атрибутов, связанных с Infoblox, для интеграций IPAM в облачных шаблонах vRealize Automation

Для проектов vRealize Automation, содержащих интеграции внешнего IPAM Infoblox, можно использовать свойства, связанные с Infoblox.

Следующие свойства Infoblox доступны для использования с интеграциями IPAM Infoblox в проектах и развертываниях облачных шаблонов. Их можно использовать в vRealize Automation для последующего управления выделением IP-адресов во время развертывания облачных шаблонов. Использование этих свойств не является обязательным.

---

**Примечание** Если используется подключаемый модуль Infoblox 1.4 или более ранней версии, глобальное свойство Infoblox переопределяет локальное свойство Infoblox для свойств `dnsSuffix`, `dnsView`, `enableDns` и `enableDhcp`. Глобальное свойство применяется ко всем сетевым адаптерам.

---

Следующие свойства доступны и включены в последнюю версию подключаемого модуля Infoblox для vRealize Automation. Информацию о версиях подключаемого модуля Infoblox и о том, где можно найти последнюю версию подключаемого модуля Infoblox для интеграции IPAM в vRealize Automation, см. в [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

- `Infoblox.IPAM.createFixedAddress`

Данное свойство позволяет создать фиксированную запись адреса в Infoblox. Возможны следующие значения: `True` и `False`. По умолчанию создается запись узла. Значение по умолчанию — «Ложь».

- `Infoblox.IPAM.Network.dnsView`

Это свойство позволяет использовать представление DNS при создании записи узла в Infoblox.

- `Infoblox.IPAM.Network.enableDns`

При выделении IP-адреса в Infoblox это свойство позволяет также создать DNS-запись. Возможны следующие значения: `True` и `False`. Значение по умолчанию — «Истина».

- `Infoblox.IPAM.Network.enableDhcp`

Это свойство позволяет настроить DHCP для адреса узла. Возможны следующие значения: `True` и `False`. Значение по умолчанию — «Истина».

- `Infoblox.IPAM.Network.dnsSuffix`

Это свойство позволяет перезаписать параметр DHCP *domain* для сети Infoblox, используя новое значение. Такая возможность полезна в том случае, если в сети Infoblox не задан параметр DHCP *domain* или если параметр DHCP *domain* должен быть перезаписан. Значение по умолчанию — `null` (пустая строка).

Если используется внешний поставщик IPAM, например Infoblox, при подготовке компьютера необходимо указать суффикс DNS. Суффикс DNS является обязательным. Его можно указать одним из следующих способов.

- Укажите суффикс DNS для подсети сети vSphere в vRealize Automation.
- Укажите свойство `Infoblox.IPAM.Network.dnsSuffix` в коде ресурса компьютера в облачном шаблоне vRealize Automation.

Пример показан ниже в разделе `Infoblox.IPAM.Network.hostnameNicSuffix`.

`Infoblox.IPAM.Network.dnsSuffix` применим только в том случае, если для `Infoblox.IPAM.Network.enableDns` задано значение `True`.

#### ■ `Infoblox.IPAM.Network.hostnameNicSuffix`

Это свойство можно использовать, чтобы указать суффикс индекса сетевого адаптера при создании имени узла.

При этом можно подготовить компьютер с несколькими сетевыми адаптерами таким образом, чтобы имена узлов для каждого сетевого адаптера различались настраиваемым суффиксом. В следующем примере показано, что можно подготовить компьютер, например *my-machine*, который содержит 2 сетевых адаптера, таким образом, чтобы в качестве суффикса имени узла для первого сетевого адаптера использовалось значение `-nic1`, а для второго — `-nic2`.

Кроме того, можно указать суффикс DNS, как показано в примере. Свойство `Infoblox.IPAM.Network.dnsSuffix` используется со значением `test.local`, поэтому, первый сетевой адаптер получает имя *my-machine-nic1.test.local* а второй — *my-machine-nic2.test.local*.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      Infoblox.IPAM.Network.dnsSuffix: test.local
      Infoblox.IPAM.Network0.hostnameNicSuffix: -nic1
      Infoblox.IPAM.Network1.hostnameNicSuffix: -nic2
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          deviceIndex: 0
        - network: '${resource.Cloud_Network_2.id}'
          deviceIndex: 1
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
```

```
Cloud_Network_2:
  type: Cloud.Network
  properties:
    networkType: existing
```

Это свойство было впервые реализовано в подключаемом модуле Infoblox версии 1.3. См. [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

- Свойства можно также задать с помощью подписки с поддержкой расширяемости.

Дополнительные сведения о расширяемых атрибутах Infoblox, относящихся к этому примеру использования, см. в разделе [Добавление обязательных расширяемых атрибутов в приложении Infoblox для интеграции с vRealize Automation](#).

## Использование свойств Infoblox на различных сетевых адаптерах компьютеров в облачном шаблоне

Следующие свойства Infoblox могут поддерживать разные значения для каждого сетевого адаптера компьютера в облачном шаблоне:

- `Infoblox.IPAM.Network.enableDhcp`
- `Infoblox.IPAM.Network.dnsView`
- `Infoblox.IPAM.Network.enableDns`
- `Infoblox.IPAM.Network.hostnameNicSuffix`

Например, чтобы использовать другое значение `Infoblox.IPAM.Network.dnsView` для каждого сетевого адаптера, используйте `Infoblox.IPAM.Network<nicIndex>.dnsView` для каждого сетевого адаптера. В следующем примере показаны различные значения `Infoblox.IPAM.Network.dnsView` для двух сетевых адаптеров.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      Infoblox.IPAM.Network0.dnsView: default
      Infoblox.IPAM.Network1.dnsView: my-net
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          deviceIndex: 0
        - network: '${resource.Cloud_Network_2.id}'
          deviceIndex: 1
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
```

```
Cloud_Network_2:
  type: Cloud.Network
  properties:
    networkType: existing
```

По умолчанию интеграция Infoblox создает запись узла DNS в представлении DNS *по умолчанию* в Infoblox. Если администратор Infoblox создал *настраиваемые* представления DNS, поведение интеграции по умолчанию можно перезаписать и указать именованное представление с помощью свойства Infoblox.IPAM.Network.dnsView в компоненте компьютера. Например, можно добавить в компонент Cloud\_Machine\_1 следующее свойство, чтобы указать именованное представление DNS в Infoblox.

```
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ubuntu
    flavor: small
    Infoblox.IPAM.Network.dnsView: <dns-view-name>
```

Сведения о настройке и использовании представлений DNS см. в разделе [Представления DNS](#) в документации по продукту Infoblox. Пример рабочего процесса интеграции Infoblox см. в разделе [Определение и развертывание облачного шаблона](#), в котором используется назначение диапазона внешнего поставщика IPAM в vRealize Automation.

## Указание свойств Infoblox

Свойство Infoblox можно задать с помощью одного из следующих методов в Cloud Assembly.

- Свойства можно задать в проекте в разделе **Настраиваемые свойства** на странице **Инфраструктура > Администрирование > Проекты**. С помощью этого метода указанные свойства применяются ко всем компьютерам, подготовленным в рамках данного проекта.
- Свойства каждого компонента компьютера можно задать в облачном шаблоне. Пример кода облачного шаблона, иллюстрирующий использование свойства Infoblox.IPAM.Network.dnsView, приведен ниже.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      Infoblox.IPAM.Network.dnsView: default
      image: ubuntu
      cpuCount: 1
      totalMemoryMB: 1024
      networks:
        - network: '${resource.Cloud_Network_1.id}'
  Cloud_Network_1:
    type: Cloud.Network
```



```
properties:
  networkType: existing
  constraints:
    - tag: mk-ipam-demo
```

## Управление сбором данных сети с помощью фильтров Infoblox в vRealize Automation

В Infoblox сбор данных можно ограничить только теми сетями, которые необходимы для работы vRealize Automation. Это позволяет уменьшить объем передаваемых данных и повысить производительность системы.

vRealize Automation собирает данные из внешней системы управления IP-адресами каждые 10 минут. Настраивая фильтры Infoblox различными способами, можно обнаруживать и собирать данные только из того подмножества сетей, которое используется для работы vRealize Automation.

Чтобы отфильтровать собираемые данные сетей, в которых используются IP-адреса, созданные Infoblox, воспользуйтесь следующими свойствами на вкладке «Интеграция IPAM». Свойства фильтра становятся доступными при создании или изменении точки интеграции внешней службы управления IP-адресами Infoblox.

Эти фильтры доступны только в vRealize Automation 8.3 и более поздних версий, а также в подключаемом модуле Infoblox 1.3 и более поздних версий (например, подключаемом модуле Infoblox 1.4).

---

**Примечание** Подключаемый модуль Infoblox 1.3 можно использовать с vRealize Automation 8.1 или 8.2, но только в определенных ситуациях и с осторожностью. См. статью базы знаний [Совместимость Infoblox 1.3 с vRealize Automation 8.x \(82142\)](#).

---

- Infoblox.IPAM.NetworkContainerFilter

Фильтры сетевых контейнеров.

- Infoblox.IPAM.NetworkFilter

Фильтр сетей.

- Infoblox.IPAM.RangeFilter

Фильтр диапазонов IP-адресов.

Следует соблюдать осторожность, применяя фильтры к сетям, которые уже участвуют в сборе данных. Если фильтры применяются, чтобы предотвратить сбор данных в некоторых сетях, то сети, которые исключаются из сбора данных, считаются ненужными и удаляются из vRealize Automation. Исключение — сети, связанные с подсетями vRealize Automation. Сети, в которых ранее собирались данные, но которые впоследствии не обнаруживаются и не участвуют в сборе данных, например потому что они не попали в фильтры задачи сбора данных, удаляются из базы данных vRealize Automation. Тем не менее если сети, в которых ранее собирались данные, используются в vRealize Automation, они не удаляются.

Эти фильтры применяются в качестве параметров в запросах поиска различных сетевых объектов. Можно использовать любые параметры поиска, которые поддерживает Infoblox. Фильтрация осуществляется по CIDR или расширяемым атрибутам на основе регулярных выражений или точных совпадений. Используется формат фильтрации Infoblox WAPI, как описано в [документации по Infoblox WAPI](#). Способы фильтрации по CIDR или расширяемым атрибутам показаны в следующих примерах.

- Фильтрация на основе CIDR для сетей и сетевых контейнеров. Примеры:
  - Точное совпадение — `Infoblox.IPAM.NetworkFilter: network=192.168.0.0`
  - Сопоставление с расширяемым атрибутом — `Infoblox.IPAM.NetworkFilter: network~=192.168`
- Фильтрация на основе CIDR для диапазона IP-адресов. Пример  
 Сопоставление с регулярным выражением и именем представления сети —  
`Infoblox.IPAM.RangeFilter: network~=192.168.&network_view=my_view`
- Фильтрация на основе расширяемых атрибутов для сетей, диапазонов IP-адресов и сетевых контейнеров.  
 В синтаксисе используется следующий формат: *имя\_фильтра* = \* *расши\_аттпр* = *знач\_расши\_аттпр*.  
 Примеры:
  - Точное совпадение — `*Building=Data Center`
  - Соответствие регулярному выражению с символом «~» — `*Building~=*Center`
  - Соответствие с учетом регистра с символом «:» — `*Building:=data center`
  - Исключение соответствия с помощью символа «!» — `*Building!=Data Center`
  - Соответствие регулярному выражению (учет регистра и исключение можно комбинировать):  
`*Building! ~:=Data Cent / *Building~:=center`
- Фильтрация на основе CIDR и расширяемых атрибутов с применением синтаксиса из рассмотренных выше способов фильтрации. Пример  
`network=192.168.&*Building=Data Center`

Дополнительные сведения об использовании расширяемых атрибутов и регулярных выражений в этих свойствах см. в разделах [Поддерживаемые Infoblox выражения для параметров поиска](#) и [Руководство по Infoblox REST API](#).

# Настройка Cloud Assembly для организации

# 3

Администратор Cloud Assembly должен разбираться в особенностях ролей пользователей и настраивать соединения с поставщиком учетных записей облачной службы и интегрируемыми приложениями.

Настройка облачных учетных записей и интеграций подразумевает определение параметров обмена данными между Cloud Assembly и этими целевыми системами.

В эту главу входят следующие разделы:

- [Что такое роли пользователей vRealize Automation](#)
- [Добавление учетных записей облачной службы в Cloud Assembly](#)
- [Интеграция vRealize Automation с другими приложениями](#)
- [Что такое планы внедрения в Cloud Assembly](#)
- [Расширенная настройка среды Cloud Assembly](#)

## Что такое роли пользователей vRealize Automation

В vRealize Automation имеется несколько уровней ролей пользователей. Эти уровни контролируют доступ к организации, службам, проектам, которые создают или используют облачные шаблоны, элементы каталога и конвейеры, а также возможность использования или просмотра отдельных частей пользовательского интерфейса. Благодаря уровням администраторы облачных систем получают различные инструменты для применения детализации любой степени, необходимой в ходе эксплуатации.

### Общее описание ролей

Роли пользователей определяются на разных уровнях. Роли уровня службы определяются для каждой службы.

Ниже под таблицей приведены дополнительные сведения о ролях служб.

Роль	Общие разрешения	Где определяется роль
Владелец организации	<p>Может получать доступ к консоли и добавлять пользователей в организацию.</p> <p>Владелец организации не может получить доступ к службе, если у него нет роли службы.</p> <p>Подробнее о <a href="#">Роли пользователей организации</a></p>	Консоль организации
Участник организации	<p>Может получить доступ к консоли.</p> <p>Участник организации не может получить доступ к службе, если у него нет роли службы.</p> <p>Подробнее о <a href="#">Роли пользователей организации</a></p>	Консоль организации
Администратор службы	<p>Может получить доступ к консоли и обладает полными правами на просмотр, обновление и удаление в службе.</p> <ul style="list-style-type: none"> <li>■ <a href="#">Роли службы Cloud Assembly</a></li> <li>■ <a href="#">Роли службы Service Broker</a></li> <li>■ <a href="#">Роли службы Code Stream</a></li> <li>■ <a href="#">Роли службы vRA Migration Assistant</a></li> <li>■ <a href="#">Роли службы Orchestrator</a></li> <li>■ <a href="#">Роль службы SaltStack Config</a></li> </ul>	Консоль организации
Пользователь службы	<p>Может получить доступ к консоли и службе с ограниченными разрешениями.</p> <p>Участнику службы доступен пользовательский интерфейс с ограниченными возможностями. Его возможности по просмотру или выполнению действий зависят от членства в проекте.</p> <ul style="list-style-type: none"> <li>■ <a href="#">Роли службы Cloud Assembly</a></li> <li>■ <a href="#">Роли службы Service Broker</a></li> <li>■ <a href="#">Роли службы Code Stream</a></li> </ul>	Консоль организации
Обозреватель службы	<p>Может получать доступ к консоли и службе в режиме просмотра.</p> <ul style="list-style-type: none"> <li>■ <a href="#">Роли службы Cloud Assembly</a></li> <li>■ <a href="#">Роли службы Service Broker</a></li> <li>■ <a href="#">Роли службы Code Stream</a></li> <li>■ <a href="#">Роли службы vRA Migration Assistant</a></li> <li>■ <a href="#">Роли службы Orchestrator</a></li> </ul>	Консоль организации
Исполнитель (только в Code Stream)	<p>Может получить доступ к консоли и управлять выполнением конвейера.</p> <p><a href="#">Роли службы Code Stream</a></p>	Консоль организации

Роль	Общие разрешения	Где определяется роль
Конструктор рабочих процессов Orchestrator (только для Orchestrator)	Может создавать, запускать, изменять и удалять собственное содержимое клиента vRealize Orchestrator. Может добавить собственное содержимое в назначенную группу. Не имеет доступа к функциям клиента vRealize Orchestrator по администрированию и устранению неполадок. <a href="#">Роли службы Orchestrator</a>	Консоль организации
Роли в проекте	Позволяет просматривать ресурсы проекта и управлять ими в зависимости от роли в проекте. Существуют следующие роли проекта: администратор, участник и обозреватель. <a href="#">Роли пользователей организаций и служб в vRealize Automation</a>	Cloud Assembly, Service Broker и Code Stream
Настраиваемые роли	Администратор Cloud Assembly определяет разрешения для всех служб. Пользователь должен иметь по крайней мере роль обозревателя в соответствующих службах, чтобы иметь доступ к ним. Настраиваемые роли имеют более высокий приоритет, чем роли служб. <a href="#">Настраиваемые роли пользователей в vRealize Automation</a>	Cloud Assembly и Service Broker
Встроенная роль администратора инфраструктуры	Предоставляет предварительно определенные разрешения для задач в vRealize Automation. <a href="#">Назначение пользователю встроенной роли администратора инфраструктуры Cloud Assembly</a>	Использование API-интерфейса

## Роли пользователей организаций и служб в vRealize Automation

Роли пользователей организаций и служб, определенные для служб Cloud Assembly, Service Broker и Code Stream, указывают, что пользователь может просматривать и делать в каждой службе.

### Роли пользователей организации

Роли пользователей определяются для организации в консоли vRealize Automation владельцем организации. Существует два типа ролей: роли организации и роли служб.

Роли организации являются глобальными и применяются ко всем службам в организации. Роли на уровне организации: «владелец организации» или «участник организации».

Подробные сведения о ролях организации см. в разделе [Администрирование vRealize Automation](#)

Роли служб Cloud Assembly, представляющие собой разрешения для конкретных служб, также назначаются на уровне организации в консоли.

## Роли служб

Эти роли службы назначаются владельцем организации.

Эта статья содержит информацию о следующих службах.

- [Роли службы Cloud Assembly](#)
- [Роли службы Service Broker](#)
- [Роли службы Code Stream](#)
- [Роли службы vRA Migration Assistant](#)
- [Роли службы Orchestrator](#)
- [Роль службы SaltStack Config](#)

## Роли службы Cloud Assembly

Роли службы Cloud Assembly определяют отображаемое содержимое и доступные действия в Cloud Assembly. Эти роли службы определяются в консоли владельцем организации.

Таблица 3-1. Описание ролей службы Cloud Assembly

Роль	Описание
Администратор Cloud Assembly	Пользователь должен иметь доступ для чтения и записи ко всему пользовательскому интерфейсу и ресурсам API. Это единственная роль пользователя, которая позволяет просматривать и выполнять все действия, в том числе добавлять облачные учетные записи, создавать новые проекты и назначать администратора проекта.
Пользователь Cloud Assembly	Пользователь, у которого нет роли администратора Cloud Assembly. В проекте Cloud Assembly администратор добавляет пользователей в качестве участников, администраторов или обозревателей проекта. Администратор также может добавить администратора проекта.
Наблюдатель Cloud Assembly	Пользователь с правом чтения может просматривать сведения, но не может создавать, обновлять или удалять какие-либо значения. Это роль «только для чтения» для всех проектов. Пользователи с ролью обозревателя могут видеть всю информацию, доступную администратору. Он не может выполнять никакие действия, пока не будет назначен в качестве администратора или участника проекта. Если пользователь связан с проектом, у него есть разрешения, связанные с ролью. Обозреватель проекта не может расширять свои разрешения так же, как администратор или участник.

В дополнение к ролям служб в Cloud Assembly доступны роли проектов. Любой проект доступен во всех службах.

Роли проекта определяются в Cloud Assembly и могут различаться в зависимости от проекта.

В следующих таблицах указаны возможности различных ролей по просмотру и выполнению задач в рамках служб и проектов. Следует помнить, что администраторы служб имеют полный доступ ко всем областям пользовательского интерфейса.

Описание ролей проекта поможет решить, какие разрешения следует предоставить пользователям.

- Администраторы проектов используют инфраструктуру, созданную администратором службы, и обеспечивают наличие ресурсов, необходимых участникам проектов на этапе разработки.
- Участники проектов работают в своих проектах над проектированием и развертыванием облачных шаблонов. Проекты могут включать в себя только ресурсы, которыми вы владеете, или ресурсы, которые используются совместно с другими участниками проекта.
- Обзорщики проекта имеют ограниченный доступ «только чтение», кроме нескольких случаев, когда они могут выполнять неразрушающие операции, такие как загрузка облачных шаблонов.
- Координаторы проекта являются утверждающими в Service Broker для своих проектов, для которых в политике утверждения определено утверждение координатором проекта. Чтобы предоставить координатору контекст для утверждений, можно также назначить ему роль участника или наблюдателя проекта.

Таблица 3-2. Роли служб и проектов Cloud Assembly

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly			
				Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Доступ к Cloud Assembly							
Консоль	Консоль vRA позволяет найти и открыть Cloud Assembly.	Да	Да	Да	Да	Да	Да
Инфраструктура							
	Просмотр и открытие вкладки «Инфраструктура»	Да	Да	Да	Да	Да	Да
Настройка — Проекты	Создание проектов	Да					

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
	Обновление или удаление значений сводки проекта, параметров предоставления, Kubernetes, интеграций и конфигураций тестовых проектов.	Да					
	Добавление пользователей и групп, назначение ролей в проектах.	Да		Да. Проекты.			
	Просмотр проектов.	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	Да. Проекты
Настройка — Облачные зоны	Создание, обновление и удаление облачных зон	Да					
	Просмотр облачных зон.	Да	Да				
	Просмотр панели управления «Ключевые сведения» для облачной зоны	Да	Да				
	Просмотр оповещений для облачных зон	Да	Да				
Настройка — зоны Kubernetes	Создание, обновление и удаление зон Kubernetes	Да					
	Просмотр зон Kubernetes	Да	Да				
Настройка — конфигурации ресурсов	Создание, обновление и удаление конфигураций ресурсов	Да					



Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
	Просмотр конфигураций ресурсов	Да	Да				
Настройка — Сопоставления образов	Создание, обновление и удаление сопоставлений образов	Да					
	Просмотр сопоставлений образов	Да	Да				
Настройка — Профили сети	Создание, обновление и удаление профилей сети	Да					
	Просмотр профилей сети образа	Да	Да				
Настройка — Профили хранилища	Создание, обновление и удаление профилей хранилища	Да					
	Просмотр профилей хранилища образа	Да	Да				
Настройка — карточки ценообразования	Создание, обновление и удаление карточек ценообразования	Да					
	Просмотр карточек ценообразования	Да	Да				
Настройка — Теги	Создание, обновление и удаление тегов	Да					
	Просмотр тегов	Да	Да				

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Ресурсы — Вычислительные ресурсы	Добавление тегов для обнаруженных вычислительных ресурсов	Да					
	Просмотр обнаруженных вычислительных ресурсов	Да	Да				
Ресурсы — сети	Изменение тегов, диапазонов IP-адресов и IP-адресов сетей	Да					
	Просмотр обнаруженных сетевых ресурсов	Да	Да				
Ресурсы — безопасность	Добавление тегов к обнаруженным группам безопасности	Да					
	Просмотр обнаруженных групп безопасности	Да	Да				
Ресурсы — Хранилище	Добавление тегов к обнаруженному хранилищу	Да					
	Просмотр хранилища	Да	Да				
Ресурсы — Kubernetes	Развертывание или добавление кластеров Kubernetes, а также создание или добавление пространств имен	Да					
	Просмотр кластеров и пространств имен Kubernetes	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Действия — Запросы	Удаление записей запросов на развертывание	Да					
	Просмотр записей запросов на развертывание	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
Действие — журналы событий	Просмотр журналов событий	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
Подключения — Облачные учетные записи	Создание, обновление и удаление облачных учетных записей	Да					
	Просмотр облачных учетных записей	Да	Да				
Подключения — Интеграции	Создание, обновление или удаление интеграций	Да					
	Просмотр интеграций	Да	Да				
Внедрение	Создание, обновление или удаление планов внедрений	Да					
	Просмотр планов внедрений	Да	Да			Да. Проекты	
<b>Расширяемость</b>							
	Просмотр и открытие вкладки «Расширяемость»	Да	Да			Да	
События	Просмотр событий расширяемости	Да	Да				

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Подписки	Создание, обновление и удаление подписок на расширяемость	Да					
	Деактивация подписок	Да					
	Просмотр подписок	Да	Да				
Библиотека — темы событий	Просмотр тем событий	Да	Да				
Библиотека — действия	Создание, обновление и удаление действий по расширению	Да					
	Просмотр действий по расширению	Да	Да				
Библиотека — рабочие процессы	Просмотр рабочих процессов расширяемости	Да	Да				
Действие — запуски действий	Отмена или удаление запусков действий по расширению	Да					
	Просмотр запусков действий по расширению	Да	Да			Да. Проекты	
Действие — запуски рабочих процессов	Просмотр запусков рабочих процессов расширяемости	Да	Да				
<b>Проектирование</b>							
Проектирование	Открытие вкладки «Проектирование»	Да	Да	Да.	Да.	Да.	Да

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Облачные шаблоны	Создание, обновление и удаление облачных шаблонов	Да		Да. Проекты	Да. Проекты		
	Просмотр облачных шаблонов	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
	Загрузка облачных шаблонов	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
	Отправка облачных шаблонов	Да		Да. Проекты	Да. Проекты		
	Развертывание облачных шаблонов	Да		Да. Проекты	Да. Проекты		
	Создание версии и восстановление облачных шаблонов	Да		Да. Проекты	Да. Проекты		
	Выпуск облачных шаблонов в каталог	Да		Да. Проекты	Да. Проекты		
Настраиваемые ресурсы	Создание, обновление и удаление настраиваемых ресурсов	Да					
	Просмотр настраиваемых ресурсов	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
Настраиваемые действия	Создание, обновление и удаление настраиваемых действий	Да					
	Просмотр настраиваемых действий	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
<b>Ресурсы</b>							

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
	Просмотр и открытие вкладки «Ресурсы»	Да	Да	Да	Да	Да	Да
Развертывания	Просмотр развертываний, включая сведения о развертывании, журнал развертывания, цену, мониторинг, оповещения, оптимизацию и информацию об устранении неполадок	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
	Управление оповещениями	Да		Да. Проекты	Да. Проекты		
	Запуск действий по регулярному обслуживанию в развертываниях на основе политик	Да		Да. Проекты	Да. Проекты		
Ресурсы — Все ресурсы	Просмотр всех обнаруженных ресурсов	Да	Да				
	Выполнение действий по регулярному обслуживанию на обнаруженных ресурсах. Действия доступны только на компьютерах и ограничены включением и выключением для всех компьютеров, а также консолью удаленного доступа для компьютеров vSphere.	Да					

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Ресурсы — Все ресурсы	Просмотр развернутых, внедренных и перенесенных ресурсов	Да	Да	Да. Проекты.	Да. Проекты.	Да. Проекты.	
	Выполнение действий по регулярному обслуживанию на развернутых, внедренных и перенесенных ресурсах в соответствии с политиками	Да	Да	Да. Проекты.	Да. Проекты.		
Ресурсы — Виртуальные машины	Просмотр обнаруженных компьютеров	Да	Да				
	Выполнение действий по регулярному обслуживанию на обнаруженных компьютерах. Действия ограничены включением и выключением, а также консолью удаленного доступа для компьютеров vSphere.	Да					
	Создание новой ВМ	Да					
	Просмотр развернутых, внедренных и перенесенных ресурсов.	Да		Да. Проекты.	Да. Проекты.	Да. Проекты.	

Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
	Выполнение действий по регулярному обслуживанию на развернутых, внедренных и перенесенных ресурсах в соответствии с политиками	Да		Да. Проекты.	Да. Проекты.		
Ресурсы — Тома	Просмотр обнаруженных томов	Да	Да				
	Нет доступных действий по регулярному обслуживанию						
	Просмотр развернутых, внедренных и перенесенных томов	Да	Да	Да. Проекты.	Да. Проекты.	Да. Проекты.	
	Выполнение действий по регулярному обслуживанию на развернутых, внедренных и перенесенных томах в соответствии с политиками	Да		Да. Проекты.	Да. Проекты.		
Ресурсы — Сети и безопасность	Просмотр обнаруженных сетей, подсистем балансировки нагрузки и групп безопасности	Да	Да				
	Нет доступных действий по регулярному обслуживанию						



Таблица 3-2. Роли служб и проектов Cloud Assembly (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Cloud Assembly	Наблюдатель Cloud Assembly	Пользователь Cloud Assembly Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором или участником проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
	Просмотр развернутых, внедренных и перенесенных сетей, подсистем балансировки нагрузки и групп безопасности	Да	Да	Да. Проекты.	Да. Проекты.	Да. Проекты.	
	Выполнение действий по регулярному обслуживанию с развернутыми, внедренными и перенесенными сетями, подсистемами балансировки нагрузки и группами безопасности в соответствии с политиками	Да		Да. Проекты.	Да. Проекты.		
<b>Оповещения</b>							
	Просмотр и открытие вкладки «Оповещения»	Да	Да	Да	Да	Да	
	Управление оповещениями	Да		Да. Проекты	Да. Проекты		
	Просмотр оповещений	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	

## Роли службы Service Broker

Роли службы Service Broker определяют отображаемое содержимое и доступные действия в Service Broker. Эти роли службы определяются в консоли владельцем организации.

Таблица 3-3. Описание ролей службы Service Broker

Роль	Описание
Администратор Service Broker	Должен иметь доступ для чтения и записи ко всему пользовательскому интерфейсу и ресурсам API. Это единственная роль пользователя, которая позволяет выполнять все задачи, в том числе создавать новые проекты и назначать администраторов проектов.
Пользователь Service Broker	Любой пользователь, у которого нет роли администратора Service Broker.  В проекте Service Broker администратор добавляет пользователей в качестве участников, администраторов или обозревателей проекта. Администратор также может добавить администратора проекта.
Наблюдатель Service Broker	Пользователь с правом чтения может просматривать сведения, но не может создавать, обновлять или удалять какие-либо значения.  Пользователи с ролью обозревателя могут видеть всю информацию, доступную администратору. Он не может выполнять никакие действия, пока не будет назначен в качестве администратора или участника проекта. Если пользователь связан с проектом, у него есть разрешения, связанные с ролью. Обозреватель проекта не может расширять свои разрешения так же, как администратор или участник.

В дополнение к ролям служб в Service Broker доступны роли проектов. Любой проект доступен во всех службах.

Роли проекта определяются в Service Broker и могут различаться в зависимости от проекта.

В следующих таблицах указаны возможности различных ролей по просмотру и выполнению задач в рамках служб и проектов. Следует помнить, что администраторы служб имеют полный доступ ко всем областям пользовательского интерфейса.

Следующее описание ролей проекта поможет решить, какие разрешения следует предоставить пользователям.

- Администраторы проектов используют инфраструктуру, созданную администратором службы, и обеспечивают наличие ресурсов, необходимых участникам проектов на этапе разработки.
- Участники проектов работают в своих проектах над проектированием и развертыванием облачных шаблонов. В следующей таблице проекты могут включать только ресурсы, которыми вы владеете, или ресурсы, которые используются совместно с другими участниками проекта.
- Наблюдатели проекта имеют ограниченный доступ «только чтение».
- Координаторы проекта являются утверждающими в Service Broker для своих проектов, для которых в политике утверждения определено утверждение координатором проекта. Чтобы предоставить координатору контекст для утверждений, можно также назначить ему роль участника или наблюдателя проекта.

Таблица 3-4. Роли службы и роли проектов Service Broker

Контекст пользовательского интерфейса	Задача	Администратор	Наблюдатель	Пользователь Service Broker			
		Service Broker	Service Broker	Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Доступ к Service Broker							
Консоль	Консоль позволяет найти и открыть Service Broker	Да	Да	Да	Да	Да	Да
Инфраструктура							
	Просмотр и открытие вкладки «Инфраструктура»	Да	Да				
Настройка — Проекты	Создание проектов	Да					
	Обновление или удаление значений сводки проекта, параметров предоставления, Kubernetes, интеграций и конфигураций тестовых проектов.	Да					
	Добавление пользователей и групп, назначение ролей в проектах.	Да		Да. Проекты.			
	Просмотр проектов.	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
Настройка — Облачные зоны	Создание, обновление и удаление облачных зон	Да					
	Просмотр облачных зон.	Да	Да				
Настройка — зоны Kubernetes	Создание, обновление и удаление зон Kubernetes	Да					
	Просмотр зон Kubernetes	Да	Да				

Таблица 3-4. Роли службы и роли проектов Service Broker (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Service Broker	Наблюдатель Service Broker	Пользователь Service Broker Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Подключения — Облачные учетные записи	Создание, обновление и удаление облачных учетных записей	Да					
	Просмотр облачных учетных записей	Да	Да				
Подключения — Интеграции	Создание, обновление или удаление интеграций	Да					
	Просмотр интеграций	Да	Да				
Действия — Запросы	Удаление записей запросов на развертывание	Да					
	Просмотр записей запросов на развертывание	Да					
Действие — журналы событий	Просмотр журналов событий	Да					
<b>Содержимое и политики</b>							
	Просмотр и открытие вкладки «Содержимое и политики»	Да	Да				
Источники содержимого	Создание, обновление и удаление источников содержимого	Да					
	Просмотр источников содержимого	Да	Да				
Общий доступ к содержимому	Добавление или удаление общего содержимого	Да					
	Просмотр общего содержимого	Да	Да				

Таблица 3-4. Роли службы и роли проектов Service Broker (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Service Broker	Наблюдатель Service Broker	Пользователь Service Broker Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Содержимое	Настройка форм и настройка элементов	Да					
	Просмотр содержимого	Да	Да				
Политики — определения	Создание, обновление и удаление определений политик	Да					
	Просмотр определений политик	Да	Да				
Политики — применение	Просмотр журнала применения	Да	Да				
Уведомления — почтовый сервер	Настройка почтового сервера	Да					
<b>Каталог</b>							
	Просмотр и открытие вкладки «Каталог»	Да	Да	Да	Да	Да	Да
	Просмотр доступных элементов каталога	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
	Запрос элемента каталога	Да		Да. Проекты	Да. Проекты		
<b>Ресурсы</b>							
	Просмотр и открытие вкладки «Ресурсы»	Да	Да	Да.	Да	Да	Да

Таблица 3-4. Роли службы и роли проектов Service Broker (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Service Broker	Наблюдатель Service Broker	Пользователь Service Broker Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Развертывания	Просмотр развертываний, включая сведения о развертывании, журнал развертывания, цену, мониторинг, оповещения, оптимизацию и информацию об устранении неполадок	Да	Да	Да. Проекты	Да. Проекты	Да. Проекты	
	Управление оповещениями	Да		Да. Проекты	Да. Проекты		
	Запуск действий по регулярному обслуживанию в развертываниях на основе политик	Да		Да. Проекты	Да. Проекты		
Ресурсы — Все ресурсы	Просмотр всех обнаруженных ресурсов	Да	Да				
	Выполнение действий по регулярному обслуживанию на обнаруженных ресурсах. Действия доступны только на компьютерах и ограничены включением и выключением для всех компьютеров, а также консолью удаленного доступа для компьютеров vSphere.	Да					

Таблица 3-4. Роли службы и роли проектов Service Broker (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Service Broker	Наблюдатель Service Broker	Пользователь Service Broker Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
Ресурсы — Все ресурсы	Просмотр развернутых, внедренных и перенесенных ресурсов	Да	Да	Да. Проекты.	Да. Проекты.	Да. Проекты.	
	Выполнение действий по регулярному обслуживанию на развернутых, внедренных и перенесенных ресурсах в соответствии с политиками	Да	Да	Да. Проекты.	Да. Проекты.		
Ресурсы — Виртуальные машины	Просмотр обнаруженных компьютеров	Да	Да				
	Выполнение действий по регулярному обслуживанию на обнаруженных компьютерах. Действия ограничены включением и выключением, а также консолью удаленного доступа для компьютеров vSphere.	Да					
	Создание новой VM	Да					
	Просмотр развернутых, внедренных и перенесенных ресурсов.	Да		Да. Проекты.	Да. Проекты.	Да. Проекты.	

Таблица 3-4. Роли службы и роли проектов Service Broker (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Service Broker	Наблюдатель Service Broker	Пользователь Service Broker Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
	Выполнение действий по регулярному обслуживанию на развернутых, внедренных и перенесенных ресурсах в соответствии с политиками	Да		Да. Проекты.	Да. Проекты.		
Ресурсы — Тома	Просмотр обнаруженных томов	Да	Да				
	Нет доступных действий по регулярному обслуживанию						
	Просмотр развернутых, внедренных и перенесенных томов	Да	Да	Да. Проекты.	Да. Проекты.	Да. Проекты.	
	Выполнение действий по регулярному обслуживанию на развернутых, внедренных и перенесенных томах в соответствии с политиками	Да		Да. Проекты.	Да. Проекты.		
Ресурсы — Сети и безопасность	Просмотр обнаруженных сетей, подсистем балансировки нагрузки и групп безопасности	Да	Да				
	Нет доступных действий по регулярному обслуживанию						



Таблица 3-4. Роли службы и роли проектов Service Broker (продолжение)

Контекст пользовательского интерфейса	Задача	Администратор Service Broker	Наблюдатель Service Broker	Пользователь Service Broker Для просмотра и выполнения задач, связанных с проектом, пользователь должен быть администратором проекта.			
				Администратор проекта	Участник проекта	Наблюдатель проекта	Координатор проекта
	Просмотр развернутых, внедренных и перенесенных сетей, подсистем балансировки нагрузки и групп безопасности	Да	Да	Да. Проекты.	Да. Проекты.	Да. Проекты.	
	Выполнение действий по регулярному обслуживанию с развернутыми, внедренными и перенесенными сетями, подсистемами балансировки нагрузки и группами безопасности в соответствии с политиками	Да		Да. Проекты.	Да. Проекты.		
<b>Подтверждения</b>							
	Просмотр и открытие вкладки «Подтверждения»	Да	Да	Да	Да	Да	Да
	Ответ на запросы подтверждения	Да		Да. Утверждающим проектов и политик является администратор проекта	Только назначенные утверждающие	Только назначенные утверждающие	Да. Утверждающим проектов и политик является координатор проекта

## Роли службы Code Stream

Роли службы Code Stream определяют отображаемое содержимое и доступные действия в Code Stream. Эти роли определяются на консоли владельцем организации. Любой проект доступен во всех службах.

Таблица 3-5. Описание ролей службы Code Stream

Роль	Описание
Администратор Code Stream	Пользователь должен иметь доступ для чтения и записи ко всему пользовательскому интерфейсу и ресурсам API. Это единственная роль пользователя, которая позволяет просматривать и выполнять все операции, включая создание проектов, интеграцию конечных точек, добавление триггеров, создание конвейеров и настраиваемых панелей управления, маркировку конечных точек и переменных как ресурсы с ограниченным доступом, запуск конвейеров, использующих такие ресурсы, а также запрашивать публикацию конвейеров в Service Broker.
Разработчик Code Stream	Пользователь, который может работать с конвейерами, но не с конечными точками или переменными с ограниченным доступом. Если конвейер содержит конечную точку или переменную с ограниченным доступом, этот пользователь должен получить утверждение задачи конвейера, которая использует эту конечную точку или переменную.
Исполнитель Code Stream	Пользователь, который может запускать конвейеры и утверждать или отклонять задачи, выполняемые пользователями. Этот пользователь может возобновить, приостановить и отменить выполнение конвейера, но не может их изменить.
Пользователь Code Stream	Пользователь, у которого есть доступ к Code Stream, но нет других прав в Code Stream.
Обозреватель Code Stream	Пользователь, который имеет доступ с правами чтения для просмотра конвейеров, конечных точек, циклов выполнения конвейера и панелей управления, но не может создавать, обновлять или удалять их. Пользователь, у которого есть роль «Обозреватель службы», может видеть всю информацию, доступную администратору. Он не может выполнять никакие действия, пока не будет назначен в качестве администратора или участника проекта. Если пользователь связан с проектом, у него есть разрешения, связанные с ролью. Обозреватель проекта не может расширять свои разрешения так же, как администратор или участник.

В дополнение к ролям служб в Code Stream доступны роли проектов. Любой проект доступен во всех службах.

Роли проекта определяются в Code Stream и могут различаться в зависимости от проекта.

Просматривая следующие таблицы, в которых указываются возможности просмотра и выполнения для различных ролей служб и проектов, следует помнить, что администраторы служб имеют полный доступ ко всем областям пользовательского интерфейса.

Чтобы определить, какие разрешения следует предоставить пользователям, используйте следующие описания ролей проекта.

- Администраторы проектов используют инфраструктуру, созданную администратором службы, и обеспечивают наличие ресурсов, необходимых участникам проектов на этапе разработки. Администратор проекта может добавлять участников.
- Участники проекта, у которых есть роль службы, могут использовать службы.
- Обозреватели проектов могут просматривать проекты, но не могут создавать, обновлять или удалять их.

Все, за исключением действий с ограниченным доступом: в этой роли разрешено выполнять действия по созданию, чтению, обновлению и удалению объектов, кроме переменных и конечных точек с ограниченным доступом.

Таблица 3-6. Возможности роли службы Code Stream

Контекст пользовательского интерфейса	Возможности	Роль администратора Code Stream	Роль разработчика Code Stream	Роль исполнителя Code Stream	Роль обозревателя Code Stream	Роль пользователя Code Stream
<b>Конвейеры</b>						
	Просмотр конвейеров	Да	Да	Да	Да	
	Создание конвейеров	Да	Да			
	Запуск конвейеров	Да	Да	Да		
	Запуск работы конвейеров, содержащих конечные точки и переменные с ограниченным доступом	Да				
	Обновление конвейеров	Да	Да			
	Удаление конвейеров	Да	Да			
<b>Циклы выполнения конвейера</b>						
	Просмотр циклов выполнения конвейера	Да	Да	Да	Да	
	Возобновление, приостановка и отмена циклов выполнения конвейера	Да	Да	Да		
	Возобновление работы конвейеров, приостановленных для утверждения использования ресурсов с ограниченным доступом	Да				
<b>Настраиваемые интеграции</b>						

Таблица 3-6. Возможности роли службы Code Stream (продолжение)

Контекст пользовательского интерфейса	Возможности	Роль администратора Code Stream	Роль разработчика Code Stream	Роль исполнителя Code Stream	Роль обозревателя Code Stream	Роль пользователя Code Stream
	Создание настраиваемых интеграций	Да	Да			
	Чтение настраиваемых интеграций	Да	Да	Да	Да	
	Обновление настраиваемых интеграций	Да	Да			
<b>Конечные точки</b>						
	Просмотр циклов выполнения	Да	Да	Да	Да	
	Создание циклов выполнения	Да	Да			
	Обновление циклов выполнения	Да	Да			
	Удаление циклов выполнения	Да	Да			
<b>Пометить как ресурсы с ограниченным доступом</b>						
	Пометка конечной точки или переменной как ресурса с ограниченным доступом	Да				
<b>Панели управления</b>						
	Просмотр панелей управления	Да	Да	Да	Да	
	Создание панелей управления	Да	Да			

Таблица 3-6. Возможности роли службы Code Stream (продолжение)

Контекст пользовательского интерфейса	Возможности	Роль администратора Code Stream	Роль разработчика Code Stream	Роль исполнителя Code Stream	Роль обозревателя Code Stream	Роль пользователя Code Stream
	Обновление панелей управления	Да	Да			
	Удаление панелей управления	Да	Да			

## Роли службы vRA Migration Assistant

Роли службы vRA Migration Assistant определяют отображаемое содержимое и доступные действия в службе vRA Migration Assistant и в Cloud Assembly. Эти роли службы определяются в консоли владельцем организации.

Таблица 3-7. Описание ролей службы vRealize Automation vRA Migration Assistant

Роль	Описание
Администратор службы Migration Assistant	Пользователь с полными правами на просмотр, обновление и удаление в vRA Migration Assistant и Cloud Assembly. У этой роли также должна быть по крайней мере роль «Обозреватель Cloud Assembly».
Наблюдатель службы Migration Assistant	Пользователь с правом чтения может просматривать сведения, но не может создавать, обновлять или удалять какие-либо значения в vRA Migration Assistant или в Cloud Assembly. У этой роли также должна быть по крайней мере роль «Обозреватель Cloud Assembly».

## Роли службы Orchestrator

Роли службы Orchestrator определяют отображаемое содержимое и доступные действия в клиенте vRealize Orchestrator. Эти роли службы определяются в консоли владельцем организации.

Таблица 3-8. Описание ролей службы vRealize Orchestrator

Роль	Описание
Администратор Orchestrator	Пользователь с полными правами на просмотр, обновление и удаление в службе vRealize Orchestrator. Администратор также может получить доступ к содержимому, созданному отдельными группами.
Наблюдатель службы Orchestrator	Пользователь, который имеет доступ с правами на чтение для просмотра компонентов и содержимого, включая содержимое всех групп, но не может создавать, обновлять, запускать, удалять значения или экспортировать содержимое.
Конструктор рабочих процессов Orchestrator	Пользователь, который может создавать, запускать, изменять и удалять собственное содержимое клиента vRealize Orchestrator. Он может добавить собственное содержимое в свою назначенную группу. Конструктор рабочих процессов не имеет доступа к функциям клиента vRealize Orchestrator по администрированию и устранению неполадок.

## Роль службы SaltStack Config

Роль службы SaltStack Config определяет отображаемое содержимое и доступные действия в vRealize Automation. Эта роль службы определяется в консоли владельцем организации.

Таблица 3-9. Описание ролей службы vRealize Automation SaltStack Config

Роль	Описание
Администратор SaltStack Config	Пользователь, который может получить доступ к плитке SaltStack Config в консоли, если настроена интеграция с Cloud Assembly. Чтобы войти в экземпляр SaltStack Config, пользователь должен иметь права администратора SaltStack, которые определены в SaltStack Config. Пользователь также должен иметь роль администратора Cloud Assembly.

## Настраиваемые роли пользователей в vRealize Automation

Администратор Cloud Assembly может создавать настраиваемые роли, определяющие пользователей, которые могут просматривать действия и выполнять их в vRealize Automation. Затем эти роли можно назначать пользователям.

## Разрешения настраиваемой роли пользователя

С помощью Cloud Assembly можно определить более детализированные роли пользователей, а затем назначить пользователям эти роли. Настраиваемые роли имеют две категории: «просмотр» и «управление».

- **Просмотр.** Пользователь, которому назначена роль с таким разрешением, может просматривать все элементы во всех проектах в выбранных разделах пользовательского интерфейса. Эта роль полезна для пользователей, которым необходимо просматривать учетные записи, конфигурации или назначенные значения.
- **Управление.** Пользователь, которому назначена роль с таким разрешением, может просматривать все элементы и имеет полные разрешения на добавление, изменение и удаление для всех проектов в выбранных разделах пользовательского интерфейса.

Эти разрешения расширяют права, предоставляемые другими ролями, и не ограничиваются членством в проекте. Например, можно расширить разрешения администратора проекта для управления отдельными частями инфраструктуры или предоставить обозревателю возможность просмотра запросов на утверждение и реагирования на них.

Чтобы определить роли пользователей и назначить пользователей, откройте Cloud Assembly или Service Broker как администратор службы и выберите **Инфраструктура > Администрирование > Настраиваемые роли**. Настраиваемые роли в Code Stream настраивать нельзя, но эти роли применяются ко всем службам.

Таблица 3-10. Настраиваемые роли

Пользовательский интерфейс	Разрешение	Описание
<b>Инфраструктура</b>		
	Просмотр облачных учетных записей.	Просмотр облачных учетных записей.
	Управление облачными учетными записями	Создание, обновление или удаление облачных учетных записей.
	Просмотр сопоставлений образов	Просмотр сопоставлений образов.
	Управление сопоставлениями образов	Создание, обновление или удаление сопоставлений образов.
	Просмотр сопоставлений конфигураций ресурсов	Просмотр сопоставлений конфигураций ресурсов.
	Управление сопоставлениями конфигураций ресурсов	Создание, обновление или удаление сопоставлений конфигураций ресурсов.
	Просмотр облачных зон	Просмотр облачных зон, ключевых сведений и оповещений.
	Управление облачными зонами	Создание, обновление или удаление облачных зон. Управление оповещениями.
	Просмотр запросов	Просмотр запросов на действия.
	Управление запросами	Удаление запросов из списка.

Таблица 3-10. Настраиваемые роли (продолжение)

Пользовательский интерфейс	Разрешение	Описание
	Просмотр интеграций	Просмотр интеграций.
	Управление интеграциями	Создание, обновление или удаление интеграций.
	Просмотр проектов	Просмотр проектов.
	Управление проектами	Создание проектов. Добавление пользователей и назначение ролей в проектах. Обновление или удаление значений сводки проекта, пользователей, параметров предоставления, Kubernetes, интеграций и конфигураций тестовых проектов.
	Просмотр планов внедрений	Просмотр планов внедрений
	Управление планами по внедрению	Создание, обновление, запуск и удаление планов внедрения
<b>Каталог</b>		
	Просмотр содержимого	
	Управление содержимым	Добавление, обновление и удаление источников содержимого. Общий доступ к содержимому. Настройка содержимого, включая значки каталога и формы запросов.
<b>Политики</b>		
	Просмотр политик	Просмотр определений политик.
	Управление политиками	Создание, обновление или удаление определений политик.
<b>Развертывания</b>		
	Просмотр развертываний	Просмотр всех развертываний, включая сведения о развертывании, журнал развертывания, оповещения и информацию об устранении неполадок.
	Управление развертываниями	Просмотр всех развертываний, реагирование на оповещения и выполнение всех действий по регулярному обслуживанию, которые администраторы могут выполнять с развертываниями и их компонентами согласно политикам регулярного обслуживания.
<b>Облачные шаблоны</b>		



Таблица 3-10. Настраиваемые роли (продолжение)

Пользовательский интерфейс	Разрешение	Описание
	Просмотр облачных шаблонов	Просмотр облачных шаблонов.
	Управление облачными шаблонами	Создание, обновление, тестирование, удаление, создание версий, предоставление общего доступа к облачным шаблонам и публикация/отмена публикации версий облачных шаблонов.
	Изменение облачных шаблонов	Создание, обновление, тестирование, создание версий, предоставление общего доступа к облачным шаблонам и публикация/отмена публикации версий облачного шаблона Роль не имеет разрешения на удаление облачных шаблонов.
	Развертывание облачных шаблонов	Тестирование и развертывание любого облачного шаблона в любом проекте.
	Развертывание содержимого встроенного облачного шаблона	Развертывание любого облачного шаблона в проектах, с которыми связаны уполномоченные. Ролями проекта могут быть «администратор», «участник» или «обозреватель».
<b>Любой ресурс как услуга</b>		
	Просмотр настраиваемых ресурсов	Просмотр настраиваемых ресурсов.
	Управление настраиваемыми ресурсами	Создание, обновление и удаление настраиваемых ресурсов.
	Просмотр действий с ресурсами	Просмотр настраиваемых действий.
	Управление действиями с ресурсами	Создание, обновление и удаление настраиваемых действий
<b>Расширяемость</b>		
	Просмотр ресурсов расширяемости	Просмотр событий, подписок, тем событий, действий, рабочих процессов, запусков действий и циклов выполнения рабочих процессов.
	Управление ресурсами расширяемости	Создание, обновление, удаление и деактивация подписок с поддержкой расширяемости Создание, обновление или удаление действий расширяемости. Отмена или удаление циклов запуска действий расширяемости.
<b>Конвейер</b>		

Таблица 3-10. Настраиваемые роли (продолжение)

Пользовательский интерфейс	Разрешение	Описание
	Управление конвейерами	Создание, изменение и удаление конфигураций конвейера, конечной точки, переменной и триггера. Модели с ограниченным доступом исключены.
	Управление конвейерами с ограниченным доступом	Создание, изменение и удаление конфигураций конвейера, конечной точки, переменной и триггера. Модели с ограниченным доступом включены.
	Управление настраиваемыми интеграциями	Добавление, изменение и удаление настраиваемых интеграций.
	Выполнение конвейеров	Запуск циклов выполнения моделей и триггеров модели конвейера, а также остановка, отмена, возобновление или повторный запуск циклов выполнения и триггеров.
	Выполнение конвейеров с ограниченным доступом	Запуск циклов выполнения моделей и триггеров модели конвейера, а также остановка, отмена, возобновление или повторный запуск циклов выполнения и триггеров. Разрешение конечных точек и переменных с ограниченным доступом.
	Управление циклами выполнения	Запуск циклов выполнения моделей и триггеров модели конвейера, а также остановка, отмена, возобновление или повторный запуск циклов выполнения и триггеров. Разрешение конечных точек и переменных с ограниченным доступом. Удаление циклов выполнения.
<b>Утверждение</b>		
	Управление утверждениями	Просмотр вкладки «Утверждения», где можно утверждать или отклонять запросы на утверждение. Утверждающий с этой ролью не будет получать уведомление по электронной почте о запросе на утверждение, если он не является утверждающим в рамках политики.

## Примеры использования. Роли пользователей как средство управления доступом в vRealize Automation

Администратор облачных служб контролирует, какие задачи могут выполняться пользователями в vRealize Automation. В зависимости от целей управления и сфер ответственности в группе разработчиков приложений существуют различные способы настройки ролей пользователей для поддержки этих целей.

Следующие примеры для Cloud Assembly и Service Broker построены на трех вариантах использования. Эти примеры содержат достаточное количество инструкций для демонстрации применения ролей пользователей.

Эти примеры использования предназначены для администратора облачных служб и администраторов служб.

Каждый последующий пример использования составлен на основе предыдущего. Даже если вы сразу готовы перейти к примеру 3, возможно, вам потребуется ознакомиться с примерами 1 и 2, чтобы лучше понять, почему следует настраивать роли указанным образом.

Задача этих примеров использования — показать применение ролей пользователей, а не предоставить подробную информацию о настройке инфраструктуры, управлении проектами, создании облачных шаблонов и работе с развертываниями.

Прежде чем начать, необходимо ознакомиться с уровнями ролей пользователей, настроенными администратором облачных служб в консоли vRealize Automation.

### ■ Роли организации

Роли организации определяют, кто может иметь доступ к консоли.

Владельцу организации необходимо проследить, чтобы пользователям всех служб была назначена как минимум роль участника организации.

Роль	Описание
Владелец организации	Администратор может добавлять пользователей, изменять роли пользователей и удалять пользователей из организации. Владелец управляет доступом пользователей к службам.
Участник организации	Обычный пользователь может войти в консоль организации. Чтобы пользователь мог получить доступ к службам, владелец организации назначает ему определенную роль службы.

### ■ Роли служб

Роли служб определяют, кто может иметь доступ к назначенным службам.

Владельцу организации необходимо проследить, чтобы всем пользователям, которым требуется доступ к службам, были назначены соответствующие роли. Роли определяют, какие действия пользователь может выполнять в каждой службе.

**Таблица 3-11. Описание ролей службы Cloud Assembly**

Роль	Описание
Администратор Cloud Assembly	Пользователь должен иметь доступ для чтения и записи ко всему пользовательскому интерфейсу и ресурсам API. Это единственная роль пользователя, которая позволяет просматривать и выполнять все действия, в том числе добавлять облачные учетные записи, создавать новые проекты и назначать администратора проекта.
Пользователь Cloud Assembly	Пользователь, у которого нет роли администратора Cloud Assembly.  В проекте Cloud Assembly администратор добавляет пользователей в качестве участников, администраторов или обозревателей проекта. Администратор также может добавить администратора проекта.
Наблюдатель Cloud Assembly	Пользователь с правом чтения может просматривать сведения, но не может создавать, обновлять или удалять какие-либо значения. Это роль «только для чтения» для всех проектов.  Пользователи с ролью обозревателя могут видеть всю информацию, доступную администратору. Он не может выполнять никакие действия, пока не будет назначен в качестве администратора или участника проекта. Если пользователь связан с проектом, у него есть разрешения, связанные с ролью. Обозреватель проекта не может расширять свои разрешения так же, как администратор или участник.

**Таблица 3-12. Описание ролей службы Service Broker**

Роль	Описание
Администратор Service Broker	Должен иметь доступ для чтения и записи ко всему пользовательскому интерфейсу и ресурсам API. Это единственная роль пользователя, которая позволяет выполнять все задачи, в том числе создавать новые проекты и назначать администраторов проектов.
Пользователь Service Broker	Любой пользователь, у которого нет роли администратора Service Broker.  В проекте Service Broker администратор добавляет пользователей в качестве участников, администраторов или обозревателей проекта. Администратор также может добавить администратора проекта.
Наблюдатель Service Broker	Пользователь с правом чтения может просматривать сведения, но не может создавать, обновлять или удалять какие-либо значения.

Таблица 3-12. Описание ролей службы Service Broker (продолжение)

Роль	Описание
	Пользователи с ролью обозревателя могут видеть всю информацию, доступную администратору. Он не может выполнять никакие действия, пока не будет назначен в качестве администратора или участника проекта. Если пользователь связан с проектом, у него есть разрешения, связанные с ролью. Обозреватель проекта не может расширять свои разрешения так же, как администратор или участник.

Таблица 3-13. Описание ролей службы Code Stream

Роль	Описание
Администратор Code Stream	Пользователь должен иметь доступ для чтения и записи ко всему пользовательскому интерфейсу и ресурсам API. Это единственная роль пользователя, которая позволяет просматривать и выполнять все операции, включая создание проектов, интеграцию конечных точек, добавление триггеров, создание конвейеров и настраиваемых панелей управления, маркировку конечных точек и переменных как ресурсы с ограниченным доступом, запуск конвейеров, использующих такие ресурсы, а также запрашивать публикацию конвейеров в Service Broker.
Разработчик Code Stream	Пользователь, который может работать с конвейерами, но не с конечными точками или переменными с ограниченным доступом. Если конвейер содержит конечную точку или переменную с ограниченным доступом, этот пользователь должен получить утверждение задачи конвейера, которая использует эту конечную точку или переменную.
Исполнитель Code Stream	Пользователь, который может запускать конвейеры и утверждать или отклонять задачи, выполняемые пользователями. Этот пользователь может возобновить, приостановить и отменить выполнение конвейера, но не может их изменить.
Пользователь Code Stream	Пользователь, у которого есть доступ к Code Stream, но нет других прав в Code Stream.
Обозреватель Code Stream	Пользователь, который имеет доступ с правами чтения для просмотра конвейеров, конечных точек, циклов выполнения конвейера и панелей управления, но не может создавать, обновлять или удалять их. Пользователь, у которого есть роль «Обозреватель службы», может видеть всю информацию, доступную администратору. Он не может выполнять никакие действия, пока не будет назначен в качестве администратора или участника проекта. Если пользователь связан с проектом, у него есть разрешения, связанные с ролью. Обозреватель проекта не может расширять свои разрешения так же, как администратор или участник.

#### ■ Роли участников проекта

Членство в проекте определяет, какие ресурсы инфраструктуры и облачные шаблоны доступны пользователю.

Членство в проекте определяется в службе пользователем с ролью администратора службы.

Администратор службы должен убедиться, что пользователям, которым необходим доступ к одному проекту или нескольким, назначены соответствующие роли на уровне каждого проекта.

Таблица 3-14. Роли в проекте

Роль	Описание
Администратор проекта	Администратор проекта может управлять своими проектами, создавать и развертывать облачные шаблоны, связанные с его проектами, а также управлять развертываниями проектов для всех участников проекта.
Участник проекта	Участник проекта может создавать и развертывать облачные шаблоны, связанные с соответствующими проектами, управлять собственными развертываниями, а также всеми общедоступными развертываниями.
Наблюдатель проекта	Обозреватель проекта — это участник проекта, который имеет доступ в режиме «только чтение» к ресурсам проекта, облачным шаблонам и развертываниям.

#### ■ Настраиваемые роли

Cloud Assembly создает настраиваемые роли для уточнения ролей участников и обозревателей.

Процедуры, описанные в этих примерах использования, предназначены для демонстрации применения ролей пользователей. Они не являются подробными или исчерпывающими процедурами по настройке vRealize Automation.

При настройке ролей следует помнить, что для пользователей, выполняющих операции API, применяются роли, назначаемые здесь.

#### Необходимые условия

- Убедитесь, что вам назначена роль владельца организации. При входе в консоль вам должна быть видна вкладка **Управление идентификацией и доступом**. Если это не так, обратитесь к владельцу организации.
- Убедитесь, что вам назначена роль администратора службы для разных служб. Если вы не уверены в своей роли, свяжитесь с владельцем организации.
- Убедитесь, что пользователи добавлены в vRealize Automation.

Пользователи Active Directory добавляются в ходе установки vRealize Automation.

- Подробный список задач и функций различных ролей см. в разделе [Роли пользователей организаций и служб в vRealize Automation](#).

## Процедура

### 1. Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений

Администратор облачных служб vRealize Automation отвечает за управление доступом к ресурсам инфраструктуры и бюджетом на них. Добавьте себя и двух других пользователей в качестве администраторов. Эта небольшая рабочая группа может создать инфраструктуру и разработать облачные шаблоны, которые отвечают бизнес-целям команд, использующих облачные шаблоны. Затем небольшая рабочая группа администраторов развертывает облачные шаблоны для потребителей, не являющихся администраторами. Пользователям, не являющимся администраторами, не предоставляется доступ к vRealize Automation.

### 2. Пример использования роли пользователя 2: настройка ролей пользователей vRealize Automation для поддержки более крупных команд разработчиков и каталога

Владелец организации vRealize Automation отвечает за управление доступом к ресурсам инфраструктуры и бюджетом на них. Есть группа разработчиков облачных шаблонов, которые создают и развертывают шаблоны для разных проектов в несколько итераций, пока они не будут готовы для предоставления потребителям. После этого пользователям в каталоге можно предоставить развертываемые ресурсы.

### 3. Пример использования роли пользователя 3. Определение настраиваемых ролей пользователей vRealize Automation для уточнения системных ролей

Владелец организации vRealize Automation или администратор службы могут управлять доступом пользователей с помощью системных ролей организации и служб. Тем не менее также может потребоваться создание настраиваемых ролей для выбранных пользователей и выполнение задач или просмотр содержимого за пределами их системных ролей.

## Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений

Администратор облачных служб vRealize Automation отвечает за управление доступом к ресурсам инфраструктуры и бюджетом на них. Добавьте себя и двух других пользователей в качестве администраторов. Эта небольшая рабочая группа может создать инфраструктуру и разработать облачные шаблоны, которые отвечают бизнес-целям команд, использующих облачные шаблоны. Затем небольшая рабочая группа администраторов развертывает облачные шаблоны для потребителей, не являющихся администраторами. Пользователям, не являющимся администраторами, не предоставляется доступ к vRealize Automation.

В этом примере вы являетесь владельцем организации и у вас есть небольшая рабочая группа, все пользователи которой имеют роль администратора службы.

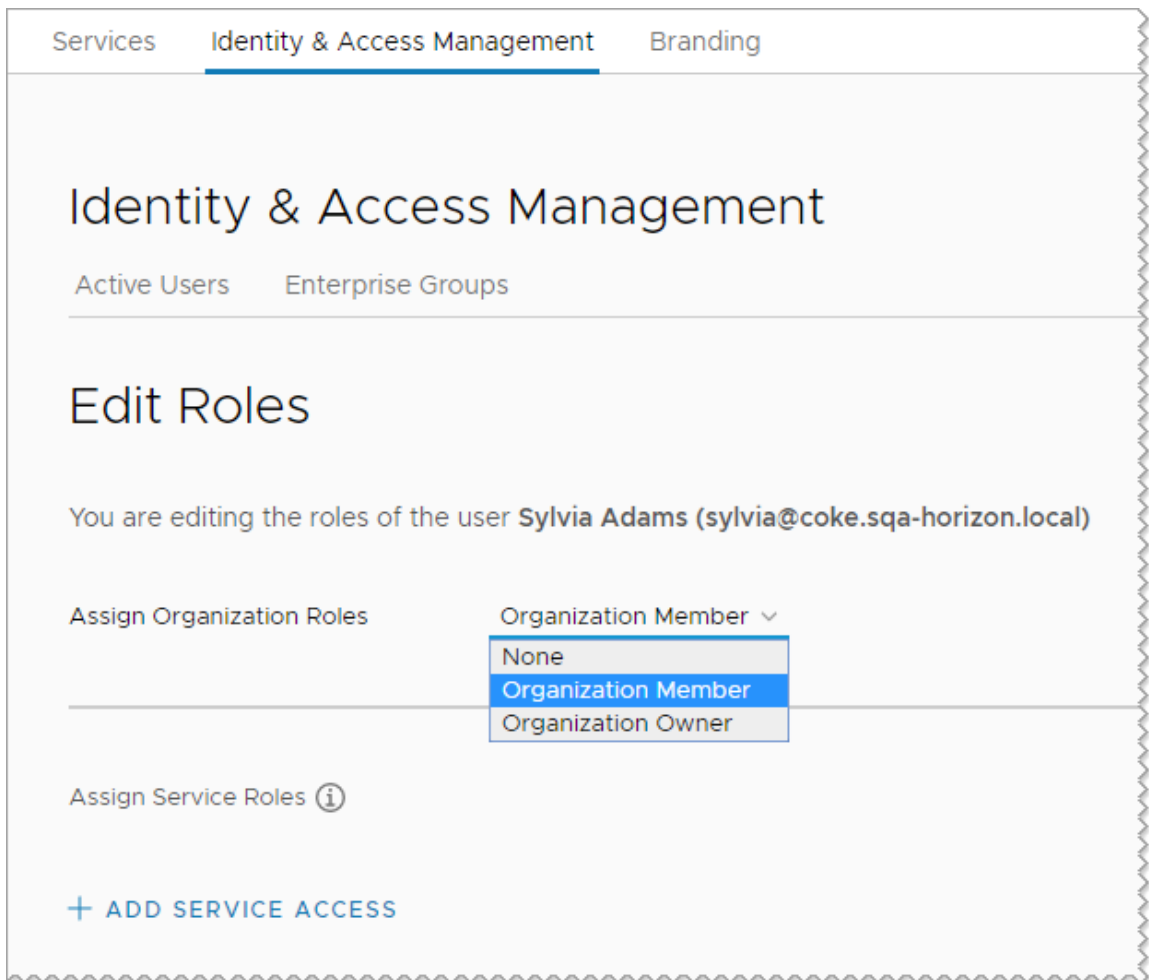
В следующей процедуре описано прохождение одного пользователя через весь процесс. Каждый шаг можно выполнять в отношении нескольких пользователей.

## Необходимые условия

- Убедитесь, что выполнены все предварительные условия, указанные во введении к примеру использования. См. раздел [Примеры использования. Роли пользователей как средство управления доступом в vRealize Automation](#).

## Процедура

1. Назначение ролей организации. Нажмите **Управление идентификацией и доступом**.
  - а) Выполните вход в консоль vRealize Automation.
  - б) Нажмите **Управление идентификацией и доступом**.
  - в) Выберите имя пользователя и щелкните **Изменить роли**.
  - г) В раскрывающемся меню **Назначение ролей организации** выберите **Участник организации**.



Пользователь с ролью участника организации имеет доступ к консоли и любым службам, в которые он добавлен. Эта роль не позволяет управлять пользователями организации.

Оставьте открытой страницу «Изменить роль» для этого пользователя и перейдите к следующему шагу.



2. Назначьте роль администратора Cloud Assembly себе и одному или двум другим пользователям, являющимся администраторами в этом сценарии.

Пользователь с ролью администратора службы имеет все права и может добавлять, изменять и удалять инфраструктуру, проекты, облачные шаблоны и развертывания. Как назначить роль администратора для одного человека и роль пользователя для другого рассматривается в сценарии 2. В этом примере используется Сильвия.

- а) Щелкните **Добавить доступ к службе**.
- б) Настройте пользователя со следующим значением.

Служба	Роль
Cloud Assembly	Администратор Cloud Assembly

[Services](#)
[Identity & Access Management](#)
[Branding](#)

### Identity & Access Management

Active Users   Enterprise Groups

#### Edit Roles

You are editing the roles of the user **Sylvia Adams** (sylvia@coke.sqa-horizon.local)

Assign Organization Roles   Organization Member ▾

---

Assign Service Roles ⓘ

Cloud Assembly ▾   with roles   Cloud Assembly Administrator ▾   ×

[+ ADD SERVICE ACCESS](#)

**SAVE**   **CANCEL**

3. Создайте проект в Cloud Assembly, который будет использоваться для группировки ресурсов и управления выставлением счетов за их использование различным бизнес-группам.

- а) В консоли перейдите на вкладку **Службы** и выберите **Cloud Assembly**.
- б) Выберите **Инфраструктура > Проекты > Создать проект**.

В этом примере настройки ролей пользователей основное внимание уделяется способам реализации ролей пользователей, а не созданию полностью определенной системы.

Сведения о настройке инфраструктуры см. в разделе [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#). Дополнительные сведения о проектах см. в разделе [Глава 5 Добавление проектов Cloud Assembly и управление ими](#).

- в) В качестве имени проекта введите **WebAppTeam**.

- г) Нажмите кнопку **Пользователи** и выберите **Добавить пользователей**.
- д) Введите адреса электронной почты лиц, которые могут помочь при сборке инфраструктуры и облачных шаблонов, а также при управлении ими.

Например, tony@mycompany.com, syliva@mycompany.com.

- е) В раскрывающемся меню **Назначить роль** выберите **Администратор**.

Как у администраторов Cloud Assembly у этих пользователей уже есть доступ к облачным учетным записям, инфраструктуре и всем проектам с правами администратора. Этот шаг помогает понять особенности ролей, используемых в следующих сценариях. В следующих сценариях определяются роли администраторов и участников проекта, которым предоставляются разные разрешения.

- ж) Перейдите на вкладку **Подготовка** и добавьте одну или несколько облачных зон.

Напомним, этот пример использования описывает роли пользователей.

#### 4. Разработайте простой облачный шаблон, позволяющий протестировать проект WebAppTeam.

Данный раздел облачного шаблона укорочен. Основное внимание здесь уделено не созданию облачного шаблона, а пользователям и ролям пользователей, определенным в проектах.

- а) Выберите **Облачные шаблоны > Создать**.
- б) Для нового облачного шаблона введите имя **WebApp**.
- в) В качестве **проекта** выберите WebAppTeam.

New Cloud Template

Name \* WebApp

Description

Project \* WebAppTeam

Cloud template sharing in Service Broker

☒ Share only with this project

☐ Allow an administrator to share with any project in this organization

CANCEL CREATE

- г) Установите флажок **Общий доступ только для проекта**.

За счет этой настройки облачный шаблон будет доступен только участникам проекта. Когда потребуется предоставить облачные шаблоны другим рабочим группам, можно установить флажок «Разрешить администратору предоставлять общий доступ любому проекту в этой организации». Возможность предоставлять другим проектам общий доступ к облачному шаблону избавляет от необходимости поддерживать дублирующиеся экземпляры одинаковых базовых шаблонов. Облачные шаблоны можно перемещать из проектов разработки в производственные проекты, чтобы потребители каталога могли выполнять развертывания на ресурсах производственной инфраструктуры.

- д) Щелкните **Создать**.
- е) В конструкторе облачных шаблонов перетащите на холст компонент **Не зависящий от облака > Компьютер**.

Дополнительные сведения о настройке облачных шаблонов см. в разделе [Глава 6 Проектирование развертываний Cloud Assembly](#).

- ж) Щелкните **Развернуть**.
- з) Продолжайте дорабатывать облачный шаблон, пока он не будет готов для предоставления потребителям.
- и) Щелкните **Версия**, чтобы опубликовать облачный шаблон и назначить ему версию.

5. Отправьте пользователям данные для входа в систему, используя наиболее распространенный способ.

### Результаты

В данном примере использования вы сделали двух коллег участниками организации. После этого вы сделали Сильвию администратором Cloud Assembly. Вы сделали Тони администратором проекта WebApp. Такая конфигурация ролей пользователей подходит только для небольших рабочих групп, которые предоставляют потребителям развернутые приложения, а не каталог или доступ в режиме самообслуживания.

### Пример использования роли пользователя 2: настройка ролей пользователей vRealize Automation для поддержки более крупных команд разработчиков и каталога

Владелец организации vRealize Automation отвечает за управление доступом к ресурсам инфраструктуры и бюджетом на них. Есть группа разработчиков облачных шаблонов, которые создают и развертывают шаблоны для разных проектов в несколько итераций, пока они не будут готовы для предоставления потребителям. После этого пользователям в каталоге можно предоставить развертываемые ресурсы.

Переходите к этому примеру, ознакомившись с первым примером использования, в котором были только администраторы. Теперь необходимо расширить систему для поддержки большего числа команд и более сложных задач.

- Разработчики могут создавать и развертывать собственные облачные шаблоны приложений во время разработки. Добавьте себя в качестве администратора, затем добавьте пользователей с ролями «Пользователь службы» и «Обозреватель службы». Затем добавьте пользователей в качестве участников проекта. Участники проекта имеют право разрабатывать и развертывать собственные облачные шаблоны.
- Опубликуйте облачные шаблоны в каталоге, где они будут доступны для развертывания пользователям, не являющимся разработчиками. Теперь назначьте роли пользователей для Service Broker. Service Broker предоставляет каталог потребителям облачных шаблонов. С ее помощью также можно создавать политики, в том числе политики аренды и предоставления прав, но эта функция не рассматривается в этом примере применения ролей пользователей.

## Необходимые условия

- Просмотрите первый пример использования. См. раздел [Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений](#).
- Определите следующих пользователей в зависимости от того, какие разрешения им нужны.
  - Разработчики облачных шаблонов, которые будут пользователями и обозревателями Cloud Assembly.
  - Администратор Service Broker.
  - Пользователи, не являющиеся разработчиками, которые будут потребителями ресурсов каталога как пользователи Service Broker.

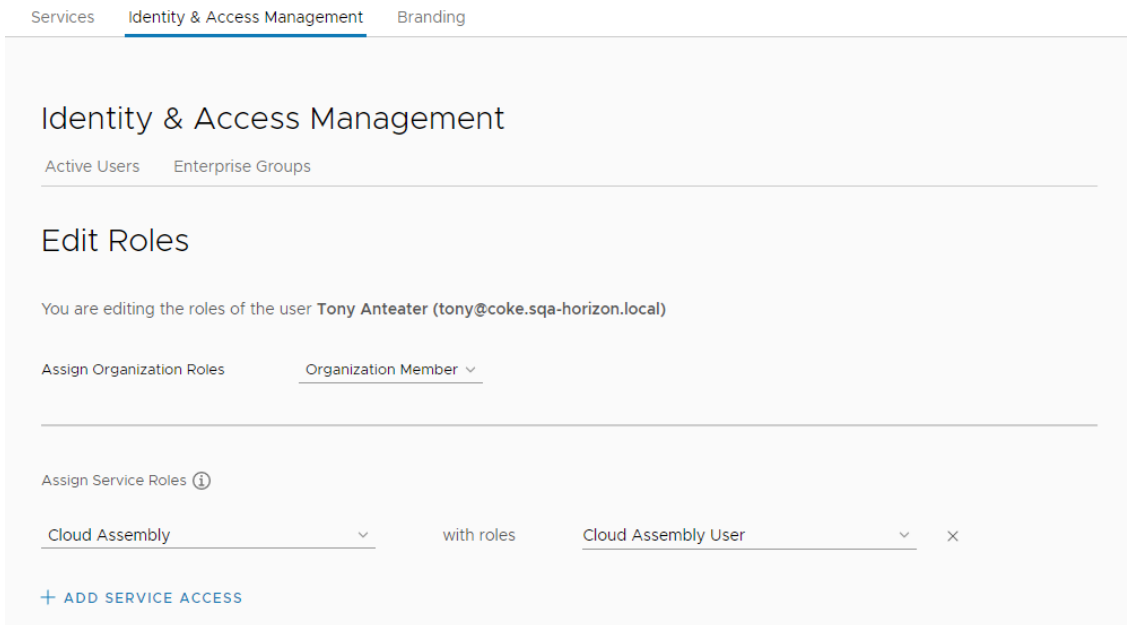
## Процедура

1. Назначьте роли участников организации своим разработчикам облачных шаблонов.

Инструкции см. в [Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений](#).

## 2. Назначьте роль участника службы Cloud Assembly разработчикам облачных шаблонов.

- а) Щелкните **Добавить доступ к службе**.



- б) Настройте пользователя со следующим значением.

Служба	Роль
Cloud Assembly	Пользователь Cloud Assembly
Cloud Assembly	Обозреватель Cloud Assembly

В этом примере использования разработчикам необходимо видеть инфраструктуру, чтобы создавать облачные шаблоны, пригодные для развертывания. Пользователи, которым на следующем шаге будут назначены роли администраторов проекта и участников проекта, не будут видеть инфраструктуру. Как обозреватели службы они могут видеть конфигурацию инфраструктуры, но не могут вносить в нее изменения. Администратор облачных служб сохраняет функции контроля и предоставляет им доступ к информации, необходимой для разработки облачных шаблонов.

## 3. Создайте проекты в Cloud Assembly, которые будут использоваться для группировки ресурсов и пользователей.

В этом случае необходимо создать два проекта. Первый проект — **PersonnelAppDev**, второй — **PayrollAppDev**.

- В консоли перейдите на вкладку **Службы** и выберите **Cloud Assembly**.
- Выберите **Инфраструктура > Проекты > Создать проект**.
- В качестве имени введите **PersonnelAppDev**.
- Нажмите кнопку **Пользователи** и выберите **Добавить пользователей**.

- д) Добавьте участников проекта и назначьте администратора проекта.

Роль проекта	Описание
Пользователь проекта	Участник проекта — это основная роль пользователя-разработчика в проекте. На уровне проектов определяется, какие облачные ресурсы будут доступны на момент готовности к тестированию результатов разработки путем развертывания облачного шаблона.
Администратор проекта	Администратор проекта управляет разработчиками путем добавления и удаления пользователей проектов. Кроме того, можно удалять свои проекты. Для создания проекта необходимо иметь права администратора службы.

- е) Укажите через запятую адреса электронной почты каждого пользователя, добавляемого в качестве участника проекта, и выберите **Пользователь** в раскрывающемся меню **Назначить роль**.

Например, tony@mycompany.com,sylvia@mycompany.com.

The screenshot shows the 'Users' configuration page for a project named 'PersonnelAppDev'. At the top, there are tabs for 'Summary', 'Users' (selected), 'Provisioning', 'Kubernetes Provisioning', and 'Integrations'. Below the tabs, there's a 'Deployment sharing' section with a toggle switch turned on, indicating that deployments are shared between all users in the project. The 'User roles' section prompts the user to 'Specify the users and groups related to this project.' It features two buttons: '+ ADD USERS' and '+ ADD GROUPS', along with a '- REMOVE' button. Below these is a search bar labeled 'Search users or groups'. A table lists the current users:

<input type="checkbox"/>	Name	Account	Role
<input type="checkbox"/>	Sylvia Adams	sylvia	Administrator
<input type="checkbox"/>	Gloria Martinez	gloria	Member
<input type="checkbox"/>	Tony Anteater	tony	Member

At the bottom of the table, it indicates '1 - 3 of 3 users'. There are 'SAVE' and 'CANCEL' buttons at the bottom left of the interface.

- ж) Для пользователей, назначаемых администраторами, выберите **Администратор** в раскрывающемся меню **Назначить роль** и укажите необходимый адрес электронной почты.

- з) Перейдите на вкладку **Подготовка** и добавьте одну или несколько облачных зон.

Когда разработчики облачных шаблонов, являющиеся участниками этого проекта, развертывают шаблон, он развертывается на ресурсах, доступных в облачных зонах. Необходимо убедиться, что ресурсов облачной зоны достаточно для удовлетворения потребностей облачных шаблонов группы разработчиков проекта.

- и) Повторите этот процесс, чтобы добавить проект PayrollAppDev с необходимыми пользователями и администратором.

4. Предоставьте пользователю службы необходимые данные для входа в систему и убедитесь, что участники каждого проекта могут выполнять следующие задачи.

- а) Открывать Cloud Assembly.
- б) Просматривать инфраструктуру по всем проектам.
- в) Создавать облачный шаблон для проекта, участниками которого они являются.
- г) Развертывать облачный шаблон на ресурсах облачной зоны, определенных в проекте.
- д) Управлять своими развертываниями.

5. Назначьте роли участников организации своим разработчикам облачных шаблонов.

Инструкции см. в [Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation](#) для поддержки небольшой группы разработчиков приложений.

6. Назначьте роли администратору каталога, потребителям каталога и разработчикам облачных шаблонов в зависимости от их обязанностей.

- а) Щелкните **Добавить доступ к службе**.
- б) Настройте роль администратора каталога со следующим значением.

Эта роль может быть назначена администратору облачных служб или другим пользователям в группе разработчиков приложений.

Служба	Роль
Service Broker	Администратор Service Broker

- в) Настройте роли потребителей облачного шаблона со следующим значением.

Служба	Роль
Service Broker	Пользователь Service Broker

### Identity & Access Management

Active Users   Enterprise Groups

## Edit Roles

You are editing the roles of the user **Gloria Martinez (gloria@coke.sqa-horizon.local)**

Assign Organization Roles   Organization Member ▾

---

Assign Service Roles ⓘ

Service Broker ▾

with roles

Service Broker User ▾

×

[+ ADD SERVICE ACCESS](#)

- г) Настройте роли разработчиков облачных шаблонов со следующим значением.

Служба	Роль
Cloud AssemblyCloud Assembly	Пользователь Cloud Assembly

7. Создайте проекты в Cloud Assembly, которые будут использоваться для группировки ресурсов и пользователей.

В этом случае необходимо создать два проекта. Первый проект — PersonnelAppDev, второй — PayrollAppDev.

Инструкции см. в [Пример использования роли пользователя 2: настройка ролей пользователей vRealize Automation для поддержки более крупных команд разработчиков и каталога](#).

8. Создайте и опубликуйте облачные шаблоны для каждой проектной группы.

Инструкции см. в [Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений](#).

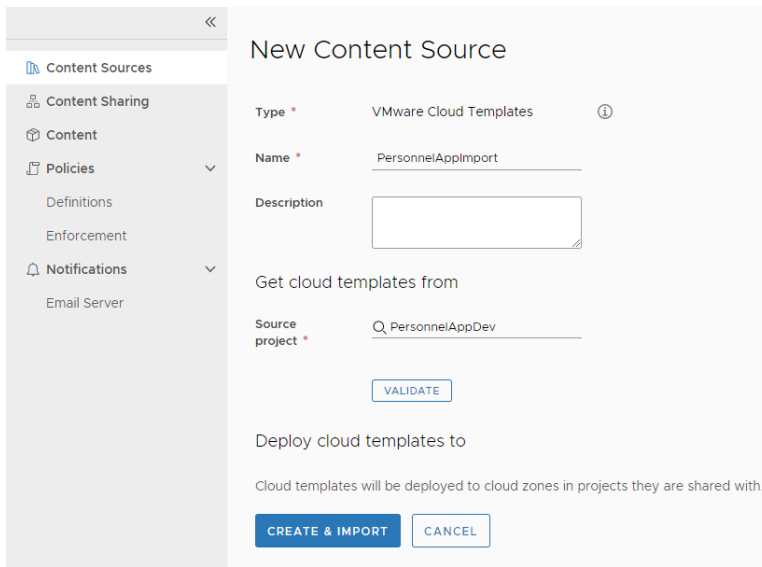
9. Импортируйте облачный шаблон Cloud Assembly в Service Broker.

Необходимо войти в систему в качестве пользователя с ролью администратора Service Broker.

- Войдите в систему как пользователь с ролью администратора Service Broker.
- В консоли щелкните Service Broker.

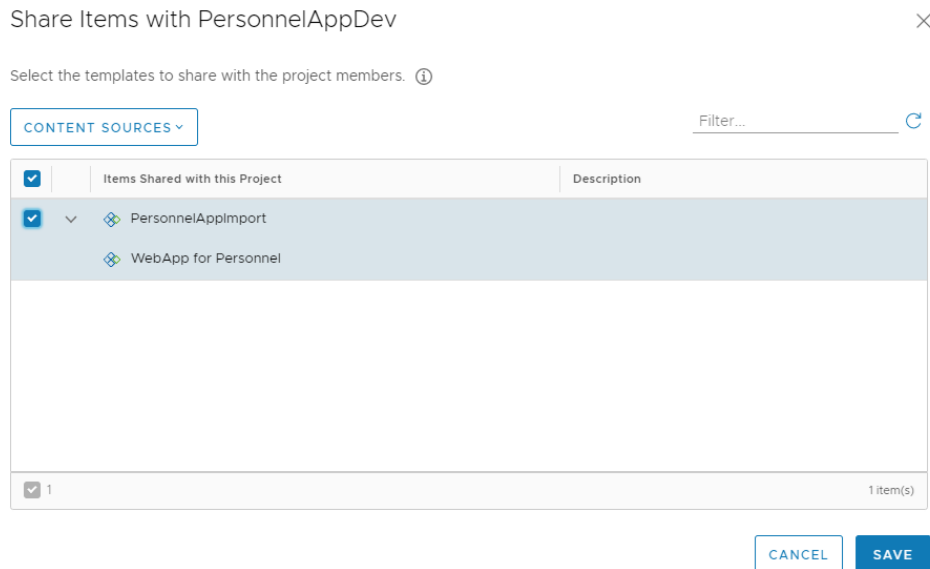


- в) Выберите **Содержимое и политики > Источники содержимого** и нажмите **Создать**.



- г) Выберите **Облачный шаблон Cloud Assembly**.
- д) В качестве имени введите **PersonnelAppImport**.
- е) В раскрывающемся меню **Исходный проект** выберите PersonnelAppDev и нажмите **Проверить**.
- ж) После проверки источника нажмите **Создать и импортировать**.
- з) Повторите эти действия для PayrollAppDev, указав PayrollAppImport в качестве имени источника содержимого.
- 10.** Предоставьте общий доступ к импортированному облачному шаблону в рамках проекта.
- Несмотря на то что облачный шаблон уже связан с проектом, можно предоставить к нему общий доступ в Service Broker, чтобы он стал доступен в каталоге.
- а) Продолжите работу в качестве пользователя с ролью администратора Service Broker.
- б) В Service Broker выберите **Содержимое и политики > Общий доступ к содержимому**.
- в) Выберите проект **PersonnelAppDev** с пользователями, которым требуется возможность развертывать облачный шаблон из каталога.

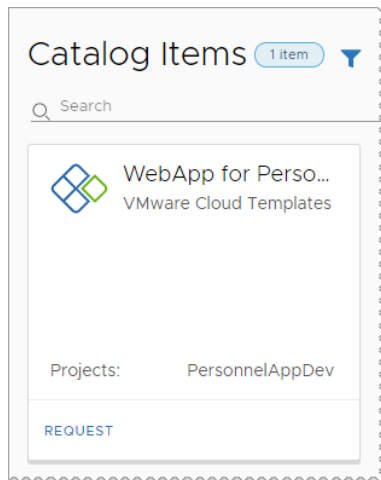
- г) Щелкните **Добавить элементы**, а затем выберите облачный шаблон PersonnelApp, чтобы предоставить общий доступ к нему участникам проекта.



- д) Нажмите **Сохранить**.

11. Убедитесь, что облачный шаблон доступен в каталоге Service Broker для участников проекта.

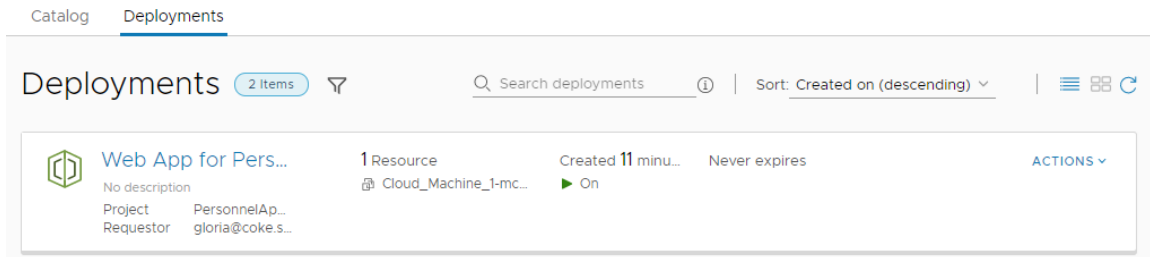
- а) Попросите участника проекта войти в систему и перейти на вкладку **Каталог**.



- б) Щелкните «Запрос» на карточке облачного шаблона PersonnelApp.
- в) Заполните форму и нажмите **Отправить**.

**12. Убедитесь, что участник проекта может отслеживать процесс развертывания.**

- а) Попросите, чтобы участник проекта выбрал **Ресурсы > Развертывания**, и найдите его запрос на подготовку.



- б) При развертывании облачного шаблона убедитесь, что к приложению получает доступ пользователь, инициировавший запрос.

**13. Повторите этот процесс для других проектов.****Результаты**

В этом примере использования была выявлена потребность в делегировании разработки облачного шаблона разработчикам. В результате были добавлены дополнительные участники организации. Им была назначена роль пользователя Cloud Assembly. Затем они были добавлены как участники в соответствующие проекты, чтобы иметь возможность создавать и развертывать облачные шаблоны. Роль участников проекта не позволяет им просматривать или изменять инфраструктуру, которую продолжает контролировать администратор. При этом им предоставлены все права обозревателя службы, чтобы они могли понимать ограничения инфраструктуры, для которой они проектируют схемы.

В этом сценарии использования были настроены пользователи с различными ролями, в том числе администратора и пользователей Service Broker. Затем необходимо предоставить пользователям, не являющимся разработчиками, каталог Service Broker.

**Следующие шаги**

Сведения о том, как определять и назначать настраиваемые роли пользователю, см. в разделе [Пример использования роли пользователя 3. Определение настраиваемых ролей пользователей vRealize Automation для уточнения системных ролей](#).

**Пример использования роли пользователя 3. Определение настраиваемых ролей пользователей vRealize Automation для уточнения системных ролей**

Владелец организации vRealize Automation или администратор службы могут управлять доступом пользователей с помощью системных ролей организации и служб. Тем не менее также может потребоваться создание настраиваемых ролей для выбранных пользователей и выполнение задач или просмотр содержимого за пределами их системных ролей.

В этом сценарии предполагается, что вы знакомы с ролями пользователя и обозревателя службы, а также с ролями участников и обозревателей проекта, которые определены в примере использования 2. Хорошо видно, что они являются более ограничивающими, чем роли администраторов служб и проектов в примере использования 1. Теперь определены несколько локальных примеров использования, в которых некоторым пользователям необходимы полные разрешения на управление одними функциями, права просмотра для других функций и запрет на просмотр третьего набора функций. Настраиваемые роли позволяют определять такие разрешения.

Этот пример использования основан на трех возможных локальных примерах использования. В данной процедуре показано, как создать разрешения для следующих настраиваемых ролей.

- Администратор инфраструктуры с ограниченными правами. Требуется, чтобы некоторые пользователи службы, которые не являются администраторами службы, имели более широкие разрешения, относящиеся к инфраструктуре. Администратору необходимо, чтобы они могли настраивать облачные зоны, образы и конфигурации ресурсов. Также необходимо, чтобы они могли внедрять обнаруженные ресурсы и управлять ими. Следует отметить, что они не могут добавлять облачные учетные записи или интеграции, а могут только определять инфраструктуру для этих конечных точек.
- Разработчик действий расширяемости. Необходимо, чтобы некоторые пользователи службы имели полные разрешения на использование действий расширяемости и подписок с поддержкой расширяемости в рамках разработки облачных шаблонов для своей проектной группы и других проектов. Им также необходимо разрабатывать настраиваемые типы ресурсов и настраиваемые действия для разных проектов.
- Разработчик любого ресурса как услуги. Необходимо, чтобы некоторые пользователи службы имели полные разрешения на разработку настраиваемых типов ресурсов и настраиваемых действий для разных проектов.
- Специалист по устранению неполадок в развертывании. Необходимо, чтобы администраторы проекта имели разрешения, требуемые для устранения неполадок и выполнения анализа основных причин сбоев при развертывании. Вы предоставляете им разрешения на управление применительно к категориям действий, не нарушающим работу, или менее затратным категориям, например на сопоставление образов и конфигураций ресурсов. Также необходимо, чтобы администраторы проекта имели разрешение на настройку утверждений и политик регулярного обслуживания в рамках роли «устранение неполадок при сбое развертывания».

#### Необходимые условия

- Информацию о ролях служб Cloud Assembly и Service Broker, а также таблицы ролей проекта см. в разделе [Что такое роли пользователей vRealize Automation](#). Необходимо разобраться в том, какие элементы служб доступны для просмотра и выполнения каждой ролью пользователя службы.
- Просмотрите описания [Настраиваемые роли пользователей в vRealize Automation](#), чтобы узнать, как уточнять разрешения для пользователей.
- Ознакомьтесь с первым примером использования, чтобы изучить возможности ролей организации и администраторов служб. См. раздел [Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений](#).

- Ознакомьтесь со вторым примером использования, чтобы изучить возможности ролей пользователей служб и участников проекта. См. раздел [Пример использования роли пользователя 2: настройка ролей пользователей vRealize Automation для поддержки более крупных команд разработчиков и каталога](#).
- Ознакомьтесь с Service Broker. См. раздел [Добавление содержимого в каталог](#).

## Процедура

1. Назначьте роли участников организации своим разработчикам облачных шаблонов.

Инструкции см. в [Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений](#).

2. Назначьте роли служб Cloud Assembly и Service Broker для разработчиков облачных шаблонов и пользователей каталога.

При необходимости см. [Пример использования роли пользователя 2: настройка ролей пользователей vRealize Automation для поддержки более крупных команд разработчиков и каталога](#).

3. Создайте проекты в Cloud Assembly, которые будут использоваться для группировки ресурсов и пользователей.

Процедура ниже для настраиваемых ролей также включает в себя роли проекта.

Инструкции по созданию проектов см. во [Пример использования роли пользователя 2: настройка ролей пользователей vRealize Automation для поддержки более крупных команд разработчиков и каталога](#).

4. Создайте и опубликуйте облачные шаблоны для каждой проектной группы.

Инструкции см. в [Роли пользователей: пример использования 1. Настройка ролей пользователей vRealize Automation для поддержки небольшой группы разработчиков приложений](#).

5. Войдите в Cloud Assembly как администратор службы и выберите **Инфраструктура > Администрирование > Настраиваемые роли**.

6. Создайте роль «Администратор инфраструктуры с ограниченными правами».

В этом примере есть пользователь по имени Тони, который является экспертом по настройке инфраструктуры для различных проектов, но ему не нужно предоставлять полные разрешения на доступ к службе. В свою очередь, Тони создает базовую инфраструктуру, которая поддерживает работу всех проектов. Ему предоставляются ограниченные разрешения на управление инфраструктурой. Тони (или внешний подрядчик) также может иметь похожие разрешения на внедрение обнаруженных компьютеров и их перевод под управление vRealize Automation.

- а) Добавьте Тони в Cloud Assembly в качестве пользователя и обозревателя службы.

Имея разрешения как обозреватель, он может видеть базовые облачные учетные записи и интеграции, если ему нужно устранять неполадки, но он не может вносить изменения.

- б) Создайте проект и добавьте Тони в качестве участника проекта.

- в) Чтобы создать настраиваемую роль, выберите **Инфраструктура > Администрирование > Настраиваемые роли** и нажмите **Создать настраиваемую роль**.

- г) Введите имя **Администратор инфраструктуры с ограниченными правами** и выберите следующие разрешения.

Выберите это разрешение...	Чтобы пользователи могли выполнять следующие операции...
Инфраструктура > Управление облачными зонами	Создание, обновление и удаление облачных зон.
Инфраструктура > Управление сопоставлениями конфигураций ресурсов	Создание, обновление и удаление сопоставлений конфигураций ресурсов.
Инфраструктура > Управление сопоставлениями образов	Создание, обновление и удаление сопоставлений образов.

- д) Щелкните **Создать**.
- е) На странице «Настраиваемые роли» выберите роль «Администратор инфраструктуры с ограниченными правами» и щелкните **Назначить**.
- ж) Введите учетную запись электронной почты Тони и щелкните **Добавить**.  
Например, введите Tony@yourcompany.com.  
Также можно ввести любые определенные группы пользователей Active Directory.
- з) Сделайте так, чтобы после входа в систему Тони мог добавлять, изменять и удалять значения в областях, определенных настраиваемой ролью.

## 7. Создание роли разработчика действий расширяемости.

В этом примере есть несколько разработчиков облачных шаблонов (Сильвия и Игорь), которые знакомы с применением действий расширяемости и подписок с поддержкой расширяемости для управления ежедневными задачами по разработке. Они также знакомы с vRealize Orchestrator, поэтому им можно поручить предоставление настраиваемых ресурсов и действий для различных проектов. Вы предоставляете им дополнительные разрешения на управление расширяемостью посредством управления настраиваемыми ресурсами и действиями, а также действиями расширяемости и подписками с поддержкой расширяемости.

- а) Добавьте Сильвию и Игоря как пользователей Cloud Assembly.
- б) Добавьте их в качестве участников проектов, где они будут применять свои навыки по расширяемости.

- в) Создайте настраиваемую роль пользователя с именем **Разработчик действий расширяемости** и выберите следующие разрешения.

Выберите это разрешение...	Чтобы пользователи могли выполнять следующие операции...
Любой ресурс как услуга > Управление настраиваемыми ресурсами	Создание, обновление или удаление настраиваемых ресурсов.
Любой ресурс как услуга > Управление действиями с ресурсами	Создание, обновление или удаление настраиваемых действий.
Расширяемость > Управление ресурсами расширяемости	Создание, обновление или удаление действий расширяемости и подписок с поддержкой расширяемости. Отключение подписок. Отмена и удаление циклов выполнения действий.

- г) Щелкните **Создать**.
- д) Назначьте Сильвии и Игорю роль «Разработчик действий расширяемости».
- е) Убедитесь, что Сильвия и Игорь могут управлять настраиваемыми ресурсами и действиями и что они могут управлять различными параметрами на вкладке «Расширяемость».
8. Создание роли «Специалист по устранению неполадок в развертывании».

В этом примере администратору проекта предоставляются дополнительные разрешения по управлению, чтобы он мог исправлять ошибки в развертывании для своих рабочих групп.

- а) Добавьте администраторов проектов (Шона, Пратап и Вей) как пользователей служб Cloud Assembly и Service Broker.
- б) Добавьте их в их проекты в качестве администраторов проектов.
- в) Создайте настраиваемую роль пользователя с именем **Специалист по устранению неполадок в развертывании** и выберите следующие разрешения.

Выберите это разрешение...	Чтобы пользователи могли выполнять следующие операции...
Инфраструктура > Управление сопоставлениями конфигураций ресурсов	Создание, обновление и удаление сопоставлений конфигураций ресурсов.
Инфраструктура > Управление сопоставлениями образов	Создание, обновление и удаление сопоставлений образов.
Развертывания > Управление развертываниями	Просмотр всех развертываний в разных проектах и выполнение всех действий по регулярному обслуживанию в развертываниях и компонентах развертывания.
Политика > Управление политиками	Создание, обновление или удаление определений политик.

- г) Щелкните **Создать**.
- д) Назначьте Шоне, Пратапу и Вею роль «Специалист по устранению неполадок в развертывании».
- е) Убедитесь, что они могут управлять сопоставлениями конфигураций ресурсов, сопоставлениями образов и политиками в Service Broker.

## Результаты

В этом примере использования необходимо настроить различных пользователей с разными ролями, в том числе настраиваемыми ролями, которые расширяют роли служб и проектов.

## Следующие шаги

Создайте настраиваемые роли, предназначенные для локальных примеров использования.

## Назначение пользователю встроенной роли администратора инфраструктуры Cloud Assembly

Администратор инфраструктуры — это встроенная роль, которую можно назначить выбранным пользователям. В пользовательском интерфейсе эту роль назначить нельзя.

### Когда нужно назначать эту роль пользователя

Разрешения можно дублировать с помощью параметров настраиваемой роли пользователя. Однако эту встроенную роль можно назначить пользователям, которые являются администраторами с ограниченными правами.

### Разрешения для роли администратора инфраструктуры

В следующей таблице приведены разрешения для управления и другие разрешения для администраторов инфраструктуры. Эти разрешения изменить нельзя. Чтобы дополнительно ограничить разрешения для пользователя, используйте настраиваемые роли, которые будут соответствовать конкретным потребностям.

**Таблица 3-15. Предоставленные разрешения для встроенной роли администратора инфраструктуры**

Разрешение на создание, изменение, обновление или удаление	Прочие разрешения
<ul style="list-style-type: none"> <li>■ Облачные учетные записи</li> <li>■ Интеграции</li> <li>■ Облачные зоны</li> <li>■ Сопоставления конфигураций ресурсов</li> <li>■ Сопоставления образов</li> <li>■ Профили сетей</li> <li>■ Профили хранилища</li> <li>■ Теги</li> <li>■ Внедрение</li> </ul>	<ul style="list-style-type: none"> <li>■ Просмотр и расстановка тегов для обнаруженных ресурсов</li> <li>■ Просмотр вычислительных ресурсов</li> <li>■ Управление IP-адресами</li> <li>■ Просмотр и расстановка тегов для подсистем балансировки нагрузки</li> <li>■ Просмотр доменов сети</li> <li>■ Просмотр безопасности</li> <li>■ Просмотр хранилища</li> <li>■ Просмотр и удаление запросов</li> </ul>

### Назначение роли администратора инфраструктуры

Эту встроенную роль можно назначить с помощью API-интерфейса RBAC. Сначала нужно получить роль, а затем назначить ее пользователю.

Перед началом работы:

- Ознакомьтесь с API-интерфейсом. См. [руководство по программированию API-интерфейсов в vRealize Automation](#).



- Ознакомьтесь с API-интерфейсом. См. [руководство по программированию API-интерфейса в vRealize Automation 8.6](#).
  - Получите маркер предьявителя API-интерфейса. См. статью «Получение маркера доступа» в [руководстве по программированию API-интерфейсов в vRealize Automation](#).
  - Получите маркер предьявителя API-интерфейса. См. статью «Получение маркера доступа» в [руководстве по программированию API-интерфейсов в vRealize Automation 8.6](#).
1. Перейдите в `$vra/project/api/swagger/swagger-ui.html?urls.primaryName=rba`, где `$vra` — базовый URL-адрес для экземпляра.
  2. В верхнем правом углу страницы в раскрывающемся списке **Выберите определение** выберите **rbac: 2020-08-10**.
  3. Чтобы получить роль пользователя, откройте раздел **Роль** и выполните команду `GET /rbac-service/api/roles`.

Результаты должны иметь примерно следующий вид.

```
"content": [
  {
    "description": "Infrastructure Administrator",
    "hidden": false,
    "id": "infrastructure_administrator",
    "name": "Infrastructure Administrator",
    "orgId": "string",
    "permissions": [
      "string"
    ],
    "projectScope": true
  }
]
```

4. Чтобы добавить пользователя в роль, откройте раздел **Назначение роли**, откройте и измените команду `PUT /rbac-service/api/role-assignments`, добавив в нее имя пользователя.

Например:

```
{
  "orgId": "string",
  "principalId": "Username@domain",
  "principalType": "user",
  "projectId": "string",
  "rolesToAdd": [
    "infrastructure_administrator"
  ],
  "rolesToRemove": [
    "string"
  ]
}
```

5. Выполните измененную команду `PUT`.

6. Чтобы проверить результаты, попросите назначенного пользователя выполнить вход и убедиться, что у него есть разрешения, определенные выше.

## Добавление учетных записей облачной службы в Cloud Assembly

Облачные учетные записи — это настроенные разрешения, которые используются в Cloud Assembly для сбора данных из регионов или центров обработки данных, а также для развертывания облачных шаблонов в этих регионах.

Собранные данные включают в себя регионы и хранилища данных, которые затем связываются с облачными зонами.

При последующей настройке облачных зон, сопоставлений и профилей необходимо будет выбрать учетную запись облачной службы, с которой они связаны.

Администратор облачных систем создает облачные учетные записи для проектов, в которых работают участники рабочей группы. Сведения о ресурсах, таких как сеть и безопасность, вычислительные ресурсы и хранилище, а также содержимое тегов извлекаются из учетных записей облачной службы.

**Примечание** Если облачная учетная запись содержит связанные компьютеры, уже развернутые в данном регионе, их можно передать в управление Cloud Assembly с помощью плана внедрения. См. раздел [Что такое планы внедрения в Cloud Assembly](#).

В случае удаления облачной учетной записи, которая используется в развертывании, ресурсы, являющиеся частью данного развертывания, становятся неуправляемыми.

## Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation

Для настройки и использования облачных учетных записей vRealize Automation убедитесь в наличии следующих учетных данных.

### Необходимые учетные данные облачной учетной записи

Задача	Действия
Зарегистрироваться и войти в Cloud Assembly	Идентификатор VMware. <ul style="list-style-type: none"> <li>■ Настройте учетную запись <a href="#">My VMware</a>, используя корпоративный адрес электронной почты.</li> </ul>
Подключитесь к службам vRealize Automation	Порт HTTPS 443, открытый для исходящего трафика, с доступом через брандмауэр к следующим узлам: <ul style="list-style-type: none"> <li>■ *.vmwareidentity.com</li> <li>■ gaz.csp-vidm-prod.com</li> <li>■ *.vmware.com</li> </ul> Дополнительные сведения о портах и протоколах см. в разделе <a href="#">VMware Ports and Protocols</a> . Дополнительные сведения о портах и протоколах см. в разделе <a href="#">Требования к портам</a> в справке по <a href="#">эталонной архитектуре</a> .

Задача	Действия
Добавить облачную учетную запись vCenter	<p>Для управления экземпляром vCenter Server агенту vSphere требуются соответствующие разрешения. Укажите учетную запись со следующими правами чтения и записи:</p> <ul style="list-style-type: none"> <li>■ vCenter IP-адрес или полное доменное имя (FQDN)</li> </ul> <p>Здесь перечислены разрешения, необходимые для облачных учетных записей VMware Cloud on AWS и vCenter. Разрешения должны быть активированы для всех кластеров vCenter Server, а не только для кластеров, в которых размещены конечные точки.</p> <p>Для всех облачных учетных записей на основе vCenter Server, в том числе NSX-V, NSX-T, vCenter и VMware Cloud on AWS, администратору требуются учетные данные конечной точки vSphere или учетные данные, которые служба агента использует в vCenter, что обеспечивает доступ к узлу vCenter Server с правами администратора.</p> <p>Подробнее требования агента vSphere описаны в <a href="#">документации по продукту VMware vSphere</a>.</p> <ul style="list-style-type: none"> <li>■ Хранилище данных <ul style="list-style-type: none"> <li>■ Выделение пространства</li> <li>■ Просмотр хранилища данных</li> <li>■ Операции с файлами на низком уровне</li> </ul> </li> <li>■ Кластер хранилища данных <ul style="list-style-type: none"> <li>■ Настройка кластера хранилища данных</li> </ul> </li> <li>■ Папка <ul style="list-style-type: none"> <li>■ Создание папок</li> <li>■ Удаление папок</li> </ul> </li> <li>■ Глобальная среда <ul style="list-style-type: none"> <li>■ Управление настраиваемыми атрибутами</li> <li>■ Задание настраиваемых атрибутов</li> </ul> </li> <li>■ Сеть <ul style="list-style-type: none"> <li>■ Назначение сети</li> </ul> </li> <li>■ Разрешения <ul style="list-style-type: none"> <li>■ Разрешение на изменение</li> </ul> </li> <li>■ Ресурс <ul style="list-style-type: none"> <li>■ Назначение виртуальной машины пулу ресурсов</li> <li>■ Перенос выключенной виртуальной машины</li> <li>■ Перенос включенной виртуальной машины</li> </ul> </li> <li>■ Хранилище на основе профилей <ul style="list-style-type: none"> <li>■ Представление хранилища на основе профилей</li> </ul> <p>Чтобы получить список политик хранилища, которые можно сопоставить с профилем хранилища, предоставьте разрешение StorageProfile.View всем учетным записям, подключающим vRealize Automation к vCenter Server.</p> </li> <li>■ Библиотека содержимого</li> </ul>

Задача	Действия
	<p>Чтобы назначить разрешение библиотеке содержимого, администратор должен предоставить пользователю глобальное разрешение. Дополнительные сведения см. в разделе <a href="#">Иерархическое наследование разрешений для библиотек содержимого</a> в документе <i>Администрирование виртуальных машин vSphere</i> в документации по VMware vSphere.</p> <ul style="list-style-type: none"> <li>■ Добавление элемента библиотеки</li> <li>■ Создание локальной библиотеки</li> <li>■ Создание библиотеки с подпиской</li> <li>■ Удаление элемента библиотеки</li> <li>■ Удаление локальной библиотеки</li> <li>■ Удаление библиотеки с подпиской</li> <li>■ Загрузка файлов</li> <li>■ Исключение элемента библиотеки</li> <li>■ Изучение информации о подписке</li> <li>■ Чтение хранилища</li> <li>■ Синхронизация элемента библиотеки</li> <li>■ Синхронизация библиотеки с подпиской</li> <li>■ Механизм интроспекции</li> <li>■ Обновление параметров конфигурации</li> <li>■ Обновление файлов</li> <li>■ Обновление библиотеки</li> <li>■ Обновление элемента библиотеки</li> <li>■ Обновление локальной библиотеки</li> <li>■ Обновление библиотеки с подпиской</li> <li>■ Просмотр параметров конфигурации</li> <li>■ Использование тегов в vSphere <ul style="list-style-type: none"> <li>■ Назначение или отмена назначения тега vSphere</li> <li>■ Назначение или отмена назначения тега vSphere объекту</li> <li>■ Создание тега vSphere</li> <li>■ Создание категории тегов vSphere</li> <li>■ Удаление тега vSphere</li> <li>■ Удаление категории тегов vSphere</li> <li>■ Редактирование тега vSphere</li> <li>■ Редактирование категории тегов vSphere</li> <li>■ Изменение поля UsedBy или категории</li> <li>■ Изменение поля UsedBy для тега</li> </ul> </li> <li>■ vApp <ul style="list-style-type: none"> <li>■ Импортировать</li> <li>■ Конфигурация приложения vApp</li> </ul> </li> </ul> <p>Для шаблонов OVF и подготовки виртуальных машин из библиотеки содержимого требуется конфигурация приложения vApp.Import.</p> <p>При использовании cloud-init для сценариев настройки облачной среды требуется указать конфигурацию приложения vApp.Import. Этот параметр позволяет изменять внутреннюю структуру vApp, например сведения о продукте и свойства.</p>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ Виртуальная машина — иерархия <ul style="list-style-type: none"> <li>■ Создание на основе существующей</li> <li>■ Создание новой</li> <li>■ Перемещение</li> <li>■ Удаление</li> </ul> </li> <li>■ Виртуальная машина — взаимодействие <ul style="list-style-type: none"> <li>■ Настройка параметров компакт-дисков</li> <li>■ Взаимодействие с консолью</li> <li>■ Подключение устройств</li> <li>■ Выключение</li> <li>■ Включение</li> <li>■ Сброс</li> <li>■ Приостановка</li> <li>■ Установка инструментов</li> </ul> </li> <li>■ Виртуальная машина — конфигурация <ul style="list-style-type: none"> <li>■ Добавление существующего диска</li> <li>■ Добавить новый элемент</li> <li>■ Удаление диска</li> <li>■ Дополнительно</li> <li>■ Изменение количества ЦП</li> <li>■ Изменение ресурсов</li> <li>■ Расширение виртуального диска</li> <li>■ Отслеживание изменений на диске</li> <li>■ Память</li> <li>■ Изменение параметров устройства</li> <li>■ Переименование</li> <li>■ Создание примечания</li> <li>■ Параметры</li> <li>■ Размещение файлов подкачки</li> </ul> </li> <li>■ Виртуальная машина — предоставление <ul style="list-style-type: none"> <li>■ Индивидуальная настройка</li> <li>■ Клонирование шаблонов</li> <li>■ Клонирование виртуальной машины</li> <li>■ Развертывание шаблонов</li> <li>■ Чтение спецификаций настройки</li> </ul> </li> <li>■ Виртуальная машина — состояние <ul style="list-style-type: none"> <li>■ Создание моментальных снимков</li> <li>■ Удаление моментальных снимков</li> <li>■ Возврат к моментальному снимку</li> </ul> </li> </ul>

Задача	Действия
Добавить облачную учетную запись Amazon Web Services (AWS)	<p>Укажите учетную запись привилегированного пользователя с правами чтения и записи. Учетная запись пользователя должна быть членом политики привилегированного доступа (PowerUserAccess) в системе «Управление учетными данными и доступом» (IAM) в AWS.</p> <ul style="list-style-type: none"> <li>■ 20-значный идентификатор ключа доступа и соответствующий секретный ключ доступа</li> </ul> <p>Если используется внешний прокси-сервер HTTP, его необходимо настроить на IPv4.</p> <p>Для расширяемости на основе действий vRealize Automation (ABX) и интеграции внешнего поставщика IPAM могут потребоваться дополнительные разрешения. Чтобы разрешить функции автоматического масштабирования, рекомендуются следующие разрешения AWS.</p> <ul style="list-style-type: none"> <li>■ Действия по автоматическому масштабированию: <ul style="list-style-type: none"> <li>■ autoscaling:DescribeAutoScalingInstances</li> <li>■ autoscaling:AttachInstances</li> <li>■ autoscaling&gt;DeleteLaunchConfiguration</li> <li>■ autoscaling:DescribeAutoScalingGroups</li> <li>■ autoscaling&gt;CreateAutoScalingGroup</li> <li>■ autoscaling:UpdateAutoScalingGroup</li> <li>■ autoscaling&gt;DeleteAutoScalingGroup</li> <li>■ autoscaling:DescribeLoadBalancers</li> </ul> </li> <li>■ Автоматическое масштабирование ресурсов: <ul style="list-style-type: none"> <li>■ *</li> </ul> </li> </ul> <p>Предоставьте все разрешения на автоматическое масштабирование ресурсов.</p> <p>Чтобы функции службы маркеров безопасности AWS (AWS STS) могли поддерживать временные учетные данные с ограниченными правами для идентификации и доступа AWS, требуются следующие разрешения.</p> <ul style="list-style-type: none"> <li>■ Ресурсы STS для AWS: <ul style="list-style-type: none"> <li>■ *</li> </ul> </li> </ul> <p>Предоставьте все разрешения для ресурсов STS.</p> <p>Чтобы разрешить функции EC2, требуются следующие разрешения AWS.</p> <ul style="list-style-type: none"> <li>■ Действия EC2: <ul style="list-style-type: none"> <li>■ ec2:AttachVolume</li> <li>■ ec2:AuthorizeSecurityGroupIngress</li> <li>■ ec2&gt;DeleteSubnet</li> <li>■ ec2&gt;DeleteSnapshot</li> <li>■ ec2:DescribeInstances</li> <li>■ ec2&gt;DeleteTags</li> <li>■ ec2:DescribeRegions</li> <li>■ ec2:DescribeVolumesModifications</li> <li>■ ec2&gt;CreateVpc</li> <li>■ ec2:DescribeSnapshots</li> <li>■ ec2:DescribeInternetGateways</li> <li>■ ec2&gt;DeleteVolume</li> <li>■ ec2:DescribeNetworkInterfaces</li> </ul> </li> </ul>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ ec2:StartInstances</li> <li>■ ec2:DescribeAvailabilityZones</li> <li>■ ec2:CreateInternetGateway</li> <li>■ ec2:CreateSecurityGroup</li> <li>■ ec2:DescribeVolumes</li> <li>■ ec2:CreateSnapshot</li> <li>■ ec2:ModifyInstanceAttribute</li> <li>■ ec2:DescribeRouteTables</li> <li>■ ec2:DescribeInstanceTypes</li> <li>■ ec2:DescribeInstanceTypeOfferings</li> <li>■ ec2:DescribeInstanceStatus</li> <li>■ ec2:DetachVolume</li> <li>■ ec2:RebootInstances</li> <li>■ ec2:AuthorizeSecurityGroupEgress</li> <li>■ ec2:ModifyVolume</li> <li>■ ec2:TerminateInstances</li> <li>■ ec2:DescribeSpotFleetRequestHistory</li> <li>■ ec2:DescribeTags</li> <li>■ ec2:CreateTags</li> <li>■ ec2:RunInstances</li> <li>■ ec2:DescribeNatGateways</li> <li>■ ec2:StopInstances</li> <li>■ ec2:DescribeSecurityGroups</li> <li>■ ec2:CreateVolume</li> <li>■ ec2:DescribeSpotFleetRequests</li> <li>■ ec2:DescribeImages</li> <li>■ ec2:DescribeVpcs</li> <li>■ ec2&gt;DeleteSecurityGroup</li> <li>■ ec2&gt;DeleteVpc</li> <li>■ ec2:CreateSubnet</li> <li>■ ec2:DescribeSubnets</li> <li>■ ec2:RequestSpotFleet</li> </ul>
	<p><b>Примечание</b> Для расширяемости на основе действий vRealize Automation (ABX) или интеграции внешнего поставщика IPAM разрешение на запрос SpotFleet не требуется.</p>
	<ul style="list-style-type: none"> <li>■ Ресурсы EC2: <ul style="list-style-type: none"> <li>■ *</li> </ul> </li> </ul> <p>Предоставьте все разрешения для ресурсов EC2.</p> <p>Для выполнения функций эластичной балансировки нагрузки требуются следующие разрешения AWS.</p> <ul style="list-style-type: none"> <li>■ Действия с подсистемой балансировки нагрузки: <ul style="list-style-type: none"> <li>■ elasticloadbalancing&gt;DeleteLoadBalancer</li> <li>■ elasticloadbalancing:DescribeLoadBalancers</li> <li>■ elasticloadbalancing:RemoveTags</li> </ul> </li> </ul>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ elasticloadbalancing:CreateLoadBalancer</li> <li>■ elasticloadbalancing:DescribeTags</li> <li>■ elasticloadbalancing:ConfigureHealthCheck</li> <li>■ elasticloadbalancing:AddTags</li> <li>■ elasticloadbalancing:CreateTargetGroup</li> <li>■ elasticloadbalancing&gt;DeleteLoadBalancerListeners</li> <li>■ elasticloadbalancing:DeregisterInstancesFromLoadBalancer</li> <li>■ elasticloadbalancing:RegisterInstancesWithLoadBalancer</li> <li>■ elasticloadbalancing&gt;CreateLoadBalancerListeners</li> <li>■ Ресурсы подсистемы балансировки нагрузки: <ul style="list-style-type: none"> <li>■ *</li> </ul> </li> </ul> <p>Предоставьте все разрешения для ресурсов подсистемы балансировки нагрузки.</p> <p>Можно включить следующие разрешения на управление идентификацией и доступом (IAM) в AWS, хотя они не являются обязательными.</p> <ul style="list-style-type: none"> <li>■ iam:SimulateCustomPolicy</li> <li>■ iam:GetUser</li> <li>■ iam:ListUserPolicies</li> <li>■ iam:GetUserPolicy</li> <li>■ iam:ListAttachedUserPolicies</li> <li>■ iam:GetPolicyVersion</li> <li>■ iam:ListGroupsForUser</li> <li>■ iam:ListGroupPolicies</li> <li>■ iam:GetGroupPolicy</li> <li>■ iam:ListAttachedGroupPolicies</li> <li>■ iam:ListPolicyVersions</li> </ul>



Задача	Действия
Добавить облачную учетную запись Microsoft Azure	<p>Настройте экземпляр Microsoft Azure и получите действующую подписку Microsoft Azure, позволяющую использовать идентификатор подписки.</p> <p>Создайте приложение Active Directory, как описано в разделе <a href="#">Использование портала для создания приложения Azure AD и субъекта службы, которые могут получать доступ к ресурсам</a> в документации по продукту Microsoft Azure.</p> <p>Если используется внешний прокси-сервер HTTP, его необходимо настроить на IPv4.</p> <p>Запишите следующую информацию.</p> <ul style="list-style-type: none"> <li>■ Идентификатор подписки <p>Обеспечивает доступ к подпискам Microsoft Azure</p> </li> <li>■ Идентификатор арендатора <p>Конечная точка авторизации для приложений Active Directory, создаваемых вами в учетной записи Microsoft Azure.</p> </li> <li>■ Идентификатор клиентского приложения <p>Предоставляет доступ к Microsoft Active Directory в вашей индивидуальной учетной записи Microsoft Azure.</p> </li> <li>■ Секретный ключ клиентского приложения <p>Этот уникальный секретный ключ генерируется для привязки к идентификатору клиентского приложения</p> </li> </ul> <p>Для создания и проверки учетных записей облачных служб Microsoft Azure требуются следующие разрешения:</p> <ul style="list-style-type: none"> <li>■ Microsoft Compute <ul style="list-style-type: none"> <li>■ Microsoft.Compute/virtualMachines/extensions/write</li> <li>■ Microsoft.Compute/virtualMachines/extensions/read</li> <li>■ Microsoft.Compute/virtualMachines/extensions/delete</li> <li>■ Microsoft.Compute/virtualMachines/deallocate/action</li> <li>■ Microsoft.Compute/virtualMachines/delete</li> <li>■ Microsoft.Compute/virtualMachines/powerOff/action</li> <li>■ Microsoft.Compute/virtualMachines/read</li> <li>■ Microsoft.Compute/virtualMachines/restart/action</li> <li>■ Microsoft.Compute/virtualMachines/start/action</li> <li>■ Microsoft.Compute/virtualMachines/write</li> <li>■ Microsoft.Compute/availabilitySets/write</li> <li>■ Microsoft.Compute/availabilitySets/read</li> <li>■ Microsoft.Compute/availabilitySets/delete</li> <li>■ Microsoft.Compute/disks/delete</li> <li>■ Microsoft.Compute/disks/read</li> <li>■ Microsoft.Compute/disks/write</li> </ul> </li> <li>■ Microsoft Network <ul style="list-style-type: none"> <li>■ Microsoft.Network/loadBalancers/backendAddressPools/join/action</li> <li>■ Microsoft.Network/loadBalancers/delete</li> <li>■ Microsoft.Network/loadBalancers/read</li> <li>■ Microsoft.Network/loadBalancers/write</li> <li>■ Microsoft.Network/networkInterfaces/join/action</li> <li>■ Microsoft.Network/networkInterfaces/read</li> </ul> </li> </ul>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ Microsoft.Network/networkInterfaces/write</li> <li>■ Microsoft.Network/networkInterfaces/delete</li> <li>■ Microsoft.Network/networkSecurityGroups/join/action</li> <li>■ Microsoft.Network/networkSecurityGroups/read</li> <li>■ Microsoft.Network/networkSecurityGroups/write</li> <li>■ Microsoft.Network/networkSecurityGroups/delete</li> <li>■ Microsoft.Network/publicIPAddresses/delete</li> <li>■ Microsoft.Network/publicIPAddresses/join/action</li> <li>■ Microsoft.Network/publicIPAddresses/read</li> <li>■ Microsoft.Network/publicIPAddresses/write</li> <li>■ Microsoft.Network/virtualNetworks/read</li> <li>■ Microsoft.Network/virtualNetworks/subnets/delete</li> <li>■ Microsoft.Network/virtualNetworks/subnets/join/action</li> <li>■ Microsoft.Network/virtualNetworks/subnets/read</li> <li>■ Microsoft.Network/virtualNetworks/subnets/write</li> <li>■ Microsoft.Network/virtualNetworks/write</li> <li>■ Microsoft Resources <ul style="list-style-type: none"> <li>■ Microsoft.Resources/subscriptions/resourcegroups/delete</li> <li>■ Microsoft.Resources/subscriptions/resourcegroups/read</li> <li>■ Microsoft.Resources/subscriptions/resourcegroups/write</li> </ul> </li> <li>■ Microsoft Storage <ul style="list-style-type: none"> <li>■ Microsoft.Storage/storageAccounts/delete</li> <li>■ Microsoft.Storage/storageAccounts/listKeys/action</li> <li>■ Microsoft.Storage/storageAccounts/read</li> <li>■ Microsoft.Storage/storageAccounts/write</li> </ul> </li> <li>■ Microsoft Web <ul style="list-style-type: none"> <li>■ Microsoft.Web/sites/read</li> <li>■ Microsoft.Web/sites/write</li> <li>■ Microsoft.Web/sites/delete</li> <li>■ Microsoft.Web/sites/config/read</li> <li>■ Microsoft.Web/sites/config/write</li> <li>■ Microsoft.Web/sites/config/list/action</li> <li>■ Microsoft.Web/sites/publishxml/action</li> <li>■ Microsoft.Web/serverfarms/write</li> <li>■ Microsoft.Web/serverfarms/delete</li> <li>■ Microsoft.Web/sites/hostruntime/functions/keys/read</li> <li>■ Microsoft.Web/sites/hostruntime/host/read</li> <li>■ Microsoft.web/sites/functions/masterkey/read</li> </ul> </li> </ul> <p>Если вы используете Microsoft Azure со средствами расширяемости на основе действий, помимо минимальных разрешений требуются следующие:</p> <ul style="list-style-type: none"> <li>■ Microsoft.Web/sites/read</li> <li>■ Microsoft.Web/sites/write</li> <li>■ Microsoft.Web/sites/delete</li> <li>■ Microsoft.Web/sites/*/action</li> <li>■ Microsoft.Web/sites/config/read</li> </ul>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ Microsoft.Web/sites/config/write</li> <li>■ Microsoft.Web/sites/config/list/action</li> <li>■ Microsoft.Web/sites/publishxml/action</li> <li>■ Microsoft.Web/serverfarms/write</li> <li>■ Microsoft.Web/serverfarms/delete</li> <li>■ Microsoft.Web/sites/hostruntime/functions/keys/read</li> <li>■ Microsoft.Web/sites/hostruntime/host/read</li> <li>■ Microsoft.Web/sites/functions/masterkey/read</li> <li>■ Microsoft.Web/apimanagementaccounts/apis/read</li> <li>■ Microsoft.Authorization/roleAssignments/read</li> <li>■ Microsoft.Authorization/roleAssignments/write</li> <li>■ Microsoft.Authorization/roleAssignments/delete</li> <li>■ Microsoft.Insights/Components/Read</li> <li>■ Microsoft.Insights/Components/Write</li> <li>■ Microsoft.Insights/Components/Query/Read</li> </ul> <p>Если вы используете Microsoft Azure со средствами расширяемости на основе действий с расширениями, также нужны следующие разрешения:</p> <ul style="list-style-type: none"> <li>■ Microsoft.Compute/virtualMachines/extensions/write</li> <li>■ Microsoft.Compute/virtualMachines/extensions/read</li> <li>■ Microsoft.Compute/virtualMachines/extensions/delete</li> </ul> <p>Дополнительные сведения о создании облачной учетной записи Microsoft Azure см. в разделе <a href="#">Настройка Microsoft Azure</a>.</p>

Задача	Действия
Добавить облачную учетную запись Google Cloud Platform (GCP)	<p>Облачная учетная запись Google Cloud Platform взаимодействует с вычислительным модулем Google Cloud Platform.</p> <p>Для создания и проверки облачных учетных записей Google Cloud Platform требуются учетные данные администратора и владельца проекта.</p> <p>Если используется внешний прокси-сервер HTTP, его необходимо настроить на IPv4.</p> <p>Необходимо включить службу вычислительного модуля. При создании облачной учетной записи в vRealize Automation используйте учетную запись службы, которая была создана при инициализации вычислительного модуля.</p> <p>Также требуются следующие разрешения вычислительного модуля, в зависимости от действий, которые может выполнить пользователь.</p> <ul style="list-style-type: none"> <li>■ <code>roles/compute.admin</code> <p>Обеспечивает полный контроль над всеми ресурсами вычислительного модуля.</p> </li> <li>■ <code>roles/iam.serviceAccountUser</code> <p>Предоставляет доступ пользователям, которые управляют экземплярами виртуальных машин, настроенными для запуска в качестве учетной записи службы. Предоставьте доступ к следующим ресурсам и службам:</p> <ul style="list-style-type: none"> <li>■ <code>compute.*</code></li> <li>■ <code>resourceManager.projects.get</code></li> <li>■ <code>resourceManager.projects.list</code></li> <li>■ <code>serviceUsage.quotas.get</code></li> <li>■ <code>serviceUsage.services.get</code></li> <li>■ <code>serviceUsage.services.list</code></li> </ul> </li> <li>■ <code>roles/compute.imageUser</code> <p>Предоставляет разрешение на перечисление и чтение изображений без других разрешений на образ. Роль <code>compute.imageUser</code> на уровне проекта позволяет пользователям перечислять все изображения в проекте. Она также позволяет пользователям создавать ресурсы, например, экземпляры и диски с сохранением состояния, на основе образов в проекте.</p> <ul style="list-style-type: none"> <li>■ <code>compute.images.get</code></li> <li>■ <code>compute.images.getFromFamily</code></li> <li>■ <code>compute.images.list</code></li> <li>■ <code>compute.images.useReadOnly</code></li> <li>■ <code>resourceManager.projects.get</code></li> <li>■ <code>resourceManager.projects.list</code></li> <li>■ <code>serviceUsage.quotas.get</code></li> <li>■ <code>serviceUsage.services.get</code></li> <li>■ <code>serviceUsage.services.list</code></li> </ul> </li> <li>■ <code>roles/compute.instanceAdmin</code> <p>Предоставляет разрешения на создание, изменение и удаление экземпляров виртуальных машин. К ним относятся разрешения на создание, изменение и удаление дисков, а также на настройку параметров экранированного VMBETA.</p> <p>Для пользователей, которые управляют экземплярами виртуальных машин (но не параметрами сети или безопасности или экземплярами, которые выполняются как учетные записи служб), предоставьте эту роль организации, папке или проекту, содержащему экземпляры, или отдельным экземплярам.</p> </li> </ul>

Задача	Действия
	<p>Пользователям, которые управляют экземплярами виртуальных машин, настроенными для работы в качестве учетной записи службы, также требуется роль <code>roles/iam.serviceAccountUser</code>.</p> <ul style="list-style-type: none"> <li>■ <code>compute.acceleratorTypes</code></li> <li>■ <code>compute.addresses.get</code></li> <li>■ <code>compute.addresses.list</code></li> <li>■ <code>compute.addresses.use</code></li> <li>■ <code>compute.autoscalers</code></li> <li>■ <code>compute.diskTypes</code></li> <li>■ <code>compute.disks.create</code></li> <li>■ <code>compute.disks.createSnapshot</code></li> <li>■ <code>compute.disks.delete</code></li> <li>■ <code>compute.disks.get</code></li> <li>■ <code>compute.disks.list</code></li> <li>■ <code>compute.disks.resize</code></li> <li>■ <code>compute.disks.setLabels</code></li> <li>■ <code>compute.disks.update</code></li> <li>■ <code>compute.disks.use</code></li> <li>■ <code>compute.disks.useReadOnly</code></li> <li>■ <code>compute.globalAddresses.get</code></li> <li>■ <code>compute.globalAddresses.list</code></li> <li>■ <code>compute.globalAddresses.use</code></li> <li>■ <code>compute.globalOperations.get</code></li> <li>■ <code>compute.globalOperations.list</code></li> <li>■ <code>compute.images.get</code></li> <li>■ <code>compute.images.getFromFamily</code></li> <li>■ <code>compute.images.list</code></li> <li>■ <code>compute.images.useReadOnly</code></li> <li>■ <code>compute.instanceGroupManagers</code></li> <li>■ <code>compute.instanceGroups</code></li> <li>■ <code>compute.instanceTemplates</code></li> <li>■ <code>compute.instances</code></li> <li>■ <code>compute.licenses.get</code></li> <li>■ <code>compute.licenses.list</code></li> <li>■ <code>compute.machineTypes</code></li> <li>■ <code>compute.networkEndpointGroups</code></li> <li>■ <code>compute.networks.get</code></li> <li>■ <code>compute.networks.list</code></li> <li>■ <code>compute.networks.use</code></li> <li>■ <code>compute.networks.useExternalIp</code></li> <li>■ <code>compute.projects.get</code></li> <li>■ <code>compute.regionOperations.get</code></li> <li>■ <code>compute.regionOperations.list</code></li> <li>■ <code>compute.regions</code></li> <li>■ <code>compute.reservations.get</code></li> </ul>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ compute.reservations.list</li> <li>■ compute.subnetworks.get</li> <li>■ compute.subnetworks.list</li> <li>■ compute.subnetworks.use</li> <li>■ compute.subnetworks.useExternalIp</li> <li>■ compute.targetPools.get</li> <li>■ compute.targetPools.list</li> <li>■ compute.zoneOperations.get</li> <li>■ compute.zoneOperations.list</li> <li>■ compute.zones</li> <li>■ resourceManager.projects.get</li> <li>■ resourceManager.projects.list</li> <li>■ serviceusage.quotas.get</li> <li>■ serviceusage.services.get</li> <li>■ serviceusage.services.list</li> <li>■ roles/compute.instanceAdmin.v1</li> </ul> <p>Обеспечивает полный контроль над экземплярами вычислительного модуля, группами экземпляров, дисками, моментальными снимками и образами. Также предоставляет доступ на чтение всем сетевым ресурсам вычислительного модуля.</p> <hr/> <p><b>Примечание</b> Если пользователю назначить эту роль на уровне экземпляра, данный пользователь не сможет создавать новые экземпляры.</p> <hr/> <ul style="list-style-type: none"> <li>■ compute.acceleratorTypes</li> <li>■ compute.addresses.get</li> <li>■ compute.addresses.list</li> <li>■ compute.addresses.use</li> <li>■ compute.autoscalers</li> <li>■ compute.backendBuckets.get</li> <li>■ compute.backendBuckets.list</li> <li>■ compute.backendServices.get</li> <li>■ compute.backendServices.list</li> <li>■ compute.diskTypes</li> <li>■ compute.disks</li> <li>■ compute.firewalls.get</li> <li>■ compute.firewalls.list</li> <li>■ compute.forwardingRules.get</li> <li>■ compute.forwardingRules.list</li> <li>■ compute.globalAddresses.get</li> <li>■ compute.globalAddresses.list</li> <li>■ compute.globalAddresses.use</li> <li>■ compute.globalForwardingRules.get</li> <li>■ compute.globalForwardingRules.list</li> <li>■ compute.globalOperations.get</li> <li>■ compute.globalOperations.list</li> </ul>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ compute.healthChecks.get</li> <li>■ compute.healthChecks.list</li> <li>■ compute.httpHealthChecks.get</li> <li>■ compute.httpHealthChecks.list</li> <li>■ compute.httpsHealthChecks.get</li> <li>■ compute.httpsHealthChecks.list</li> <li>■ compute.images</li> <li>■ compute.instanceGroupManagers</li> <li>■ compute.instanceGroups</li> <li>■ compute.instanceTemplates</li> <li>■ compute.instances</li> <li>■ compute.interconnectAttachments.get</li> <li>■ compute.interconnectAttachments.list</li> <li>■ compute.interconnectLocations</li> <li>■ compute.interconnects.get</li> <li>■ compute.interconnects.list</li> <li>■ compute.licenseCodes</li> <li>■ compute.licenses</li> <li>■ compute.machineTypes</li> <li>■ compute.networkEndpointGroups</li> <li>■ compute.networks.get</li> <li>■ compute.networks.list</li> <li>■ compute.networks.use</li> <li>■ compute.networks.useExternalIp</li> <li>■ compute.projects.get</li> <li>■ compute.projects.setCommonInstanceMetadata</li> <li>■ compute.regionBackendServices.get</li> <li>■ compute.regionBackendServices.list</li> <li>■ compute.regionOperations.get</li> <li>■ compute.regionOperations.list</li> <li>■ compute.regions</li> <li>■ compute.reservations.get</li> <li>■ compute.reservations.list</li> <li>■ compute.resourcePolicies</li> <li>■ compute.routers.get</li> <li>■ compute.routers.list</li> <li>■ compute.routes.get</li> <li>■ compute.routes.list</li> <li>■ compute.snapshots</li> <li>■ compute.sslCertificates.get</li> <li>■ compute.sslCertificates.list</li> <li>■ compute.sslPolicies.get</li> <li>■ compute.sslPolicies.list</li> <li>■ compute.sslPolicies.listAvailableFeatures</li> <li>■ compute.subnetworks.get</li> </ul>

Задача	Действия
	<ul style="list-style-type: none"> <li>■ compute.subnetworks.list</li> <li>■ compute.subnetworks.use</li> <li>■ compute.subnetworks.useExternalIp</li> <li>■ compute.targetHttpProxies.get</li> <li>■ compute.targetHttpProxies.list</li> <li>■ compute.targetHttpsProxies.get</li> <li>■ compute.targetHttpsProxies.list</li> <li>■ compute.targetInstances.get</li> <li>■ compute.targetInstances.list</li> <li>■ compute.targetPools.get</li> <li>■ compute.targetPools.list</li> <li>■ compute.targetSslProxies.get</li> <li>■ compute.targetSslProxies.list</li> <li>■ compute.targetTcpProxies.get</li> <li>■ compute.targetTcpProxies.list</li> <li>■ compute.targetVpnGateways.get</li> <li>■ compute.targetVpnGateways.list</li> <li>■ compute.urlMaps.get</li> <li>■ compute.urlMaps.list</li> <li>■ compute.vpnTunnels.get</li> <li>■ compute.vpnTunnels.list</li> <li>■ compute.zoneOperations.get</li> <li>■ compute.zoneOperations.list</li> <li>■ compute.zones</li> <li>■ resourcemanager.projects.get</li> <li>■ resourcemanager.projects.list</li> <li>■ serviceusage.quotas.get</li> <li>■ serviceusage.services.get</li> <li>■ serviceusage.services.list</li> </ul>
Добавить облачную учетную запись NSX-T	<p>Укажите учетную запись со следующими правами чтения и записи:</p> <ul style="list-style-type: none"> <li>■ NSX-T IP-адрес или полное доменное имя (FQDN)</li> <li>■ ЦОД NSX-T — роль администратора предприятия и учетные данные для доступа</li> </ul> <p>Администраторам <i>также</i> требуется доступ к vCenter Server, как описано в разделе <i>Добавление облачной учетной записи vCenter</i> в этой таблице.</p>
Добавить облачную учетную запись NSX-V	<p>Укажите учетную запись со следующими правами чтения и записи:</p> <ul style="list-style-type: none"> <li>■ NSX-V Роль администратора предприятия и учетные данные для доступа</li> <li>■ NSX-V IP-адрес или полное доменное имя (FQDN)</li> </ul> <p>Администраторам <i>также</i> требуется доступ к vCenter Server, как описано в разделе <i>Добавление облачной учетной записи vCenter</i> в этой таблице.</p>



Задача	Действия
Добавить облачную учетную запись VMware Cloud on AWS (VMC)	<p>Укажите учетную запись со следующими правами чтения и записи:</p> <ul style="list-style-type: none"> <li>■ Учетная запись cloudadmin@vmc.local или любая учетная запись пользователя в группе CloudAdmin</li> <li>■ NSX Роль администратора предприятия и учетные данные для доступа</li> <li>■ NSX Доступ с правами администратора облачных служб к среде программно-определяемого ЦОД VMware Cloud on AWS вашей организации</li> <li>■ Доступ с правами администратора к среде программно-определяемого ЦОД VMware Cloud on AWS вашей организации</li> <li>■ VMware Cloud on AWS Маркер API-интерфейса для среды VMware Cloud on AWS в службе VMware Cloud on AWS вашей организации</li> <li>■ vCenter IP-адрес или полное доменное имя (FQDN)</li> </ul> <p>Администраторам также требуется доступ к vCenter Server, как описано в разделе <i>Добавление облачной учетной записи vCenter</i> в этой таблице.</p> <p>Разрешения, необходимые для создания и использования облачных учетных записей VMware Cloud on AWS, подробно описаны в разделе <i>Управление центром обработки данных VMware Cloud on AWS</i> в <a href="#">документации по продукту VMware Cloud on AWS</a>.</p>
Интеграция с vRealize Operations Manager	<p>Предоставьте следующие права чтения для локальной или внешней учетной записи в vRealize Operations Manager.</p> <ul style="list-style-type: none"> <li>■ Экземпляр адаптера — адаптер vCenter &gt; Экземпляр адаптера VC для vCenter-FQDN</li> </ul> <p>Возможно, необходимо импортировать внешнюю учетную запись перед назначением роли «только для чтения».</p>

## Настройка Microsoft Azure для использования с Cloud Assembly

Чтобы создать облачную учетную запись Microsoft Azure в Cloud Assembly, необходимо собрать сведения и настроить некоторые параметры.

### Процедура

- Найдите и запишите идентификаторы подписки и арендатора Microsoft Azure.
  - Идентификатор подписки. Щелкните значок «Подписки» на левой панели инструментов на портале Azure, чтобы посмотреть идентификатор подписки.
  - Идентификатор арендатора. Щелкните значок «Справка» и выберите «Показать данные диагностики» на портале Azure. Найдите арендатора и запишите его идентификатор.
- Чтобы начать работу, можно создать новую учетную запись хранилища и группу ресурсов. Их также можно создать позже в схемах элементов.
  - Учетная запись хранилища. Для настройки учетной записи выполните следующие действия.
    - На портале Azure найдите на боковой панели значок «Учетные записи хранилища». Убедитесь, что выбрана верная подписка, и щелкните **Добавить**. Для поиска учетной записи хранилища также можно использовать поле поиска Azure.

2. Введите необходимую информацию для учетной записи хранилища. Вам понадобится идентификатор подписки.
3. Укажите, следует ли использовать существующую группу ресурсов или создать новую. Запишите имя группы ресурсов: она понадобится вам позже.

---

**Примечание** Сохраните расположение учетной записи хранилища (оно понадобится вам позже).

---

3. Создайте виртуальную сеть. Если у вас уже есть подходящая сеть, можно выбрать ее.

При создании сети необходимо выбрать параметр «Использовать существующую группу ресурсов» и указать группу, созданную на предыдущем этапе. Кроме того, необходимо выбрать расположение, указанное ранее. Microsoft Azure не сможет развернуть виртуальные машины и другие объекты, если для всех применимых компонентов, которые будут использоваться объектом, не будет указано одно и то же расположение.

- а) Найдите значок «Виртуальная сеть» на левой панели и щелкните его или выполните поиск виртуальной сети. Убедитесь, что выбрана верная подписка, и нажмите **Добавить**.
- б) Введите уникальное имя для новой виртуальной сети и запишите его для последующего использования.
- в) Введите соответствующий IP-адрес для виртуальной сети в поле **Адресное пространство**.
- г) Убедитесь, что выбрана верная подписка, и щелкните **Добавить**.
- д) Введите остальные базовые сведения о конфигурации.
- е) Остальные параметры можно изменить, но в большинстве конфигураций можно оставить значения по умолчанию.
- ж) Щелкните **Создать**.

4. Настройте приложение Azure Active Directory для проверки подлинности в vRA.

- а) Найдите значок Active Directory в левом меню Azure и щелкните его.
- б) Щелкните **Регистрация приложений** и нажмите **Добавить**.
- в) Введите имя приложения, которое соответствует требованиям проверки имени в Azure.
- г) Оставьте в параметре «Тип приложения» значение «Веб-приложение/API-интерфейс» (Web app/API).
- д) Можно указать любой URL-адрес входа, подходящий для данного варианта использования.
- е) Щелкните **Создать**.

5. Создайте секретный ключ, чтобы выполнять аутентификацию приложения в Cloud Assembly.

- а) Щелкните имя приложения в Azure.  
Запишите идентификатор приложения, чтобы использовать его в дальнейшем.
- б) Щелкните **Все настройки** в следующей панели и выберите в списке настроек элемент «Ключи».
- в) Введите описание нового ключа и выберите продолжительность.

- г) Щелкните **Сохранить** и скопируйте значение ключа в безопасное место, поскольку в дальнейшем получить его будет невозможно.
  - д) В меню слева выберите **Разрешения API-интерфейса** для данного приложения и щелкните **Добавить разрешение**, чтобы создать новое разрешение.
  - е) Выберите «Управление службами Azure» на странице «Выбор API-интерфейса».
  - ж) Щелкните **Делегированные разрешения**.
  - з) В разделе «Выбор разрешений» выберите `user_impersonation`, а затем щелкните **Добавить разрешения**.
6. Выполните проверку подлинности для подключения приложения Active Directory к подписке Azure, чтобы иметь возможность развертывать виртуальные машины и управлять ими.
- а) В меню слева щелкните значок «Подписки» и выберите новую подписку.  
Возможно, понадобится щелкнуть текст названия, чтобы выполнить прокрутку панели.
  - б) Чтобы отобразить разрешения для подписки, выберите параметр «Управление доступом (IAM)».
  - в) Щелкните **Добавить** в разделе «Добавить назначение ролей».
  - г) Выберите «Участник» в раскрывающемся списке «Роль».
  - д) Оставьте значение по умолчанию в раскрывающемся списке «Предоставить доступ».
  - е) В поле «Выбрать» введите имя приложения.
  - ж) Нажмите кнопку **Сохранить**.
  - з) Добавьте дополнительные роли: «Владелец», «Участник» и «Читатель».
  - и) Нажмите кнопку **Сохранить**.

#### Следующие шаги

Необходимо установить средства интерфейса командной строки Microsoft Azure. Эти средства доступны бесплатно для операционных систем Windows и Mac. Дополнительные сведения о загрузке и установке этих средств см. в документации Microsoft.

После установки интерфейса командной строки необходимо пройти проверку подлинности в новой подписке.

1. Откройте окно терминала и введите имя пользователя Microsoft Azure. Вам будет отправлен URL-адрес и шорткод для проверки подлинности.
2. Введите в браузере код, полученный из приложения на устройстве.
3. Введите код проверки подлинности и щелкните **Продолжить**.
4. Выберите учетную запись Azure и войдите в нее.

При наличии нескольких подписок убедитесь, что выбран правильный вариант, с помощью команды `azure account set <subscription-name>`.

- Прежде чем продолжить, необходимо зарегистрировать поставщика Microsoft.Compute в новой подписке Azure с помощью команды `azure provider register microsoft.compute`.

Если при первом выполнении команды время ожидания истекает и появляется сообщение об ошибке, выполните команду снова.

После завершения настройки можно использовать команду `azure vm image list` для получения имен доступных образов виртуальных машин. Можно выбрать нужный образ, записать его URN, а затем использовать его в схемах элементов.

## Создание облачной учетной записи Microsoft Azure в vRealize Automation

Пользователи с правами администратора облачных систем могут создать облачную учетную запись Microsoft Azure для регионов учетной записи, в которых рабочая группа будет развертывать облачные шаблоны vRealize Automation.

Пример использования облачной учетной записи Microsoft Azure в службе vRealize Automation см. в [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).

### Необходимые условия

- Убедитесь, что у вас есть необходимые учетные данные администратора и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена необходимая роль пользователя. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Настройте учетную запись Microsoft Azure для использования с vRealize Automation. См. раздел [Настройка Microsoft Azure для использования с Cloud Assembly](#).
- Если у вас нет внешнего доступа к Интернету, настройте прокси-сервер Интернета. См. раздел [Настройка прокси-сервера Интернета для vRealize Automation](#).

### Процедура

- Выберите пункт **Инфраструктура > Подключения > Облачные учетные записи** и щелкните **Добавить облачную учетную запись**.
- Выберите тип учетной записи Microsoft Azure и введите учетные данные и другие необходимые значения.
- Щелкните **Проверить**.  
Выполняется поиск регионов учетных записей, связанных с этой учетной записью.
- Выберите регионы, в которых будет предоставлен данный ресурс.
- Для повышения эффективности можно нажать кнопку **Создать облачную зону для выбранных регионов**.

6. Введите теги возможностей, если этого требуют правила установки тегов. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).



Дополнительные сведения о том, как теги возможностей и теги ограничений помогают управлять размещением развертывания, см. в обучающем видео [Теги ограничений и размещение](#).

7. Нажмите **Сохранить**.

## Результаты

Учетная запись добавляется в службу vRealize Automation, и выбранные регионы становятся доступны для указанной облачной зоны.

## Следующие шаги

Создайте ресурсы инфраструктуры для этой облачной учетной записи.

При добавлении облачной учетной записи Azure в облачный шаблон при необходимости группы доступности можно использовать повторно. В подписках может быть не больше 2000 групп доступности, а также не больше 25 000 виртуальных машин. Поэтому по возможности группы доступности рекомендуется использовать повторно. Существуют два свойства YAML, которые можно использовать для управления применением групп доступности в развертываниях. Свойство `availabilitySetName` позволяет указать группу доступности, которую необходимо использовать. Второе свойство — `doNotAttachAvailabilitySet`, которое по умолчанию имеет значение «ложь». Если для этого свойства задано значение «истина», служба vRealize Automation создаст развертывание без группы доступности.

Создать развертывание без группы доступности невозможно, если используется подсистема балансировки нагрузки, подключенная к виртуальной машине.

В следующей таблице показано поведение службы vRealize Automation в зависимости от того, указаны ли группа ресурсов и группа доступности в облачном шаблоне.

Группа доступности не может существовать, если она не является частью группы ресурсов. В заданной группе ресурсов группы доступности должны иметь уникальные имена. Группы доступности могут иметь одинаковое имя только в том случае, если они входят в разные группы ресурсов.

Если имя группы ресурсов не указано, в службе vRealize Automation будет создана новая группа ресурсов. Это означает, что необходимо создать новую группу доступности, даже если имя передается. Для новой группы доступности будет использоваться переданное имя.

Таблица 3-16.

Указана группа ресурсов	Указана группа доступности	Результат
Нет	Нет	vRealize Automation создает новую группу ресурсов и новую группу доступности для виртуальной машины.
Да	Нет	vRealize Automation повторно использует существующую группу ресурсов и создает новую группу доступности для виртуальной машины.

Таблица 3-16. (продолжение)

Указана группа ресурсов	Указана группа доступности	Результат
Нет	Да	vRealize Automation создает новую группу ресурсов и новую группу доступности с указанным именем.
Да	Да	vRealize Automation повторно использует существующую группу ресурсов. Если в группе уже есть группа доступности с указанным именем, она также будет повторно использоваться. Если в группе нет группы доступности с указанным именем, будет создана новая группа с таким именем.

Cloud Assembly поддерживает моментальные снимки дисков Azure для развернутых виртуальных машин. Дополнительные сведения см. в [Работа с моментальными снимками дисков виртуальных машин Microsoft Azure в vRealize Operations Manager](#).

Cloud Assembly поддерживает несколько параметров диагностики загрузки для развертываний Azure. Диагностика загрузки поддерживает отладку виртуальных машин Azure и включает в себя сбор сведений журнала и соответствующих снимков экрана. Дополнительные сведения см. в [Использование функций диагностики загрузки и аналитической обработки журналов на виртуальной машине Microsoft Azure](#).

## Использование функций диагностики загрузки и аналитической обработки журналов на виртуальной машине Microsoft Azure

Для вызова и настройки функции диагностики загрузки Microsoft Azure из экземпляра Azure можно использовать облачный шаблон. Кроме того, для экземпляра виртуальной машины Azure можно настроить функцию аналитической обработки журналов. Диагностика загрузки — это функция отладки для виртуальных машин Azure, которая упрощает диагностику сбоев в ходе их загрузки. Используя функцию диагностики загрузки, пользователь может контролировать состояние виртуальной машины в ходе ее загрузки путем последовательного сбора журнальных записей и снимков экранов.

### Диагностика загрузки

Функция диагностики загрузки получает последовательные данные о журнальных записях и снимки экранов. Эти данные и снимки должны быть сохранены на диске. Существует два типа дисков: управляемый и неуправляемый диск Azure.

Свойство YAML `bootDiagnostics` поддерживается в облачных шаблонах Azure. Если для этого свойства задано значение `true`, диагностика загрузки включается в соответствующем развертывании виртуальной машины Azure.

В следующем фрагменте кода YAML приведен пример использования свойства `bootDiagnostics`.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Azure_Machine_1:
    type: Cloud.Azure.Machine
    metadata:
      layoutPosition:
```

```

- 0
- 0
properties:
  image: ubuntu
  flavor: small
  bootDiagnostics: true

```

Диагностику загрузки также можно вызвать на развернутой виртуальной машине Azure в рамках операции по регулярному обслуживанию. Перейдите на страницу «Развертывания» в Cloud Assembly и выберите развертывание Azure. Меню «Действия» на этой странице позволяет переключаться между параметрами «Включить диагностику загрузки» и «Отключить диагностику загрузки».

После развертывания облачного шаблона с включенной функцией диагностики загрузки на странице «Развертывания Cloud Assembly» для развертывания будет показано, что диагностика загрузки включена. Если функцию диагностики загрузки необходимо отключить, щелкните меню «Действия» на странице «Развертывания» и нажмите «Отключить диагностику загрузки».

### Аналитическая обработка журналов

Функция аналитической обработки журналов Azure позволяет изменять и запускать запросы журналов в отношении данных, полученных из журналов монитора Azure, а затем анализировать результаты в интерактивном режиме. С помощью таких запросов можно извлекать записи, которые соответствуют определенным критериям, искать тенденции и шаблоны и предоставлять различные аналитические данные. Если включить аналитику журналов на виртуальной машине Azure, она станет источником данных.

Перед настройкой функции аналитической обработки журналов в облачном шаблоне Cloud Assembly необходимо создать рабочую область «Аналитическая обработка журналов Azure» и настроить ее. Это можно сделать с помощью пункта «Виртуальные машины» в меню «Монитор Azure». Дополнительные сведения см. в документации по Microsoft Azure.

Для настройки функции аналитической обработки журналов необходимо иметь идентификатор рабочей области Azure и ключ рабочей области. Они указаны на вкладке «Управление агентами» в Azure в разделе «Рабочая область аналитики журналов».

В следующем примере облачного шаблона показан способ настройки функции аналитической обработки журналов с помощью расширений.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_Azure_Machine_1:
    type: Cloud.Azure.Machine
    properties:
      image: ubuntu
      flavor: small
    extensions:
      - autoUpgradeMinorVersion: true
        name: test-loga
        protectedSettings:
          workspaceKey: xxxxxxxxxx
        publisher: Microsoft.EnterpriseCloud.Monitoring
        settings:

```

```
workspaceId: aaaaaaaaaa
type: OmsAgentForLinux
typeHandlerVersion: '1.0'
```

После развертывания облачного шаблона с включенной функцией аналитической обработки журналов ее можно включить или отключить с помощью пунктов меню «Действия» на странице «Развертывания Cloud Assembly» для развертывания.

## Работа с моментальными снимками дисков виртуальных машин Microsoft Azure в vRealize Operations Manager

Для управляемых дисков Microsoft Azure можно создавать полные или инкрементные моментальные снимки.

На странице «Развертывания» в Cloud Assembly для развертывания на платформе Azure есть меню «Действия», содержащее разные команды для создания и удаления моментальных снимков управляемых дисков виртуальных машин и независимых управляемых дисков из развертываний в Azure. В следующем списке описаны поддерживаемые функции моментальных снимков.

- Создание моментального снимка диска — поддерживается как для внешних дисков, так и для дисков вычислительной системы. Кроме того, можно создать моментальные снимки диска в другой группе ресурсов.
- Удаление моментального снимка диска — поддерживается только для внешних дисков.
- Шифрование моментальных снимков с помощью набора средств шифрования диска Azure.
- Возможность предоставления пары «ключ-значение» в виде тегов во время создания моментального снимка.

В настоящее время моментальные снимки на неуправляемых дисках не поддерживаются.

Если используется шифрование, текущий способ реализации моментальных снимков поддерживает шифрование ключей под управлением платформы. По умолчанию политика сети разрешает доступ из любого места, поэтому ограничить доступ к моментальным снимкам через эту политику невозможно.

Дополнительные сведения об использовании страницы «Действия и развертывания» в Cloud Assembly см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

Дополнительные сведения о поддержке моментальных снимков Microsoft Azure см. в разделе [Создание моментального снимка виртуального жесткого диска](#) в документации по продукту Microsoft.

## Создание облачной учетной записи Amazon Web Services в vRealize Automation

Администраторы облачных систем могут создать облачную учетную запись Amazon Web Services (AWS) для регионов учетной записи, в которых рабочая группа будет развертывать облачные шаблоны vRealize Automation.

Ниже представлена процедура настройки облачной учетной записи AWS.



## Необходимые условия

- Убедитесь, что у вас есть необходимые учетные данные администратора и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена необходимая роль пользователя. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь в наличии необходимых учетных данных администратора AWS.
- Если у вас нет внешнего доступа к Интернету, настройте прокси-сервер Интернета. См. раздел [Настройка прокси-сервера Интернета для vRealize Automation](#).

## Процедура

1. Выберите **Инфраструктура > Подключения > Облачные учетные записи** и нажмите **Добавить облачную учетную запись**.
2. Выберите тип учетной записи AWS и введите учетные данные и другие необходимые значения.
3. Щелкните **Проверить**.  
Выполняется поиск регионов учетных записей, связанных с этой учетной записью.
4. Выберите регионы, в которых будет предоставлен данный ресурс.
5. Для повышения эффективности можно нажать кнопку **Создать облачную зону для выбранных регионов**.
6. Введите теги возможностей, если этого требуют правила установки тегов. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).



Дополнительные сведения о том, как теги возможностей и теги ограничений помогают управлять размещением развертывания, см. в обучающем видео [Теги ограничений и размещение](#).

7. Нажмите кнопку **Добавить**.

## Результаты

Учетная запись добавляется в службу vRealize Automation, и выбранные регионы становятся доступны для указанной облачной зоны.

## Следующие шаги

Выполните настройку ресурсов инфраструктуры для этой облачной учетной записи.

## Создание облачной учетной записи Google Cloud Platform в vRealize Automation

Администраторы облачных систем могут создать облачную учетную запись Google Cloud Platform (GCP) для регионов учетной записи, в которых рабочая группа будет развертывать облачные шаблоны vRealize Automation.

### Необходимые условия

- Убедитесь, что у вас есть необходимые учетные данные администратора и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена необходимая роль пользователя. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь в наличии доступа к ключу безопасности JSON для Google Cloud Platform.
- Убедитесь в наличии сведений по безопасности, необходимых для экземпляра Google Cloud Platform. Большую часть этих сведений можно найти в самом экземпляре либо в документации Google.
- Если у вас нет внешнего доступа к Интернету, настройте прокси-сервер Интернета. См. раздел [Настройка прокси-сервера Интернета для vRealize Automation](#).

### Процедура

1. В Cloud Assembly выберите пункт **Инфраструктура > Подключения > Облачные учетные записи** и щелкните **Добавить облачную учетную запись**.
2. Выберите тип учетной записи Google Cloud Platform и введите соответствующие учетные данные и сопутствующие сведения. Используйте учетную запись службы, созданную при инициализации вычислительного модуля исходной учетной записи GCP.

Как указано в разделе **Предварительные требования** выше, требования к учетным данным доступны в [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#). Чтобы успешно создать облачную учетную запись в vRealize Automation, для исходной учетной записи GCP должна быть включена служба вычислительного модуля.

В vRealize Automation идентификатор проекта является частью конечной точки Google Cloud Platform. Этот параметр указывается при создании облачной учетной записи. В процессе сбора данных о частных образах для конкретного проекта адаптер vRealize Automation GCP запрашивает API-интерфейс Google Cloud Platform.

3. Щелкните **Проверить**.  
Выполняется поиск регионов учетных записей, связанных с этой учетной записью.
4. Выберите регионы, в которых будет предоставлен данный ресурс.
5. Для повышения эффективности можно нажать кнопку **Создать облачную зону для выбранных регионов**.

6. Если политика установки тегов требует добавления тегов, введите теги возможностей. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).



Дополнительные сведения о том, как теги возможностей и теги ограничений помогают управлять размещением развертывания, см. в обучающем видео [Теги ограничений и размещение](#).

7. Нажмите кнопку **Добавить**.

## Результаты

Учетная запись добавляется в службу vRealize Automation, и выбранные регионы становятся доступны для указанной облачной зоны.

## Следующие шаги

Создайте ресурсы инфраструктуры для этой облачной учетной записи.

Ниже приведены сведения о развертывании виртуальной машины Google Cloud Platform в Cloud Assembly.

При добавлении облачной учетной записи Google Cloud Platform в облачный шаблон Cloud Assembly можно использовать свойство `useSoleTenant` YAML, чтобы указать, что виртуальную машину необходимо развернуть только на одном узле арендатора. Такая конфигурация позволяет изолировать виртуальные машины для обеспечения безопасности, конфиденциальности и проч.

Чтобы упростить эту функцию, метки сходства узла Google Cloud Platform преобразовываются в теги в Cloud Assembly, которые применяются в соответствующих зонах доступности vRealize Automation, где находятся группы узлов. Если для свойства `useSoleTenant` задано значение «истина», в качестве тегов ограничений должна использоваться одна из меток сходства узла. Кроме того, для развертывания компьютера в режиме единственного арендатора (sole tenant) в облачный шаблон необходимо добавить свойство `useSoleTenant`, а также теги ограничений.

Прежде чем использовать эту функцию, в Google Cloud Platform необходимо создать соответствующий шаблон узла и метки сходства узла, а затем создать группу узлов.

В следующем примере кода YAML показано, как можно использовать свойство `useSoleTenant` в облачных шаблонах Cloud Assembly. Теги ограничений — это метки сходства узла, которые были автоматически получены с сервера Google Cloud Platform.

```
resources:
  Cloud_GCP_Machine_1:
    type: Cloud.GCP.Machine
    properties:
      image: ubuntu
      flavor: c2-family
      name: demo-vm
      useSoleTenant: true
      constraints:
        -tag: 'env:prod'
        -tag: 'region:asia-east1'
```

## Создание облачной учетной записи vCenter в vRealize Automation

Облачную учетную запись vCenter можно добавить для тех регионов учетной записи, в которых планируется выполнить развертывание облачных шаблонов vRealize Automation.

В целях безопасности и оптимального использования сети облачную учетную запись vCenter можно связать с облачной учетной записью NSX-T или NSX-V.

Облачную учетную запись NSX-T можно связать с одной или несколькими облачными учетными записями vCenter. Тем не менее облачную учетную запись NSX-V можно связать только с одной облачной учетной записью vCenter.

### Необходимые условия

- Убедитесь, что у вас есть необходимые учетные данные администратора и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что порты и протоколы корректно настроены для поддержки облачной учетной записи. См. раздел *Порты и протоколы для vRealize Automation* в документе *Установка vRealize Automation с помощью vRealize Easy Installer* и раздел *Требования к портам* в документе *Руководство по эталонной архитектуре vRealize Automation* в документации по продукту vRealize Automation.

### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Облачные учетные записи** и щелкните **Добавить облачную учетную запись**.
2. Выберите тип учетной записи vCenter и введите IP-адрес узла vCenter Server.
3. Введите учетные данные администратора vCenter Server и нажмите кнопку **Проверить**.

Выполняется сбор данных в центрах обработки данных, связанных с этой учетной записью. Данные, а также все теги vSphere собираются по следующим элементам.

- Компьютеры
  - Кластеры и узлы
  - Группы портов
  - Хранилища данных
4. Выберите по крайней мере один из доступных центров обработки данных в указанном экземпляре vCenter Server, чтобы обеспечить предоставление ресурсов для этой облачной учетной записи.
  5. Чтобы повысить эффективность, можно создать облачную зону для предоставления ресурсов в выбранных центрах обработки данных.

Создание облачных зон можно также вынести в отдельный этап, если этого требует принятая в организации политика работы с облаком.

Дополнительные сведения об облачных зонах см. в разделе [Дополнительные сведения об облачных зонах Cloud Assembly](#).

**6. Выберите существующую облачную учетную запись NSX.**

Учетную запись NSX можно выбрать сейчас или позже при редактировании ее параметров.

Дополнительные сведения об облачных учетных записях NSX-V см. в разделе [Создание облачной учетной записи NSX-V в vRealize Automation](#).

Дополнительные сведения об облачных учетных записях NSX-T см. в разделе [Создание облачной учетной записи NSX-T в vRealize Automation](#).

Дополнительные сведения о внесении изменений в связи после развертывания облачного шаблона см. в разделе [Что произойдет, если удалить связь облачной учетной записи NSX в vRealize Automation](#).

**7. Если для политики тегирования необходимо добавить теги, введите теги возможностей.**

Теги можно выбрать сейчас или позже при редактировании облачной учетной записи. Сведения о тегировании см. в разделе [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#).



Дополнительные сведения о том, как теги возможностей и теги ограничений помогают управлять размещением развертывания, см. в обучающем видео [Теги ограничений и размещение](#).

**8. Нажмите **Сохранить**.**

**Результаты**

Облачная учетная запись добавлена, и выбранные центры обработки данных доступны в указанной облачной зоне. Собранные данные, в том числе сведения о компьютерах, сетях, хранилище и томах, перечислены в разделе **Ресурсы** на вкладке **Инфраструктура**.

**Следующие шаги**

Выполните настройку остальных ресурсов инфраструктуры для этой облачной учетной записи. См. раздел [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).

## Создание облачной учетной записи NSX-V в vRealize Automation

В целях безопасности и оптимального использования сети можно создать облачную учетную запись NSX-V и связать ее с облачной учетной записью vCenter.

Облачную учетную запись NSX-V можно связать только с одной облачной учетной записью vCenter.

Связь между NSX-V и облачной учетной записью vCenter должна быть настроена за пределами vRealize Automation — в приложении NSX. vRealize Automation не создает связи между NSX и vCenter. В vRealize Automation необходимо указать связь, которая уже существует в NSX.

## Необходимые условия

- Убедитесь, что у вас есть необходимые учетные данные администратора и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь в наличии облачной учетной записи vCenter, которая будет использоваться вместе с этой облачной учетной записью NSX. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).
- Убедитесь, что порты и протоколы корректно настроены для поддержки облачной учетной записи. См. раздел *Порты и протоколы для vRealize Automation* в документе *Установка vRealize Automation с помощью vRealize Easy Installer* и раздел *Требования к портам* в документе *Руководство по эталонной архитектуре vRealize Automation* в [документации по продукту vRealize Automation](#).

## Процедура

1. Выберите пункт **Инфраструктура > Подключения > Облачные учетные записи** и щелкните **Добавить облачную учетную запись**.
2. Выберите тип учетной записи NSX-V и введите IP-адрес узла NSX-V.
3. Введите учетные данные администратора NSX и нажмите кнопку **Проверить**.  
Выполняется поиск ресурсов, связанных с этой учетной записью.  
Если IP-адрес узла NSX недоступен, проверка завершится ошибкой.
4. При наличии выберите конечную точку vCenter, представляющую облачную учетную запись vCenter, которую необходимо привязать к этой облачной учетной записи NSX-V.  
Для выбора доступны только облачные учетные записи vCenter, которые в данный момент не связаны с облачной учетной записью NSX-T или NSX-V.  
Дополнительные сведения о внесении изменений в связи после развертывания облачного шаблона см. в разделе [Что произойдет, если удалить связь облачной учетной записи NSX в vRealize Automation](#).
5. Если для политики тегирования необходимо добавить теги, введите теги возможностей.  
Теги возможностей можно добавить или удалить позже. См. раздел [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#).



Дополнительные сведения о том, как теги возможностей и теги ограничений помогают управлять размещением развертываний, см. в обучающем видео [Теги ограничений и размещение](#).

6. Нажмите **Сохранить**.

## Следующие шаги

Можно создать новую или изменить существующую облачную учетную запись vCenter, чтобы связать ее с этой учетной записью NSX. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).

Создайте и настройте одну или несколько облачных зон для использования с центрами обработки данных, которые используются этой облачной учетной записью. См. раздел [Дополнительные сведения об облачных зонах Cloud Assembly](#).

Выполните настройку ресурсов инфраструктуры для этой облачной учетной записи. См. раздел [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).

## Создание облачной учетной записи NSX-T в vRealize Automation

В целях безопасности и оптимального использования сети можно создать облачную учетную запись NSX-T и связать ее с одной или несколькими облачными учетными записями vCenter.

Облачную учетную запись NSX-T можно связать с одной или несколькими облачными учетными записями vCenter. Тем не менее облачную учетную запись NSX-V можно связать только с одной облачной учетной записью vCenter.

Связь между NSX-T и одним или несколькими облачными учетными записями vCenter должна быть настроена за пределами vRealize Automation, прежде всего, в приложении NSX. vRealize Automation не создает связи между NSX и vCenter. В vRealize Automation нужно указать одну или несколько связей конфигурации, которые уже существуют в NSX.

При создании облачной учетной NSX-T в vRealize Automation указывается тип диспетчера и режим NSX. После создания облачной учетной записи эти настройки будет нельзя изменить.

Можно подключиться к глобальному диспетчеру NSX-T и настроить связь между глобальным диспетчером NSX-T и локальными диспетчерами в контексте федерации NSX-T.

Дополнительные сведения о параметрах и возможностях NSX-T см. в [документации по продукту NSX-T Data Center](#).

Чтобы обеспечить отказоустойчивость и высокую доступность в развертываниях, каждая конечная точка центра обработки данных NSX-T представляет собой кластер из трех диспетчеров NSX.

- vRealize Automation может указывать на одну из подсистем NSX Manager. При использовании этого варианта один NSX Manager получает вызовы API-интерфейса из vRealize Automation.
- vRealize Automation может указывать на виртуальный IP-адрес кластера. При использовании этого варианта один NSX Manager берет на себя управление виртуальным IP-адресом. NSX Manager получает вызовы API-интерфейса из vRealize Automation. В случае сбоя другой узел в кластере принимает на себя управление виртуальным IP-адресом и получает вызовы API-интерфейса из vRealize Automation.

Дополнительные сведения о настройке виртуального IP-адреса для NSX см. в разделе *Настройка виртуального IP-адреса (VIP) для кластера в руководстве по установке NSX-T Data Center* в [документации по VMware по NSX-T Data Center](#).

- vRealize Automation может указывать на виртуальный IP-адрес подсистемы балансировки нагрузки для равномерного распределения вызовов между тремя подсистемами NSX Manager. В этом случае все три подсистемы NSX Manager получают вызовы API-интерфейса из vRealize Automation.

Виртуальный IP-адрес можно настроить в сторонней подсистеме балансировки нагрузки или в подсистеме балансировки нагрузки NSX-T.

Данный вариант рекомендуется использовать для крупных сред, чтобы распределять вызовы API-интерфейса vRealize Automation между тремя подсистемами NSX Manager.

Подробные сведения об использовании NSX-T 3.2 с vRealize Automation см. в записи блога VMware [VMware Network Automation с NSX-T 3.2 и vRealize Automation](#).

#### Необходимые условия

- Убедитесь, что у вас есть необходимые учетные данные администратора и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь в наличии облачной учетной записи vCenter, которая будет использоваться вместе с этой облачной учетной записью NSX. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).
- Убедитесь, что порты и протоколы корректно настроены для поддержки облачной учетной записи. См. раздел *Порты и протоколы для vRealize Automation* в документе *Установка vRealize Automation с помощью vRealize Easy Installer* и раздел *Требования к портам* в документе *Руководство по эталонной архитектуре vRealize Automation* в [документации по продукту vRealize Automation](#).

#### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Облачные учетные записи** и щелкните **Добавить облачную учетную запись**.
2. Выберите тип учетной NSX-T и введите имя облачной учетной записи, а также ее описание.
3. Введите IP-адрес узла для экземпляра NSX-T Manager или виртуальный IP-адрес (сведения о предполагаемых действиях для вариантов использования NSX Manager и виртуального IP-адреса см. выше).
4. Введите имя пользователя NSX и пароль для администратора.
5. В поле **Тип диспетчера** выберите **Глобальный** или **Локальный** (значение по умолчанию).

- Глобальный диспетчер

Параметр «Глобальный диспетчер» доступен для использования только с параметром политики **Режим NSX**. При использовании параметра диспетчера **Режим NSX** он будет недоступен.

Параметр «Глобальный» относится к возможностям федерации NSX-T, включая глобальные сегменты сети. Федерацию NSX-T поддерживают только облачные учетные записи NSX-T с параметром «Глобальный».



При использовании параметра «Глобальный диспетчер» система предлагает определить облачную учетную запись NSX-T локального диспетчера и связанную облачную учетную запись vCenter Server.

Облачную учетную запись NSX-T глобального диспетчера нельзя связать с облачной учетной записью vCenter, но можно связать с облачной учетной записью NSX-T локального диспетчера. Подобно тому как облачная учетная запись NSX-T локального диспетчера может быть связана с несколькими облачными учетными записями vCenter, облачная учетная запись NSX-T глобального диспетчера может быть связана с несколькими облачными учетными записями NSX-T локального диспетчера.

- Локальный диспетчер

Используйте параметр «Локальный», чтобы определить традиционную облачную учетную запись NSX-T, которую можно связать с одной облачной учетной записью vSphere или несколькими. Облачную учетную запись NSX-T глобального диспетчера можно связать с облачными учетными записями локального NSX-T. Следует отметить, что этот параметр также нужно использовать при создании новой пустой целевой облачной учетной записи NSX-T для переноса с NSX-V на NSX-T.

После создания облачной учетной записи параметр **Тип диспетчера** изменить нельзя.

## 6. Для режима NSX выберите **Policy** или **Manager**.

- Режим Policy (по умолчанию)

Режим Policy доступен в NSX-T 3.0, NSX-T 3.1 и последующих версиях. Благодаря этому варианту vRealize Automation может использовать дополнительные возможности, доступные в API-интерфейсе NSX-T Policy.

Если вы используете NSX-T с облачной учетной записью VMware Cloud on AWS в облачном шаблоне, то облачная учетная запись NSX-T должна использовать **режим NSX Policy**.

Параметр Policy относится к форме API-интерфейса Policy NSX-T для NSX-T.

- Режим Manager

Существующие конечные точки или облачные учетные записи NSX-T, обновленные с более ранней версии vRealize Automation и не поддерживающие вариант Policy, воспринимаются как облачные учетные записи NSX-T в режиме Manager.

Режим Manager поддерживается в версиях NSX-T 2.4, NSX-T 3.0 и NSX-T 3.1.

Если указан режим Manager, используйте вариант «Режим Manager» для других облачных учетных записей NSX-T, пока в vRealize Automation в варианты переноса режима Policy не будет добавлен режим Manager.

Для некоторых функций vRealize Automation для NSX-T требуется NSX-T 3.0 или более поздней версии, в том числе для функции добавления тегов в компоненты сетевых адаптеров виртуальных машин в облачном шаблоне.

Параметр Manager относится к форме API-интерфейса NSX-T Manager для NSX-T.

Если у вас есть существующие облачные учетные записи NSX-T, созданные до реализации режима Policy в vRealize Automation 8.2, в них используется метод API-интерфейса Manager. Рекомендуется подождать, пока в vRealize Automation не появится средство перехода с API-интерфейс Manager на API-интерфейс Policy. Если вы не хотите ждать, замените существующие облачные учетные записи NSX-T новыми облачными учетными записями NSX-T, в которых задан метод API-интерфейса Policy.

После создания облачной учетной записи **режим NSX** изменить нельзя.

7. Щелкните **Проверить**, чтобы подтвердить учетные данные для выбранного типа NSX Manager и режима NSX.

Выполняется поиск ресурсов, связанных с этой учетной записью.

Если IP-адрес узла NSX недоступен, проверка завершится ошибкой.

8. В разделе **Связи** добавьте одну или несколько облачных учетных записей vCenter для связи с этой облачной учетной записью NSX-T. Можно удалить существующие связи облачных учетных записей vCenter.

Для выбора доступны только облачные учетные записи vCenter, которые в данный момент не связаны в vRealize Automation с облачной учетной записью NSX-T или NSX-V.

См. раздел [Что можно делать с сопоставлением NSX-T с несколькими системами vCenter в vRealize Automation](#).

Сведения о внесении изменений в облачную учетную запись связи или об ее удалении после развертывания облачного шаблона см. в разделе [Что произойдет, если удалить связь облачной учетной записи NSX в vRealize Automation](#).

9. Если для политики тегирования необходимо добавить теги, введите теги возможностей.

Теги возможностей можно добавить или удалить позже. См. раздел [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#).



Дополнительные сведения о том, как теги возможностей и теги ограничений помогают управлять размещением развертывания, см. в обучающем видео [Теги ограничений и размещение](#).

10. Нажмите **Сохранить**.

#### Следующие шаги

Можно создать новую или изменить существующую облачную учетную запись vCenter, чтобы связать ее с этой учетной записью NSX. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).

Создайте и настройте одну или несколько облачных зон для использования с центрами обработки данных, которые используются этой облачной учетной записью. См. раздел [Дополнительные сведения об облачных зонах Cloud Assembly](#).

Выполните настройку ресурсов инфраструктуры для этой облачной учетной записи. См. раздел [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).

Примеры использования вариантов NSX-T в облачных шаблонах vRealize Automation см. в [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Создание облачной учетной записи VMware Cloud on AWS в vRealize Automation

Пользователи с правами администратора облачных систем могут создать облачную учетную запись VMware Cloud on AWS для регионов учетной записи, в которых рабочая группа будет развертывать облачные шаблоны vRealize Automation.

Для использования VMware Cloud on AWS в vRealize Automation требуется выполнить специальную настройку. Сведения о точной настройке vRealize Automation для использования VMware Cloud on AWS, в том числе о задании значений маркера API-интерфейса для облачной учетной записи и настройке правил брандмауэра шлюза для соответствующего облачного прокси-сервера, см. в описании рабочего процесса [Учебник «Настройка VMware Cloud on AWS для vRealize Automation»](#).

### Необходимые условия

- Убедитесь, что у вас есть необходимые учетные данные администратора VMware Cloud on AWS, в том числе учетные данные CloudAdmin для VMware Cloud on AWS для целевого программно-определяемого ЦОД в vCenter, и что на порте 443 включен доступ HTTPS. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Если у вас нет внешнего доступа к Интернету, настройте прокси-сервер Интернета. См. раздел [Настройка прокси-сервера Интернета для vRealize Automation](#).
- Убедитесь, что в программно-определяемом ЦОД настроены необходимые правила доступа и брандмауэра. См. раздел [Подготовка программно-определяемого ЦОД VMware Cloud on AWS к подключению с помощью облачных учетных записей VMware Cloud on AWS в vRealize Automation](#).

### Процедура

1. Выберите **Инфраструктура > Подключения > Облачные учетные записи**, нажмите **Добавить облачную учетную запись** и выберите тип учетной записи VMware Cloud on AWS.
2. Добавьте **маркер API-интерфейса VMware Cloud on AWS**, чтобы предоставить организации доступ к имеющимся программно-определяемым ЦОД.

Можно создать новый маркер или использовать существующий маркер, назначенный организации (см. страницу [Маркеры API-интерфейса](#) по ссылке). Подробности см. в разделе [Создание облачной учетной записи VMware Cloud on AWS в службе vRealize Automation в рамках образца рабочего процесса](#).

3. Выберите программно-определяемый ЦОД, который будет доступен для развертываний.

Программно-определяемые ЦОД NSX-V не поддерживаются и не отображаются в списке

Значения IP-адреса и полного доменного имени vCenter и NSX-TManager автоматически заполняются в соответствии с выбранным программно-определяемым ЦОД.

4. Введите имя пользователя vCenter и пароль для заданного программно-определяемого ЦОД, если используется имя, отличное от значения по умолчанию (cloudadmin@vmc.local).
5. Нажмите **Проверить**, чтобы убедиться в наличии прав доступа к заданному экземпляру vCenter и в том, что этот экземпляр vCenter запущен и работает.

Выполняется поиск центров обработки данных, связанных с этой учетной записью.

6. Для повышения эффективности можно создать облачную зону для предоставления ресурсов в выбранном программно-определяемом ЦОД.

Создание облачных зон можно также вынести в отдельный этап, если этого требует принятая в организации политика работы с облаком.

7. Если для политики тегирования необходимо добавить теги, введите теги возможностей.

Теги возможностей можно добавить или удалить позже. См. раздел [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#).



Дополнительные сведения о том, как теги возможностей и теги ограничений помогают управлять размещением развертывания, см. в обучающем видео [Теги ограничений и размещение](#).

Как и для виртуальных машин, развертываемых в vSphere, теги компьютера можно настроить для виртуальной машины, развертываемой в VMware Cloud on AWS. Тег компьютера также можно обновить после первоначального развертывания. Эти теги компьютера позволяют vRealize Automation динамически назначать виртуальную машину соответствующей группе безопасности NSX-T во время развертывания. Дополнительные сведения см. в разделе [Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation](#).

8. Нажмите **Сохранить**.

#### Результаты

Облачная учетная запись добавлена, и выбранный программно-определяемый ЦОД доступен в указанной облачной зоне.

#### Следующие шаги

Сведения о точной настройке vRealize Automation для использования VMware Cloud on AWS см. в разделе [Учебник «Настройка VMware Cloud on AWS для vRealize Automation»](#).

Сопутствующие сведения об использовании VMware Cloud on AWS вне среды vRealize Automation см. в [документации по VMware Cloud on AWS](#).

## Создание облачной учетной записи VMware Cloud Foundation

Чтобы использовать домены рабочей нагрузки, можно настроить VMware Cloud Foundation (VCF) как облачную учетную запись в Cloud Assembly.

Облачная учетная запись VCF позволяет встроить рабочую нагрузку VCF в Cloud Assembly, чтобы обеспечить комплексное решение для управления гибридной облачной средой. Cloud Assembly предлагает несколько точек входа, в которых можно активировать страницу настройки облачной учетной записи VCF. Если доступ к этой странице осуществляется с помощью кнопки **Добавить облачную учетную запись** на вкладке «Домен рабочей нагрузки» в интеграции SDDC, рабочая нагрузка, включая основные данные для vCenter и NSX Manager, выбраны заранее.

### Необходимые условия

Для использования с этой облачной учетной записью требуется экземпляр VMware SDDC Manager 4.1 или более поздней версии, настроенный в качестве интеграции с Cloud Assembly. Дополнительные сведения см. в разделе [Настройка интеграции VMware SDDC Manager](#).

### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Облачные учетные записи** и щелкните **Добавить облачную учетную запись**.
2. Выберите тип «Облачная учетная запись VCF» и заполните поля **Имя** и **Описание**.
3. Введите полное доменное имя и учетные данные для экземпляра SDDC Manager, который используется с этой облачной учетной записью.  
  
Этот шаг можно пропустить, если экземпляр SDDC Manager, который будет использоваться с этой учетной записью, уже настроен.
4. Выберите один или несколько доменов рабочей нагрузки, которые нужно использовать с этой облачной учетной записью VCF.
5. Если необходимо, чтобы служба Cloud Assembly использовала учетные данные управляемой службы Cloud Foundation для vCenter и NSX, выберите **Автоматически создавать учетные данные службы**. Затем, если эти учетные данные нужно будет изменить, для управления паролями необходимо использовать механизм VCF.  
  
При выборе этого параметра можно пропустить шаги 7 и 8.
6. Введите учетные данные, необходимые для доступа к экземпляру vCenter, связанному с этой облачной учетной записью.
7. Если учетные данные для облачной учетной записи VCF требуется ввести вручную, в заголовке NSX Manager введите учетные данные NSX. Если требуется, чтобы учетные данные NSX были созданы и проверены с помощью Cloud Assembly, щелкните «Создать и проверить учетные данные службы».
8. Введите учетные данные, необходимые для доступа к сети NSX-T, связанной с этой облачной учетной записью.
9. При необходимости выберите режим NSX.
10. Щелкните **Проверить**, чтобы подтвердить подключение к SDDC Manager.
11. При необходимости выберите центры обработки данных, которые нужно подготовить, в разделе «Конфигурация». Установите флажок, если необходимо создать облачную зону для выбранных центров обработки данных.

12. Если для политики тегирования используются теги, введите теги возможностей. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).
13. Нажмите **Сохранить**.

### Результаты

Эта облачная учетная запись позволяет использовать выбранный домен рабочей нагрузки, связанный с указанным экземпляром SDDC Manager, в Cloud Assembly.

Если требуются возможности управления дополнительными доменами рабочих нагрузок с помощью vRealize Automation, эти действия нужно повторить для каждого домена.

### Следующие шаги

После настройки облачной учетной записи VCF можно выбрать учетную запись на главной странице облачных учетных записей и щелкнуть **Настройка облачной среды**, чтобы открыть мастер быстрого запуска VMware Cloud Foundation для настройки облачной среды.

Дополнительные сведения о мастере быстрого запуска см. в статье [Начало работы с vRealize Automation с помощью быстрого запуска VMware Cloud Foundation](#) в разделе «Начало работы».

## Создание облачной учетной записи VMware Cloud Director в vRealize Automation

Для развертывания виртуальных машин Cloud Director с использованием объектов, независимых от облачной среды, можно создать облачную учетную запись VMware Cloud Director в vRealize Automation. Cloud Director поддерживает гибкое предоставление сетевых и вычислительных ресурсов, а также ресурсов хранения. Интерфейс портала позволяет управлять экземплярами vCenter, их сетевыми устройствами NSX-T и NSX-V, а также связанными виртуальными центрами обработки данных посредством каталога.

Облачная учетная запись VMware Cloud Director поддерживает создание автономных виртуальных машин Cloud Director без vApp. Поддерживается три сценария подготовки виртуальных машин Cloud Director посредством облачных шаблонов Cloud Assembly.

- Виртуальные машины
- Подключенные сети виртуальных машин
- Виртуальные машины с дополнительными дисками

Дополнительные сведения о работе с VMware Cloud Director, включая сведения о настройке нескольких серверов для обеспечения высокой доступности, см. в официальной документации на странице <https://docs.vmware.com/ru/VMware-Cloud-Director/index.html>.

Облачная запись VMware Cloud Director поддерживает до 1000 виртуальных машин с vRealize Automation в устойчивом режиме.

Ниже описана процедура настройки облачной учетной записи VMware Cloud Director в vRealize Automation Cloud Assembly.

## Необходимые условия

- Настройте развертывание VMware Cloud Director 10.2.0, 10.2.1, 10.2.2, 10.3 или 10.3.1 для одной или нескольких соответствующих организаций.
- Пользователи, указанные для этой интеграции, должны иметь права администратора организации, чтобы читать применимые шаблоны, создавать виртуальные машины и просматривать другие ресурсы, такие как политики вычислительных ресурсов, диски, виртуальные центры обработки данных и т. д. Облачная учетная запись VCD для vRealize Automation работает в контексте арендатора в Cloud Director, поэтому подключаться к отдельной организации в Cloud Director нужно с учетными данными арендатора. Дополнительные сведения о необходимых учетных данных см. в разделе [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- До или после настройки интеграции необходимо настроить соответствующее хранилище, сеть, образ и конфигурации ресурсов или политику размера в экземпляре VMware Cloud Director и сопоставить эти объекты с Cloud Assembly в vRealize Automation. В следующем списке объясняется, как виртуальные объекты VMware Cloud Director должны быть сопоставлены с объектами vRealize Automation в Cloud Assembly.
  - Сети организации VMware Cloud Director (изолированные, прямые, маршрутизируемые) сопоставляются с сетями vRealize Automation. Для сетевого адаптера нельзя задать пул статических IP-адресов.
  - Политики размера виртуальных машин VMware Cloud Director сопоставляются с конфигурациями vRealize Automation.
  - Политики хранилища VMware Cloud Director сопоставляются с профилями хранилища vRealize Automation.
  - Образы VMware Cloud Director (OVF, загрузочный носитель ISO) сопоставляются с образами vRealize Automation. Образы могут быть шаблонами vApp или носителями, например файлами ISO. При использовании ISO создается пустая виртуальная машина и носитель присоединяется как загрузочный носитель.
  - Виртуальные машины VMware Cloud Director сопоставляются с вычислительными ресурсами vRealize Automation.
  - Диски виртуальных машин VMware Cloud Director сопоставляются с облачными томами vRealize Automation.

Эти объекты VMware Cloud Director сопоставляются с объектами vRealize Automation с помощью параметров на страницах в разделе **Инфраструктура > Конфигурация >** в Cloud Assembly.

Подробные сведения о сопоставлении объектов в vRealize Automation см. в соответствующих темах в разделе [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).

## Процедура

1. Выберите пункт **Инфраструктура > Подключения > Облачные учетные записи** и щелкните **Добавить облачную учетную запись**.



2. Выберите тип «Облачная учетная запись VMware Cloud Director» и заполните поля **Имя** и **Описание**.
3. Введите соответствующие сведения учетной записи, необходимые для доступа к серверу VMware Cloud Director.
4. Введите основной URL-адрес, который будет использоваться для подключения к серверу VMware Cloud Director.
5. Заполните поля **Имя пользователя** и **Пароль** для действующей учетной записи с доступом к указанному экземпляру Cloud Director.
6. Введите желаемое имя **организации**, которое будет использоваться с этой интеграцией.  
 В vCloud Director организация содержит пользователей, объекты vApp, которые они создают, и ресурсы, которые используются в объектах vApp.
7. Щелкните **Проверить**.  
 Во время проверки может появиться запрос на принятие сертификата. После проверки подключения можно выбрать дополнительные параметры.
8. Если для политики тегирования используются теги, введите теги возможностей. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).
9. После проверки на странице отобразится список виртуальных центров обработки данных Cloud Director. Выберите соответствующий центр обработки данных. От этого выбора зависит, в каких регионах можно будет проводить развертывание.
10. Щелкните **Добавить**, чтобы добавить облачную учетную запись VMware Cloud Director в vRealize Automation.

## Результаты

Облачная учетная запись VMware Cloud Director доступна для настройки в vRealize Automation. Сети, связанные с экземпляром Cloud Director, доступны для настройки на странице **Ресурсы > Сети** в Cloud Assembly. Можно настроить соответствующие профили хранилища, а затем использовать облачную учетную запись для создания развертываний в облачных шаблонах. Кроме того, убедитесь, что в Cloud Assembly настроен соответствующий проект для использования с экземпляром Cloud Director.

## Следующие шаги

Облачная учетная запись VMware Cloud Director готова к использованию в облачных шаблонах Cloud Assembly.

Ниже приведен пример облачного шаблона для базового развертывания VMware Cloud Director.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Network_1:
    type: Cloud.Network
    properties:
```



```

    networkType: existing
    constraints:
      - tag: net1:isolated
  Cloud_Volume_1:
    type: Cloud.Volume
    properties:
      capacityGb: 2
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: image1
      flavor: small
      storage:
        constraints:
          - tag: storage:development
      attachedDisks:
        - source: '${resource.Cloud_Volume_1.id}'
      networks:
        - network: '${resource.Cloud_Network_1.id}'

```

На развернутых виртуальных машинах VMware Cloud Director поддерживаются следующие действия по регулярному обслуживанию.

- Включение
- Выключение
- Приостановка
- Создание моментальных снимков
- Возврат к моментальному снимку
- Удаление моментальных снимков
- Добавление диска
- Удаление диска
- Изменение размера диска (Примечание. Поддерживается только увеличение размера диска.)
- Изменение размера загрузочного диска

После развертывания схемы элементов пользователи могут применить теги к недавно подготовленным компьютерам в vRealize Automation. Эти теги vRealize Automation сопоставляются с метаданными VMware Cloud Director, которые можно получить с помощью API-интерфейса VMware Cloud Director. Пользователи также могут помечать тегами другие ресурсы vRealize Automation, но обновляться будут только компьютеры в среде VMware Cloud Director, так как это единственный поддерживаемый тип ресурса данной функции.

После развертывания схемы элементов пользователи могут изменить размер загрузочного диска виртуальной машины. Также поддерживаются обычные диски. В этом случае клиентам необходимо только присоединить дисковый ресурс к ресурсу компьютера. Для всех развернутых компонентов можно использовать вариант «обновления загрузочного диска» или «обновления диска», чтобы увеличить (но не уменьшить) размер требуемого диска.

После развертывания схемы элементов пользователи могут изменить политику размера виртуальной машины с помощью параметра изменения размера конфигурации ресурсов vRealize Automation. После выбора виртуальная машина VMware Cloud Director будет использовать заданную политику размера.

Для этой функции требуется, чтобы **Пакет прав по умолчанию**, назначенный роли администратора организации, содержал право «Изменение политики вычислительных ресурсов», для которого внутренний код имеет значение `VAPP_EDIT_VM_COMPUTE_POLICY`. Это право должно быть активировано для администратора организации. В противном случае операция изменения размера завершится ошибкой 403: `Either you need some or all of the following rights [VAPP_EDIT_VM_COMPUTE_POLICY] to perform operations.`

Чтобы изменить размер загрузочного диска виртуальной машины VMware Cloud Director в рамках операции по регулярному обслуживанию, выберите виртуальную машину на странице «Развертывания». Однако перед изменением размера загрузочного диска необходимо отключить быструю подготовку. В противном случае может появиться следующее сообщение об ошибке.

```
Request timed out after 120 minutes. Please configure project request timeout parameter for long running resource requests.
```

Следует отметить, что это требование относится только к виртуальным машинам, созданным с помощью дисков с шаблоном vApp. Оно не применяется к виртуальным машинам, созданным с помощью файлов ISO.

Ниже описана процедура отключения быстрой подготовки.

1. Войдите в VMware Cloud Director как системный администратор: `https://vcd_url/поставщик` с именем системного пользователя
2. Щелкните «Виртуальные центры обработки данных организации».
3. Выберите целевую организацию.
4. Щелкните «Хранилище» (в разделе «Политики»).
5. Отключите **Быстрая подготовка**.

## Использование журналов и других ресурсов для устранения неполадок с учетными записями VMware Cloud Director в vRealize Automation

В случае возникновения проблем при настройке или использовании облачной учетной записи VMware Cloud Director в vRealize Automation можно обратиться к журналам и другим ресурсам, как описано ниже.

### Устранение неполадок подключения облачной учетной записи VMware Cloud Director

Если адаптер VMware Cloud Director отсутствует на экране создания облачной учетной записи или не отвечает, можно использовать следующую команду для проверки состояния, войдя на узел Kubernetes в vRealize Automation и проверив состояние модуля адаптера.

```
root@host [ ~ ]# kubectl -n prelude get pods | grep adapter-host-service-app
adapter-host-service-app-65f5c945bb-p6hpn      1/1      Running    0          4dlh
```

Если адаптер VMware Cloud Director не может обмениваться данными с физическим компьютером Cloud Director, на экране облачной учетной записи отображается ошибка со сведениями о подключении и обработке исключений. Ошибка также появляется в журналах.

## Работа с журналами VMware Cloud Director

Основной файл журнала адаптера VMware Cloud Director находится в локальном каталоге (модуль) `/var/log/adapter-host-service-app.log`, а если адаптер запущен внутри узла устройства vRealize Automation, этот журнал также копируется в каталог `/services-logs/prelude/adapter-host-service-app/file-logs/`. По умолчанию ведение большинства журналов ограничено уровнями `DEBUG` или `INFO`. Чтобы включить более подробное ведение журнала в целях отладки, можно изменить конфигурацию для следующих средств ведения журнала.

- `org.apache.cxf.services=INFO` — это средство ведения журнала обеспечивает подробные сведения для обмена данными между адаптером и VMware Cloud Director.
- `com.vmware.vra.vcloud.director.adapter=TRACE` — это средство ведения журнала обеспечивает подробные сведения для обмена данными между адаптером и vRealize Automation.

Существуют три способа доступа к журналам.

- Доступ к журналу посредством входа в модуль адаптера

```
root@host [ ~ ]# kubectl -n prelude exec -ti adapter-host-service-app-65f5c945bb-p6hpn --
bash
root [ / ]# less /var/log/adapter-host-service-app.log
```

- Доступ к журналу с помощью `kubectl`

```
root@host [ ~ ]# kubectl -n prelude get logs adapter-host-service-app-65f5c945bb-p6hpn
```

- Доступ к журналу с использованием локальной копии узла адаптера Kubernetes

```
root@host [ ~ ]# less /services-logs/prelude/adapter-host-service-app/file-logs/adapter-
host-service-app.log
```

Можно запросить или изменить конфигурацию средств ведения журнала с помощью конечной точки API-интерфейса REST `/actuator/loggers`.

- Пример включения трассировки обмена данными для клиента VMware Cloud Director с помощью `curl`:

```
curl -i -X POST -H 'Content-Type: application/json' -d '{"configuredLevel": "INFO"}'
http://{adapter-url}/actuator/loggers/org.apache.cxf.services
```

- Пример выключения трассировки обмена данными для клиента VMware Cloud Director с помощью `curl`:

```
curl -i -X POST -H 'Content-Type: application/json' -d '{"configuredLevel": "OFF"}'
http://{adapter-url}/actuator/loggers/org.apache.cxf.services
```

- Пример получения текущей конфигурации обмена данными для клиента VMware Cloud Director с помощью curl:

```
curl http://{adapter-url}/actuator/loggers/org.apache.cxf.services
...
{"configuredLevel":"OFF","effectiveLevel":"INFO"}
```

Существуют другие параметры, которые можно настроить, чтобы изменить поведение VMware Cloud Director.

- `vcd.max.thread.count` — этот параметр определяет максимальную степень параллельности при выполнении вызовов API-интерфейса VMware Cloud Director. Значение по умолчанию — 128.

---

**Примечание** Уменьшение значения этого параметра уменьшит нагрузку на северную часть VMware Cloud Director при выполнении подсчета, но может снизить производительность подсчета.

---

- `VCD_ADAPTER_PAGINATION_SIZE_IMAGES` — этот параметр определяет размер страницы при подсчете образов. Значение по умолчанию — 50.

---

**Примечание** Уменьшите значение этого параметра, если во время подсчета образов возникают ошибки времени ожидания адаптера.

---

## Интеграция vRealize Automation с другими приложениями

Интеграция позволяет добавлять внешние системы в vRealize Automation.

К интеграциям относятся vRealize Orchestrator, системы управления конфигурацией и другие внешние системы, такие как GitHub, Ansible, Puppet, и внешние поставщики IPAM, такие как Infoblox.

---

**Примечание** Если у вас нет внешнего интернет-доступа, но он требуется для интеграции, можно настроить прокси-сервер Интернета. См. раздел [Настройка прокси-сервера Интернета для vRealize Automation](#).

---

## Как использовать интеграцию Git в Cloud Assembly

Cloud Assembly поддерживает интеграцию с различными конфигурациями ресурсов репозитория Git, чтобы можно было управлять шаблонами VMware Cloud Templates и сценариями действий в системе управления версиями. Это помогает повысить прозрачность процессов в рамках развертывания и оптимизировать их аудит.

Cloud Assembly поддерживает различные конфигурации ресурсов интеграции Git, как указано в следующем списке. Каждый из этих вариантов представляет собой отдельную интеграцию.

- Облако GitHub, локальный GitHub Enterprise
- Облако GitLab, локальный GitLab Enterprise
- Локальный BitBucket

Для настройки интеграции Git со службой Cloud Assembly требуется наличие соответствующего локального репозитория Git, открытого для доступа всем назначенным пользователям. Кроме того, чтобы облачные шаблоны распознавались Git, они должны быть сохранены с использованием определенной структуры. Чтобы создать интеграцию с GitLab или GitHub, выберите в Cloud Assembly **Инфраструктура > Подключения > Интеграции**, а затем нужный вариант. Потребуется ввести URL-адрес и маркер целевого репозитория.

Если при настройке интеграции с Git используется существующий репозиторий, все облачные шаблоны, связанные с выбранными проектами, становятся доступными пользователям с соответствующими правами. Эти шаблоны можно использовать в существующих развертываниях или в качестве основы для нового развертывания. При добавлении проекта необходимо указать определенные свойства, касающиеся места и способа его хранения в Git.

Действия можно сохранить в репозиторий Git непосредственно из Cloud Assembly. Версиями сценариев действий можно либо управлять непосредственно в Git, либо создать версии в Cloud Assembly. Если в Cloud Assembly создается версия действия, она автоматически сохраняется в Git в качестве версии. Облачные шаблоны немного сложнее, так как их нельзя добавить непосредственно в интеграцию с Git из Cloud Assembly. Их необходимо сохранить в экземпляре Git, а затем их можно извлечь из Git при работе со страницей управления облачными шаблонами в Cloud Assembly.

## Перед началом работы

Чтобы платформа GitLab или GitHub могла распознать облачные шаблоны, они должны быть созданы и сохранены с использованием определенной структуры.

- Убедитесь, что все облачные шаблоны, которые планируется интегрировать с GitLab, настроены и сохранены требуемым образом. В GitLab импортируются только соответствующие требованиям шаблоны.
  - Создайте одну папку, назначенную для облачных шаблонов, или несколько.
  - Все облачные шаблоны должны храниться в файлах `blueprint.yaml`.
  - Убедитесь, что в верхней части шаблонов содержатся свойства `name:` и `version:.`
- Извлеките ключ API для соответствующего репозитория. Откройте учетную запись Git, щелкните имя пользователя в правом верхнем углу и перейдите в меню настроек. Нажмите кнопку **Маркеры доступа**, введите имя маркера и задайте срок действия. Затем выберите API и создайте маркер. Скопируйте и сохраните полученное значение.

Приведенные ниже рекомендации касаются всех облачных шаблонов, которые используются в рамках интеграции с Git.

- Каждый облачный шаблон должен находиться в отдельной папке.
- Все облачные шаблоны должны называться `blueprint.yaml`.
- Во всех файлах YAML облачных шаблонов должны присутствовать поля `name` и `version`.
- Импортируются только действительные облачные шаблоны.

- Если после обновления чернового облачного шаблона, который был импортирован из Git, его содержимое будет отличаться от содержимого самой последней версии, данный черновик не будет обновляться при последующих синхронизациях и будет создана новая версия. Если требуется обновить шаблон и обеспечить дальнейшую синхронизацию с Git, следует создать новую версию после внесения окончательных изменений.
- **Настройка интеграции облачного шаблона GitLab в Cloud Assembly**  
Эта процедура позволяет настроить интеграцию GitLab в Cloud Assembly, чтобы можно было работать с облачными шаблонами в репозитории и автоматически загружать сохраненные шаблоны, связанные с указанными проектами. Для доступа к возможностям GitLab при работе с облачными шаблонами необходимо настроить подключение к соответствующему экземпляру GitLab, а затем сохранить нужные шаблоны в этом экземпляре.
- **Настройка интеграции с GitHub в службе Cloud Assembly**  
Служба Cloud Assembly поддерживает интеграцию с облачной службой размещения репозитория GitHub.
- **Настройка интеграции Bitbucket в Cloud Assembly**  
Cloud Assembly поддерживает интеграцию с Bitbucket для использования в качестве репозитория на основе Git для сценариев действий ABX и шаблонов VMware Cloud Templates.

## Настройка интеграции облачного шаблона GitLab в Cloud Assembly

Эта процедура позволяет настроить интеграцию GitLab в Cloud Assembly, чтобы можно было работать с облачными шаблонами в репозитории и автоматически загружать сохраненные шаблоны, связанные с указанными проектами. Для доступа к возможностям GitLab при работе с облачными шаблонами необходимо настроить подключение к соответствующему экземпляру GitLab, а затем сохранить нужные шаблоны в этом экземпляре.

Если при настройке интеграции с GitLab используется существующий репозиторий, все облачные шаблоны, связанные с выбранными проектами, становятся доступными пользователям с соответствующими правами. Эти шаблоны можно использовать в существующих развертываниях или в качестве основы для нового развертывания. При добавлении проекта необходимо указать определенные свойства, касающиеся места и способа его хранения в GitLab.

---

**Примечание** Отправка новых или обновленных облачных шаблонов из службы Cloud Assembly в репозиторий Git не поддерживается. Отправка новых шаблонов из службы Cloud Assembly в репозиторий также не поддерживается. Для добавления облачных шаблонов в репозиторий разработчики должны использовать интерфейс Git.

---

Если после обновления чернового облачного шаблона, который был импортирован из Git, его содержимое будет отличаться от содержимого самой последней версии, данный черновик не будет обновляться при последующих синхронизациях и будет создана новая версия. Если требуется обновить облачный шаблон и обеспечить дальнейшую синхронизацию с Git, то следует создать новую версию после внесения окончательных изменений.

После настройки облачных шаблонов для использования с GitLab и сбора обязательных сведений следует настроить интеграцию с экземпляром GitLab. После этого можно импортировать в GitLab указанные облачные шаблоны. Видео ролик с демонстрацией этой процедуры можно посмотреть здесь: <https://www.youtube.com/watch?v=h0vqo63Sdgg>.

#### Необходимые условия

- Извлеките ключ API для соответствующего репозитория. Откройте учетную запись GitLab, щелкните имя пользователя в правом верхнем углу и перейдите в меню настроек. Нажмите кнопку «Маркеры доступа», введите имя маркера и задайте срок действия. Затем выберите API и создайте маркер. Скопируйте и сохраните полученное значение.

Для настройки интеграции Git со службой Cloud Assembly требуется наличие соответствующего локального репозитория Git, открытого для доступа всем назначенным пользователям. Кроме того, чтобы платформа GitLab могла распознать облачные шаблоны, они должны быть созданы и сохранены с использованием определенной структуры.

- Убедитесь, что все облачные шаблоны, которые планируется интегрировать с GitLab, настроены и сохранены требуемым образом. В GitLab импортируются только соответствующие требованиям шаблоны. См. раздел [Как использовать интеграцию Git в Cloud Assembly](#).

#### Процедура

1. Настройте интеграцию со средой GitLab в службе Cloud Assembly.
  - а) Выберите **Инфраструктура > Интеграции > Добавить новый элемент**, а затем GitLab.
  - б) Введите **URL-адрес** для экземпляра GitLab. Для экземпляров GitLab в формате «программное обеспечение как услуга» данный адрес в большинстве случаев будет иметь вид gitlab.com.
  - в) В поле **Маркер** введите маркер, также называемый ключом API-интерфейса, для заданного экземпляра GitLab. Дополнительные сведения о получении маркера из экземпляра GitLab см. в приведенных выше требованиях.
  - г) Введите подходящее имя и описание.
  - д) Нажмите **Проверить**, чтобы проверить подключение.
  - е) При необходимости добавьте теги возможностей. Дополнительные сведения см. в разделе [Использование тегов возможностей в Cloud Assembly](#).
  - ж) Нажмите кнопку **Добавить**.
2. Настройте подключение GitLab, чтобы система принимала облачные шаблоны в соответствующем репозитории.
  - а) Выберите **Инфраструктура > Интеграции**, а затем соответствующую интеграцию GitLab.
  - б) Нажмите **Проекты**.
  - в) Нажмите **Создать проект** и введите имя проекта.
  - г) В поле **Репозиторий** укажите путь к репозиторию в системе GitLab. Как правило, путь к репозиторию состоит из имени пользователя основной учетной записи и имени репозитория.

- д) В поле **Ветвь** укажите нужную ветвь GitLab, которую требуется использовать.
- е) При необходимости введите имя **папки**. Если оставить это поле пустым, будут доступны все папки.
- ж) Введите соответствующий тип в поле **Тип**. При необходимости введите имя папки. Если оставить это поле пустым, будут доступны все папки.
- з) Нажмите **Далее**, чтобы завершить процесс добавления репозитория.

При нажатии кнопки **Далее** запускается автоматическая синхронизация, которая импортирует облачные шаблоны на платформу.

После выполнения задач синхронизации появляется сообщение о том, что облачные шаблоны импортированы.

## Результаты

Теперь облачные шаблоны можно извлекать из GitLab.

## Настройка интеграции с GitHub в службе Cloud Assembly

Служба Cloud Assembly поддерживает интеграцию с облачной службой размещения репозитория GitHub.

Для настройки интеграции с GitHub в службе Cloud Assembly потребуется действующий маркер GitHub. Сведения о создании и поиске маркеров см. в документации по GitHub.

### Необходимые условия

- Для этого потребуется доступ к GitHub.
- Убедитесь, что все облачные шаблоны, которые планируется интегрировать с GitHub, настроены и сохранены требуемым образом. В GitHub импортируются только соответствующие требованиям облачные шаблоны. См. раздел [Как использовать интеграцию Git в Cloud Assembly](#).

### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите GitHub.
3. Введите необходимую информацию на странице настройки GitHub.
4. Нажмите **Проверить**, чтобы проверить интеграцию.
5. Введите теги возможностей, если этого требуют правила установки тегов. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).
6. Нажмите кнопку **Добавить**.



7. Настройте подключение GitHub, чтобы система принимала облачные шаблоны в соответствующем репозитории.

- а) Выберите **Инфраструктура > Интеграции**, а затем выберите соответствующую интеграцию GitHub.
- б) Нажмите **Проекты**.
- в) Нажмите **Создать проект** и введите имя проекта.
- г) В поле **Репозиторий** укажите путь к репозиторию в системе GitHub. Как правило, путь к репозиторию состоит из имени пользователя основной учетной записи и имени репозитория.
- д) В поле **Ветвь** укажите нужную ветвь GitHub, которую требуется использовать.
- е) При необходимости введите имя **папки**. Если оставить это поле пустым, будут доступны все папки.
- ж) Введите соответствующий тип в поле **Тип**.
- з) Нажмите **Далее**, чтобы завершить процесс добавления репозитория.

Запускается задача автоматической синхронизации, которая импортирует облачные шаблоны на платформу.

После выполнения задач синхронизации появляется сообщение о том, что облачные шаблоны импортированы.

#### Результаты

Служба GitHub доступна для использования при работе со схемами элементов Cloud Assembly.

#### Следующие шаги

Теперь облачные шаблоны можно извлекать из GitHub.

## Настройка интеграции Bitbucket в Cloud Assembly

Cloud Assembly поддерживает интеграцию с Bitbucket для использования в качестве репозитория на основе Git для сценариев действий ABX и шаблонов VMware Cloud Templates.

В Cloud Assembly с помощью интеграции Bitbucket можно работать с двумя типами элементов репозитория: шаблонами VMware Cloud Templates и сценариями действий ABX. Перед использованием интеграции Bitbucket требуется синхронизировать проекты, с которыми необходимо работать. Действия ABX поддерживают обратную запись в репозиторий Bitbucket, но записывать обратно облачные шаблоны из интеграции нельзя. Если необходимо создать новые версии файлов облачных шаблонов, это нужно сделать вручную.

#### Необходимые условия

- Настройте локальное развертывание сервера Bitbucket с помощью одного или нескольких проектов на основе ABX или облачных шаблонов, которые необходимо использовать с развертываниями. В настоящее время облачная среда Bitbucket не поддерживается.
- Создайте или выберите проект Cloud Assembly, чтобы привязать интеграцию Bitbucket.

- Файлы облачного шаблона, которые нужно синхронизировать с интеграцией Bitbucket, должны называться `blueprint.yaml`.

## Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите Bitbucket.
3. Введите сводную информацию и учетные данные Bitbucket на странице «Сводка по новой интеграции Bitbucket».
4. Для проверки интеграции нажмите **Проверить**.
5. Если для политики тегирования необходимо добавить теги, введите теги возможностей. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).
6. Нажмите кнопку **Добавить**.
7. Перейдите на вкладку «Проекты» на главной странице интеграции Bitbucket, чтобы связать проект с этой интеграцией Bitbucket.
8. Выберите проект, который нужно связать с интеграцией Bitbucket.
9. Нажмите кнопку **Далее**, чтобы добавить репозиторий в проект Bitbucket и указать тип добавляемого репозитория, а затем укажите имя **репозитория**, **ветвь** и **папку**.
10. Нажмите кнопку **Добавить**.

Если в проект необходимо добавить один или несколько репозитория, щелкните **Добавить репозиторий**.

## Результаты

Интеграция Bitbucket настраивается с указанной конфигурацией репозитория; можно просматривать и использовать действия ABX и облачные шаблоны, которые содержатся в настроенных репозиториях. При добавлении проекта в интеграцию Bitbucket выполняется операция синхронизации, позволяющая получить последние версии сценариев действий ABX и файлов облачных шаблонов из назначенного репозитория. На вкладке «Журнал» на странице «Интеграция Bitbucket» отображаются записи обо всех операциях синхронизации для интеграции. По умолчанию файлы автоматически синхронизируются каждые 15 минут, но файл можно синхронизировать вручную: для этого выберите его и нажмите **СИНХРОНИЗИРОВАТЬ**.

## Следующие шаги

На странице «Расширяемость» Cloud Assembly можно работать с действиями ABX, а на странице «Проект» — с облачными шаблонами. При сохранении измененной версии действия ABX в области «Расширяемость» в Cloud Assembly создается новая версия сценария, которая записывается обратно в репозиторий.

## Настройка интеграции внешнего поставщика IPAM в vRealize Automation

Для управления IP-адресами, используемыми в развертываниях облачных шаблонов, можно создать точку интеграции внешнего IPAM для определенного поставщика. При использовании точки интеграции внешнего поставщика IPAM IP-адреса поступают от указанного поставщика и управляются в его среде, а не в vRealize Automation.

Чтобы управлять IP-адресами и параметрами DNS для развертываний облачных шаблонов и виртуальных машин в vRealize Automation, можно создать точку интеграции IPAM для конкретного поставщика.

Сведения о настройке необходимых условий, а также пример создания точки интеграции определенного внешнего поставщика IPAM в рамках типового рабочего процесса см. в разделе [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#). Следует отметить, что этот рабочий процесс предназначен для интеграции с Infoblox IPAM, но его можно использовать в качестве образца для любого внешнего поставщика IPAM.

Дополнительные сведения о создании необходимых активов, позволяющих внешним партнерам и поставщикам IPAM интегрировать свое решение IPAM с vRealize Automation, см. в разделе [Использование IPAM SDK для создания внешнего пакета интеграции IPAM для определенного поставщика для vRealize Automation](#).

### Необходимые условия

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что у вас есть учетная запись с внешним поставщиком IPAM, например [Infoblox](#) или [Bluecat](#), и корректные учетные данные для доступа к учетной записи организации с помощью поставщика IPAM.
- Убедитесь, что у вас есть доступ к развернутому пакету интеграции для поставщика IPAM, например Infoblox или BlueCat. Этот пакет сначала получают путем загрузки ZIP-файла с сайта поставщика IPAM или с [VMware Marketplace](#), а затем развертывают в vRealize Automation.
- Убедитесь, что у вас есть доступ к настроенной и запущенной среде для поставщика IPAM.
- Если вы используете локальную встроенную запущенную среду расширяемости на основе действий (ABX), убедитесь, что в сети vRealize Automation имеется прокси-сервер HTTP, способный передавать исходящий трафик на внешние сайты, такие как gcr.io и storage.googleapis.com. Дополнительные сведения см. в статье [Получение образов Docker из среды, защищенной прокси-сервером, в vRealize Automation 8.x \(75180\)](#).
- Убедитесь, что у вас есть необходимые учетные данные пользователя для продукта поставщика IPAM. Дополнительные сведения о необходимых разрешениях пользователей см. в документации по продуктам поставщика интеграции.

## Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Щелкните **IPAM**.
3. В раскрывающемся списке **Поставщик** выберите настроенный пакет поставщика IPAM в списке.

Если список пуст, нажмите **Импортировать пакет поставщика**, найдите существующий ZIP-файл пакета поставщика и выберите его. Если у вас нет ZIP-файла, его можно получить на [VMware Marketplace](#).

4. Введите имя пользователя и пароль администратора для своей учетной записи с внешним поставщиком IPAM, а также заполните все остальные обязательные поля (при наличии), например имя узла поставщика.
5. В раскрывающемся списке **Запущенная среда** выберите существующую запущенную среду, например локальную точку интеграции расширяемости на основе действий.

Запущенная среда поддерживает связь между vRealize Automation и поставщиком IPAM.

Платформа IPAM поддерживает только локальную встроенную запущенную среду расширяемости на основе действий (ABX).

---

**Примечание** Если в качестве запущенной среды интеграции используется облачная учетная запись Amazon Web Services или Microsoft Azure, убедитесь, что устройство поставщика IPAM доступно из Интернета и не находится под защитой NAT или брандмауэра и что оно имеет общедоступное разрешимое DNS-имя. Если поставщик IPAM недоступен, функции Amazon Web Services Lambda или Microsoft Azure не смогут подключиться к нему и интеграция завершится ошибкой.

---

6. Щелкните **Проверить**.
7. При появлении запроса на доверие самозаверяющего сертификата от внешнего поставщика IPAM нажмите кнопку **Принять**.
8. Введите имя для этой точки интеграции с IPAM и нажмите **Добавить**, чтобы сохранить новую точку интеграции IPAM.

Выполняется имитация действия сбора данных. Данные о сетях и IP-адресах поступают от внешнего поставщика IPAM.

## Обновление до новой версии пакета интеграции внешнего поставщика IPAM в vRealize Automation

Можно обновить существующую точку интеграции внешнего поставщика IPAM, чтобы получить новую версию пакета интеграции IPAM для конкретного поставщика.

Внешний поставщик IPAM или VMware могут обновить исходный пакет интеграции IPAM для конкретного поставщика. Например, пакет интеграции внешнего IPAM Infoblox был обновлен несколько раз. Для сохранения существующих параметров инфраструктуры vRealize Automation, в которых используется именованная точка интеграции IPAM, можно изменить эту точку интеграции, чтобы получить обновленный пакет интеграции IPAM, не создавая новую точку интеграции IPAM.

### Необходимые условия

В этой процедуре предполагается, что вы уже создали точку интеграции внешнего поставщика IPAM и хотите ее обновить, чтобы использовать новую версию пакета интеграции IPAM поставщика.

Сведения о процедуре создания внешней точки интеграции IPAM см. в разделе [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#).

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- У вас должна быть учетная запись в системе внешнего поставщика IPAM, а также правильные учетные данные для доступа к учетной записи организации в системе IPAM.
- Убедитесь, что у вас есть доступ к развернутому пакету интеграции для поставщика IPAM. Этот пакет сначала получают путем загрузки ZIP-файла с сайта поставщика IPAM или с [Vmware Marketplace](#), а затем развертывают в vRealize Automation.

Дополнительные сведения о том, как загрузить и развернуть ZIP-файл пакета поставщика и сделать его доступным в виде значения **Поставщик** на странице «Интеграция IPAM», см. в разделе [Загрузка и развертывание пакета внешнего поставщика IPAM для использования в vRealize Automation](#).

- Убедитесь, что у вас есть доступ к настроенной и запущенной среде для поставщика IPAM. Как правило, запущенная среда является локальной встроенной точкой интеграции средств расширяемости на основе действий (ABX).

Сведения о характеристиках запущенной среды см. в разделе [Создание запущенной среды для точки интеграции IPAM в vRealize Automation](#).

### Процедура

1. Выберите **Инфраструктура > Подключения > Интеграции IPAM** и откройте существующую точку интеграции IPAM.
2. Щелкните **Управление поставщиками**.
3. Выберите обновленный пакет интеграции IPAM и импортируйте его.
4. Нажмите **Проверить**, а затем **Сохранить**.

## Настройка интеграции с приложением My VMware в службе Cloud Assembly

Можно интегрировать My VMware с Cloud Assembly для поддержки действий и возможностей, связанных с VMware и загружаемыми компонентами, для которых требуется учетная запись.

Для каждой организации можно создать только одну интеграцию My VMware.

### Необходимые условия

У вас должна быть учетная запись пользователя с соответствующими разрешениями для My VMware.

- Дополнительные сведения о приглашении пользователя в учетную запись My VMware см. в статье [базы знаний 2070555](#).
- Сведения о назначении разрешений пользователей в учетной записи My VMware см. в статье [базы знаний 2006977](#).

### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите My VMware.
3. Введите необходимую информацию на странице настройки My VMware.
4. Если для политики тегирования требуются теги, введите теги возможностей. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).
5. Нажмите кнопку **Добавить**.

### Результаты

Приложение My VMware доступно для использования.

### Следующие шаги

При необходимости можно осуществлять доступ к компонентам My VMware.

## Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly

Для использования рабочих процессов в рамках расширяемости и облачных шаблонов можно настроить одну или несколько интеграций vRealize Orchestrator.

vRealize Automation содержит предварительно настроенный экземпляр vRealize Orchestrator. Доступ к клиенту встроенного экземпляра vRealize Orchestrator можно получить в консоли облачных служб vRealize Automation.

---

**Примечание** Чтобы получить доступ к центру управления встроенного экземпляра vRealize Orchestrator, перейдите по ссылке [https://полное\\_доменное\\_имя\\_vRA/vco-controlcenter](https://полное_доменное_имя_vRA/vco-controlcenter) и войдите в систему как пользователь **root**.

---

Кроме того, можно интегрировать внешний экземпляр vRealize Orchestrator для использования в подписках на действия расширяемости vRealize Automation и в операциях со службой «Любой ресурс как услуга», применяемых для облачных шаблонов.

#### Необходимые условия

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Выполните обновление или перенос в vRealize Orchestrator 8.3. См. раздел *Обновление и перенос VMware vRealize Orchestrator*.

#### Процедура

1. Выберите **Инфраструктура > Подключения > Интеграции**.
2. Щелкните **Добавить интеграцию**.
3. Выберите **vRealize Orchestrator**.
4. Введите имя интеграции vRealize Orchestrator.
5. (дополнительно) Введите описание интеграции vRealize Orchestrator.
6. В поле **URL-адрес vRealize Orchestrator** введите полное доменное имя внешнего экземпляра vRealize Orchestrator.  
  
Например, `https://my_vRO_FQDN.com:443`.
7. Для проверки интеграции нажмите **Проверить**.
8. (дополнительно) При появлении запроса проверьте сведения о сертификате и нажмите кнопку **Принять**.
9. (дополнительно) Добавьте теги возможностей. Дополнительные сведения о тегах возможностей см. в разделе [Использование тегов возможностей в Cloud Assembly](#).

---

**Примечание** Теги возможностей можно использовать для управления несколькими интеграциями vRealize Orchestrator. См. раздел [Управление несколькими интеграциями vRealize Orchestrator с проектными ограничениями](#).

---

10. Нажмите кнопку **Добавить**.  
  
Интеграция vRealize Orchestrator сохранена.
11. Чтобы убедиться, что интеграция настроена и рабочие процессы добавлены, выберите пункты **Расширяемость > Библиотека > Рабочие процессы**.

#### Следующие шаги

Для доступа к интегрированному внешнему клиенту vRealize Orchestrator выполните следующие действия.

1. Перейдите в консоль облачных служб vRealize Automation.
2. Выберите **Orchestrator**.

3. Выберите вкладку, которая соответствует интегрированному экземпляру vRealize Orchestrator.

---

**Примечание** Для пользователей Cloud Assembly, не имеющих учетных данных администратора облачных служб, не отображается вкладка интегрированного экземпляра vRealize Orchestrator.

---

## Отключение и включение интеграций vRealize Orchestrator

Можно вручную отключить или включить интеграцию vRealize Orchestrator, чтобы выполнять операции обслуживания, когда интеграция продолжает функционировать.

Для выполнения обслуживания интеграцию vRealize Orchestrator можно отключить. В случае отключения интеграция vRealize Orchestrator по-прежнему находится в состоянии **ВЫПОЛНЯЕТСЯ**, поэтому такие задачи, как мониторинг ресурсов и сбор данных, можно продолжать выполнять.

---

**Примечание** В дополнение к отключению вручную служба vRealize Orchestrator Gateway периодически проверяет состояние работоспособности, чтобы определить, активны интеграции vRealize Orchestrator или нет. Все неактивные интеграции vRealize Orchestrator отключаются автоматически и переводятся в состояние **ОТКЛЮЧЕНО**. В отключенных интеграциях такие задачи, как сбор данных или мониторинг ресурсов, выполнять будет нельзя.

---

После деактивации интеграции vRealize Orchestrator или ее отключения с помощью средства проверки состояния работоспособности рабочие процессы будут выполняться только в остальных интеграциях, которые включены. Если среда содержит несколько включенных интеграций vRealize Orchestrator, для управления которыми теги ограничений или возможностей проекта не используются, для выполнения рабочего процесса будет выбрана интеграция vRealize Orchestrator в случайном порядке.

---

**Примечание** Так как выбор интеграции vRealize Orchestrator выполняется случайным образом, убедитесь, что информация, необходимая для выполнения данной операции, доступна во всех интеграциях. Для сущностей содержимого, например рабочих процессов, это означает, что они должны быть синхронизированы во всех интеграциях. Для объектов иерархии не гарантируется, что они будут иметь один и тот же идентификатор объекта во всех интеграциях. Поэтому попытка запуска рабочего процесса, который включает в себя такой объект иерархии в качестве входного параметра, может завершиться сбоем.

---

Сведения об управлении несколькими интеграциями vRealize Orchestrator с помощью тегов ограничений и возможностей проекта см. в [Управление несколькими интеграциями vRealize Orchestrator с проектными ограничениями](#) и [Управление несколькими интеграциями vRealize Orchestrator с тегами возможностей облачной учетной записи](#).

### Необходимые условия

Настройте одну интеграцию vRealize Orchestrator или несколько в Cloud Assembly. См. раздел [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).

### Процедура

1. Отключите интеграцию vRealize Orchestrator.
  - а) Перейдите в раздел **Инфраструктура > Подключения > Интеграции**.
  - б) Выберите интеграцию vRealize Orchestrator, которую необходимо отключить.



- в) В разделе **Учетные данные сервера vRealize Orchestrator** отключите параметр **Включить конечную точку**.
  - г) Щелкните **Проверить**.
  - д) После успешной проверки нажмите кнопку **Сохранить**.
2. Выполните необходимые задачи по обслуживанию для отключенной интеграции vRealize Orchestrator.
  3. Включите интеграцию vRealize Orchestrator.
    - а) Перейдите в раздел **Инфраструктура > Подключения > Интеграции**.
    - б) Выберите ранее отключенную интеграцию vRealize Orchestrator.
    - в) В разделе **Учетные данные сервера vRealize Orchestrator** включите параметр **Включить конечную точку**.
    - г) Щелкните **Проверить**.
    - д) После успешной проверки нажмите кнопку **Сохранить**.

## Управление несколькими интеграциями vRealize Orchestrator с проектными ограничениями

С помощью ограничений проекта можно управлять тем, какие интеграции vRealize Orchestrator используются в подписках на рабочие процессы.

Cloud Assembly поддерживает интеграцию нескольких серверов vRealize Orchestrator, которые можно использовать в подписках на рабочие процессы. Благодаря мягким или жестким ограничениям проекта можно управлять тем, какие интеграции vRealize Orchestrator используются в облачных шаблонах, подготовленных в рамках проекта. Дополнительные сведения об ограничениях проекта см. в разделе [Использование тегов проекта и настраиваемых свойств в Cloud Assembly](#).

### Необходимые условия

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Настройте две или более интеграций vRealize Orchestrator в Cloud Assembly. См. раздел [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).
- Добавьте теги возможностей в интеграции vRealize Orchestrator. См. раздел [Использование тегов возможностей в Cloud Assembly](#).

### Процедура

1. Выберите **Инфраструктура > Администрирование > Проекты**, а затем проект.
2. Перейдите на вкладку **Предоставление**.
3. Введите теги возможностей для интеграций vRealize Orchestrator в текстовом поле **Ограничения расширяемости** и настройте их как мягкие или жесткие ограничения проекта.
4. Нажмите **Сохранить**.

## Результаты

При развертывании облачного шаблона Cloud Assembly использует ограничения проекта для управления тем, какие интеграции vRealize Orchestrator используются в подписках на рабочие процессы.

## Следующие шаги

Кроме того, с помощью тегов возможностей можно управлять несколькими интеграциями vRealize Orchestrator на уровне облачной учетной записи. Дополнительные сведения см. в разделе [Управление несколькими интеграциями vRealize Orchestrator с тегами возможностей облачной учетной записи](#).

## Управление несколькими интеграциями vRealize Orchestrator с тегами возможностей облачной учетной записи

С помощью тегов возможностей можно управлять тем, какие интеграции vRealize Orchestrator используются в подписках на рабочие процессы.

Cloud Assembly поддерживает интеграцию нескольких серверов vRealize Orchestrator, которые можно использовать в подписках на рабочие процессы. Благодаря добавлению тегов возможностей в облачную учетную запись можно управлять тем, какие интеграции vRealize Orchestrator используются в подписках на рабочие процессы.

## Необходимые условия

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Настройте две или более интеграций vRealize Orchestrator в Cloud Assembly. Дополнительные сведения см. в разделе [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).
- Добавьте теги возможностей в интеграции vRealize Orchestrator. См. раздел [Использование тегов возможностей в Cloud Assembly](#).

## Процедура

1. Выберите **Инфраструктура > Подключения > Облачные учетные записи**.
2. Выберите облачную учетную запись.
3. Введите теги возможностей интеграций vRealize Orchestrator, которые необходимо использовать.  
Теги возможностей автоматически преобразуются в мягкие ограничения. Чтобы использовать жесткие ограничения при управлении интеграциями, необходимо применить ограничения для проекта. Дополнительные сведения см. в разделе [Управление несколькими интеграциями vRealize Orchestrator с проектными ограничениями](#).
4. Нажмите **Сохранить**.

## Результаты

При развертывании облачного шаблона Cloud Assembly использует теги в связанной облачной учетной записи, чтобы управлять тем, какие интеграции vRealize Orchestrator используются в подписках на рабочие процессы.

## Сбор данных для интеграций vRealize Orchestrator

vRealize Automation Выполняет периодический сбор данных для интеграций vRealize Orchestrator.

Сбор данных для интеграций vRealize Orchestrator запускается каждые 10 минут. В ходе данной операции выполняется сбор данных о рабочих процессах, включенных в библиотеку каждой интеграции vRealize Orchestrator.

---

**Важно!** По завершении редактирования убедитесь, что вы назначили версию рабочему процессу. Сборщик данных не фиксирует изменения рабочих процессов, для которых не указана версия.

---

Чтобы получить сведения о последнем сборе данных, выполненном для интеграции vRealize Orchestrator, выберите пункт **Инфраструктура > Подключения > Интеграции**, а затем выберите нужную интеграцию. Для запуска сбора данных вручную нажмите **Запуск сбора данных**.

Дополнительные сведения о сборе данных vRealize Automation см. в разделе [Как сбор данных работает в vRealize Automation](#).

## Работа с Kubernetes в Cloud Assembly

Cloud Assembly обеспечивает несколько вариантов настройки и развертывания виртуальных нагрузок Kubernetes, а также управления ими.

В Cloud Assembly предлагается два варианта работы с ресурсами Tanzu Kubernetes. Можно создать конфигурацию vSphere with Tanzu Kubernetes, при которой для доступа к встроенным возможностям vSphere Tanzu Kubernetes требуются только подходящая облачная учетная запись vCenter и план кластера. В этом варианте можно использовать облачную учетную запись vCenter для доступа к пространствам имен управляющей программы с целью развертывания рабочих нагрузок vSphere на основе Kubernetes. Кроме того, можно интегрировать внешние ресурсы Kubernetes в Cloud Assembly.

Кроме того, можно интегрировать VMware Tanzu Kubernetes Grid Integrated Edition (TKGI) (ранее — PKS). Для этого типа реализации Kubernetes требуется интеграция PKS в Cloud Assembly. Для этого не требуется план кластера Cloud Assembly.

Кроме того, для настройки и развертывания ресурсов Kubernetes, а также управления ими можно создать интеграцию Red Hat OpenShift с Cloud Assembly.

## Работа с кластерами vSphere with Tanzu Kubernetes

vSphere 7.x содержит значительные улучшения, позволяющие работать с Kubernetes для управления виртуальными машинами и контейнерами из одного интерфейса. Cloud Assembly позволяет пользователям использовать встроенные в vSphere возможности vSphere with Tanzu Kubernetes. Доступ к возможностям vSphere with Tanzu Kubernetes можно получить с помощью облачной учетной записи vCenter с реализацией vSphere, содержащей кластеры управляющей программы. Такая реализация позволяет управлять традиционными виртуальными машинами и кластерами Kubernetes из vCenter.

Для пространств имен управляющей программы Tanzu Kubernetes пользователи должны иметь доступ к применимой службе единого входа vSphere, чтобы выполнять вход для просмотра сведений о пространстве имен управляющей программы по предоставленной ссылке. Затем они могут загрузить настроенный инструмент Kubectl с проверкой подлинности vSphere, чтобы использовать пространство имен управляющей программы.

Чтобы использовать эту функцию, требуется vCenter с облачной учетной записью vSphere, для которой настроены пространства имен управляющей программы. После входа в систему пользователи могут начать работу с соответствующими пространствами имен.

## Работа с интеграциями VMware Tanzu Kubernetes Grid Integrated Edition (TKGI) или OpenShift

Для конфигураций TKGI, внешних кластеров и OpenShift в Cloud Assembly предоставляется доступ к файлу Kubeconfig, который обеспечивает пользователям доступ к применимым кластерам Kubernetes.

После создания интеграции TKGI или OpenShift применимые кластеры Kubernetes становятся доступны в Cloud Assembly. В Cloud Assembly можно добавлять и создавать компоненты Kubernetes для управления приложениями кластеров и контейнеров. Эти приложения формируют основу развертываний по принципу самообслуживания, доступных в каталоге Service Broker.

- [Настройка интеграции VMware Tanzu Kubernetes Grid Integrated Edition в Cloud Assembly](#)

Подключение к ресурсам VMware Tanzu Kubernetes Grid Integrated Edition (TKGI, предыдущее название — PKS) можно настроить локально и в облаке, чтобы обеспечить поддержку возможностей интеграции и управления Kubernetes в Cloud Assembly.

- [Подготовка развертывания vSphere with Tanzu Kubernetes в vRealize Automation](#)

vRealize Automation позволяет подготовить развертывание vSphere with Tanzu Kubernetes из Cloud Assembly, чтобы использовать собственные возможности vSphere 7.x для развертывания кластеров Tanzu Kubernetes и управления ими. Это обеспечивает независимый от инфраструктуры уровень подготовки виртуальной инфраструктуры и управления ей.

- [Настройка интеграции Red Hat OpenShift в Cloud Assembly](#)

Локальное подключение ресурсов Red Hat OpenShift можно настроить на локальном уровне и в облаке, чтобы обеспечить поддержку возможностей интеграции и управления Kubernetes корпоративного уровня в Cloud Assembly.

## ■ [Настройка зоны Kubernetes в Cloud Assembly](#)

Зоны Kubernetes позволяют администраторам облачных систем определять размещение кластеров Kubernetes и пространств имен, а также пространств имен управляющей программы, используемых в развертываниях Cloud Assembly, на основе политик. На этой странице администратор может указать кластеры, доступные для предоставления пространств имен Kubernetes, и свойства, применимые для кластеров.

## ■ [Создание плана кластера в vRealize Automation Cloud Assembly для использования с развертыванием vSphere with Tanzu Kubernetes](#)

Необходимо создать план кластера для использования с развертываниями vSphere with Tanzu Kubernetes в vRealize Automation. План кластера функционирует как шаблон конфигурации для подготовки экземпляров кластера Tanzu Kubernetes в определенном экземпляре облачной учетной записи vSphere.

## ■ [Использование управляющих кластеров и пространств имен Tanzu в Cloud Assembly](#)

Администраторы могут сделать пространства имен управляющей программы в интеграции vSphere с поддержкой Tanzu доступными для пользователей, чтобы те могли добавлять эти пространства имен в развертывания Kubernetes с помощью облачных шаблонов или запрашивать их из каталога Service Broker.

## ■ [Работа с кластерами и пространствами имен Kubernetes в Cloud Assembly](#)

Администраторы облачных систем могут добавлять и просматривать конфигурацию развернутых кластеров Kubernetes и пространств имен, как универсальных, так и основанных на Pacific, а также управлять ими в Cloud Assembly.

## ■ [Добавление компонентов Kubernetes в облачные шаблоны в Cloud Assembly](#)

При добавлении компонентов Kubernetes в облачный шаблон Cloud Assembly можно добавить кластеры или разрешить пользователям создавать пространства имен в различных конфигурациях. Как правило, выбор операции зависит от требований по контролю доступа, от способа настройки компонентов Kubernetes и требований к развертыванию.

## ■ [Использование расширяемости Cloud Assembly с помощью Kubernetes](#)

Cloud Assembly предоставляет набор тем событий, которые соответствуют типовым действиям, связанным с развертыванием кластера Kubernetes и пространства имен. При необходимости пользователи могут подписаться на эти темы, и они будут запускаться в соответствующее время. Пользователи получают уведомление, когда произойдет событие, связанное с темой подписки. Кроме того, можно настроить запуск рабочих процессов vRO на основе уведомлений о событиях.

## Настройка интеграции VMware Tanzu Kubernetes Grid Integrated Edition в Cloud Assembly

Подключение к ресурсам VMware Tanzu Kubernetes Grid Integrated Edition (TKGI, предыдущее название — PKS) можно настроить локально и в облаке, чтобы обеспечить поддержку возможностей интеграции и управления Kubernetes в Cloud Assembly.

Интеграции TKGI позволяют управлять экземплярами TKGI локально, а также в облачной среде и кластерах Kubernetes, которые подготовлены в TKGI и во внешних кластерах. Для поддержки размещения ресурсов на основе политик необходимо создать профиль Kubernetes и связать его с проектом.

#### Необходимые условия

- Требуется правильно настроенный сервер TKGI со службой проверки подлинности UAA.
- Убедитесь в наличии учетных данных администратора облака. Дополнительные сведения см. в разделе [Что такое роли пользователей vRealize Automation](#).

#### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите VMware Tanzu Kubernetes Grid Integrated Edition.
3. Введите IP-адрес или полное доменное имя и адрес TKGI для создаваемой облачной учетной записи TKGI.
  - IP-адрес — это полное доменное имя или IP-адрес сервера проверки подлинности пользователей TKGI.
  - Адрес TKGI — это полное доменное имя или IP-адрес основного сервера TKGI.
4. Укажите, является ли этот сервер TKGI локальным или находится в общедоступном или частном облаке.
5. Введите соответствующее **имя пользователя** и **пароль** для сервера TKGI и другую необходимую информацию.
6. Если для политики тегирования используются теги, введите теги возможностей. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).
7. Нажмите кнопку **Добавить**.

#### Результаты

Можно создать зоны Kubernetes и назначить их проекту либо найти и назначать внешние кластеры Kubernetes. Кроме того, можно добавить или создать пространства имен Kubernetes, которые упрощают управление кластерами в крупных группах и организациях.

#### Следующие шаги

Создайте или выберите соответствующие зоны Kubernetes, затем укажите один или несколько кластеров либо пространств имен и назначьте их проекту. После этого можно создать и опубликовать облачные шаблоны, чтобы пользователи могли создавать развертывания, использующие Kubernetes, по принципу самообслуживания.

## Подготовка развертывания vSphere with Tanzu Kubernetes в vRealize Automation

vRealize Automation позволяет подготовить развертывание vSphere with Tanzu Kubernetes из Cloud Assembly, чтобы использовать собственные возможности vSphere 7.x для развертывания кластеров Tanzu Kubernetes и управления ими. Это обеспечивает независимый от инфраструктуры уровень подготовки виртуальной инфраструктуры и управления ей.

В основе функциональных возможностей vSphere with Tanzu Kubernetes лежат встроенные возможности Kubernetes в vSphere 7.x. Для работы не требуется интеграция vRealize Automation PKS.

### Необходимые условия

- Чтобы подготовить развертывание vSphere with Tanzu Kubernetes из Cloud Assembly, необходимо иметь доступ к vSphere 7.x. В vRealize Automation доступ к vSphere предоставляется в рамках облачной учетной записи vCenter в Cloud Assembly. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).
- Служба Tanzu должна быть включена в облачной учетной записи vSphere и должна содержать соответствующие пространства имен управляющей программы.
- Необходимо иметь соответствующий план кластера для использования с интеграцией. См. раздел [Создание плана кластера в vRealize Automation Cloud Assembly для использования с развертыванием vSphere with Tanzu Kubernetes](#).

### Процедура

1. Если в Cloud Assembly еще нет подходящей облачной учетной записи vCenter, создайте ее. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).
2. Выберите **Инфраструктура > Настроить > Зона Kubernetes** в vRealize Automation Cloud Assembly, чтобы создать или выбрать зону Kubernetes.  
  
Можно использовать существующую зону Kubernetes, если подходящая зона уже настроена, однако администратор должен добавить в нее одно или несколько пространств имен управляющей программы. Эти пространства имен служат в качестве вычислительных ресурсов, в которых в рамках зоны создаются подготовленные кластеры Tanzu Kubernetes. Дополнительные сведения о зонах Kubernetes см. в разделе [Настройка зоны Kubernetes в Cloud Assembly](#).
3. Перейдите на вкладку «Подготовка Kubernetes» на странице **Инфраструктура > Администрирование > Проекты** в Cloud Assembly и свяжите зону Kubernetes с соответствующим проектом.
4. Создайте или выберите план кластера для соответствующей облачной учетной записи vSphere 7.x.  
  
Дополнительные сведения см. в [Создание плана кластера в vRealize Automation Cloud Assembly для использования с развертыванием vSphere with Tanzu Kubernetes](#).

5. Выберите **Проект > Облачные шаблоны** и создайте облачный шаблон для проекта с доступом к соответствующей зоне Kubernetes. Затем перетащите компонент K8s Cluster в схему облачного шаблона и укажите его имя и план кластера.

Можно также указать количество рабочих узлов.

6. Запустите облачный шаблон, а затем по завершении выполнения найдите адрес подготовленного кластера Tanzu в развертывании в свойствах ресурсов на странице «Развертывания Cloud Assembly».
7. Найдите и просмотрите кластер Tanzu в Cloud Assembly на странице **Инфраструктура > Настроить > Kubernetes**.

### Результаты

Кластер Tanzu Kubernetes подготавливается в соответствии с параметрами, указанными в облачном шаблоне.

### Следующие шаги

После развертывания кластера Tanzu можно использовать несколько вариантов работы с ним.

- Перейдите на страницу **Ресурсы > Развертывания** в Cloud Assembly, найдите и загрузите связанный файл Kubeconfig, чтобы получить доступ к подготовленному кластеру Tanzu. С помощью файла Kubeconfig можно управлять развернутым кластером Tanzu Kubernetes как любым другим совместимым кластером Kubernetes.
- Кластер Tanzu можно найти и просмотреть в Cloud Assembly на странице **Инфраструктура > Ресурсы > Kubernetes**.
- Чтобы создать новое пространство имен в применимом кластере Tanzu, перейдите на вкладку «Пространства имен» на странице Cloud Assembly **Инфраструктура > Ресурсы > Kubernetes** и щелкните **Создать пространство имен**. Чтобы убедиться в том, что пространство имен было создано, найдите его на вкладке «Пространства имен» на странице Kubernetes.

## Настройка интеграции Red Hat OpenShift в Cloud Assembly

Локальное подключение ресурсов Red Hat OpenShift можно настроить на локальном уровне и в облаке, чтобы обеспечить поддержку возможностей интеграции и управления Kubernetes корпоративного уровня в Cloud Assembly.

Cloud Assembly поддерживает интеграцию с OpenShift (версии 3.x).

### Необходимые условия

- У вас должна быть платформа Red Hat OpenShift, настроенная соответствующим образом.
- Убедитесь в наличии учетных данных администратора облака. Дополнительные сведения см. в разделе [Что такое роли пользователей vRealize Automation](#).



- VMware предоставляет ресурсы, которые можно использовать для создания кластера OpenShift с облачным шаблоном в следующем расположении: <https://flings.vmware.com/enterprise-openshift-as-a-service-on-cloud-automation-services>. Кластеры, созданные с этими ресурсами, можно использовать в качестве глобальных кластеров в зонах Kubernetes для создания пространств имен самообслуживания.

### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите Red Hat OpenShift.
3. Введите **адрес** и **расположение** для сервера OpenShift.
4. Выберите соответствующий **тип учетных данных** и введите необходимые учетные данные.

Интеграция OpenShift поддерживает проверку подлинности по имени пользователя или паролю OAuth, открытому ключу или маркеру предъявителя.

5. Введите соответствующее **имя** и **описание** для интеграции OpenShift.
6. Если для поддержки политики тегирования используются теги, введите соответствующие теги возможностей. См. разделы [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#) и [Создание политики расстановки тегов](#).
7. Нажмите кнопку **Добавить**.

### Результаты

После создания интеграции новые кластеры Kubernetes появляются в соответствующем разделе на странице Kubernetes. Можно создать зоны Kubernetes и назначить их проекту. Кроме того, можно настроить пространства имен Kubernetes, которые упрощают управление кластерами в крупных группах и организациях.

### Следующие шаги

Создайте или выберите соответствующие зоны Kubernetes, затем укажите один или несколько кластеров либо пространств имен и назначьте их проекту. После этого можно создать и опубликовать облачные шаблоны, чтобы пользователи могли создавать развертывания, использующие Kubernetes, по принципу самообслуживания.

## Настройка зоны Kubernetes в Cloud Assembly

Зоны Kubernetes позволяют администраторам облачных систем определять размещение кластеров Kubernetes и пространств имен, а также пространств имен управляющей программы, используемых в развертываниях Cloud Assembly, на основе политик. На этой странице администратор может указать кластеры, доступные для предоставления пространств имен Kubernetes, и свойства, применимые для кластеров.

Администраторы облачных систем могут связать зоны Kubernetes с облачными учетными записями TKGI, настроенными для Cloud Assembly, или с внешними кластерами Kubernetes, которые не связаны с проектом.

При создании зоны Kubernetes ей можно назначить несколько ресурсов для конкретного поставщика. Эти ресурсы будут определять, какие свойства можно настроить для новых подготовленных кластеров с точки зрения количества рабочих процессов, основных серверов, доступных ЦП, памяти и других параметров конфигурации. Для поставщиков TKGI они соответствуют планам TKGI. Администратор также может назначить несколько кластеров в зону Kubernetes, которая будет использоваться для размещения новых подготовленных пространств имен Kubernetes. Администратор может назначить только те кластеры, которые не внедрены или не управляются системой CMX и подготовлены с помощью предварительно выбранного поставщика кластеров. Администратор может назначить несколько зон Kubernetes в проект, что сделает их доступными для операций размещения, выполняемых в рамках данного проекта.

Администратор облачных систем может назначать приоритеты на разных уровнях.

- Приоритет зоны Kubernetes в рамках проекта.
- Приоритет ресурса в зоне Kubernetes.
- Приоритет кластера в зоне Kubernetes.

Администратор облачных систем также может назначать теги на разных уровнях.

- Теги возможностей для каждой зоны Kubernetes.
- Теги для каждого назначения ресурсов.
- Теги для каждого назначения кластера.

В vSphere зоны Kubernetes с пространствами имен управляющей программы можно создать таким же образом, как и при работе с универсальными пространствами имен Kubernetes. Чтобы добавить пространство имен управляющей программы в зону Kubernetes, необходимо связать эту зону с конечной точкой vSphere 7, которая содержит требуемые ресурсы пространства имен Pacific.

Service Broker содержит версию страницы зоны Kubernetes, с помощью которой администраторы Service Broker могут получать доступ к существующим зонам Kubernetes и создавать политики размещения для пространств имен и кластеров Kubernetes, подготавливаемых из каталога.

#### Необходимые условия

Настройте интеграцию с подходящим развертыванием

VMware Tanzu Kubernetes Grid Integrated Edition (TKGI). См. раздел [Настройка интеграции VMware Tanzu Kubernetes Grid Integrated Edition в Cloud Assembly](#).

#### Процедура

1. Выберите пункты **Инфраструктура > Настроить > Зона Kubernetes**, затем нажмите **Создать зону Kubernetes**.
2. Введите имя **учетной записи** для интеграции TKGI, к которой должна относиться эта зона.  
 Это определяет облачную учетную запись или конечную точку, связанную с зоной. В каждой зоне можно назначать только одну конечную точку. При работе с пространством имен управляющей программы в vSphere можно выбрать только те экземпляры vSphere, которые содержат пространства имен управляющей программы.
3. Добавьте **имя** и **описание** для зоны Kubernetes.

4. При необходимости добавьте теги возможностей. Дополнительные сведения см. в разделе [Использование тегов возможностей в Cloud Assembly](#).
5. Нажмите **Сохранить**.
6. Перейдите на вкладку «По требованию» и при необходимости добавьте для зоны планы TKGI, чтобы использовать их при предоставлении кластера.

Можно выбрать один или несколько планов и назначить им приоритеты. Чем меньше значение, тем выше приоритет. Назначения приоритетов не имеют большого значения для выбора на основе тегов.

7. Перейдите на вкладку «Кластер» и нажмите кнопку **Добавить вычислительный ресурс**, чтобы добавить кластеры Kubernetes или управляющей программы в зону. При работе с внешним кластером он автоматически внедряется в Cloud Assembly после выбора.

Пространства имен Kubernetes можно добавить в кластер на странице «Кластеры Kubernetes» в Cloud Assembly.

## Результаты

Зоны Kubernetes настраиваются для использования с развертываниями Cloud Assembly.

## Следующие шаги

Назначьте в проект зону Kubernetes.

1. Выберите пункты **Инфраструктура > Администрирование > Проекты**, а затем проект, который нужно связать с зоной Kubernetes.
2. Перейдите на вкладку «Предоставление Kubernetes» на странице «Проект».
3. Нажмите **Добавить зону Kubernetes** и добавьте только что созданную зону. При необходимости можно использовать несколько зон, а также задать их приоритет.
4. Нажмите **Сохранить**.

На вкладке «Предоставление Kubernetes» страницы «Проект» в Cloud Assembly можно установить ограничения по типу и количеству пространств имен, которые можно будет использовать для зоны Kubernetes. Можно также выбрать для зоны тип пространств имен: обычные пространства или пространства имен управляющей программы. В столбцах таблицы зон Kubernetes на вкладке «Предоставление Kubernetes» указаны текущие настройки ограничений. Чтобы установить ограничения, выберите нужную зону в таблице. Откроется диалоговое окно, в котором можно задать ограничения для пространств имен и пространств имен управляющей программы.

Щелкните столбец «Поддерживает» в таблице зон Kubernetes, чтобы выбрать тип пространства имен для зоны.

После назначения зоны Kubernetes проекту страницу «Облачные шаблоны» на вкладке «Проект» Cloud Assembly можно использовать для подготовки развертывания на основе зоны Kubernetes и конфигурации проекта. На этой странице содержатся параметры, позволяющие добавить кластер K8S, пространство имен K8S и пространство имен управляющей программы. Выберите соответствующий параметр для ресурса Kubernetes, с которым работаете.

## Создание плана кластера в vRealize Automation Cloud Assembly для использования с развертыванием vSphere with Tanzu Kubernetes

Необходимо создать план кластера для использования с развертываниями vSphere with Tanzu Kubernetes в vRealize Automation. План кластера функционирует как шаблон конфигурации для подготовки экземпляров кластера Tanzu Kubernetes в определенном экземпляре облачной учетной записи vSphere.

План кластера определяет сопоставление конфигурации, подобное сопоставлению конфигурации ресурсов, для набора экземпляров облачной учетной записи vSphere. Как правило, в плане кластера кодируется значимый набор свойств конфигурации, таких как классы виртуальных машин, классы хранилища и т. д., которые используются при подготовке кластеров Tanzu Kubernetes в конкретной облачной учетной записи сервера vSphere.

Обратите внимание, что один план кластера может иметь определенное сопоставление свойств конфигурации в одной облачной учетной записи vSphere и другое сопоставление конфигурации в другом экземпляре vSphere. Например, если имеются две соответствующие требованиям облачные учетные записи vSphere, одна из которых с большим объемом ресурсов, а другая с ограниченными ресурсами, в плане кластера `large` может быть указан класс `guaranteed-xlarge` для крупного сервера vSphere и класс `best-effort-medium` для ограниченного экземпляра vSphere. В общем, спецификация `large` сопоставляет набор свойств конфигурации с каждым подходящим экземпляром сервера vSphere.

После создания плана кластера для одного или нескольких экземпляров vSphere все подходящие пространства имен управляющей программы, которые администратор назначает для размещения кластера Tanzu Kubernetes посредством назначения зон Kubernetes, должны соответствовать конфигурации, определенной в спецификации плана кластера. Например, политику хранения, указанную в плане кластера, следует добавить в качестве класса хранилища ко всем пространствам имен управляющей программы vSphere, выделенным для подготовки кластеров Tanzu.

### Необходимые условия

- Чтобы создать развертывание vSphere with Tanzu Kubernetes в Cloud Assembly, необходимо иметь доступ к vSphere 7.x, который предоставляется в рамках облачной учетной записи vCenter. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).
- Служба Tanzu должна быть включена в облачной учетной записи vSphere с одним или несколькими пространствами имен управляющей программы.
- Все кластеры управляющей программы в зарегистрированной облачной учетной записи vSphere, подходящие для подготовки кластеров Tanzu, должны быть добавлены в качестве управляемых объектов в Cloud Assembly на странице **Инфраструктура > Kubernetes > Кластеры управляющей программы** с помощью параметра **Добавить кластер управляющей программы**.

### Процедура

1. Выберите **Инфраструктура > Настроить > План кластера** и щелкните **Новый план кластера**.
2. Введите **учетную запись, имя и описание** для плана кластера. Учетная запись определяет облачную учетную запись, к которой применяется этот план кластера.

3. Введите сведения о кластере, в том числе **версии Kubernetes** и **плоскость управления**. К этим сведениям относятся выделения для узлов, класс компьютера и класс хранилища.

- Введите версию Kubernetes, которая применяется к этому плану кластера. Это версия Kubernetes подготовленных кластеров Tanzu Kubernetes, например 1.19 или 1.20.
- Количество плоскостей управления определяет спецификацию узлов сервера API-интерфейса Kubernetes.
- Класс виртуальной машины — это запрос резервирования на виртуальной машине определенной вычислительной мощности. Существует множество предварительно определенных классов компьютеров, которые соответствуют различным уровням вычислительной мощности. Дополнительные сведения см. в разделе [Классы виртуальных машин для кластеров Tanzu Kubernetes](#).
- Рабочие процессы указывают узлы рабочих процессов Tanzu Kubernetes, которые будут развернуты с этим планом.

4. Введите и выберите дополнительные настройки для плана кластера.

- Введите **класс PVC по умолчанию**, который будет использоваться с этим кластером.
- Используйте переключатели, чтобы указать поведение, связанное с использованием классов хранилища и настроек сети.

5. Щелкните **Создать**.

#### Результаты

План кластера создан и доступен для использования в облачных шаблонах Cloud Assembly.

#### Следующие шаги

После создания плана кластера его можно использовать для создания развертывания vSphere with Tanzu Kubernetes в Cloud Assembly. См. раздел [Подготовка развертывания vSphere with Tanzu Kubernetes в vRealize Automation](#).

## Использование управляющих кластеров и пространств имен Tanzu в Cloud Assembly

Администраторы могут сделать пространства имен управляющей программы в интеграции vSphere с поддержкой Tanzu доступными для пользователей, чтобы те могли добавлять эти пространства имен в развертывания Kubernetes с помощью облачных шаблонов или запрашивать их из каталога Service Broker.

В этой задаче описывается добавление управляющих кластеров Tanzu в Cloud Assembly для использования в развертываниях, а также создание и добавление пространств имен, определяющих, какие проекты и пользователи Cloud Assembly могут получать доступ к определенным ресурсам Kubernetes. Эта функция основана на применении подходящей облачной учетной записи vSphere, а не интеграции, например VMware Tanzu Kubernetes Grid Integrated Edition (TKGI) или Openshift. Управляющие кластеры — это настроенные кластеры Kubernetes, связанные с vSphere. Они предоставляют конечным пользователям API-интерфейсы Kubernetes и используют в качестве платформы для узлов рабочих

процессов не Linux, а ESXi. Пространства имен управляющей программы упрощают контроль доступа к ресурсам Kubernetes, так как политики, как правило, проще применять к пространствам имен, а не к отдельным виртуальным машинам. Для каждого кластера управляющей программы можно создать несколько пространств имен.

Кроме того, в развертываниях с поддержкой Tanzu можно использовать гостевые кластеры, созданные решением vSphere. Гостевой кластер — это кластер Kubernetes, который запускается внутри виртуальных машин в управляющем кластере. Гостевой кластер полностью совместим с Kubernetes, поэтому он гарантированно работает со всеми приложениями Kubernetes. Гостевые кластеры в vSphere используют проект API-интерфейса кластера с открытым исходным кодом для управления жизненным циклом кластеров Kubernetes, который, в свою очередь, использует оператор ВМ для управления виртуальными машинами, входящими в гостевой кластер.

При использовании с экземплярами vSphere, поддерживающими Tanzu, зоны Kubernetes определяют, какие управляющие кластеры доступны для подготовки пространства имен управляющей программы. В экземплярах vSphere с поддержкой Tanzu используются особые пространства имен управляющей программы. Для экземпляра vSphere с поддержкой Tanzu нельзя подготовить универсальный ресурс Kubernetes.

Пользователи Cloud Assembly, называемые обозревателями проекта, имеют доступ к пространствам имен только для просмотра, в то время как участники проекта могут их изменять.

При необходимости можно настроить управляющие кластеры, связанные с пространствами имен.

#### Необходимые условия

- Для использования управляющих кластеров и пространств имен в Cloud Assembly необходима настроенная конечная точка vSphere 7.x. В vRealize Automation решение vSphere устанавливается в составе облачной учетной записи vCenter. См. раздел [Создание облачной учетной записи vCenter в vRealize Automation](#).
- Служба Tanzu должна быть включена в облачной учетной записи vSphere и должна содержать соответствующие пространства имен управляющей программы.
- Для синхронизации пользователей экземпляр vCenter и развертывание vRealize Automation должны использовать один и тот же каталог Active Directory. Если это не так, предоставление будет выполняться, но пользователи vRealize Automation не будут автоматически получать доступ к пространству имен.

#### Процедура

1. Выберите пункты **Инфраструктура > Настроить > Зона Kubernetes** в Cloud Assembly.

На этой странице отображаются управляемые кластеры, доступные для использования. Здесь также можно добавлять дополнительные кластеры. Для просмотра сведений о кластере щелкните его.

2. Выберите **Создать зону Kubernetes**.
3. Введите данные в поле **Учетная запись** для целевой облачной учетной записи vSphere.
4. Щелкните значок поиска в текстовом поле, чтобы просмотреть все учетные записи vSphere или выполнить поиск учетной записи по имени.

5. Введите **имя** и **описание** для новой зоны.
6. При необходимости добавьте теги возможностей. Дополнительные сведения см. в разделе [Использование тегов возможностей в Cloud Assembly](#).
7. Перейдите на вкладку «Предоставление», чтобы выбрать управляющий кластер, который будет связан с пространствами имен.
8. Щелкните **Добавить вычислительный ресурс**, чтобы просмотреть и выбрать доступные управляющие кластеры.
9. Нажмите кнопку **Добавить**.
10. Выберите пункты **Инфраструктура > Администрирование > Проекты**, а затем проект, который нужно связать с зоной Kubernetes.
11. Перейдите на вкладку «Предоставление Kubernetes» на странице «Проект».
12. Нажмите **Добавить зону Kubernetes** и добавьте только что созданную зону. При необходимости можно использовать несколько зон, а также задать их приоритет.
13. Нажмите **Сохранить**.

#### Следующие шаги

После настройки пространства имен оно отображается для соответствующих пользователей на странице **Инфраструктура > Ресурсы > Kubernetes** в Cloud Assembly. Пользователи могут щелкнуть ссылку «Адрес» на вкладке «Сводка», чтобы открыть «Средства Kubernetes CLI в vSphere» для управления пространством имен. Для доступа к сведениям о пространстве имен управляющей программы по ссылке пользователь должен быть администратором облачных систем или членом пространства имен для заданного проекта. Кроме того, пользователи могут загрузить настраиваемый инструмент Kubectl, чтобы использовать пространство имен управляющей программы. Пользователи могут войти в пространство имен управляющей программы и использовать его так же, как и любое другое пространство имен, а затем создать облачные шаблоны и развернуть приложения.

Чтобы добавить пространство имен в облачный шаблон, выберите пункты **Проектирование > Облачный шаблон**, а затем выберите существующий облачный шаблон или создайте новый. Затем выберите элемент «Пространство имен управляющей программы» в меню слева и перетащите его на холст.

Политики хранилища можно назначить пространству имен управляющей программы с помощью тегов. Можно добавить теги, например `location:local`, чтобы указать зону Kubernetes, которая будет применяться для развертывания, и другие теги в профилях хранилища, такие как `speed:fast` и `speed:slow`.

```
formatVersion: 1
resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: 'a'
      storage:
        -profile:
          constraints:
```

```

    - tag: 'speed:fast'
  -profile:
    liimitMB:1000
    constraints:
      -tag: 'speed:slow'

```

Этот облачный шаблон запрашивает пространство имен управляющей программы без ограничений и указывает для него два профиля хранилища.

После развертывания облачных шаблонов, содержащих пространство имен управляющей программы, пользователи могут также запрашивать пространства имен управляющей программы из каталога Service Broker. Кроме того, можно щелкнуть страницу «Развертывания» в Cloud Assembly, чтобы просмотреть сведения о развертывании и получить доступ к ссылке, которая содержит команду для запуска инструмента Kubectl для пространства имен в vSphere.

Классы виртуальных машин можно указать для пространств имен управляющей программы в облачном шаблоне с помощью свойства `vmclasses`, которое позволяет задать имя класса. См. следующий пример облачного шаблона.

```

resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: demo-vmclass1
      vmclasses:
        - name: vmclass1

```

## Работа с кластерами и пространствами имен Kubernetes в Cloud Assembly

Администраторы облачных систем могут добавлять и просматривать конфигурацию развернутых кластеров Kubernetes и пространств имен, как универсальных, так и основанных на Pacific, а также управлять ими в Cloud Assembly.

Пользователи с правами администратора облачных систем могут просматривать и добавлять те кластеры и пространства имен Kubernetes, к которым есть доступ на странице **Инфраструктура > Ресурсы > Kubernetes**, а также управлять ими. На этой странице содержатся вкладки для кластеров, пространств имен, кластеров управляющей программы и пространств имен управляющей программы. Одну из этих вкладок можно использовать для просмотра и управления аналогичными ресурсами. Как правило, эта страница позволяет упростить процесс управления развернутыми кластерами и пространствами имен.

- **Кластер.** Кластер — это группа узлов Kubernetes, распределенных по одному или нескольким физическим компьютерам. На этой странице отображаются подготовленные и неразвернутые кластеры, которые были настроены для использования в экземпляре Cloud Assembly. Чтобы просмотреть сведения о текущем состоянии кластера, щелкните его. При развертывании кластера имеется ссылка на файл `Kubconfig`, который доступен только администраторам облачных систем. Этот файл предоставляет полные права администратора для всего кластера, в том числе список пространств имен.

Кластеры управляющей программы являются уникальными для экземпляров vSphere и используют ESXi в качестве рабочих узлов вместо Linux.



- **Пространства имен.** Пространства имен — это виртуальные кластеры, с помощью которых администраторы могут объединять или разделять ресурсы кластера. Они упрощают управление ресурсами в крупных группах пользователей и организациях. В качестве формы контроля доступа на основе ролей администратор облачных систем может разрешить пользователям добавлять пространства имен в проект при запросе на развертывание, а затем управлять этими пространствами имен на странице «Кластеры Kubernetes». При развертывании пространства имен имеется ссылка на файл `kubecconfig`, который позволяет авторизованным пользователям, например разработчикам, просматривать некоторые аспекты пространства имен и управлять ими.

Кластеры управляющей программы и пространства имен управляющей программы существуют только в экземплярах vSphere и предоставляют доступ к объектам vSphere, аналогично доступу к Kubernetes.

Администратор облачных систем может изменить на этой странице проект, связанный с пространством имен или кластером Kubernetes. Таким образом, администратор может выполнять подготовку ресурсов Kubernetes с помощью облачных шаблонов и Service Broker, а затем назначать их конкретным проектам для последующего использования. Администратор может изменить область кластера, чтобы сделать его глобальным или выделенным для проекта. Для глобальных кластеров отображается вкладка «Кластеры» для всех зон Kubernetes. Такие кластеры доступны для выбора и подготовки. Если кластер является глобальным, его можно добавить в зону Kubernetes, а затем использовать для подготовки пространств имен с помощью каталога.

При настройке нового или существующего кластера необходимо указать, что используется при подключении: основной IP-адрес или основное имя узла.

### Работа с универсальными кластерами Kubernetes в Cloud Assembly

В Cloud Assembly можно добавлять новые, существующие или внешние кластеры, используя параметры на этой странице.

1. Выберите **Инфраструктура > Ресурсы > Kubernetes** и убедитесь, что вкладка «Кластеры» активна.  
Если в данный момент для экземпляра Cloud Assembly имеются настроенные кластеры, они будут отображаться на этой странице.

2. В случае добавления нового или существующего кластера либо развертывания кластера выберите соответствующий параметр в соответствии со следующей таблицей.

Параметр	Описание	Сведения
Развернуть	Добавление новых кластеров в Cloud Assembly	Необходимо указать облачную учетную запись TKGI, в которой будет развернут этот кластер, а также требуемый план и количество узлов.
Добавить существующий	Настройте существующий кластер для работы с проектом.	Необходимо указать облачную учетную запись TKGI, используемый кластер и соответствующий проект для конкретного разработчика. Кроме того, необходимо указать область предоставления общего доступа. Если общий доступ необходимо предоставить глобально, нужно соответствующим образом настроить зоны и пространства имен Kubernetes.
Добавить внешний	Добавьте в Cloud Assembly кластер Vanilla Kubernetes, который может быть связан с TKGI.	Необходимо указать проект, с которым связан кластер, ввести IP-адрес для нужного кластера и выбрать облачный прокси-сервер, а также сведения о сертификате, необходимые для подключения к этому кластеру.

3. Нажмите **Добавить**, чтобы сделать кластер доступным в Cloud Assembly.

### Работа с пространствами имен Kubernetes в Cloud Assembly

С помощью пространств имен администратор облачных систем может группировать ресурсы кластера Kubernetes и управлять ими. Для пользователя пространства имен представляют собой область в кластерах Kubernetes для развертываний. Администраторы и пользователи могут получать доступ к пространствам имен, используя вкладку «Пространства имен» на странице **Инфраструктура > Ресурсы > Kubernetes**.

Для добавления пространств имен Kubernetes в ресурсы в Cloud Assembly можно использовать несколько способов. Ниже представлена типовая процедура.

1. Выберите **Инфраструктура > Ресурсы > Kubernetes** и перейдите на вкладку «Пространства имен».
2. Чтобы добавить новое пространство имен, щелкните **Создать пространство имен**. Чтобы добавить существующее пространство имен, нажмите **Добавить пространство имен**.

3. Введите **имя** и **описание** для пространства имен.

Сейчас пространство имен добавлено для использования с ресурсами Kubernetes, но оно ни с чем не связано.

4. Укажите **кластер**, который необходимо связать с этим пространством имен.
5. Щелкните **Создать**, чтобы добавить пространство имен в Cloud Assembly.

В пространствах имен Kubernetes можно добавить настраиваемые свойства для поддержки расширяемости различными способами. Настраиваемые свойства можно добавить при подготовке пространства имен путем создания облачного шаблона Cloud Assembly. Свойства можно добавить в пространство имен при указании пространства имен Kubernetes в облачном шаблоне. Свойства можно щелкнуть в шаблоне правой кнопкой мыши, чтобы получить доступ к свойствам по умолчанию, которые являются частью схемы облачного шаблона. С другой стороны, определяемые пользователем свойства можно добавить в разделе свойств пространства имен в облачном шаблоне.

После развертывания эти настраиваемые свойства появляются на странице «Развертывания» в Cloud Assembly для соответствующего развертывания.

Наконец, настраиваемые свойства можно добавить в пространство имен с помощью действий, настроенных на странице **Расширяемость > Действия** в Cloud Assembly.

### Работа с кластерами управляющей программы и пространствами имен управляющей программы

Администраторы облачных систем могут просматривать и изменять конфигурацию кластеров и пространств имен управляющей программы на странице Kubernetes в Cloud Assembly.

1. Выберите **Инфраструктура > Ресурсы > Kubernetes** в Cloud Assembly.
2. Нажмите **Добавить кластер управляющей программы**.
3. Введите сведения об учетной записи для целевой облачной учетной записи vSphere.
4. Щелкните значок «Поиск» в текстовом поле кластера управляющей программы, чтобы просмотреть все кластеры управляющей программы или выполнить поиск кластера по имени.
5. Выберите нужный кластер и нажмите **Добавить**.
6. Перейдите на вкладку «Пространства имен управляющей программы» и нажмите кнопку **Создать пространство имен управляющей программы**, чтобы добавить новое пространство имен.
7. Перейдите на вкладку «Пространства имен управляющей программы» и нажмите кнопку **Создать пространство имен управляющей программы**, чтобы добавить новое пространство имен.
  - а) При создании нового пространства имен заполните поля **Имя** и **Описание**.
  - б) Выберите соответствующую **облачную учетную запись** для связи с пространством имен.
  - в) Выберите **Кластер управляющей программы** для связи с этим пространством имен.
  - г) Выберите **Проект** для связи с пространством имен.
  - д) Чтобы добавить политики хранилища для применения с пространством имен, используйте вариант **Доступные политики хранилища**.

Можно добавить все доступные политики хранилища или выбрать конкретные политики для использования с пространством имен управляющей программы. Кроме того, можно дополнительно установить ограничение на размер хранилища для каждой доступной политики хранилища.

- е) Щелкните **Создать**.
8. Просмотрите соответствующие сведения для нового пространства имен. При необходимости конфигурацию политики хранилища можно изменить.

Пользователи и группы, у которых в данный момент есть доступ к пространству имен в vSphere, перечислены на вкладке «Пользователи». Если в проект добавляются новые пользователи или группы, нажмите кнопку **Обновить пользователей** на этой вкладке, чтобы обновить список. Список не обновляется автоматически, поэтому для обновления необходимо нажать кнопку.

---

**Примечание** Синхронизация пользователей имеет смысл только в том случае, если Cloud Assembly и vCenter настроены с общей службой Active Directory или LDAP.

---

После настройки кластера или пространства имен на странице **Инфраструктура > Ресурсы > Kubernetes** в Cloud Assembly отображаются кластеры и пространства имен, доступные пользователю. Можно щелкнуть отдельное пространство имен или кластер, чтобы открыть страницу с несколькими вкладками, на которых отображается статистика и другая информация о ресурсе, и настроить различные параметры.

С помощью вкладки «Сводка» для кластеров на странице Kubernetes администраторы могут просматривать и в ряде случаев обновлять конфигурацию кластера, в том числе изменять область. С помощью переключателей «Общий доступ» можно выбрать режим «Глобальный» (общий доступ внутри зоны Kubernetes) или «Проект» (доступ возможен только для одного проекта). Если выбрано значение «Проект», также необходимо указать применимый проект на этапе выбора проекта.

---

**Примечание** Изменение конфигурации общего доступа может повлиять на пространства имен, доступные в кластере.

---

Пользователи могут щелкнуть ссылку «Адрес» на вкладке «Сводка», чтобы открыть «Средства Kubernetes CLI в vSphere» для управления пространством имен. Для доступа к сведениям о пространстве имен управляющей программы по ссылке пользователь должен быть администратором облачных систем или членом пространства имен для заданного проекта. Кроме того, пользователи могут загрузить настраиваемый инструмент Kubectl, чтобы использовать пространство имен управляющей программы. Пользователи могут войти в пространство имен управляющей программы и использовать его так же, как и любое другое пространство имен, а затем создать облачные шаблоны и развернуть приложения.

## Добавление компонентов Kubernetes в облачные шаблоны в Cloud Assembly

При добавлении компонентов Kubernetes в облачный шаблон Cloud Assembly можно добавить кластеры или разрешить пользователям создавать пространства имен в различных конфигурациях. Как правило, выбор операции зависит от требований по контролю доступа, от способа настройки компонентов Kubernetes и требований к развертыванию.

Чтобы добавить компонент Kubernetes в облачный шаблон в Cloud Assembly, выберите

**Проектирование > Облачные шаблоны**, щелкните **Создать**, а затем найдите и разверните параметр Kubernetes в меню слева. Затем выберите нужный вариант («Кластер» или «Пространство имен KBS»), перетаскив его на холст.

Добавление кластера Kubernetes, связанного с проектом, в облачный шаблон — это наиболее простой способ предоставить соответствующим пользователям доступ к ресурсам Kubernetes. Для управления местом развертывания можно использовать теги в кластерах, так же как в случае с другими ресурсами Cloud Assembly. Теги можно использовать для выбора зоны и плана VMware Tanzu Kubernetes Grid Integrated Edition (TKGI) на этапе выделения в ходе развертывания кластера.

После добавления кластера таким образом он автоматически становится доступен всем допустимым пользователям.

### Примеры облачных шаблонов

В первом примере показан облачный шаблон для простого развертывания Kubernetes, управляемого с помощью тегов. Зона Kubernetes создана с двумя планами развертывания, настроенными на странице «Создать зону Kubernetes». В этом случае тег с именем `placement:tag` добавлен в зону как возможность, и он используется для сопоставления аналогичного ограничения в облачном шаблоне. Если тег присвоен нескольким зонам, будет выбрано значение с наименьшим приоритетом.

```
formatVersion: 1
inputs: {}
resources:
  Cluster_provisioned_from_tag:
    type: Cloud.K8S.Cluster
    properties:
      hostname: 109.129.209.125
      constraints:
        -tag: 'placement tag'
      port: 7003
      workers: 1
      connectBy: hostname
```

Во втором примере облачного шаблона показано, как настроить шаблон с переменной `$(input.hostname)`, чтобы пользователи могли ввести требуемое имя узла кластера при запросе развертывания. Теги также можно использовать для выбора зоны и плана TKGI на этапе выделения ресурсов в ходе развертывания кластера.

```
formatVersion: 1
inputs:
  hostname:
    type: string
    title: Cluster hostname
resources:
  Cloud_K8S_Cluster_1:
    type: Cloud.K8S.Cluster
    properties:
      hostname: $(input.hostname)
      port: 8443
      connectBy: hostname
      workers: 1
```

Если необходимо использовать пространства имен для управления использованием кластера, в облачном шаблоне можно настроить переменную `name: ${input.name}` для подстановки имени пространства имен, вводимого пользователем при запросе развертывания. Для развертывания такого типа необходимо создать облачный шаблон, как в примере ниже.

```

1 formatVersion: 1
2 inputs:
3   name:
4     type: string
5     title: "Namespace name"
6 resources:
7   Cloud_K8S_Namespace_1:
8     type: Cloud.K8S.Namespace
9     properties:
10      name: ${input.name}

```

Пользователи могут управлять развернутыми кластерами с помощью файлов `kubecfg`, доступных на странице **Инфраструктура > Ресурсы > Кластеры Kubernetes**. Найдите карточку на странице для нужного кластера и нажмите **Kubecfg**.

### Пространства имен управляющей программы в шаблонах VMware Cloud Templates

Ниже приведена схема для основного пространства имен управляющей программы в облачном шаблоне Cloud Assembly.

```

{
  "title": "Supervisor namespace schema",
  "description": "Request schema for provisioning of Supervisor namespace resource",
  "type": "object",
  "properties": {
    "name": {
      "title": "Name",
      "description": "Alphabetic (a-z and 0-9) string with maximum length of 63 characters. The character '-' is allowed anywhere except the first or last position of the identifier.",
      "type": "string",
      "pattern": "^[a-z0-9-]{1,63}$",
      "ignoreOnUpdate": true
    },
    "description": {
      "title": "Description",
      "description": "An optional description of this Supervisor namespace.",
      "type": "string",
      "ignoreOnUpdate": true
    },
    "content": {
      "title": "Content",
      "description": "Kubernetes Yaml Content",
      "type": "string",
      "maxLength": 65000
    },
    "constraints": {
      "title": "Constraints",
      "description": "To target the correct resources, blueprint constraints are matched

```

```

against infrastructure capability tags. Constraints must include the key name. Options
include value, negative [!], and hard or soft requirement.",
  "type": "array",
  "recreateOnUpdate": true,
  "items": {
    "type": "object",
    "properties": {
      "tag": {
        "title": "Tag",
        "description": "Constraint definition in syntax `[!]tag_key[:tag_value]`
[:hard|soft]` \nExamples:\n```\n!location:eu:hard\n location:us:soft\n!pci\n```,
        "type": "string",
        "recreateOnUpdate": true
      }
    }
  },
  "limits": {
    "title": "Limits",
    "description": "Defines namespace resource limits such as pods, services, etc.",
    "type": "object",
    "properties": {
      "stateful_set_count": {
        "title": "stateful_set_count",
        "description": "This represents the new value for 'statefulSetCount' option which
is the maximum number of StatefulSets in the namespace.",
        "type": "integer",
        "recreateOnUpdate": false
      },
      "deployment_count": {
        "title": "deployment_count",
        "description": "This represents the new value for 'deploymentCount' option which is
the maximum number of deployments in the namespace.",
        "type": "integer",
        "recreateOnUpdate": false
      },
      "cpu_limit_default": {
        "title": "cpu_limit_default",
        "description": "This represents the new value for the default CPU limit (in Mhz)
for containers in the pod. If specified, this limit should be at least 10 Mhz.",
        "type": "integer",
        "recreateOnUpdate": false
      },
      "config_map_count": {
        "title": "config_map_count",
        "description": "This represents the new value for 'configMapCount' option which is
the maximum number of ConfigMaps in the namespace.",
        "type": "integer",
        "recreateOnUpdate": false
      },
      "pod_count": {
        "title": "pod_count",
        "description": "This represents the new value for 'podCount' option which is the
maximum number of pods in the namespace.",
        "type": "integer",

```

```

    "recreateOnUpdate": false
  },
  "job_count": {
    "title": "job_count",
    "description": "This represents the new value for 'jobCount' option which is the
maximum number of jobs in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "secret_count": {
    "title": "secret_count",
    "description": "This represents the new value for 'secretCount' option which is the
maximum number of secrets in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "cpu_limit": {
    "title": "cpu_limit",
    "description": "This represents the new value for 'limits.cpu' option which is
equivalent to the maximum CPU limit (in MHz) across all pods in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "cpu_request_default": {
    "title": "cpu_request_default",
    "description": "This represents the new value for the default CPU request (in Mhz)
for containers in the pod. If specified, this field should be at least 10 MHz.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "memory_limit_default": {
    "title": "memory_limit_default",
    "description": "This represents the new value for the default memory limit (in
mebibytes) for containers in the pod.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "memory_limit": {
    "title": "memory_limit",
    "description": "This represents the new value for 'limits.memory' option which is
equivalent to the maximum memory limit (in mebibytes) across all pods in the namespace.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "memory_request_default": {
    "title": "memory_request_default",
    "description": "This represents the new value for the default memory request (in
mebibytes) for containers in the pod.",
    "type": "integer",
    "recreateOnUpdate": false
  },
  "service_count": {
    "title": "service_count",
    "description": "This represents the new value for 'serviceCount' option which is
the maximum number of services in the namespace.",

```



```

        "type": "integer",
        "recreateOnUpdate": false
    },
    "replica_set_count": {
        "title": "replica_set_count",
        "description": "This represents the new value for 'replicaSetCount' option which is
the maximum number of ReplicaSets in the namespace.",
        "type": "integer",
        "recreateOnUpdate": false
    },
    "replication_controller_count": {
        "title": "replication_controller_count",
        "description": "This represents the new value for 'replicationControllerCount'
option which is the maximum number of ReplicationControllers in the namespace.",
        "type": "integer",
        "recreateOnUpdate": false
    },
    "storage_request_limit": {
        "title": "storage_request_limit",
        "description": "This represents the new value for 'requests.storage' which is the
limit on storage requests (in mebibytes) across all persistent volume claims from pods in the
namespace.",
        "type": "integer",
        "recreateOnUpdate": false
    },
    "persistent_volume_claim_count": {
        "title": "persistent_volume_claim_count",
        "description": "This represents the new value for 'persistentVolumeClaimCount'
option which is the maximum number of PersistentVolumeClaims in the namespace.",
        "type": "integer",
        "recreateOnUpdate": false
    },
    "daemon_set_count": {
        "title": "daemon_set_count",
        "description": "This represents the new value for 'daemonSetCount' option which is
the maximum number of DaemonSets in the namespace.",
        "type": "integer",
        "recreateOnUpdate": false
    }
},
"additionalProperties": false
},
"vm_classes": {
    "title": "VM classes",
    "description": "Defines set of Virtual Machine classes to be assigned to the namespace",
    "type": "array",
    "recreateOnUpdate": false,
    "items": {
        "type": "object",
        "properties": {
            "name": {
                "title": "Name",
                "description": "Name of the Virtual Machine class.",
                "type": "string",
                "recreateOnUpdate": false
            }
        }
    }
}

```

```

    }
  }
},
"storage": {
  "title": "Storage policies",
  "description": "Defines set of storage profiles to be used to assign storage policies
to the namespace.",
  "type": "array",
  "recreateOnUpdate": false,
  "items": {
    "type": "object",
    "properties": {
      "profile": {
        "type": "object",
        "title": "Storage profile",
        "description": "Defines storage policies to be assigned to the namespace",
        "recreateOnUpdate": false,
        "properties": {
          "constraints": {
            "title": "Constraints",
            "description": "To target the correct storage profiles, blueprint constraints
are matched against storage profile capability tags.",
            "type": "array",
            "recreateOnUpdate": false,
            "items": {
              "type": "object",
              "properties": {
                "tag": {
                  "title": "Tag",
                  "description": "Constraint definition in syntax `[!]tag_key[:tag_value]
[:hard|:soft]` \nExamples:\n```\nlocation:eu:hard\n location:us:soft\n```,
                  "type": "string",
                  "recreateOnUpdate": false
                }
              }
            },
            "minItems": 1
          },
          "limitMb": {
            "title": "Limit",
            "description": "The maximum amount of storage (in mebibytes) which can be
utilized by the namespace for this storage policy. Optional. If unset, no limits are placed.",
            "type": "integer"
          }
        },
        "required": [
          "constraints"
        ]
      }
    }
  },
  "required": [

```

```

    "name"
  ]
}

```

Шаблоны VMware Cloud Templates поддерживают использование предельных значений для пространств имен управляющей программы. Предельные значения позволяют контролировать использование ресурсов ЦП и памяти, а также максимальное количество модулей, разрешенных в пространстве имен, в зависимости от числа развернутых компьютеров.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: '${env.deploymentName}'
      limits:
        - cpu_limit: 1000
          cpu_request_default: 800
          memory_limit: 2000
          memory_limit_default: 1500
          pod_count: 200

```

В следующем примере показано, как можно указать политику хранилища с помощью тегов.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: 'ns-with-storage-policy'
      description: 'sample'
      storage:
        - profile:
            limitMb: 1000
            constraints:
              - tag: 'storage:fast'
        - profile:
            constraints:
              - tag: 'storage:cheap'

```

## Использование произвольных кодов YAML с пространством имен самообслуживания или VST кластера

В ходе создания кластера или пространства имен пользователям часто требуется выполнить дополнительную настройку. Например, может потребоваться добавление пользователей (привязки «роль — роль»), создание политики безопасности модуля или установка агентов. С помощью свойства YAML `content` пользователи могут определить настраиваемые пакеты, которые нужно подготовить в данном пространстве имен кластера или управляющей программы.

Каждый пакет содержимого YAML, связанный со свойством `content`, должен быть отделен тройным тире (`---`). Сведения о содержимом должны представлять собой строку с несколькими подстроками. Дополнительные сведения о настройке пакетов содержимого см. в следующем примере кода YAML.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Tanzu_Cluster_1:
    type: Cloud.Tanzu.Cluster
    properties:
      name: ddonchev-tkc
      plan: small
      content: |-
        apiVersion: rbac.authorization.k8s.io/v1
        kind: ClusterRoleBinding
        metadata:
          name: psp:authenticated-from-yaml
        subjects:
        - apiGroup: rbac.authorization.k8s.io
          kind: Group
          name: system:authenticated
        roleRef:
          apiGroup: rbac.authorization.k8s.io
          kind: ClusterRole
          name: psp:vmware-system-privileged
        ---
        apiVersion: apiextensions.k8s.io/v1
        kind: CustomResourceDefinition
        metadata:
          # name must match the spec fields below, and be in the form: <plural>.<group>
          name: crontabs.stable.example.com
        spec:
          # group name to use for REST API: /apis/<group>/<version>
          group: stable.example.com
          # list of versions supported by this CustomResourceDefinition
          versions:
            - name: v1
              # Each version can be enabled/disabled by Served flag.
              served: true
              # One and only one version must be marked as the storage version.
              storage: true
              schema:
                openAPIV3Schema:
                  type: object
                  properties:
                    spec:
                      type: object
                      properties:
                        cronSpec:
                          type: string
                        image:
                          type: string
                        replicas:
                          type: integer
```

```
# either Namespaced or Cluster
scope: Namespaced
names:
  # plural name to be used in the URL: /apis/<group>/<version>/<plural>
  plural: crontabs
  # singular name to be used as an alias on the CLI and for display
  singular: crontab
  # kind is normally the CamelCased singular type. Your resource manifests use this.
  kind: CronTab
  # shortNames allow shorter string to match your resource on the CLI
  shortNames:
    - ct
```

Код YAML, определенный в свойстве содержимого, также отображается на вкладке «Свойства» для развертывания.

Cloud Assembly может создавать ресурсы содержимого только в пределах развертываемого ресурса. Например, при подготовке пространства имен kubernetes Cloud Assembly не может создать развертывание в другом пространстве имен. Пользователи имеют те же права, как если бы они использовали kubeconfig с kubectl.

После подготовки виртуальной машины начнется процесс установки объектов kubernetes внутри свойства content. Если не удастся подготовить один из ресурсов, ссылки на которые содержатся в свойстве содержимого YAML, Cloud Assembly выполнит откат и удалит из ресурса все предыдущие объекты kubernetes, а развертывание будет иметь состояние «Ошибка». Этот ресурс будет по-прежнему подготовлен и будет отображаться. Также можно использовать действия по регулярному обслуживанию, в том числе попытаться повторно применить содержимое.

Для расширения свойства content можно использовать входные данные из облачного шаблона, как показано в следующем примере.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: sv-namespace-with-vm-classes
      vm_classes:
        - name: best-effort-2xlarge
        - name: best-effort-4xlarge
        - name: best-effort-8xlarge
```

Кроме того, можно подготовить настраиваемые ресурсы, например TanzuKubernetesCluster. Это действие, представляющее собой операцию в ходе настройки, завершится сбоем, так как пространство имен управляющей программы не будет содержать необходимые классы виртуальной машины и классы хранилища. Если классы виртуальной машины и классы хранилища привязаны к пространству имен управляющей программы, то TanzuKubernetesCluster (или другой ресурс) можно создать с помощью действия по регулярному обслуживанию.

Примечание. Ресурс можно подготовить без содержимого. Объекты kubernetes в виде кода YAML можно будет пр-прежнему добавить с помощью действия по регулярному обслуживанию.

Содержимое, отображаемое в свойстве YAML, определяет, что именно подготовлено в ресурсе. В следующей таблице показаны возможные результаты в случае изменения этого содержимого.

Действие	Результат
Если добавляется объект kubernetes и выполняется отправка.	Указанный объект создается в ресурсе.
Если удаляется объект kubernetes и выполняется отправка.	Указанный объект удаляется из ресурса.
Если изменяется объект kubernetes и выполняется отправка.	Указанный объект исправляется в ресурсе.

Важно пояснить, какие действия считаются изменением текущего объекта. Например, если изменить поле пространства имен для объекта, то вместо исправления старого объекта будет создан новый объект.

Уникальность ресурса определяется следующими полями: `apiVersion`, `kind`, `metadata.name`, `metadata.namespace`.

## Использование расширяемости Cloud Assembly с помощью Kubernetes

Cloud Assembly предоставляет набор тем событий, которые соответствуют типовым действиям, связанным с развертыванием кластера Kubernetes и пространства имен. При необходимости пользователи могут подписаться на эти темы, и они будут запускаться в соответствующее время. Пользователи получают уведомление, когда произойдет событие, связанное с темой подписки. Кроме того, можно настроить запуск рабочих процессов vRO на основе уведомлений о событиях.

На странице **Расширяемость > Библиотека > Темы событий** в Cloud Assembly для подписки доступны следующие темы. Чтобы просмотреть эти темы, выполните поиск Kubernetes в текстовом поле «Поиск тем событий».

- Выделение кластера Kubernetes
- Пост-подготовка кластера Kubernetes
- Пост-удаление кластера Kubernetes
- Подготовка кластера Kubernetes
- Удаление кластера Kubernetes
- Выделение пространства имен Kubernetes
- Пост-подготовка пространства имен Kubernetes
- Пост-удаление пространства имен Kubernetes
- Удаление пространства имен Kubernetes
- Выделение пространства имен Kubernetes
- Выделение пространства имен управляющей программы Kubernetes
- Пост-подготовка пространства имен управляющей программы Kubernetes
- Пост-удаление пространства имен управляющей программы Kubernetes

- Удаление пространства имен управляющей программы Kubernetes
- Выделение пространства имен управляющей программы Kubernetes

Щелкните одну из тем, чтобы просмотреть схему для данной темы, в которой отображаются все собранные и переданные сведения. Для пространств имен Kubernetes и управляющей программы существуют отдельные темы. Для настройки различных уведомлений, а также задач по управлению и отчетности можно использовать любую из этих схем.

Сценарии действий, связанных с CMX, можно настроить на странице **Расширяемость > Библиотека > Действия**. Сценарии действий можно использовать для различных целей, например для создания DNS-записи для предоставления кластера Kubernetes. При создании DNS-записи можно использовать поле `masternodeips` в теме «Пост-подготовка кластера Kubernetes» с помощью команды REST в сценарии «Действие» для создания DNS-записи.

На странице «Подписки» определяется связь между темами событий и сценариями действий. Для просмотра этих компонентов и управления ими можно использовать страницу «Подписки» в Cloud Assembly

Дополнительные сведения см. в документации по расширяемости Cloud Assembly в разделе [Увеличение и автоматизация жизненных циклов приложений с помощью действий расширяемости](#).

## Что такое управление конфигурацией в Cloud Assembly

Cloud Assembly поддерживает интеграцию с Puppet Enterprise, Ansible Open Source и Ansible Tower, позволяя управлять конфигурациями развертываний и дрейфом конфигураций.

### Интеграция Puppet

Для интеграции управления конфигурацией на основе Puppet необходимо установить допустимый экземпляр Puppet Enterprise в общедоступном или частном облаке с рабочей нагрузкой vSphere. Между этой внешней системой и экземпляром Cloud Assembly необходимо установить соединение. После этого можно предоставить Cloud Assembly доступ к управлению конфигурацией Puppet путем его добавления в соответствующие схемы элементов.

Поставщик Puppet для службы схем элементов Cloud Assembly устанавливает, настраивает и запускает агент Puppet на развернутом вычислительном ресурсе. Поставщик Puppet поддерживает подключения по протоколу SSH и WinRM со следующими предварительными условиями.

- Подключения по SSH:
  - В качестве пользователя должен быть указан либо привилегированный пользователь, либо пользователь с разрешениями `sudo` для выполнения команд в режиме `NOPASSWD`.
  - Деактивируйте `requiretty` для заданного пользователя.
  - В вычислительном ресурсе развертывания должна быть доступна служба `cURL`.
- Подключения по WinRM:
  - в вычислительном ресурсе развертывания должен быть доступен инструмент PowerShell 2.0.
  - Настройте шаблон Windows, как описано в документации по vRealize Orchestrator.

Администратор DevOps отвечает за управление подключениями к Puppet Master и применение ролей Puppet или правил настройки к определенным развертываниям. После развертывания виртуальные машины, в которых настроена поддержка управления конфигурацией, регистрируются с помощью назначенного сервера Puppet Master.

При развертывании виртуальных машин пользователи могут добавлять или удалять Puppet Master как внешнюю систему или обновлять проекты, назначенные серверу Puppet Master. Наконец, соответствующие пользователи могут отменять регистрацию развернутых виртуальных машин в Puppet Master после вывода их из эксплуатации.

## Интеграция Ansible Open Source

При настройке интеграции Ansible установите Ansible Open Source в соответствии с инструкциями по установке Ansible. Дополнительные сведения об установке см. в документации по Ansible.

Ansible по умолчанию включает функцию проверки ключа узла. Если узел установлен повторно с другим ключом в файле `known_hosts`, появится сообщение об ошибке. Если узел не указан в файле `known_hosts`, необходимо ввести ключ при запуске. Для деактивации проверки ключа узла используйте следующий параметр в файле `/etc/ansible/ansible.cfg` или `~/.ansible.cfg`:

```
[defaults]
host_key_checking = False
localhost_warning = False

[paramiko_connection]
record_host_keys = False

[ssh_connection]
#ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
ssh_args = -o UserKnownHostsFile=/dev/null
```

Для предотвращения ошибок при проверке ключей узла задайте для параметров `host_key_checking` и `record_host_keys` значение `False`, а также добавьте дополнительный параметр `UserKnownHostsFile=/dev/null`, заданный в `ssh_args`. Кроме того, если изначально иерархия пуста, Ansible выведет предупреждение, что список узлов пуст. Это приведет к сбою проверки синтаксиса сценария.

Хранилище Ansible Vault позволяет хранить конфиденциальные данные, например пароли или ключи, в зашифрованных файлах, а не в виде обычных текстовых файлов. Хранилище шифруется с помощью пароля. В системе Cloud Assembly Ansible использует хранилище для шифрования таких данных, как пароли SSH для компьютеров узлов. Предполагается, что задан путь к паролю хранилища.

Файл `ansible.cfg` можно изменить, указав расположение файла с паролем в следующем формате.

```
vault_password_file = /path to/file.txt
```

Кроме того, можно задать переменную среды `ANSIBLE_VAULT_PASSWORD_FILE`, чтобы Ansible автоматически выполняла поиск пароля. Например, `ANSIBLE_VAULT_PASSWORD_FILE=~/.vault_pass.txt`.



Cloud Assembly управляет файлом иерархии Ansible, поэтому необходимо убедиться, что пользователь Cloud Assembly имеет доступ rwx к файлу иерархии.

```
cat ~/var/tmp/vmware/provider/user_defined_script/$(ls -t ~/var/tmp/vmware/provider/
user_defined_script/ | head -1)/log.txt
```

Если необходимо использовать пользователя, не являющегося привилегированным (root), с интеграцией Cloud Assembly с открытым исходным кодом, то пользователям потребуется набор разрешений для запуска команд, используемых поставщиком Cloud Assembly с открытым исходным кодом. В файле sudoers пользователя должны быть настроены следующие команды.

```
Defaults:myuser !requiretty
```

Если пользователь не является членом группы администраторов, для которой приложение askpass не указано, настройте следующую команду в файле sudoers пользователя.

```
myuser ALL=(ALL) NOPASSWD: ALL
```

Если при настройке интеграции Ansible возникают ошибки или иные проблемы, используйте файл log.txt в 'cat~/var/tmp/vmware/provider/user\_defined\_script/\$(ls -t ~/var/tmp/vmware/provider/user\_defined\_script/ | head -1)/' на управляющем компьютере Ansible.

## Интеграция Ansible Tower

Поддерживаемые типы операционных систем

- Red Hat Enterprise Linux 8.0 или более поздней версии, 64-разрядная (x86), поддерживает только Ansible Tower 3.5 или более поздней версии.
- Red Hat Enterprise Linux 7.4 или более поздней версии, 64-разрядная (x86).
- CentOS 7.4 или более поздней версии, 64-разрядная (x86).

Ниже приведен пример файла иерархии, который создается при установке Ansible Tower. Возможно, для использования интеграции Cloud Assembly его потребуется изменить.

```
[root@cava-env8-dev-001359 ansible-tower-setup-bundle-3.5.2-1.el8]# pwd

/root/ansible-tower-install/ansible-tower-setup-bundle-3.5.2-1.el8

[root@cava-env8-dev-001359 ansible-tower-setup-bundle-3.5.2-1.el8]# cat inventory

[tower]

localhost ansible_connection=local


[database]
```

```

[all:vars]

admin_password='VMware1!'

pg_host=''

pg_port=''

pg_database='awx'

pg_username='awx'

pg_password='VMware1!'

rabbitmq_port=5672

rabbitmq_vhost=tower

rabbitmq_username=tower

rabbitmq_password='VMware1!'

rabbitmq_cookie=cookiemonster

# Needs to be true for fqdns and ip addresses

rabbitmq_use_long_name=false

# Isolated Tower nodes automatically generate an RSA key for authentication;

# To deactivate this behavior, set this value to false

# isolated_key_generation=true

```

## Настройка интеграции Puppet Enterprise в службе Cloud Assembly

Служба Cloud Assembly поддерживает интеграцию с Puppet Enterprise — программным обеспечением для управления конфигурациями.

Когда инструмент Puppet Enterprise добавляется в Cloud Assembly в качестве внешней системы, по умолчанию его возможности доступны для всех проектов. Можно ограничить доступ, сделав систему Puppet доступной только определенным проектам.

Чтобы добавить интеграцию Puppet Enterprise, потребуется имя главного сервера Puppet, а также его имя узла или IP-адрес.

Журналы Puppet можно найти в следующем расположении в том случае, если необходимо проверить их на наличие ошибок или определенной информации.

Описание	Расположение журнала
Журнал для создания и установки связанных событий	Журналы находятся на развернутом компьютере по адресу <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/   head -1)/`</code> . Полные журналы см. в файле <b>log.txt</b> . Дополнительные сведения о журналах агента Puppet см. в разделе <a href="https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging">https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging</a>
Журнал для задач Puppet, связанных с удалением и запуском	Журналы находятся в Puppet Enterprise по адресу <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/   head -1)/`</code> . Полные журналы см. в файле <b>log.txt</b> .

## Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите Puppet.
3. Введите необходимую информацию на странице настройки Puppet.

Чтобы интеграция Puppet функционировала правильно, указанные учетные данные должны быть действительными как для учетной записи SSH, так и для учетной записи API-интерфейса. Также у указанных учетных записей пользователей ОС и приложений должны совпадать имена пользователей и пароли.

4. Нажмите **Проверить**, чтобы проверить интеграцию.
5. Нажмите кнопку **Добавить**.

## Результаты

Приложение Puppet доступно для использования при работе с облачными шаблонами.

## Следующие шаги

Добавьте компоненты Puppet в нужные облачные шаблоны.

1. В разделе «Облачные шаблоны» в Cloud Assembly выберите Puppet в разделе «Управление содержимым» в меню облачного шаблона и перетащите компонент Puppet на холст.
2. Введите свойства Puppet на панели справа.

Свойство	Описание
Сервер	Введите имя основного компьютера Puppet, используемого с этим облачным шаблоном.
Среда	Выберите среду для основного компьютера Puppet.
Роль	Выберите роль Puppet, которая будет использоваться с этим облачным шаблоном.
Интервал между запусками агента	Частота, с которой агент Puppet выполняет опрос основного компьютера Puppet для получения сведений о конфигурации, применяемых к развернутым виртуальным машинам, связанным с этим облачным шаблоном.

3. Перейдите на вкладку Код на панели справа, чтобы просмотреть код YAML для свойств конфигурации Puppet.

При добавлении компонента Puppet в облачный шаблон в файл YAML можно добавить свойство `installMaster`, чтобы оно указывало на компонент Puppet Install Master, также называемый `Compile Master`. Значением этого свойства может быть IP-адрес или имя узла компонента Puppet Compile Master. Это свойство позволяет получить доступ к расширенным возможностям для развернутых виртуальных машин Puppet, а также обеспечивает поддержку дополнительных действий по регулярному обслуживанию.

```
Puppet_Agent:
  type: Cloud.Puppet
  properties:
    account: PEIntegrationAccount
    environment: production
    role: 'role::linux_webserver'
    host: '${CentOS-Puppet.*}'
    username: root
    password: password123!
    installMaster: my-pe-compile-master.example.com
    agentConfiguration:
      certName: '${CentOS-Puppet.address}'
    osType: linux
    count: 1
```

**Примечание** Хотя пользователь определен здесь как пользователь `root`, в облачном шаблоне может быть настроен любой пользователь, включенный в список `sudoers`.

В некоторых случаях vRealize Automation по умолчанию передает определенные сведения о компьютерах на виртуальные машины Puppet в виде фактов. Настраиваемые факты не поддерживаются для компьютеров с Windows. На компьютерах с Linux некоторая информация передается по умолчанию, а дополнительную информацию пользователи могут передавать с помощью настраиваемых свойств.

На данные, передаваемые на компьютеры Puppet, работающие под управлением Linux, накладываются определенные ограничения. На виртуальные машины Puppet передаются настраиваемые свойства ресурсов узла и агента Puppet. Настраиваемые свойства сетевых ресурсов не передаются на виртуальную машину. Передаваемые элементы включают в себя простые, логические свойства, а также настраиваемые именованные и сложные типы, например вложенные карты с массивами.

В следующем примере показано, как различные настраиваемые ресурсы могут вызываться для ресурсов узлов.

```
resources:
  Puppet-Host:
    type: Cloud.AWS.EC2.Instance
    properties:
      customer_specified_property_on_ec2_resource: "property"

customer_specified_property_on_network_resource_that_should_also_be_a_fact_and_is_boolean:
true
  CustomerNameStuff: "zone A"
  try_map:
    key: value
    keytwo: value
  nested_array:
    - one
    - two
    - true
  try_array:
    - one
    - two
    -three:
      inner_key: value
```

Если команда очистки Puppet (purge) приводит к ошибкам, то в большинстве случаев служба vRealize Automation будет игнорировать такие ошибки для узлов и выполнять операцию удаления узла. Даже если для определенного узла сертификат не будет найден, vRealize Automation продолжит выполнять операцию удаления. Если по какой-либо причине vRealize Automation не сможет продолжить операцию удаления узла, можно нажать кнопку «Удалить» в меню «Действия» на странице «Развертывания», чтобы открыть диалоговое окно, в котором будет можно продолжить операцию удаления узла. Аналогичный рабочий процесс выполняется при удалении интеграции Puppet из облачного шаблона и последующем применении шаблона к развертыванию. Этот рабочий процесс запускает операцию очистки узла, которая выполняется, как описано выше.

Для интеграции с Puppet Enterprise требуется общедоступный IP-адрес. Если для компьютера Puppet Enterprise не настроен общедоступный IP-адрес, будет использоваться IP-адрес первого сетевого адаптера.

Если сетевой адаптер компьютера, подготовленного Puppet и работающего на компьютере vSphere, имеет несколько IP-адресов, для указания IP-адреса, который нужно использовать для подключений, можно применить свойство YAML `primaryAddress` в облачных шаблонах. Если свойство `primaryAddress` назначено сетевому адаптеру, Puppet будет использовать IP-адрес этого сетевого адаптера. В качестве основного можно назначить только один сетевой адаптер. Пример использования свойства `primaryAddress` см. в следующем фрагменте кода YAML.

```
BaseVM:
  type: Cloud.vSphere.Machine
  properties:
    image: photon
    count: 2
    customizationSpec: Linux
    cpuCount: 1
    totalMemoryMB: 1024
    networks:
      - network: '${resource.dev.id}'
        deviceIndex: 0
        primaryAddress: true
        assignment: static
      - network: '${resource.prod.id}'
        deviceIndex: 1
        assignment: static
```

Если свойство `primaryAddress` не задано ни для одного сетевого адаптера виртуальной машины, логика облачных шаблонов при выборе IP-адресов по умолчанию будет соответствовать текущему поведению.

## Настройка интеграции с приложением Ansible в службе Cloud Assembly

Служба Cloud Assembly поддерживает интеграцию с Ansible — программным обеспечением с открытым исходным кодом для управления конфигурациями. Выполнив настройку данной интеграции, можно добавлять компоненты Ansible к новым и существующим развертываниям.

Интеграция приложения Ansible со службой Cloud Assembly позволяет настроить автоматический запуск одного или нескольких сценариев Ansible в определенном порядке при подготовке нового компьютера, что помогает автоматизировать управление конфигурациями. Требуемые сценарии следует указать в облачном шаблоне, который будет использоваться для развертывания.

При настройке интеграции с Ansible необходимо указать компьютер узла Ansible, а также путь к файлу иерархии, в котором определены данные, используемые для управления ресурсами. Кроме того, необходимо указать имя и пароль для доступа к экземпляру Ansible. Настройки подключения можно будет обновить позже при добавлении компонента Ansible в развертывание, если требуется задать проверку подлинности на основе ключей.

По умолчанию Ansible использует протокол SSH для подключения к физическим компьютерам. Если используются компьютеры Windows, указанные в облачном шаблоне со свойством `osType` (для Windows), для переменной `connection_type` автоматически устанавливается значение `winrm`.

Изначально в интеграции Ansible используются учетные данные в виде «пользователь/пароль» или «пользователь/ключ», указанные для подключения к управляющему компьютеру Ansible. После успешного подключения для сценариев, заданных в облачном шаблоне, выполняется проверка синтаксиса.

Если проверка прошла успешно, на управляющем компьютере Ansible создается папка выполнения по адресу `~/var/tmp/vmware/provider/user_defined_script/`. Это расположение, в котором запускаются сценарии для добавления узла в иерархию, создания файлов vars для узла, включая настройку режима проверки подлинности для подключения к узлу, и выполнение сценариев. На этом этапе учетные данные, указанные в облачном шаблоне, используются для подключения к узлу с управляющего компьютера Ansible.

Интеграция Ansible поддерживает физические компьютеры, на которых не используется IP-адрес. Для компьютеров, подготовленных в общедоступных облаках, таких как AWS, Azure и GCP, в свойство `address` для созданного ресурса общедоступный IP-адрес компьютера подставляется только в том случае, если этот компьютер подключен к общедоступной сети. Для компьютеров, не подключенных к общедоступной сети, интеграция Ansible ищет IP-адрес в сети, подключенной к компьютеру. Если подключено несколько сетей, интеграция Ansible выполняет поиск сети по наименьшему значению свойства `deviceIndex`, которое представляет собой индекс сетевого адаптера (NIC), подключенного к компьютеру. Если свойство `deviceIndex` не задано в схеме элементов, интеграция использует первую подключенную сеть.

Дополнительные сведения по настройке Ansible Open Source для интеграции в Cloud Assembly см. в разделе [Что такое управление конфигурацией в Cloud Assembly](#).

#### Необходимые условия

- На управляющем компьютере Ansible должна использоваться версия Ansible. Сведения о поддерживаемых версиях см. в [Таблице поддержки vRealize Automation](#).
- По умолчанию для уровня детализации журнала Ansible должно быть задано нулевое значение.
- Пользователь должен иметь доступ с правом чтения и записи к каталогу, в котором хранится файл иерархии Ansible. Пользователь также должен иметь доступ с правом чтения и записи к самому файлу иерархии, если он уже существует.
- Если используется учетная запись непривилегированного пользователя с параметром `sudo`, убедитесь, что в файле `sudoers` заданы следующие параметры:

```
Defaults:user_name !requiretty
```

и

```
username ALL=(ALL) NOPASSWD: ALL
```

- Убедитесь, что проверка ключа узла деактивирована, установив значение `host_key_checking = False` в `/etc/ansible/ansible.cfg` или `~/.ansible.cfg`.
- Убедитесь в наличии пароля для хранилища, добавив в файл `/etc/ansible/ansible.cfg` или `~/.ansible.cfg` следующую строку кода:

```
vault password_file = /path/to/password_file
```

Файл пароля для хранилища содержит пароль в виде обычного текста и используется только в том случае, если в облачных шаблонах или развертываниях предусмотрена комбинация имени пользователя и пароля для соединений между АСМ и узлом, как показано в следующем примере.

```
echo 'myStr0ng9@88w0rd' > ~/.ansible_vault_password.txt
echo 'ANSIBLE_VAULT_PASSWORD_FILE=~/.ansible_vault_password.txt' > ~/.profile
# Instead of this way, you can also set it setting
'vault_password_file=~/.ansible_vault_password.txt' in either /etc/ansible/ansible.cfg or
~/.ansible.cfg
```

- Чтобы избежать ошибок с ключами узла при запуске сценариев, рекомендуется включить следующие параметры в папке /etc/ansible/ansible config.

```
[paramiko_connection]
record_host_keys = False

[ssh_connection]
#ssh_args = -C -o ControlMaster=auto -o ControlPersist=60s
ssh_args = -o UserKnownHostsFile=/dev/null # If you already have any
options set for ssh_args, just add the additional option shown here at the end.
```

## Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите **Ansible**.  
Отобразится страница настройки Ansible.
3. Введите имя узла, путь к файлу иерархии и другую информацию, необходимую для настройки экземпляра Ansible.
4. Нажмите **Проверить**, чтобы проверить интеграцию.
5. Нажмите кнопку **Добавить**.

## Результаты

Приложение Ansible доступно для использования при работе с облачными шаблонами.

## Следующие шаги

Добавьте компоненты Ansible в нужные облачные шаблоны.

1. На странице холста облачного шаблона выберите Ansible в меню параметров облачного шаблона в разделе «Управление конфигурацией» и перетащите компонент Ansible на холст.
2. На панели справа можно настроить соответствующие свойства Ansible, в том числе указать сценарии, которые будут выполняться.



В Ansible пользователи могут назначать переменные одиночному узлу, а затем использовать его позже в сценариях. Интеграция Ansible Open Source позволяет задать эти переменные узлов в облачных шаблонах. Свойство `hostVariables` должно иметь корректный формат YAML, соответствующий требованиям управляющего компьютера Ansible, и эти сведения будут размещены в следующем каталоге.

```
parent_directory_of_inventory_file/host_vars/host_ip_address/vra_user_host_vars.yml
```

Расположение по умолчанию файла иерархии Ansible определяется в учетной записи Ansible на странице «Интеграции» в Cloud Assembly. Интеграция Ansible не будет проверять синтаксис YAML свойства `hostVariable` в облачном шаблоне, но при запуске сценария в случае некорректного формата или синтаксиса управляющий компьютер Ansible будет выдавать ошибку.

В следующем фрагменте YAML облачного шаблона приведен пример использования свойства `hostVariables`.

```
Cloud_Ansible_1:
  type: Cloud.Ansible
  properties:
    host: '${resource.AnsibleLinuxVM.*}'
    osType: linux
    account: ansible-CAVA
    username: ${input.username}
    password: ${input.password}
    maxConnectionRetries: 20
    groups:
      - linux_vms
  playbooks:
    provision:
      - /root/ansible-playbooks/install_web_server.yml
  hostVariables: |
    message: Hello ${env.requestedBy}
    project: ${env.projectName}
```

Интеграции Ansible предполагают наличие учетных данных для проверки подлинности в облачном шаблоне. Возможны следующие варианты.

- Имя пользователя и пароль в ресурсе Ansible.
- Имя пользователя и файл `privateKeyFile` в ресурсе Ansible.
- Имя пользователя в ресурсе Ansible и элемент `privatekey` в вычислительном ресурсе (для `remoteAccess` указано значение `generatedPublicPrivateKey`).

При создании интеграции Ansible Open Source ее пользователю необходимо предоставить учетные данные для входа, чтобы он мог подключаться к управляющему компьютеру Ansible по протоколу SSH. Для запуска сценариев с интеграцией в коде YAML интеграции можно указать другого пользователя. Свойство `username` является обязательным и требуется для подключения к виртуальной машине, на которой Ansible будет вносить изменения. Свойство `playbookRunUsername` является необязательным и может быть предоставлено для выполнения сценария на узле Ansible. Значением параметра `playbookRunUsername` по умолчанию является имя пользователя интеграции конечной точки Ansible.

Если указать другого пользователя, у такого пользователя должен быть доступ на запись к файлу узлов Ansible, а также ему необходимо разрешение на создание файлов закрытых ключей.

При добавлении плитки Ansible Open Source в облачный шаблон vRealize Automation создает запись об узле для подключенной виртуальной машины. По умолчанию vRealize Automation будет использовать имя ресурса виртуальной машины для создания записи об узле, но с помощью свойства `hostName` в коде YAML схемы элементов можно указать любое имя. Для обеспечения связи с компьютером служба vRealize Automation создаст переменную узла `ansible_host: IP Address` для записи об узле. Поведение по умолчанию можно переопределить и настроить вариант связи с использованием полного доменного имени, указав ключевое слово `ansible_host` в разделе `hostVariables` и введя полное доменное имя в качестве его значения. В следующем фрагменте кода YAML приведен пример настройки связи с использованием имени узла и полного доменного имени.

```
Cloud_Ansible:
  type: Cloud Ansible
  properties:
    osType: linux
    username: ubuntu
    groups:
      - sample
    hostName: resource name
    host: name of host
    account: name of account
    hostVariables:
      ansible_host:Host FQDN
```

В этом примере выполняется переопределение значения `ansible_host` по умолчанию путем указания полного доменного имени. Это может быть полезно для пользователей, которые хотят, чтобы служба Ansible Open Source подключалась к компьютеру узла с помощью полного доменного имени.

Значением по умолчанию для параметра `hostVariables` в коде YAML будет `ansible_host:IP_address`, а IP-адрес используется для связи с сервером.

Если свойство `count` в YAML для Ansible Open Source имеет значение больше 1, имя узла можно сопоставить с любым из соответствующих свойств виртуальной машины. В следующем примере показано сопоставление для ресурса виртуальной машины с именем Ubuntu-VM, если нужно сопоставить ее свойство `address` с именем узла.

```
hostname: '${resource.Ubuntu-VM.address[count.index]}'
```

Убедитесь, что путь к сценарию Ansible в облачных шаблонах доступен пользователю, указанному в учетной записи интеграции. Можно использовать абсолютный путь для указания расположения сценария, но это необязательно. Абсолютный путь к домашней папке пользователя рекомендуется использовать для того, чтобы путь оставался действительным даже в случае последующего изменения учетных данных интеграции Ansible.

## Настройка интеграции Ansible Tower в Cloud Assembly

Ansible Tower можно интегрировать с Cloud Assembly для управления конфигурацией развернутых ресурсов. После настройки интеграции виртуальные компоненты Ansible Tower можно добавлять в новые или существующие развертывания в редакторе облачных шаблонов.

### Необходимые условия

- Предоставьте пользователям, не являющимся администраторами, соответствующие разрешения на доступ к Ansible Tower. Существует два варианта, которые можно использовать в большинстве конфигураций. Выберите наиболее подходящий для своей конфигурации.
  - Предоставьте пользователям роли «Администратор иерархии» и «Администратор шаблонов заданий» на уровне организации.
  - Предоставьте пользователям права администратора на определенную иерархию и роль «Выполнение» для всех шаблонов заданий, используемых для предоставления ресурсов.
- Для использования с развертываниями в Ansible Tower необходимо настроить соответствующие учетные данные и шаблоны. Можно использовать шаблоны заданий или рабочих процессов. Шаблоны заданий определяют иерархию и сценарий, используемые для развертывания. Задания и сценарии сопоставляются один к одному. Для определения задач, связанных с шаблоном, в сценариях используется синтаксис, похожий на YAML. В большинстве типовых развертываний для проверки подлинности используйте учетные данные компьютера.

Шаблоны рабочих процессов позволяют пользователям создавать последовательности, состоящие из любой комбинации шаблонов заданий, результатов синхронизации проекта и синхронизации иерархии, которые связаны друг с другом, что позволяет выполнять их единым блоком. Благодаря средству Ansible Tower Workflow Visualizer пользователи могут проектировать шаблоны рабочих процессов. В большинстве типовых развертываний для проверки подлинности можно использовать учетные данные компьютера.

а) Войдите в Ansible Tower и перейдите в раздел «Шаблоны».

б) Выберите «Добавить новый шаблон задания».

- Выберите созданные вами учетные данные. Это учетные данные компьютера, для управления которыми будет использоваться Ansible Tower. Для каждого шаблона задания может существовать один объект учетных данных.
- Для выбора ограничения выберите «Запрос при запуске». Это гарантирует, что шаблон задания будет выполняться для узла, процесс подготовки или отмены подготовки которого выполняется из Cloud Assembly. Если этот параметр не выбран, то при развертывании схемы элементов, содержащей шаблон задания, будет выводиться сообщение об ошибке «Ограничение не установлено».

в) Выберите «Добавить новый шаблон рабочего процесса».

- Выберите созданные ранее учетные данные, а затем определите иерархию. С помощью Workflow Visualizer спроектируйте шаблон рабочего процесса.

В поле «Ограничение» для шаблонов рабочих процессов или заданий обычно можно выбрать вариант «Запрос при запуске». В этом случае шаблон рабочих процессов или заданий будет выполняться для узла, процесс подготовки или отмены подготовки которого выполняется в Cloud Assembly.

- Выполнение шаблонов заданий или шаблонов рабочих процессов, вызываемых из Cloud Assembly, можно просматривать на вкладке «Задания Ansible Tower».

#### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Нажмите **Ansible Tower**.  
Отобразится страница настройки Ansible.
3. Введите **имя узла**, которое может быть IP-адресом, и другую необходимую информацию для экземпляра Ansible Tower.
4. Введите **имя пользователя и пароль**, применяемые для проверки подлинности с помощью пользовательского интерфейса, для соответствующего экземпляра Ansible Tower.
5. Нажмите **Проверить**, чтобы проверить интеграцию.
6. Введите соответствующее **имя и описание** для интеграции.
7. Нажмите кнопку **Добавить**.

#### Результаты

Приложение Ansible Tower доступно для использования в облачных шаблонах.

#### Следующие шаги

Добавьте компоненты Ansible Tower в нужные облачные шаблоны. Для пользователя, указанного в учетной записи интеграции, необходимо определить применимый шаблон задания с разрешением на выполнение.

1. На странице холста облачного шаблона выберите Ansible в разделе «Управление конфигурацией» в меню параметров схемы элементов и перетащите компонент Ansible Tower на холст.
2. На панели справа можно настроить соответствующие свойства Ansible Tower, например шаблоны заданий.

При добавлении плитки Ansible Tower в облачный шаблон vRealize Automation создает запись об узле для подключенной виртуальной машины в Ansible Tower. По умолчанию vRealize Automation будет использовать имя ресурса виртуальной машины для создания записи об узле, но с помощью свойства `hostName` в коде YAML схемы элементов можно указать любое имя. Для обеспечения связи с компьютером служба vRealize Automation создаст переменную узла `ansible_host`: IP Address для записи об

узле. Поведение по умолчанию можно переопределить и настроить вариант связи с использованием полного доменного имени, указав ключевое слово `ansible_host` в разделе `hostVariables` и введя полное доменное имя в качестве его значения. В следующем фрагменте кода YAML приведен пример настройки связи с использованием имени узла и полного доменного имени.

```
Cloud_Ansible_Tower_1:
  type: Cloud Ansible Tower
  properties:
    host: name of host
    account: name of account
    hostName: resource name
    hostVariables:
      ansible_host: Host FQDN
```

В этом примере выполняется переопределение значения `ansible_host` по умолчанию путем указания полного доменного имени. Это может быть полезно для пользователей, которые хотят, чтобы служба Ansible Tower подключалась к компьютеру узла с помощью полного доменного имени.

Значением по умолчанию для параметра `hostVariables` в коде YAML будет `ansible_host:IP_address`, а IP-адрес используется для связи с сервером.

Если свойство `count` в YAML для Ansible Tower имеет значение больше 1, имя узла можно сопоставить с любым из соответствующих свойств виртуальной машины. В следующем примере показано сопоставление для ресурса виртуальной машины с именем Ubuntu-VM, если нужно сопоставить ее свойство `address` с именем узла.

```
hostname: '${resource.Ubuntu-VM.address[count.index]}'
```

При добавлении компонента Ansible Tower в облачный шаблон в коде YAML этого шаблона можно указать шаблон задания, который будет вызываться. Также можно указать шаблоны рабочих процессов или комбинацию шаблонов заданий и шаблонов рабочих процессов. Если тип шаблона не указан, vRealize Automation по умолчанию предполагает, что вы вызываете шаблон задания.

В следующем фрагменте кода YAML приведен пример вызова комбинации шаблонов задания и рабочего процесса в облачном шаблоне Ansible Tower.

```
Cloud_Ansible_1:
  type: Cloud.Ansible.Tower
  properties:
    host: '${resource.CentOS_Machine.*}'
    account:
    maxConnectionRetries: 2
    maxJobRetries: 2
    templates:
      provision:
        - name: My workflow
          type: workflow
        - name: My job template
```

Для обработки ошибок, связанных с Ansible, мы добавили параметры `maxConnectionsRetries` и `maxJobRetries`. Облачный шаблон принимает настраиваемое значение. Если значение не указано, шаблон будет использовать значение по умолчанию. Значение по умолчанию для `maxConnectionRetries` — 10, а значение по умолчанию для `maxJobRetries` — 3.

**Примечание** Предыдущие версии службы vRealize Automation поддерживали выполнение шаблонов заданий только с помощью схемы `jobTemplate` в облачном шаблоне. Теперь схема `jobTemplate` признана устаревшей и, возможно, будет удалена в будущих выпусках. В текущем выпуске свойство `jobTemplate` работает как раньше. Для запуска шаблонов рабочих процессов и использования дополнительных функций рекомендуется применять схему шаблонов.

Облачные шаблоны Cloud Assembly для интеграций Ansible Tower содержат свойство `useDefaultLimit` со значением «истина» или «ложь», которое позволяет определить, где выполняются шаблоны Ansible. Шаблоны Ansible могут быть шаблонами заданий или шаблонами рабочих процессов. Если задано значение «истина», указанные шаблоны будут запускаться для компьютера, указанного в поле «Ограничение» на странице «Шаблоны Ansible». Если задано значение «ложь», шаблоны запускаются для подготовленного компьютера, но пользователи должны установить флажок «Запрос при запуске» на странице «Шаблоны Ansible Tower». По умолчанию это свойство имеет значение «ложь». В следующем примере кода YAML показано, как свойство `useDefaultLimit` отображается в облачных шаблонах.

```
templates:
  provision:
    - name: ping aws_credentials
      type: job
      useDefaultLimit: false
      extraVars: '{"rubiconSurveyJob" : "checkSurvey"}'
```

Кроме того, как показано в примере выше, свойство `extraVars` можно использовать для указания дополнительных переменных или переменных опроса. Такая возможность может быть полезна для запуска шаблонов, для которых требуется ввод данных. Если пользователь использует переменную опроса, то во избежание ошибок эта переменная должна передаваться в разделе `extraVars` облачного шаблона.

## Создание интеграции SaltStack Config в vRealize Automation

Для доступа к службе SaltStack Config и использования объектов и действий SaltStack Config в vRealize Automation можно создать интеграцию SaltStack Config.

Благодаря применению vRealize Automation SaltStack Config можно подготавливать, настраивать и развертывать программное обеспечение (ПО) на виртуальных машинах в среде любого масштаба в режиме автоматизации на основе событий. Кроме того, SaltStack Config можно использовать для определения оптимальных состояний ПО, соответствующих нормативным требованиям, и их принудительного применения в рамках всей среды.

### Установка

Прежде чем интегрировать SaltStack Config с vRealize Automation, необходимо сначала установить его в своей среде. Дополнительные сведения см. в разделе [Установка и настройка SaltStack Config](#).

## Примечания

Интегрированный экземпляр vRealize Automation SaltStack Config доступен для vRealize Automation при следующих условиях:

- Интеграция SaltStack Config связывается с конкретным узлом во время установки.
- В настоящее время vRealize Automation не поддерживает мультитенантность для SaltStack Config.
- Арендатор vRealize Automation может поддерживать одну интеграцию SaltStack Config и один главный сервер Salt. Главный сервер Salt может поддерживать несколько служебных серверов.
- Прежде чем удалить интеграцию SaltStack Config в vRealize Automation, необходимо удалить все существующие развертывания, в которых используется эта интеграция SaltStack Config.

## Необходимые условия

- Убедитесь в наличии учетных данных администратора vRealize Automation и SaltStack Config (для доступа с правами root).

Учетные данные администратора vRealize Automation и SaltStack Config (для доступа с правами root) необходимы для создания интеграции SaltStack Config.

Учетные данные администратора SaltStack Config также требуются для открытия службы SaltStack Config и работы в ней.

Учетные данные vRealize Automation используются для доступа к vRealize Automation, а учетные данные SaltStack Config — для доступа к SaltStack Config.

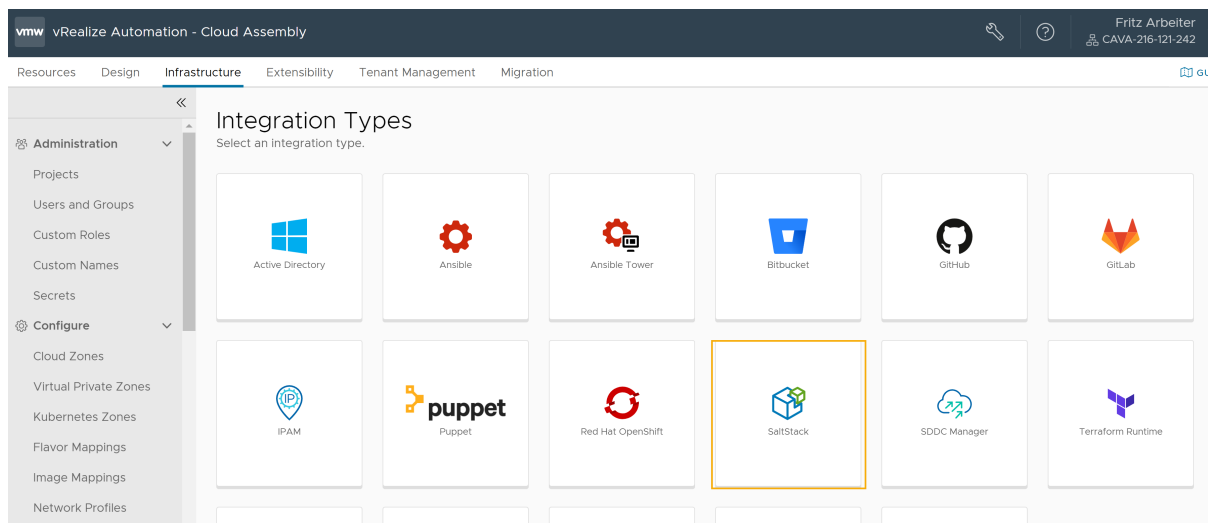
Сведения об учетных данных администратора SaltStack Config см. в руководстве [Установка и настройка SaltStack Config](#).

- Убедитесь, что служба SaltStack Config установлена.
- Убедитесь, что главный сервер Salt, который будет использоваться в интеграции SaltStack Config, содержит подключаемый модуль Master.
- Убедитесь, что вам назначена роль администратора службы SaltStack Config в vRealize Automation. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что вам назначена роль администратора службы Cloud Assembly в vRealize Automation. См. раздел [Роли пользователей организаций и служб в vRealize Automation](#).

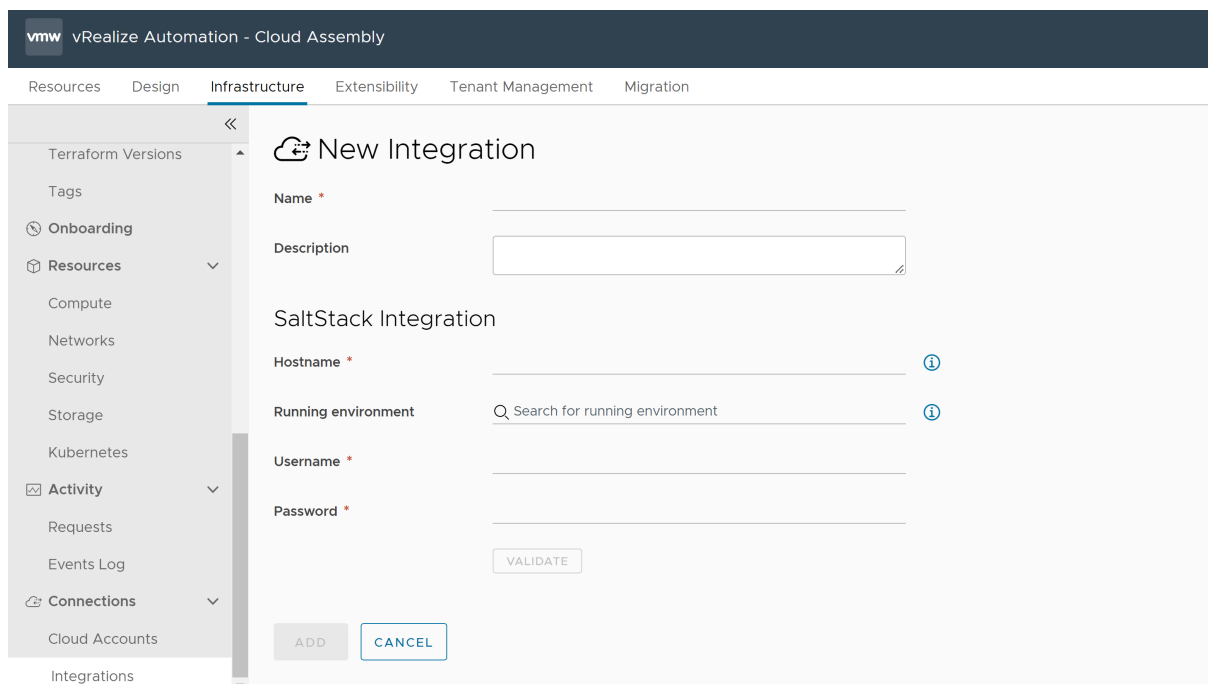
## Настройка интеграции SaltStack Config в службе vRealize Automation

После установки SaltStack Config для vRealize Automation можно настроить интеграцию в Cloud Assembly.

1. В Cloud Assembly выберите **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите тип интеграции SaltStack Config.



### 3. Заполните форму.



- а) Введите имя интеграции.
- б) (Необязательно.) Введите описание для интеграции.
- в) Введите имя узла для сервера SaltStack Config.
- г) Укажите запущенную среду для интеграции SaltStack Config.

Если для развертывания служебных серверов и применения файлов состояния на виртуальных машинах используется свойство `saltConfiguration`, настраивать запущенную среду не нужно. Для использования ресурса SaltStack Config рекомендуется обновить облачные шаблоны. В следующем выпуске свойство `saltConfiguration` уже не будет поддерживаться.



Если для развертывания служебных серверов и применения файлов состояния на виртуальных машинах используется ресурс SaltStack Config, выберите запущенную среду **embedded-ABX-onprem**.

- д) Введите имя администратора SaltStack Config и пароль, используемые для доступа к указанному узлу.
- е) Щелкните **Проверить**, чтобы подтвердить доступ администратора к узлу интеграции SaltStack Config.

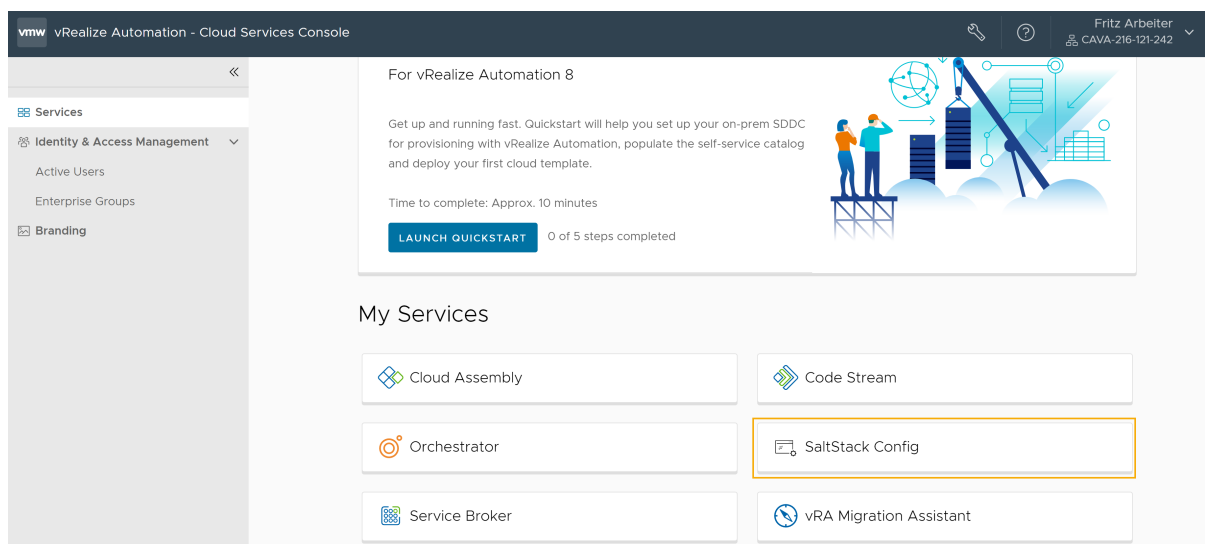
Если проверка завершается сбоем, убедитесь, что введены правильные имя узла, имя пользователя и пароль.

- ж) Нажмите **Сохранить**.

## Доступ к интеграции SaltStack Config

После сохранения точки интеграции SaltStack Config можно открыть службу интеграции SaltStack Config.

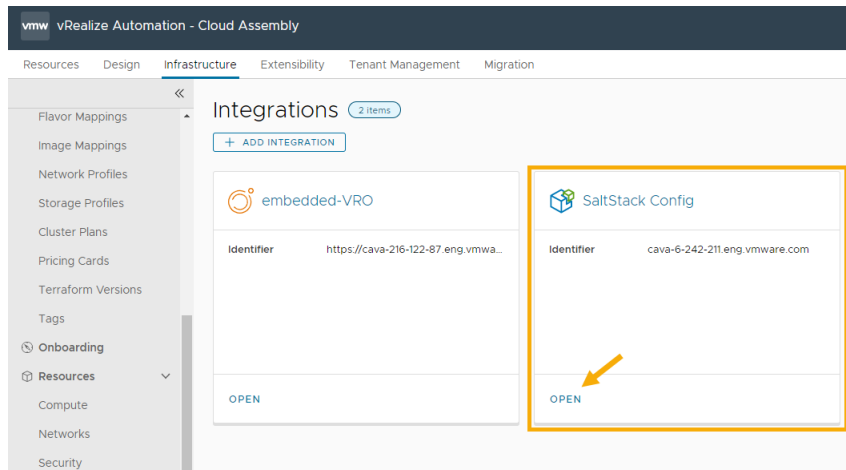
- В случае развертывания SaltStack Config посредством vRealize Suite Lifecycle Manager можно щелкнуть плитку службы в служебной консоли vRealize Automation, чтобы открыть интеграцию и получить доступ к узлу.



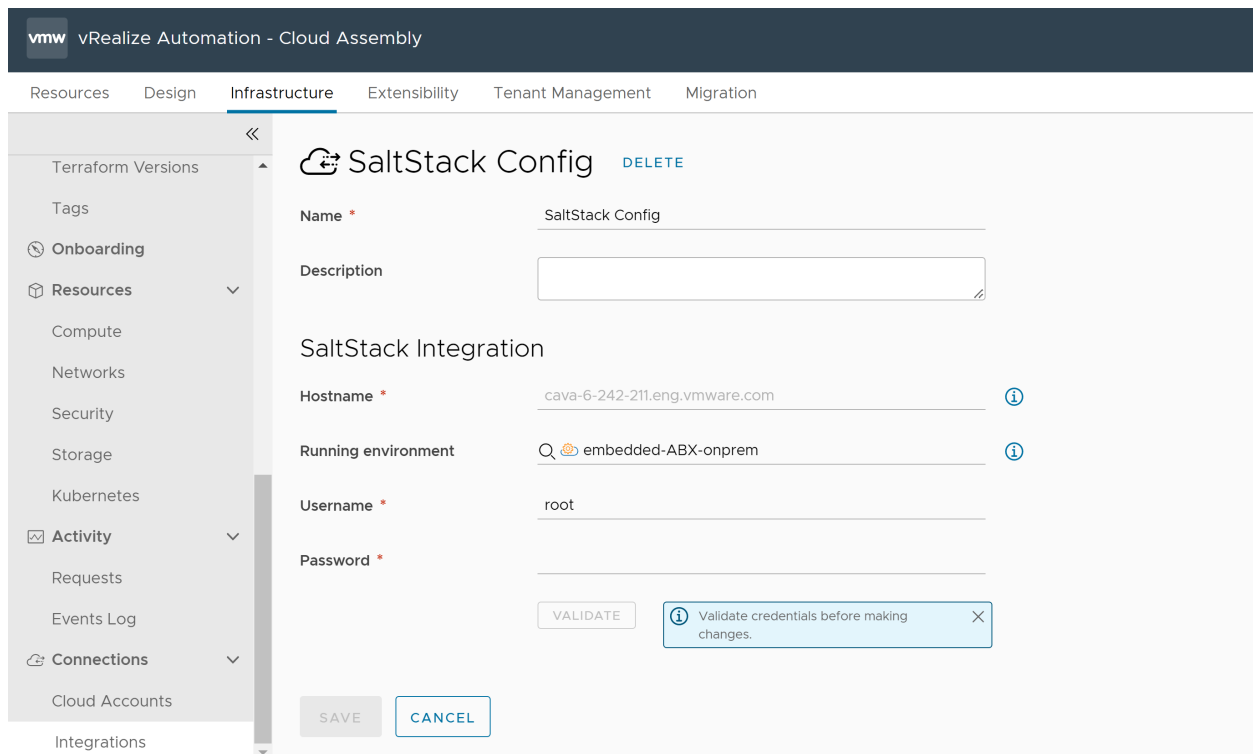
Если выполнена автономная установка SaltStack Config, для доступа к службе можно использовать имя узла SaltStack Config.

- При появлении запроса на вход в службу SaltStack Config введите имя администратора и пароль SaltStack Config.

Чтобы внести какие-либо изменения в интеграцию, выберите **Инфраструктура > Подключения > Интеграции**, нажмите доступную плитку интеграции SaltStack Config, а затем кнопку **Открыть**.



После настройки интеграции имя узла изменить нельзя. Для интеграции можно изменить только имя, описание, запущенную среду и учетные данные.



## Дополнительные сведения об использовании SaltStack Config

SaltStack Config — это автономный продукт, который можно интегрировать со службой vRealize Automation и использовать в ней.

- Ознакомьтесь с процедурой добавления [ресурса SaltStack Config](#) для установки служебных серверов на виртуальных машинах в развертываниях Cloud Assembly.
- Ознакомьтесь с процедурой [развертывания служебных серверов с помощью API-интерфейса \(RaaS\)](#) в среде Linux или Windows.

## Создание интеграции Active Directory в Cloud Assembly

Cloud Assembly поддерживает интеграцию с серверами Active Directory, чтобы поддерживать готовый режим создания учетных записей компьютеров в указанном организационном подразделении (OU) на сервере Active Directory до подготовки виртуальной машины. Active Directory поддерживает подключение LDAP к серверу Active Directory.

Политика Active Directory, связанная с проектом, применяется ко всем виртуальным машинам, которые были подготовлены в рамках этого проекта. Пользователи могут указать один или несколько тегов для выборочного применения политики к виртуальным машинам, которые подготовлены для облачных зон с соответствующими тегами возможностей.

Для локальных развертываний интеграция Active Directory позволяет настроить функцию проверки работоспособности, которая показывает состояние интеграции, и базовую интеграцию ABX, на которой она основана, включая требуемый облачный прокси-сервер расширяемости. Прежде чем применить политику Active Directory, Cloud Assembly проверяет состояние базовых интеграций. Если интеграция исправна, Cloud Assembly создает развернутые объекты компьютера в заданном каталоге Active Directory. Если интеграция неисправна, процедура развертывания пропускает этап Active Directory во время предоставления.

### Необходимые условия

- Для интеграции Active Directory требуется подключение LDAP к серверу Active Directory.
- При настройке интеграции Active Directory с vCenter в локальной среде необходимо настроить интеграцию ABX с облачным прокси-сервером расширяемости. Выберите **Расширяемость > Действия > Интеграции**, а затем **Действия расширяемости, локальн.**
- При настройке интеграции Active Directory в облачной среде необходимо иметь учетную запись Microsoft Azure или Amazon Web Services.
- Необходимо иметь настроенный проект с соответствующими облачными зонами, а также сопоставления образов и конфигураций ресурсов, которые следует использовать с интеграцией Active Directory.
- Прежде чем связать интеграцию Active Directory с проектом, необходимо предварительно создать нужное подразделение в Active Directory.

### Процедура

1. Выберите **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Щелкните **Active Directory**.
3. На вкладке **Сводка** введите соответствующие имена узлов LDAP и среды.

Указанный узел LDAP используется для проверки интеграции Active Directory, а также для последующих развертываний, если альтернативные узлы не указаны или не вызываются из-за ошибок или недоступности.

4. Введите имя пользователя и пароль для сервера LDAP.

- Введите соответствующее базовое DN, которое указывает корневой элемент для ресурсов Active Directory.

---

**Примечание** Для каждой интеграции Active Directory можно указать только одно различающееся имя.

---

- Щелкните **Проверить**, чтобы проверить работоспособность интеграции.
- Введите название и описание интеграции.
- Нажмите **Сохранить**.
- Перейдите на вкладку **Проект**, чтобы добавить проект в интеграцию Active Directory.

В диалоговом окне **Добавление проектов** необходимо выбрать имя проекта и относительное различающееся имя, представляющее собой различающееся имя, которое существует в базовом различающемся имени, указанном на вкладке «Сводка».

- В разделе «Расширенные параметры» укажите разделенный запятыми список **альтернативных узлов**, которые будут использоваться, если изначально выбранный сервер окажется недоступным во время развертывания. Для начальной проверки интеграции всегда используется основной сервер.

---

**Примечание** Если для основного узла используется формат LDAP, то для альтернативных узлов формат LDAPS не поддерживается.

---

- В поле **Время ожидания подключения** введите время ожидания ответа начального сервера перед попыткой использования альтернативного сервера (в секундах).
- Нажмите **Сохранить**.

## Результаты

Теперь можно связать проект, для которого настроена интеграция Active Directory, с облачным шаблоном. Если компьютер подготовлен с помощью этого облачного шаблона, он предварительно сохраняется в указанном каталоге Active Directory и организационном подразделении.

Изначально развертывание интеграций Active Directory проводится в организационной единице по умолчанию с небольшими ограничениями для пользователей. Организационная единица задается по умолчанию при сопоставлении интеграции Active Directory с проектом. Чтобы изменить организационную единицу для развертываний Active Directory, в схемы элементов можно добавить свойство `FinalRelativeDN`. Это свойство позволяет указать организационную единицу, которая будет использоваться для развертывания Active Directory.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: CenOS8
```

```

flavor: tiny
activeDirectory:
  finalRelativeDN: ou=test
  securityGroup: TestSecurityGroup

```

Как показано в предыдущем примере кода YAML, пользователи могут добавить в развертывание интеграции Active Directory свойство, которое позволяет добавить учетную запись компьютера в группу безопасности для назначения соответствующих разрешений с целью доступа к общему ресурсу по сети. Развертывание виртуальной машины Active Directory изначально проводится в заданной организационной единице, но когда машина готова к выпуску, она перемещается в другую организационную единицу с соответствующей политикой, применимой к пользователям.

Если после развертывания учетная запись компьютера перемещается в другую организационную единицу, Cloud Assembly предпринимается попытка удалить учетные записи в исходной организационной единице. Учетные записи компьютеров успешно удаляются только в том случае, если виртуальные машины перемещены в другую организационную единицу в пределах того же самого домена.

Также можно применить проверку работоспособности на основе тегов для локальных интеграций Active Directory, как показано ниже.

1. Создайте интеграцию Active Directory, как описано выше.
2. Перейдите на вкладку **Проект**, чтобы добавить проект в интеграцию Active Directory.
3. В диалоговом окне «Добавление проектов» выберите имя проекта и относительное различающееся имя. Относительное различающееся имя должно существовать в указанном базовом различающемся имени.

В этом диалоговом окне есть два переключателя, которые позволяют управлять конфигурацией Active Directory с помощью облачных шаблонов. Оба переключателя по умолчанию отключены.

- **Переопределить.** Этот переключатель позволяет переопределить свойства Active Directory, в частности, относительное различающееся имя в облачных шаблонах. Если данный переключатель включен, можно изменить организационное подразделение, указанное в свойстве `relativeDN` в облачном шаблоне. После подготовки компьютер будет добавлен в подразделение, указанное в свойстве `relativeDN` в облачном шаблоне. В следующем примере показана иерархия облачного шаблона, в которой появляется данное свойство.

```

activeDirectory:
  relativeDN: OU=ad_integration_machine_override

```

- **Игнорировать.** Этот переключатель позволяет игнорировать конфигурацию Active Directory для проекта. Если данный переключатель включен, он добавляет свойство в облачный шаблон с именем `ignoreActiveDirectory` для связанной виртуальной машины. Если для этого свойства задано значение `true`, это означает, что компьютер не добавляется в Active Directory при развертывании.
4. Добавьте соответствующие теги. Эти теги применимы к облачной зоне, к которой может применяться политика Active Directory.
  5. Щелкните «Сохранить».

Состояние интеграции Active Directory отображается для каждой интеграции на странице **Инфраструктура > Подключения > Интеграции** в Cloud Assembly.

Проект с интеграцией Active Directory можно связать с облачным шаблоном. Если компьютер подготовлен с помощью этого шаблона, он предварительно сохраняется в указанном каталоге Active Directory и организационном подразделении.

## Настройка интеграции VMware SDDC Manager

Чтобы упростить использование доменов рабочей нагрузки в рамках облачных учетных записей VMware Cloud Foundation (VCF) в vRealize Automation, в vRealize Automation можно добавить интеграцию с VMware SDDC Manager.

### Необходимые условия

- В vRealize Automation поддерживается интеграция только с VMware SDDC Manager версии 4.1 и более поздних.

### Процедура

1. Выберите пункт **Инфраструктура > Подключения > Интеграции** и нажмите **Добавить интеграцию**.
2. Выберите SDDC Manager.

Откроется страница «Конфигурация интеграции SDDC Manager».

3. В разделе «Сводка» введите **имя** и **описание** для интеграции.
4. В разделе «Учетные данные SDDC Manager» введите **IP-адрес** или **полное доменное имя SDDC Manager** для компьютера сервера SDDC Manager.
5. Введите имя пользователя и пароль для учетной записи администратора, которая будет использоваться для первоначального подключения к SDDC Manager. Рекомендуется не использовать для подключения учетную запись администратора. Для создания ролей службы используйте другую учетную запись с правами администратора в SDDC Manager.

Эти учетные данные используются для первоначальной настройки подключения к SDDC Manager, а затем создаются учетные данные службы, применяемые при подключении из облачной учетной записи VCF.

6. Щелкните **Проверить**, чтобы проверить подключение к SDDC Manager.
7. Нажмите кнопку **Добавить**.

### Результаты

После создания интеграции можно просмотреть рабочие нагрузки, связанные с программно-определяемым ЦОД (SDDC), на вкладке «Домен рабочей нагрузки», которая отображается на странице выполненной интеграции. Кроме того, можно просмотреть и выбрать рабочие нагрузки, связанные с интеграцией, а затем нажать кнопку **Добавить облачную учетную запись**, чтобы открыть страницу для создания облачной учетной записи VCF, которая будет использовать выбранную рабочую нагрузку.

## Следующие шаги

После настройки облачной учетной записи VCF в верхней части страницы появится кнопка **Настройка облачной среды**. Нажмите эту кнопку, чтобы запустить мастер настройки облачной среды VCF.

## Интеграция с vRealize Operations Manager

vRealize Automation может взаимодействовать с vRealize Operations Manager для выполнения расширенных операций размещения рабочих нагрузок, предоставления показателей проверки работоспособности развертывания и виртуальных машин, а также отображения цен.

### Количество и тип интеграций

Интеграция между двумя продуктами должна быть «локальная — локальная», а не комбинацией локальной и облачной среды.

Можно интегрировать один экземпляр vRealize Automation с несколькими экземплярами vRealize Operations Manager, но экземпляр vRealize Operations Manager можно подключить только к одному экземпляру vRealize Automation.

Объединенный кластер vRealize Operations Manager нельзя подключить к vRealize Automation.

### Основные требования для интеграции

Для интеграции с vRealize Operations Manager выберите **Инфраструктура > Подключения > Интеграции**. Чтобы добавить интеграцию, требуются URL-адрес и учетные данные vRealize Operations Manager для учетной записи, описанной в следующем разделе. Кроме того, vRealize Automation и vRealize Operations Manager должны управлять одной и той же конечной точкой vSphere.

### Учетная запись входа для интеграции

В vRealize Operations Manager необходимо иметь локальную или внешнюю учетную запись vRealize Operations Manager для интеграции. Для учетной записи требуются права «только чтение» на доступ к экземпляру адаптера vCenter для конечной точки vSphere. Возможно, потребуется импортировать внешнюю учетную запись в vRealize Operations Manager и назначить ей роль «только чтение». Для интеграции формат имени пользователя для внешней учетной записи имеет следующий вид: *имя\_пользователя@домен@источник с проверкой подлинности*, например *jdoe@company.com@workspaceone*. Источники с проверкой подлинности определяются во время начальной настройки сервера vRealize Operations Manager.

Дополнительные сведения см. в следующих разделах. Сведения о ценообразовании см. в разделе [Использование карт ценообразования в vRealize Automation](#).

## Расширенное размещение рабочих нагрузок с помощью vRealize Operations Manager

vRealize Automation и vRealize Operations Manager могут взаимодействовать для оптимального размещения рабочих нагрузок развертывания.

Размещение рабочей нагрузки активируется на уровне облачной зоны на основе vSphere.

Усовершенствованное размещение с помощью vRealize Operations Manager возможно только в кластерах, поддерживающих компонент Distributed Resource Scheduler (DRS) и входящих в облачную зону.

- **Размещение vRealize Automation.** Модуль размещения vRealize Automation основан на цели применения. Он учитывает ограничения на основе тегов, членство в проекте и связанные облачные зоны, а также фильтры сходства, относящиеся к сети, хранилищу и вычислительным ресурсам. Размещение ресурсов зависит от всех этих факторов, а также от наличия других связанных целевых ресурсов в том же развертывании.

- **Размещение vRealize Operations Manager.** Для оптимального размещения vRealize Operations Manager учитывает эксплуатационную задачу. Эксплуатационная задача может учитывать как прошлые, так и будущие рабочие нагрузки, а также результаты анализа «что если».

При использовании расширенного размещения рабочей нагрузки необходимо применить тегирование vRealize Automation, чтобы реализовать принятие решений на основе бизнес-задач, а не использовать бизнес-варианты для vRealize Operations Manager.

При интеграции с vRealize Operations Manager служба vRealize Automation продолжает учитывать цель применения и связанные с ней ограничения для фильтрации целевых размещений. Затем на основании полученных результатов она использует рекомендации vRealize Operations Manager для дальнейшей оптимизации размещения.

### Если рекомендации отсутствуют

Если активна функция оптимизированного размещения рабочей нагрузки и средство анализа vRealize Operations Manager не выдает никаких рекомендаций, можно настроить vRealize Automation для возврата к размещению на основе цели применения (режим по умолчанию).

### Ограничения на размещение рабочих нагрузок

При использовании vRealize Operations Manager для размещения рабочих нагрузок применяются определенные ограничения.

- vRealize Operations Manager не поддерживает размещение рабочих нагрузок в пулах ресурсов в vCenter Server.
- Если vRealize Operations Manager отключен, интервал времени ожидания вызова vRealize Operations Manager для размещения рабочих нагрузок может истечь.
- Размещение не включает в себя несколько облачных зон. vRealize Automation отправляет одну облачную зону в vRealize Operations Manager, чтобы получить рекомендации по размещению в этой облачной зоне.

### Включение размещения рабочей нагрузки

Чтобы включить размещение рабочей нагрузки, необходимо выполнить определенные действия для vSphere, vRealize Operations Manager и vRealize Automation.

1. В Cloud Assembly подключитесь к своей облачной учетной записи vCenter Server.



Для доступа к параметрам выберите **Инфраструктура > Подключения > Облачные учетные записи**.

2. В vCenter Server убедитесь, что кластеры с поддержкой DRS существуют и настроены на функционирование в полностью автоматизированном режиме.
3. В vRealize Operations Manager убедитесь, что обеспечивается управление одним и тем же vCenter Server.

Требуется vRealize Operations Manager 8 или более поздней версии.

4. В Cloud Assembly добавьте интеграцию vRealize Operations Manager.

Для доступа к параметрам выберите **Инфраструктура > Подключения > Интеграции**.

Чтобы добавить интеграцию, необходимо указать URL-адрес основного узла vRealize Operations Manager, а также имя пользователя и пароль для входа.

`https://operations-manager-IP-address-or-FQDN/suite-api`

После ввода значений щелкните **ПРОВЕРИТЬ**.

5. Для синхронизации интеграции с vCenter Server нажмите кнопку **СИНХРОНИЗИРОВАТЬ**.

Также синхронизируйте любое время, когда Cloud Assembly и vRealize Operations Manager начинают управлять новым vCenter Server.

6. В Cloud Assembly создайте облачную зону для учетной записи vCenter Server.

Для доступа к параметрам выберите **Инфраструктура > Настроить > Облачные зоны**.

7. На вкладке «Сводка» для облачной зоны установите для политики размещения значение **ДОПОЛНИТЕЛЬНО**.
8. В разделе «Политика размещения» укажите, должна ли vRealize Automation возвращаться в размещение по умолчанию, если vRealize Operations Manager не возвращает рекомендации.

### Устранение проблем при размещении рабочей нагрузки

Если рекомендации vRealize Operations Manager по размещению рабочих нагрузок не соответствуют вашим ожиданиям, проверьте данные в запросе на развертывание в Cloud Assembly или vRealize Automation Service Broker.

1. Выберите **Инфраструктура > Действия > Запросы** и щелкните запрос.
2. В разделе «Сведения о запросе» просмотрите этапы выделения.  
Найдите целевые объекты, которые были успешно или неудачно идентифицированы.
3. В разделе «Сведения о запросе» в правом верхнем углу включите «Режим разработки».
4. Используя путь к запросу, найдите блоки фильтра.

## 5. Щелкните блок фильтра и просмотрите следующий раздел.

```
filterName: ComputePlacementPolicyAffinityHostFilter
  v computeLinksBefore
  v computeLinksAfter
  v filteredOutHostsReasons
```

Запись	Описание
computeLinksBefore	Список возможных узлов размещения на основе алгоритмов vRealize Automation.
computeLinksAfter	Выбранный узел размещения.
filteredOutHostsReasons	Сообщения, описывающие причины выбора или отклонения узла. Если vRealize Operations Manager выбирает узел, появляется следующее сообщение. advance policy filter: Filtered hosts based on recommendation from vROPS.

### Дополнительные сведения о размещении рабочих нагрузок

Чтобы найти оптимальную инфраструктуру для размещения развертывания, в vRealize Automation принимается несколько решений о фильтрации. Интеграция vRealize Automation с vRealize Operations Manager может обеспечить дополнительное уточнение решения о размещении.

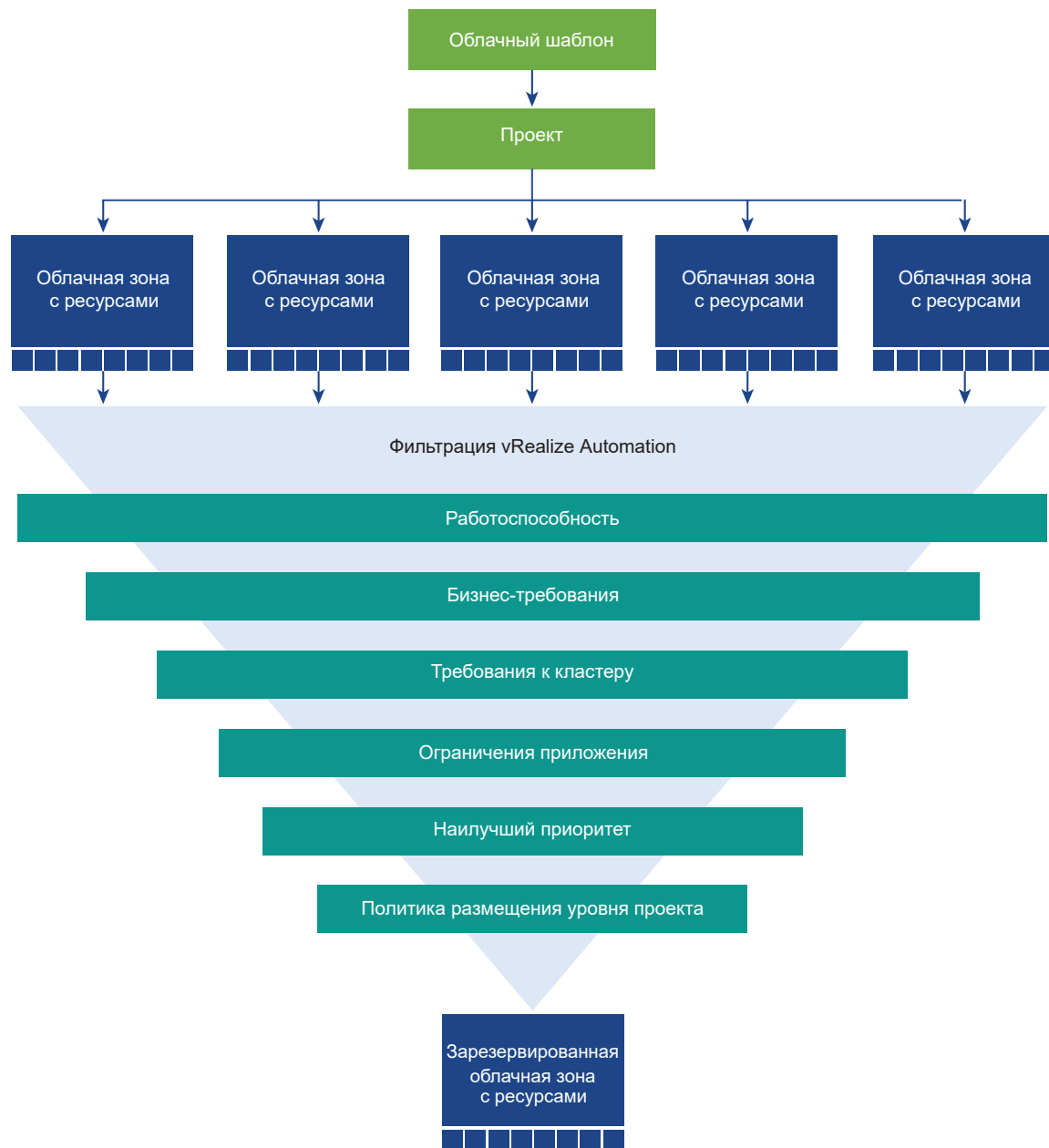
vRealize Operations Manager может помочь оптимально разместить рабочие нагрузки при условии, что в облачных зонах на основе vSphere включен параметр «Расширенная политика размещения».

Кроме того, облачные учетные записи vSphere для облачных зон должны отслеживаться vRealize Operations Manager.

### Фаза 1. Резервирование

**Примечание** Хотя название такое же, резервирование не связано с функцией резервирования в vRealize Automation 7.

В vRealize Automation фаза резервирования одинаковая независимо от того, включено ли расширенное размещение в vRealize Operations Manager.



1. Резервирование начинается со связывания облачного шаблона с проектом. Проект в свою очередь связан с облачными зонами.
2. Облачные зоны состоят из узлов, пулов и кластеров вычислительных ресурсов, а также подключенного хранилища.

Изначально любая облачная зона в проекте может быть потенциальной целевой зоной размещения.

3. vRealize Automation исключает облачные зоны, в которых не хватает работоспособных ресурсов для развертывания.

Например, если слишком много ресурсов выключено или находится в режиме обслуживания, облачная зона исключается.

4. vRealize Automation исключает облачные зоны, которые не соответствуют бизнес-требованиям.

Например, для развертывания может быть превышено ограничение цены или бюджета для зоны.

5. vRealize Automation исключает облачные зоны, которые не соответствуют требованиям к кластеру.

Например, для ресурсов облачной зоны могут быть установлены слишком сильные ограничения по использованию ЦП или памяти.

6. vRealize Automation исключает облачные зоны, которые не имеют сходства с ограничениями приложения.

Для сходства необходимо, чтобы теги ограничений на уровне проекта или облачного шаблона соответствовали тегам возможностей, найденным в ресурсах облачной зоны.

Например, если в облачном шаблоне или проекте есть ограничение хранилища, которое предполагает использование хранилища с тегом `pci`, то облачная зона, в которой ни один из ресурсов хранилища не имеет этого тега возможностей, будет исключена.

7. vRealize Automation выбирает облачные зоны с наилучшим приоритетом подготовки.

8. Если политика размещения на уровне проекта не является политикой по умолчанию, vRealize Automation выбирает облачную зону, в которой поддерживается политика размещения, которая не является политикой по умолчанию.

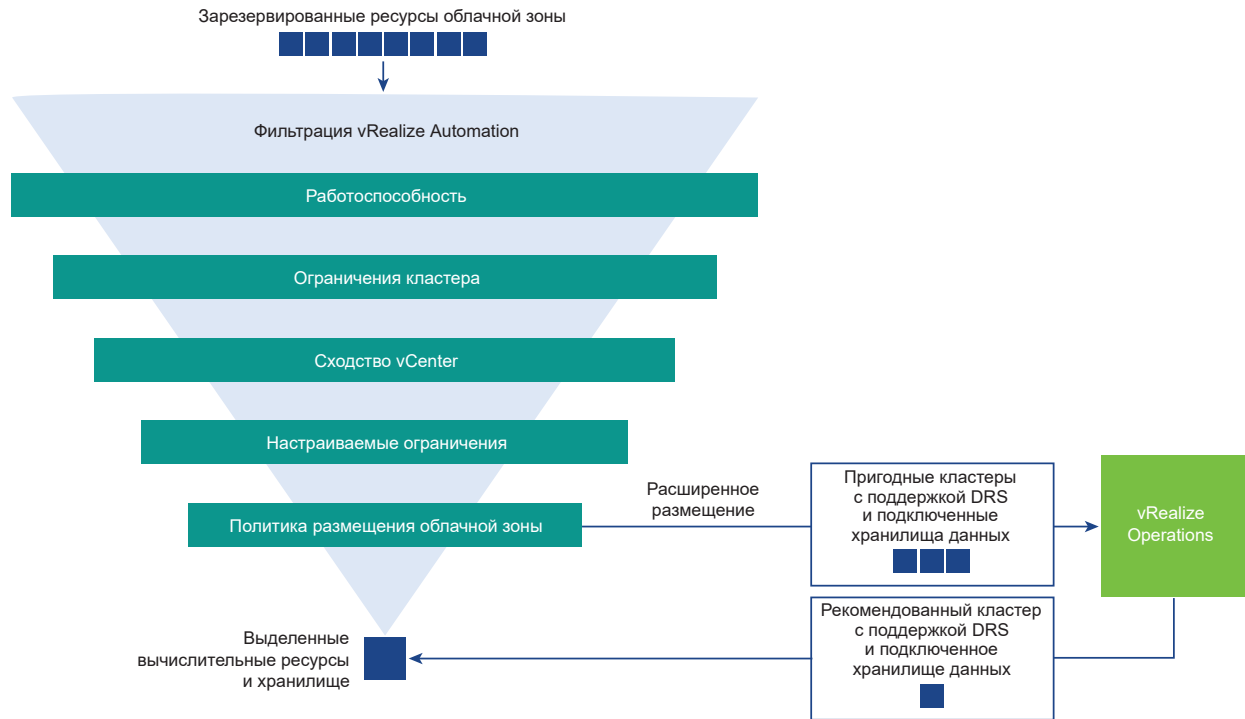
В этом выпуске политика распределения — единственная политика, которая не является политикой по умолчанию. Согласно политике распределения нагрузка распределяется посредством выбора облачной зоны с более низким соотношением виртуальных машин к узлам. Согласно политике по умолчанию развертывание проводится в первой доступной зоне.

Политика размещения проекта является лишь одним из факторов фазы резервирования облачной зоны. Она не влияет на политику размещения облачной зоны на этапе выделения и не имеет отношения к ней.

По завершении фазы резервирования будет выбрана одна облачная зона и ее ресурсы. vRealize Automation резервирует первую доступную зону, которая остается пригодной после применения предыдущих фильтров.

## Фаза 2. Выделение

vRealize Automation проверяет вычислительные ресурсы зарезервированной облачной зоны и связанное хранилище.



1. vRealize Automation исключает ресурсы облачной зоны, которые находятся в состоянии обслуживания или выключены.

Обратите внимание, что для развертывания по-прежнему достаточно работоспособных ресурсов. В противном случае вся облачная зона была бы исключена на этапе резервирования.

2. vRealize Automation исключает ресурсы, которые не соответствуют ограничениям на уровне кластера, найденным в облачном шаблоне или проекте.

Например, ресурс в облачной зоне может иметь тег `test` в разделе **Инфраструктура > Ресурсы > Вычислительные ресурсы**.

Если в облачном шаблоне или проекте есть тег ограничения использования ресурса `dev`, ресурс `test` исключается.

Кроме того, профили хранилища или сети в облачной зоне могут быть помечены тегами так, что они не соответствуют ограничениям хранилища или сети на уровне кластера в облачном шаблоне или проекте.

3. vRealize Automation исключает ресурсы на основе параметров сходства, определенных в vCenter.

Например, в vCenter может быть правило, согласно которому присутствие виртуальной машины в одном кластере может блокировать использование другого кластера.

4. vRealize Automation исключает ресурсы, которые не соответствуют оставшимся настраиваемым ограничениям, найденным в облачном шаблоне или проекте.

Например, если облачный шаблон содержит ограничение, предполагающее использование образа с тегом `ubuntu`, то облачная зона, в которой ни одно из сопоставлений образов не помечено тегом `ubuntu`, будет исключена.

5. vRealize Automation выполняет поиск наилучших возможных вычислительных ресурсов и хранилища в соответствии с политикой размещения облачной зоны.

vRealize Automation задействует vRealize Operations Manager только в том случае, если выполнены два следующих условия.

- Для политики размещения облачной зоны установлено значение «Расширенная».
- После фильтрации на шаге 4 по крайней мере один кластер с поддержкой DRS и связанное с ним хранилище остаются пригодными.

В противном случае vRealize Automation переходит к использованию собственного алгоритма размещения без входных данных из vRealize Operations Manager.

### Рекомендация по размещению, предоставляемая vRealize Operations Manager

Если обеспечивается возможность приема входных данных из vRealize Operations Manager, vRealize Automation связывается с vRealize Operations Manager для получения рекомендации относительно наилучших возможных вычислительных ресурсов и хранилища для развертывания. vRealize Automation отправляет в vRealize Operations Manager следующие данные.

- Пригодные целевые кластеры с поддержкой DRS и подключенные к ним хранилища данных или кластер хранилищ данных
- Количество ресурсов или размер кластера развертывания
- Требования к ЦП и памяти для виртуальных машин в развертывании
- Требования к диску для виртуальных машин в развертывании

Если для пригодных целевых объектов vRealize Operations Manager может вернуть оптимальное размещение для каждой из виртуальных машин, vRealize Automation выделяет вычислительные ресурсы и хранилище согласно рекомендации vRealize Operations Manager.

Дополнительные сведения о том, как в vRealize Operations Manager обрабатываются рабочие нагрузки, см. в [документации по vRealize Operations](#).

Если vRealize Operations Manager не удастся найти рекомендацию или vRealize Automation не удастся найти кластер и хранилище с поддержкой DRS, vRealize Automation проверяет параметр перехода в аварийный режим для облачной зоны.

- С переходом в аварийный режим

vRealize Automation выделяет вычислительные ресурсы и хранилище, которые являются пригодными даже без рекомендации vRealize Operations Manager.

- Без перехода в аварийный режим

vRealize Automation отменяет запрос и не переходит к подготовке.

### Фаза 3. Подготовка

vRealize Automation проводит развертывание запрошенных виртуальных машин, хранилища и сети посредством адаптера для целевого объекта размещения, выбранного в конце фазы выделения.

Целевой объект размещения состоит из узлов, кластеров или пулов вычислительных ресурсов, а также подключенного хранилища данных или кластера хранилищ данных.

### Непрерывная оптимизация с помощью vRealize Operations Manager

При добавлении адаптера vRealize Automation в vRealize Operations Manager vRealize Operations Manager автоматически создает новый настраиваемый центр обработки данных (CDC) для рабочих нагрузок на основе vRealize Automation.

Непрерывная оптимизация обеспечивает повторную балансировку, перемещение рабочей нагрузки и использование vRealize Automation при помощи vRealize Operations Manager после исходного размещения рабочей нагрузки. Если ресурсы виртуализации перемещаются или изменяется нагрузка на них, то подготовленные рабочие нагрузки vRealize Automation можно перемещать по мере необходимости.

- Непрерывная оптимизация автоматически создает новый CDC в vRealize Operations Manager.  
Для каждой облачной зоны vRealize Automation vSphere создается новый CDC.
- Новый созданный CDC содержит каждый кластер, управляемый vRealize Automation и связанный с облачной зоной.

---

**Примечание** Не нужно вручную создавать смешанный CDC для кластеров vRealize Automation и кластеров не на базе vRealize Automation.

---

- vRealize Operations Manager используется для запуска непрерывной оптимизации для только что созданного CDC на базе vRealize Automation.
- Рабочие нагрузки можно повторно сбалансировать или повторно разместить в той же облачной зоне или CDC.
- Оптимизация не приводит к новому нарушению размещения vRealize Automation или vRealize Operations Manager.
  - Если в данный момент существуют нарушения размещения, процесс оптимизации может устранить проблемы, связанные с эксплуатацией vRealize Operations Manager.
  - Если в данный момент существуют нарушения размещения, процесс оптимизации не может устранить проблемы, связанные с бизнес-задачами vRealize Operations Manager.

Например, если для перемещения вручную виртуальной машины в кластер, который не поддерживает ограничения, был использован vRealize Operations Manager, то vRealize Operations Manager не обнаружит нарушение или не устранит проблему.
- Данный выпуск подчиняется эксплуатационной задаче на уровне CDC. Все кластеры — участники vRealize Automation оптимизируются в соответствии с теми же параметрами.

Чтобы задать различные эксплуатационные задачи для кластеров, необходимо настроить их в отдельных центрах CDC vRealize Automation, связанных с отдельными облачными зонами vSphere. Одним из примеров такой ситуации может быть разделение кластера тестирования и производственного кластера.

- Цели применения и ограничения vRealize Automation, определенные в vRealize Automation, соблюдаются во время всех операций по повторной балансировке или перемещению для оптимизации.
- Теги размещения vRealize Operations Manager не могут применяться к рабочим нагрузкам, подготовленным с помощью vRealize Automation.

Кроме того, поддерживается запланированная оптимизация с участием нескольких компьютеров.

Регулярные оптимизации, происходящие по графику, не являются процессами категории «все или ничего».

Если возникают условия, прерывающие перемещение компьютеров, успешно перемещенные компьютеры остаются в новой среде, а в ходе следующего цикла vRealize Operations Manager пытается переместить оставшиеся компьютеры в режиме, обычном для vRealize Operations Manager. Такая частично выполненная оптимизация не приводит к отрицательным последствиям для vRealize Automation.

#### Включение непрерывной оптимизации

При добавлении адаптера vRealize Automation в vRealize Operations Manager vRealize Operations Manager автоматически создает новый выделенный центр обработки данных для рабочих нагрузок на основе vRealize Automation.

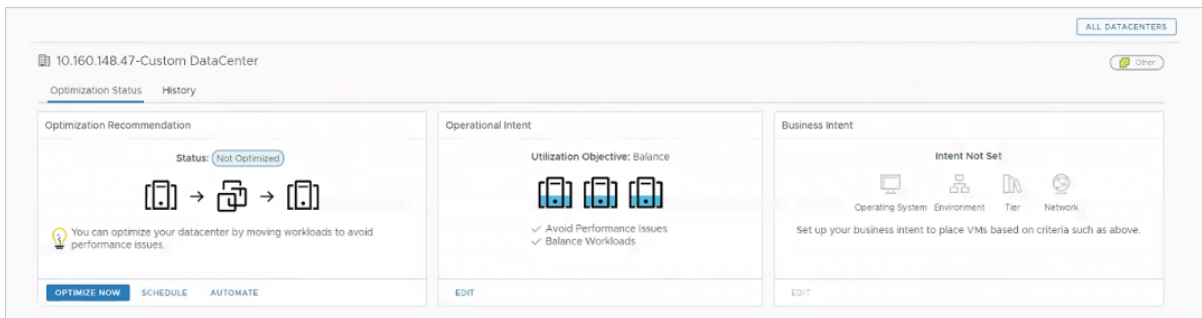
Кроме добавления интеграции в Cloud Assembly, другие действия по установке для организации непрерывной оптимизации не требуются. В созданном центре обработки данных можно настраивать и использовать vRealize Operations Manager для перемещения рабочих нагрузок. См. раздел [Пример непрерывной оптимизации](#).

#### Пример непрерывной оптимизации

В следующем примере показан рабочий процесс повторной балансировки для непрерывной оптимизации vRealize Automation при помощи vRealize Operations Manager.

1. На главной странице vRealize Operations Manager выберите **Оптимизация рабочей нагрузки**.
2. Выберите автоматически созданный центр обработки данных vRealize Automation.
3. В разделе **Эксплуатационная задача** нажмите **Изменить** и выберите **Балансировка**.

Нельзя выбрать или изменить бизнес-задачу, которая отключена, если центр обработки данных используется для оптимизации vRealize Automation.



4. В разделе **Рекомендация по оптимизации** нажмите **Оптимизировать сейчас**.  
vRealize Operations Manager отобразит схему состояния до и после предлагаемой операции.
5. Нажмите кнопку **Далее**.
6. Нажмите **Начать действие**.



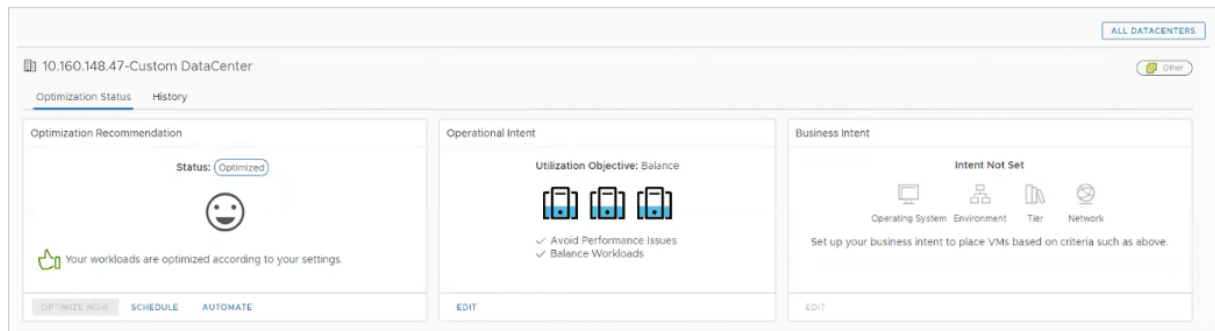
7. В vRealize Automation можно отслеживать выполнение этой операции: нажмите **Ресурсы > Развертывания** и посмотрите состояние события.

The screenshot shows the 'Events' tab in the vRealize Automation interface. It displays a table of tasks for a specific request. The request is titled '#7 - Relocate RRD-WLP-003' and is currently 'In Progress'. It was requested by 'System User' for 'Fritz Arbeiter' on 'August 13, 2018 11:43 AM'. The table lists five tasks: Submitted, Pre-approval, Relocate, Post-approval, and Completed. The 'Relocate' task is the one currently in progress.

Tasks	Component	Status	Depends On
Submitted	Deployment	Successful	
Pre-approval	Deployment	Approved	
Relocate	Deployment	In Progress	
Post-approval	Deployment		
Completed	Deployment		

Когда повторная балансировка завершается, vRealize Automation обновляется. На странице «Вычислительные ресурсы» будет указано, что компьютеры перемещены.

В vRealize Operations Manager результаты следующего сбора данных изменятся с учетом завершения данной оптимизации.



В vRealize Operations Manager можно посмотреть данную операцию, перейдя в меню:

**Администрирование > Журнал > Последние задачи.**

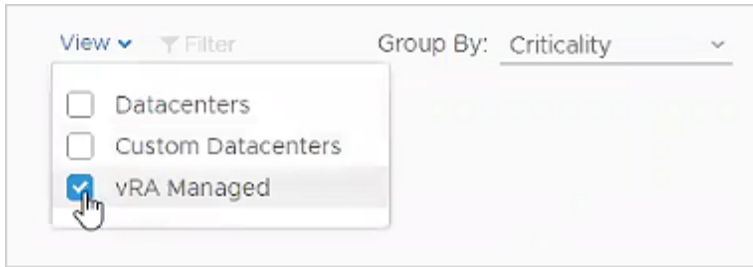
**Поиск центров обработки данных под управлением vRealize Automation**

vRealize Operations Manager можно использовать для отображения только тех центров обработки данных, которые находятся под управлением vRealize Automation.

#### Процедура

1. На главной странице vRealize Operations Manager выберите **Оптимизация рабочей нагрузки**.
2. В правом верхнем углу нажмите раскрывающееся меню **Просмотр**.

3. Выберите только центры обработки данных под управлением vRealize Automation.



## Мониторинг развертывания при помощи vRealize Operations Manager

В vRealize Automation могут отображаться данные vRealize Operations Manager о развертываниях.

Благодаря просмотру отфильтрованного набора показателей непосредственно в vRealize Automation упрощается задача получения доступа к vRealize Operations Manager или поиска в нем. Хотя контекстный запуск в vRealize Operations Manager невозможен, можно в любой момент войти в систему и использовать vRealize Operations Manager для получения необходимых дополнительных данных.

### Включение данных vRealize Operations Manager

Чтобы vRealize Automation показывала данные vRealize Operations Manager, должны иметься специальные интеграции. Для интеграции необходимо ввести адрес и учетные данные для входа для vRealize Automation, vRealize Operations Manager и vCenter.

#### Процедура

1. В vRealize Operations Manager перейдите в раздел **Источники данных > Интеграции**, а затем проверьте или добавьте интеграцию учетной записи vCenter.
2. В Cloud Assembly перейдите в раздел **Инфраструктура > Подключения > Облачные учетные записи**, а затем проверьте или добавьте свою учетную запись vCenter.

vRealize Operations Manager и vRealize Automation должны быть подключены к одной и той же системе vCenter.

3. В vRealize Operations Manager перейдите в раздел **Источники данных > Интеграции** и добавьте интеграцию учетной записи адаптера vRealize Automation 8.x.
4. В Cloud Assembly перейдите в раздел **Инфраструктура > Подключения > Интеграции** и добавьте интеграцию vRealize Operations Manager.

Введите адрес vRealize Operations Manager в следующем формате.

`https://operations-manager-IP-address-or-FQDN/suite-api`

Дополнительные сведения см. в [Интеграция с vRealize Operations Manager](#).

#### Следующие шаги

В Cloud Assembly выберите **Ресурсы > Развертывания**, затем развертывание в своем vCenter и убедитесь, что появляется вкладка «Мониторинг».

## Работоспособность и оповещения, предоставляемые vRealize Operations Manager

При включенном режиме мониторинга vRealize Automation извлекает сведения о работоспособности и соответствующие оповещения vRealize Operations Manager о ваших развертываниях.

Для доступа к средству мониторинга нажмите нужное развертывание и перейдите на вкладку **Мониторинг**. Если такая вкладка отсутствует, см. [Включение данных vRealize Operations Manager](#).

Чтобы просмотреть оповещения, выделите имя развертывания в верхней части дерева компонентов в левой части окна.

- В оповещениях можно просмотреть уровень важности и текст оповещений.
- Чтобы легче находить нужные оповещения, используйте фильтрацию и сортировку данных в столбцах.
- Появляются только значки работоспособности и оповещения о работоспособности. Другие типы оповещений, например эффективность или риск, не поддерживаются.

## Показатели, предоставляемые решением vRealize Operations Manager

При включенном режиме мониторинга vRealize Automation получает из vRealize Operations Manager показатели о развертываниях.

Для доступа к средству мониторинга нажмите нужное развертывание и перейдите на вкладку **Мониторинг**. Если такая вкладка отсутствует, см. [Включение данных vRealize Operations Manager](#).

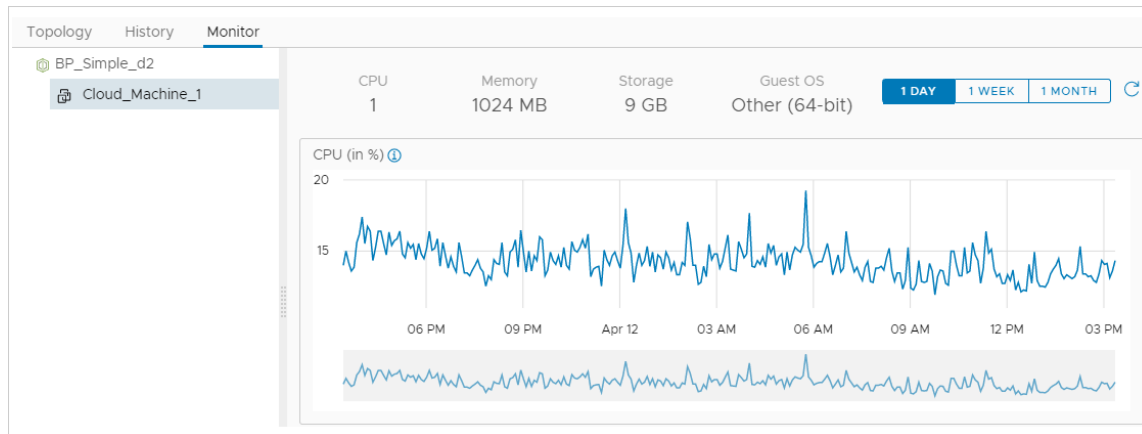
Чтобы просмотреть показатели, разверните дерево компонентов в левой части экрана и выберите нужную виртуальную машину.

- Показатели не кэшируются. Они передаются непосредственно из vRealize Operations Manager, и этот процесс может занять несколько минут.
- Отображаются только показатели виртуальных машин. Показатели других компонентов, например vCloud Director, программного обеспечения или любого ресурса как услуги не отслеживаются.
- Отображаются только показатели виртуальных машин vSphere. Другие поставщики облачных служб, например AWS или Azure, не поддерживаются.

Показатели отображаются как графики на временной шкале, на которых показаны минимальные и максимальные значения для следующих измеряемых свойств.

- ЦП
- Память
- Число операций ввода-вывода в сек. для хранилища
- Пропускная способность сети в МБ/с

Чтобы отобразить имя определенного показателя, нажмите синий значок информации в верхнем левом углу временной шкалы.

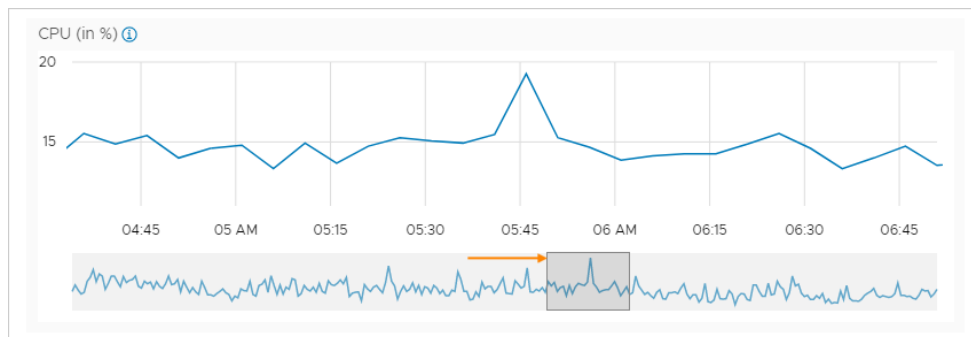


### Действия с данными, предоставляемыми из vRealize Operations Manager

Если показатели, предоставленные из vRealize Operations Manager, свидетельствуют о проблеме, проблемные области можно определить непосредственно в vRealize Automation.

Для просмотра показателей, предоставляемых vRealize Operations Manager, нажмите нужное развертывание и откройте вкладку **Мониторинг**. Если такая вкладка отсутствует, см. [Включение данных vRealize Operations Manager](#).

Предоставляются показатели за последний день, неделю и месяц. Чтобы подробнее увидеть нужный интервал времени, выберите соответствующую область в нижней затененной части графика какого-либо показателя.



### Управление ресурсами и оптимизация развертывания с помощью показателей vRealize Operations Manager в vRealize Automation

В интегрированной среде vRealize Automation и vRealize Operations Manager можно получить доступ к ключевым сведениям и оповещениям для объектов vRealize Automation, которые отслеживаются vRealize Operations Manager.

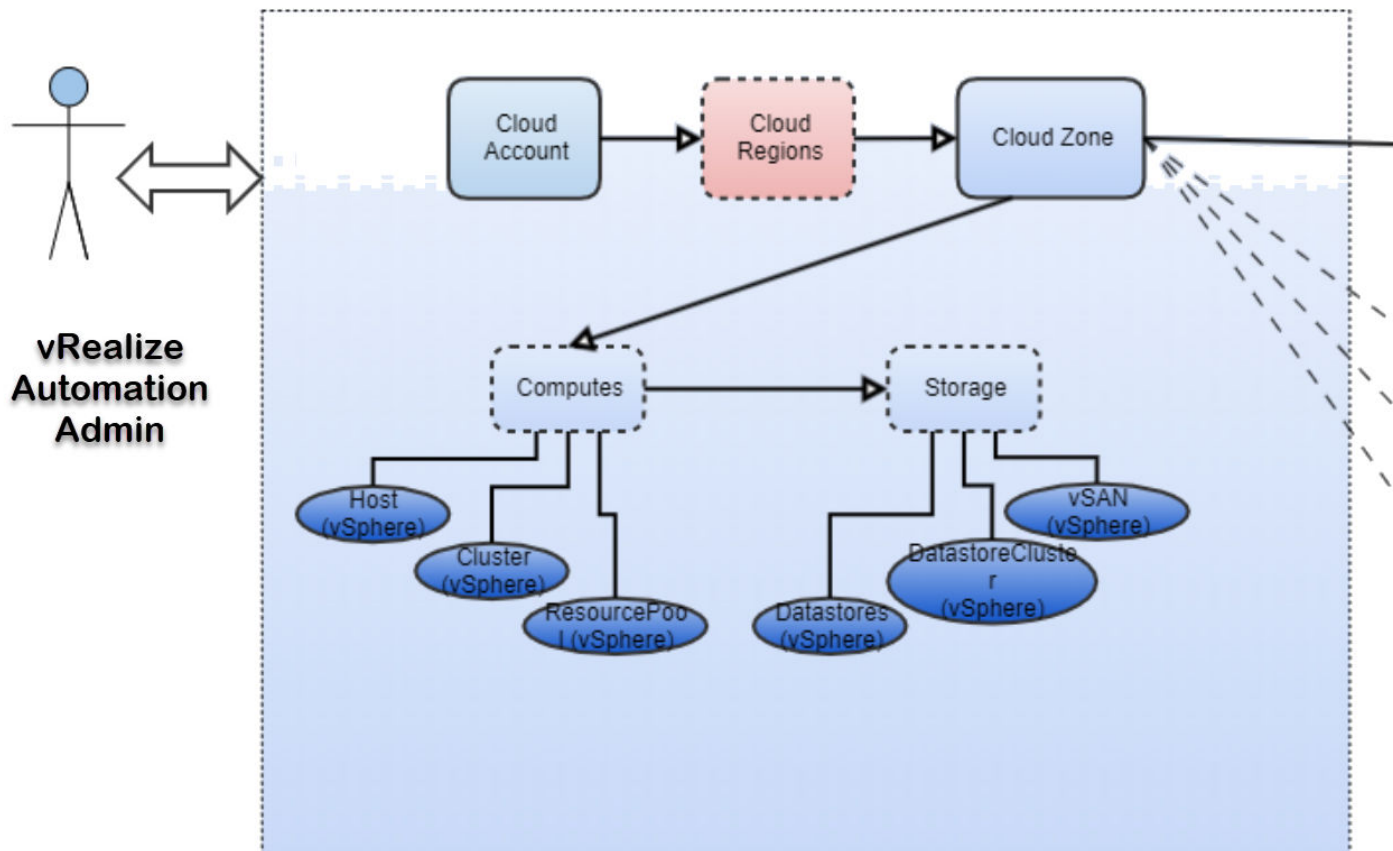
На панели управления **Ключевые сведения** и вкладке **Оповещения** в режиме реального времени отображаются информация о ресурсах и прочие связанные показатели, которые необходимы для принятия управленческих решений в vRealize Automation без открытия vRealize Operations Manager. Эту информацию предоставляет связанное приложение vRealize Operations Manager.

## Работа с панелью управления «Ключевые сведения» и с оповещениями о ресурсах

Панель управления **Ключевые сведения** передает сведения о потреблении ресурсов всеми вычислительными устройствами в облачной зоне применительно к разным проектам. На ней также могут отображаться развертывания проектов, которые необходимо оптимизировать.

На страницах **Оповещения** отображаются сведения о потенциальных проблемах, связанных с объемом ресурсов и производительностью, для таких объектов, как облачные зоны, проекты, развертывания и виртуальные машины. Они также содержат сведения для владельцев проектов в отношении развертываний, которые можно оптимизировать. Если нажать ссылку на развертывание, открывается соответствующая вкладка **Оптимизация**, где приведены необходимые инструкции.

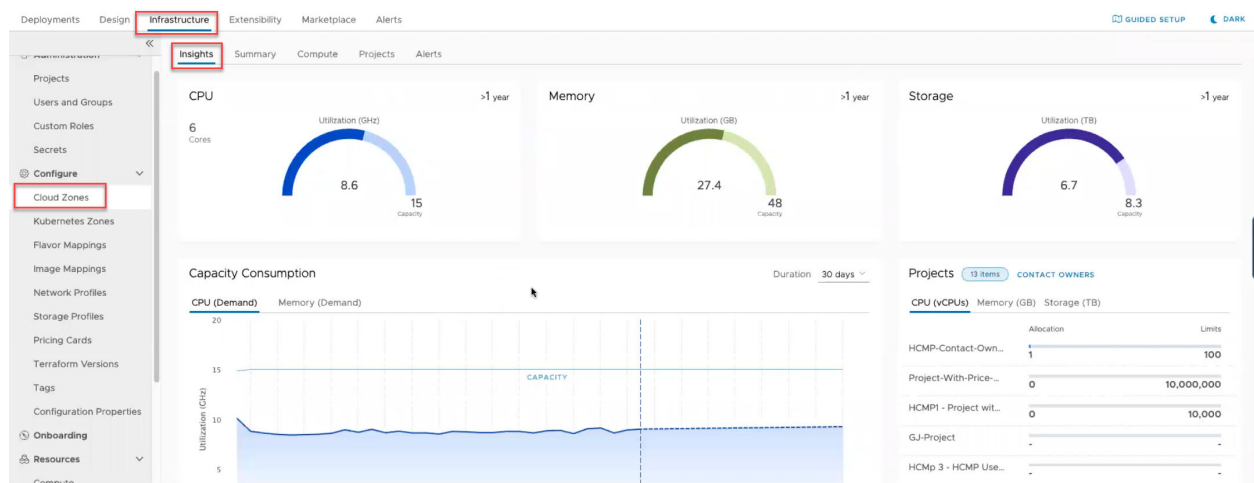
На следующем рисунке показана связь между ресурсами и развертываниями vRealize Automation, а также данными, которые предоставляет связанное приложение vRealize Operations Manager в vRealize Automation.



## Работа с панелью управления «Ключевые сведения»

На панели управления **Ключевые сведения**, которая доступна на каждой странице облачной зоны, отображаются следующие показатели vRealize Operations Manager.

- Использование ЦП, памяти и хранилища (в процентах от объема ресурсов)
- Сводка о потреблении ресурсов
- Требования к ЦП и памяти, а также журнал использования
- Потребление ресурсов в проектах.
- Объем освобождаемых ресурсов, включая экономию затрат, для развертываний и проектов в облачной зоне



Кроме того, с помощью данной панели можно оповещать владельцев проектов о возможности оптимизации развертываний.

Панель управления **Ключевые сведения** доступна для облачных зон vSphere и VMware Cloud on AWS при условии, что облачные учетные записи настроены в vRealize Automation и vRealize Operations Manager и отслеживаются в vRealize Operations Manager.

Подробности см. в [Использование панели управления «Ключевые сведения» для мониторинга объема ресурсов и уведомления владельцев проектов в vRealize Automation](#).

## Работа с оповещениями

На страницах **Оповещения** можно применять различные фильтры. Категории фильтрации определяются связанным приложением vRealize Operations Manager.

- Важность
- Состояние
- Влияние
- Тип
- Подтип

## ■ Ресурс

Для каждого фильтра можно использовать соответствующий быстрый фильтр. Например, для фильтра ресурсов можно использовать быстрые фильтры разного типа: по облачной зоне, виртуальной машине, развертыванию и проекту.

Для контроля отображаемых оповещений необходимо использовать комбинации фильтров и быстрых фильтров.

The screenshot shows the vRealize Automation Cloud Assembly Alerts page. The navigation bar includes tabs for Deployments, Design, Infrastructure, Extensibility, Marketplace, and Alerts. Below the navigation bar, there's a 'Resource Type' dropdown menu with a 'Quick filters' section. The 'Quick filters' section has checkboxes for 'Cloud Zone', 'Virtual Machine', 'Deployment', and 'Project'. The 'Alerts' section displays a list of alerts. The first two alerts are 'Virtual machine is powered off for more than 5 days' with a warning icon. The next three alerts are 'AlertDefinition\_...' with an error icon. The right sidebar shows details for the selected alert, including a warning icon, the alert title, creation and update times, a link to the virtual machine, and a 'Suggestions' section with two items: 'Delete powered off machines' and 'Manually power on the virtual machine.' There is also a 'Notes' section with a text input and an 'ADD NOTE' button.

Некоторые **Оповещения** содержат информацию о развертываниях (включая ссылки на них), которые можно оптимизировать. Отдельное оповещение позволяет связаться с владельцем проекта, посмотреть панель управления «Ключевые сведения» или принять меры.

Deployments Design Infrastructure Extensibility Marketplace **Alerts** GUIDED

Severity ▾ Q Quick filters Status: Active X Severity: Critical X

**Today**

- The project has some deployments t...** 6:17 PM  
Project » vc65 project  
The project has some deployments that contain optimizable resources.
- Cloud Zone has less than 60 days r...** 11:46 AM  
Cloud Zone » sq-a-vc67.sqa.local / Datacenter  
The time remaining on cloud zone is less than 60 days until capacity demand runs out.

**Yesterday**

- Cloud Zone has less than 60 days re...** 1:35 PM  
Cloud Zone » 测试Zone-g11n  
The time remaining on cloud zone is less than 60 days until capacity demand runs out.
- Cloud Zone has less than 60 days re...** 1:35 PM  
Cloud Zone » vmc staging-vsphere / SDDC-Datacenter  
The time remaining on cloud zone is less than 60 days until capacity demand runs out.
- AlertDefinition\_ff4d3d96-fa4f-4022-...** 1:26 PM

**The project has some deployments that contain optimizable resources**  
Created: Dec 14, 2020, 6:17:44 PM | Updated: Dec 14, 2020, 6:17:44 PM  
Project » vc65 project

The project has some deployments that contain optimizable resources.

Severity: Critical Status: Active Impact: Efficiency Type: Application Subtype: Performance

**Suggestions** 1 [REVIEW PROJECT](#)

- If the project is experiencing increased provisioning, you can review the project to understand deployments and poweroff/delete the ones that are no longer in use.

**Deployments to review**

Name	Owner
contact-owner-test-dep-2	

Items per page 10 ▾

**Notes**

Investigating

[ADD NOTE](#)

Оповещения доступны только для объектов ресурсов vSphere и VMware Cloud on AWS.

Дополнительные сведения о настройке и использовании встроенных оповещений см. в [Использование оповещений для управления объемом, производительностью и доступностью ресурсов в vRealize Automation](#) и [Использование оповещений для оптимизации развертываний в vRealize Automation](#).

## Что такое планы внедрения в Cloud Assembly

План внедрения рабочей нагрузки используется для идентификации компьютеров, данных о которых были собраны из типа учетной записи облачной службы в целевой области или центре обработки данных, но проект Cloud Assembly ими еще не управляет.

При добавлении учетной записи облачной службы, содержащей компьютеры, которые были развернуты за пределами Cloud Assembly, Cloud Assembly начнет управлять этими компьютерами только после их внедрения. Для перевода неуправляемых компьютеров под управление системой Cloud Assembly можно использовать план внедрения рабочей нагрузки. Необходимо создать план, добавить в него компьютеры, а затем запустить план, чтобы импортировать компьютеры. С помощью плана внедрения можно создать облачный шаблон, а также одно развертывание или несколько.



В рамках одного плана можно внедрять один или несколько неуправляемых компьютеров, выбираемых вручную.

- В рамках одного плана внедрения в час можно внедрить до 3500 неуправляемых компьютеров.
- В рамках нескольких планов внедрения можно одновременно внедрять до 17 000 неуправляемых компьютеров в час.

Компьютеры, доступные для внедрения рабочей нагрузки, перечислены в разделе **Ресурсы > Ресурсы > Виртуальные машины** и имеют метку *Discovered* в столбце «Источник». Здесь перечисляются только те компьютеры, по которым собраны данные. После внедрения компьютеров они отображаются в столбце «Источник» с меткой *Deployed*. Чтобы отфильтровать обнаруженные или развернутые компьютеры,

щелкните значок фильтра



Пользователю, запускающему план внедрения рабочей нагрузки, автоматически назначается статус владельца компьютеров.

Кроме того, внедрение поддерживает настраиваемые свойства внедрения, подключенные диски, возможность изменения владельцев развертываний и сети vSphere.

- **Настраиваемые свойства.** Настраиваемые свойства можно настраивать как на уровне плана, так и на уровне отдельных компьютеров. Настраиваемое свойство, настроенное на уровне компьютера, переопределяет то же самое свойство на уровне плана.
- **Подключенные диски.** Если у компьютера есть диски, не являющиеся загрузочными, они автоматически внедряются с родительским компьютером. Чтобы просмотреть диски, не являющиеся загрузочными, щелкните имя компьютера в плане, а затем перейдите на вкладку **Хранилище**.

- Владение развертыванием. Процесс внедрения позволяет изменить владельца развертывания, заданного по умолчанию. Чтобы изменить владельца, выберите развертывание на вкладке **Развертывание**, щелкните **Действия > Изменить владельца** и выберите нужного пользователя, связанного с проектом.

### Примеры внедрения

Примеры способов внедрения см. в разделе [Пример: внедрение выбранных компьютеров в качестве отдельного развертывания в Cloud Assembly](#).

### Подписки на события внедрения

При запуске плана создается событие `Deployment Onboarded`. С помощью параметров вкладки «Расширяемость» можно подписаться на такие события развертывания и выполнять над ними действия.

После внедрения можно обновить проект с помощью действия по регулярному обслуживанию для внедренных развертываний. Чтобы использовать действие изменения проекта, целевой проект должен использовать те же ресурсы облачной зоны, что и развертывание. Действие изменения проекта во внедренных развертываниях, в которые были внесены изменения после внедрения, выполнить нельзя.

## Пример: внедрение выбранных компьютеров в качестве отдельного развертывания в Cloud Assembly

В этом примере рассматривается внедрение двух неуправляемых компьютеров в виде отдельного развертывания Cloud Assembly и создание единого облачного шаблона для всех компьютеров в плане.

При создании облачной учетной записи со всех компьютеров, связанных с ней, собираются данные, после чего эти компьютеры отображаются на странице **Ресурсы > Ресурсы > Виртуальные машины**. Если в облачной учетной записи есть компьютеры, развернутые за пределами Cloud Assembly, можно воспользоваться планом подключения, чтобы служба Cloud Assembly могла управлять развертываниями этих компьютеров.

---

**Примечание** Развертывания можно переименовывать только до их внедрения. После внедрения параметр **Переименовать** отключен.

---

### Необходимые условия

- Убедитесь в том, что вам назначена необходимая роль пользователя. См. раздел [Что такое роли пользователей vRealize Automation](#).
- См. раздел [Что такое планы внедрения в Cloud Assembly](#).
- Создайте и подготовьте проект Cloud Assembly.

Эта процедура включает в себя некоторые действия из базового примера использования для Wordpress. См. раздел [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).

- Создайте проект, добавьте пользователей и назначьте роли пользователей в проекте. См. раздел [Часть 2. Создание примера проекта Cloud Assembly](#).

- Создайте учетную запись облачной службы Amazon Web Services для проекта. См. раздел, посвященный облачной учетной записи, в [Часть 1. Настройка примера инфраструктуры Cloud Assembly](#).

В этой процедуре учетная запись облачной службы Amazon Web Services содержит компьютеры, которые были развернуты до добавления учетной записи облачной службы в Cloud Assembly, причем развертывание выполнено приложением, отличным от Cloud Assembly.

- Убедитесь, что на странице **Ресурсы > Ресурсы > Виртуальные машины** имеются компьютеры для внедрения. Дополнительные сведения см. в разделе [Управление ресурсами в Cloud Assembly](#).

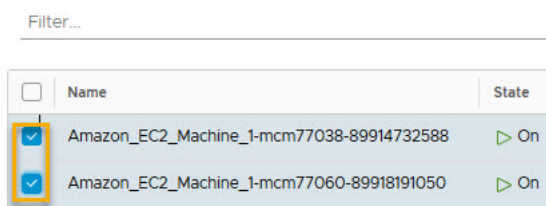
## Процедура

1. Выберите **Инфраструктура > Внедрение**.
2. Щелкните **Создать план подключения** и введите примерные значения.

Настройка	Пример значения
Имя плана	VC-sqa-deployments
Описание	Пример плана подключения для компьютера AWS для учетной записи облачной службы OurCo-AWS
Учетная запись облачной службы	OurCo-AWS
Проект по умолчанию	WordPress

3. Щелкните **Создать**.
4. В плане на вкладке **Развертывания** щелкните **Выбрать компьютеры**, выберите один или несколько компьютеров и нажмите **ОК**.

## Select Machines



5. Выберите **Создать одно развертывание, содержащее все компьютеры** и щелкните **Создать**.
6. Установите флажок рядом с именем нового развертывания и щелкните **Облачный шаблон....**
7. Щелкните **Создать облачный шаблон в формате Cloud Assembly** и введите имя облачного шаблона или щелкните **Назначить существующий облачный шаблон** и выберите облачный шаблон для назначения.

**Примечание** Сопоставление облачных шаблонов с внедренными развертываниями используется только с целью визуального представления для конечных пользователей. Внедренные развертывания несовместимы с облачными шаблонами.

8. Нажмите **Сохранить**.

**Cloud Template Configuration**

Mapping of Cloud Templates to onboarding deployments is only for visual parity for end consumers. Onboarded deployments are not compatible with Cloud Templates.

Deployment: Demo

☐ None (use runtime snapshot)  
☐ Create Cloud Template in Cloud Assembly format  
☒ Assign an existing Cloud Template

	Name	Project	Last Updated
<input checked="" type="radio"/>	Demo	onboarding	Oct 21, 2021, 1:36:15 PM
<input type="radio"/>	171	onboarding	Jun 10, 2021, 8:21:55 AM
<input type="radio"/>	asdf	onboarding	May 25, 2021, 9:24:07 AM
<input type="radio"/>	asdf	onboarding	Dec 7, 2020, 3:03:53 PM

**CANCEL** **SAVE**

**Примечание** Если в плане внедрения используется компьютер vSphere, необходимо отредактировать облачный шаблон после завершения процесса внедрения. Процесс внедрения не может связать исходный компьютер vSphere и свой шаблон компьютера. В результирующем облачном шаблоне в коде будет содержаться запись `imageRef: "no image available"`. Облачный шаблон нельзя развернуть, пока не будет указано правильное имя шаблона в поле `imageRef:`. Чтобы упростить поиск и обновление облачного шаблона после завершения процесса внедрения, используйте параметр **Имя облачного шаблона** на странице **Настройка облачного шаблона** развертывания. Запишите автоматически созданное имя облачного шаблона или введите имя и запишите его. По окончании внедрения найдите и откройте облачный шаблон, а затем замените запись `"no image available"` в поле `imageRef:` на правильное имя шаблона.

9. Установите флажок рядом с именем развертывания, щелкните **Запустить**, а затем снова нажмите **Запустить** на странице **Запуск плана**.

Выбранные компьютеры внедряются в качестве отдельного развертывания вместе с соответствующим облачным шаблоном.

10. Откройте и проверьте облачный шаблон: перейдите на страницу **Проект > Облачные шаблоны** и щелкните имя облачного шаблона.

11. Откройте и проверьте развертывание: перейдите на страницу **Ресурсы > Развертывания** и щелкните имя развертывания.

## Расширенная настройка среды Cloud Assembly

В среде Cloud Assembly можно задать различные параметры для работы над проектом, в частности его настройки, интеграции и развертывания.

Дополнительные сведения о методах администрирования, в том числе о работе с пользователями и журналами, а также о присоединении к программе улучшения качества или выходе из нее, см. в справке [Администрирование vRealize Automation](#).

## Настройка прокси-сервера Интернета для vRealize Automation

Для установленных систем vRealize Automation в изолированных сетях без прямого доступа к Интернету можно использовать прокси-сервер Интернета, чтобы разрешить функцию Интернета через прокси-сервер. Прокси-сервер Интернета поддерживает HTTP и HTTPS.

Чтобы настроить и использовать поставщиков общедоступных облачных служб, таких как Amazon Web Services (AWS), Microsoft Azure и Google Cloud Platform (GCP), а также внешние точки интеграции, такие как IPAM, Ansible и Puppet, с vRealize Automation, необходимо настроить прокси-сервер Интернета для доступа к внутреннему прокси-серверу Интернета vRealize Automation.

vRealize Automation содержит внутренний прокси-сервер, который взаимодействует с прокси-сервером Интернета. Данный сервер взаимодействует с прокси-сервером, если он настроен с помощью команды `vracli proxy set ...`. Если для организации прокси-сервер Интернета не настроен, то внутренний прокси-сервер vRealize Automation будет пытаться подключиться к Интернету напрямую.

vRealize Automation можно настроить на использование прокси-сервера Интернета с помощью служебной программы командной строки `vracli`. Сведения об использовании API-интерфейса `vracli` доступны при использовании аргумента `--help` в командной строке `vracli`, например `vracli proxy --help`.

Для доступа к прокси-серверу Интернета требуется использовать локальные элементы управления расширяемости на основе действий (ABX), встроенные в vRealize Automation.

---

**Примечание** Доступ к Workspace ONE Access (который раньше назывался VMware Identity Manager) через прокси-сервер Интернета не поддерживается. Для доступа к Workspace ONE Access через прокси-сервер Интернета команду `vracli set vidm` использовать нельзя.

---

Для внутреннего прокси-сервера в качестве IP-формата по умолчанию требуется IPv4. Данный протокол не требует ограничений для IP-протокола, проверки подлинности или действий по защите «злоумышленник посередине» для трафика сертификата TLS (HTTPS).

### Необходимые условия

- Убедитесь, что в сети vRealize Automation, способной передавать исходящий трафик на внешние сайты, у вас есть существующий сервер HTTP или HTTPS, который можно использовать в качестве прокси-сервера Интернета. Подключение должно быть настроено для IPv4.

- Убедитесь, что целевой прокси-сервер в Интернете настроен на поддержку протокола IPv4 в качестве IP-формата по умолчанию, а не IPv6.
- Если прокси-сервер Интернета использует протокол TLS и требует подключение по протоколу HTTPS к своим клиентам, то перед его настройкой необходимо импортировать сертификат сервера с помощью одной из следующих команд.
  - `vracli certificate proxy --set path_to_proxy_certificate.pem`
  - `vracli certificate proxy --set stdin`

Для интерактивного ввода используйте параметр `stdin`.

## Процедура

1. Создайте конфигурацию прокси-сервера для модулей или контейнеров, которые используются Kubernetes. В этом примере доступ к прокси-серверу осуществляется с помощью схемы HTTP.

```
vracli proxy set --host http://proxy.vmware.com:3128
```

2. Отобразите конфигурацию прокси-сервера.

```
vracli proxy show
```

Результат будет примерно следующим.

```
{
  "enabled": true,
  "host": "10.244.4.51",
  "java-proxy-exclude": "/*.local|*.localdomain|localhost|10.244.*|
192.168.*|172.16.*|kubernetes|sc2-rdops-vm06-dhcp-198-120.eng.vmware.com|10.192.204.9|
*.eng.vmware.com|sc2-rdops-vm06-dhcp-204-9.eng.vmware.com|10.192.213.146|sc2-rdops-vm06-
dhcp-213-146.eng.vmware.com|10.192.213.151|sc2-rdops-vm06-dhcp-213-151.eng.vmware.com",
  "java-user": null,
  "password": null,
  "port": 3128,
  "proxy-
exclude": ".local,.localdomain,localhost,10.244.,192.168.,172.16.,kubernetes,sc2-
rdops-vm06-dhcp-198-120.eng.vmware.com,10.192.204.9,.eng.vmware.com,sc2-
rdops-vm06-dhcp-204-9.eng.vmware.com,10.192.213.146,sc2-rdops-vm06-
dhcp-213-146.eng.vmware.com,10.192.213.151,sc2-rdops-vm06-dhcp-213-151.eng.vmware.com",
  "scheme": "http",
  "upstream_proxy_host": null,
  "upstream_proxy_password_encoded": "",
  "upstream_proxy_port": null,
  "upstream_proxy_user_encoded": "",
  "user": null,
  "internal.proxy.config": "dns_v4_first on \nhttp_port
0.0.0.0:3128\nlogformat squid %ts.%03tu %6tr %>a %Ss/%03>Hs
%<st %rm %ru %[un %Sh/%<a %mt\naccess_log stdio:/tmp/logger squid\ncoredump_dir /\ncache
deny all \nappend_domain .prelude.svc.cluster.local\nacl mylan src 10.0.0.0/8\nacl mylan
src 127.0.0.0/8\nacl mylan src 192.168.3.0/24\nacl proxy-exclude dstdomain .local\nacl
proxy-exclude dstdomain .localdomain\nacl proxy-exclude dstdomain localhost\nacl
proxy-exclude dstdomain 10.244.\nACL proxy-exclude dstdomain 192.168.\nACL proxy-exclude
dstdomain 172.16.\nACL proxy-exclude dstdomain kubernetes\nACL proxy-exclude dstdomain
10.192.204.9\nACL proxy-exclude dstdomain .eng.vmware.com\nACL proxy-exclude dstdomain
```

```
10.192.213.146\nacl proxy-exclude dstdomain 10.192.213.151\nalways_direct allow proxy-exclude\nhttp_access allow mylan\nhttp_access deny all\n# End autogen configuration\n",
  "internal.proxy.config.type": "default"
}
```

**Примечание** Если для организации настроен прокси-сервер Интернета, то в примере выше отображается "internal.proxy.config.type": "non-default", а не 'default'. В целях безопасности пароль не отображается.

**Примечание** В случае использования параметра `-proxy-exclude` необходимо изменить значения по умолчанию. Например, если необходимо добавить `acme.com` как домен, к которому нельзя получить доступ с помощью прокси-сервера Интернета, выполните следующие действия.

- а) Введите `vracli proxy default-no-proxy`, чтобы получить параметры исключения прокси-сервера по умолчанию. Это список автоматически создаваемых доменов и сетей.
  - б) Измените значение, чтобы добавить `.acme.com`.
  - в) Введите `vracli proxy set .... --proxy-exclude ...`, чтобы обновить параметры конфигурации.
  - г) Выполните команду `/opt/scripts/deploy.sh`, чтобы повторно развернуть среду.
3. (дополнительно) Исключите домены DNS, полные доменные имена и IP-адреса из доступа через прокси-сервер Интернета.

Всегда изменяйте значения по умолчанию переменной `proxy-exclude` с помощью параметра `--proxy-exclude`. Чтобы добавить `exclude.vmware.com` для домена, сначала используйте команду `vracli proxy show`, затем скопируйте переменную `proxy-exclude` и добавьте значение домена с помощью команды `vracli proxy set ...`, как показано ниже.

```
vracli proxy set --host http://
proxy.vmware.com:3128 --proxy-exclude "exclude.vmware.com,docker-
registry.prelude.svc.cluster.local,localhost,.local,.cluster.local,10.244.,192.,172.16.,sc-
rdops-vm11-dhcp-75-38.eng.vmware.com,10.161.75.38,.eng.vmware.com"
```

**Примечание** Добавьте элементы в `proxy-exclude` вместо замены значений. Если удалить значения по умолчанию `proxy-exclude`, то vRealize Automation не будет работать должным образом. В этом случае необходимо удалить конфигурацию прокси-сервера и начать заново.

4. После настройки прокси-сервера в Интернете с помощью команды `vracli proxy set ...` можно использовать команду `vracli proxy apply` для обновления конфигурации прокси-сервера Интернета и активации новых настроек прокси-сервера.
5. Если это еще не сделано, активируйте изменения сценария, выполнив следующую команду.

```
/opt/scripts/deploy.sh
```

6. (дополнительно) При необходимости настройте прокси-сервер на поддержку внешнего доступа через порт 22.

Для поддержки интеграций, например Puppet и Ansible, прокси-сервер должен разрешать порту 22 получать доступ к соответствующим узлам.

## Пример. Образец настройки Squid

Применительно к шагу 1: если вы настраиваете прокси-сервер Squid, можно настроить конфигурацию в `/etc/squid/squid.conf` с учетом следующих изменений:

```
acl localnet src 192.168.11.0/24

acl SSL_ports port 443

acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 # https
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT

http_access allow !Safe_ports
http_access allow CONNECT !SSL_ports
http_access allow localnet

http_port 0.0.0.0:3128

maximum_object_size 5 GB
cache_dir ufs /var/spool/squid 20000 16 256
coredump_dir /var/spool/squid
refresh_pattern ^ftp: 1440 20% 10080
refresh_pattern ^gopher: 1440 0% 1440
refresh_pattern -i (/cgi-bin/|\?) 0 0% 0
refresh_pattern (Release|Packages(.gz)*)$ 0 20% 2880
refresh_pattern . 0 20% 4320

client_persistent_connections on
server_persistent_connections on
```

## Что можно делать с сопоставлением NSX-T с несколькими системами vCenter в vRealize Automation

Для поддержки различных целей развертывания облачную учетную запись NSX-T можно связать с одной или несколькими облачными учетными записями vCenter.



Одну и ту же существующую сеть NSX-T можно связать с профилями сети для разных систем vCenter и подготовить развертывание в любой системе vCenter на основе ограничений. Ниже приведены несколько примеров.

- Облачные шаблоны, содержащие один компьютер с несколькими сетевыми адаптерами, в которых используется один профиль сети, содержащий сеть NSX-T, охватывающую несколько систем vCenter.
- Облачные шаблоны, содержащие компьютер в *частной* сети, в которой используется профиль сети с изоляцией на основе подсети, а также *существующая* сеть NSX-T, охватывающая несколько систем vCenter.
- Облачные шаблоны, содержащие один компьютер в *частной* сети, в которой используется профиль сети с изоляцией на основе группы безопасности, а также сеть NSX-T, охватывающая несколько систем vCenter.
- Облачные шаблоны, содержащие один компьютер в *маршрутизируемой* сети, в которой используется профиль сети, содержащий сеть NSX-T, охватывающую несколько систем vCenter.
- Облачные шаблоны, содержащие подсистему балансировки нагрузки по требованию, которая определена в профиле сети, где данная подсистема применяется ко всем компьютерам vCenter в сети.
- Облачные шаблоны, содержащие сеть по требованию, которая определена в профиле сети, где данная сеть используется всеми системами vCenter, применяющими данный профиль сети.
- Облачные шаблоны, содержащие группу безопасности по требованию, которая может содержать правила брандмауэра, где группа безопасности связана со всеми системами vCenter в сети.

Для компьютеров, которые подготовлены в разных системах vCenter, можно настроить внутреннего или внешнего поставщика IPAM в vRealize Automation в сети NSX-T и использовать один и тот же IP-адрес.

Если профиль сети не определен в системе, можно подготовить облачный шаблон, содержащий несколько компьютеров в разных системах vCenter, которые используют одну *существующую* сеть NSX-T.

## Что произойдет, если удалить связь облачной учетной записи NSX в vRealize Automation

При удалении связи между облачной учетной записью NSX и облачной учетной записью vCenter необходимо также обновить связанные профили сетей, чтобы удалить связанные объекты NSX.

При удалении связи между облачной учетной записью NSX и облачной учетной записью vCenter элементы инфраструктуры не обновляются автоматически с помощью vRealize Automation. Чтобы удалить связанные объекты NSX, необходимо обновить существующие профили сетей.

В пользовательском интерфейсе содержатся сведения, позволяющие выявить элементы профиля сети, на которые оказывается влияние.

- Если в профиле сети выбрана существующая сеть NSX.
  - Объект помечается как *недействительный*, и отображается сообщение *Некоторые сетевые объекты отсутствуют или недействительны..*
  - При сохранении профиля сети объекты удаляются.

- Если в профиле сети настроена изоляция приложений, необходимо обновить параметры политики изоляции, прежде чем можно будет сохранить профиль сети.
- Если в профиле сети выбраны группы безопасности или подсистемы балансировки нагрузки, при сохранении профиля сети объекты удаляются.

Существующие развертывания продолжают работать в штатном режиме для существующих компонентов, но при создании новых компонентов будет происходить сбой, например в ходе горизонтального масштабирования.

Если повторно установить связь, профиль сети будет заново заполнен, а существующие развертывания будут работать в штатном режиме.

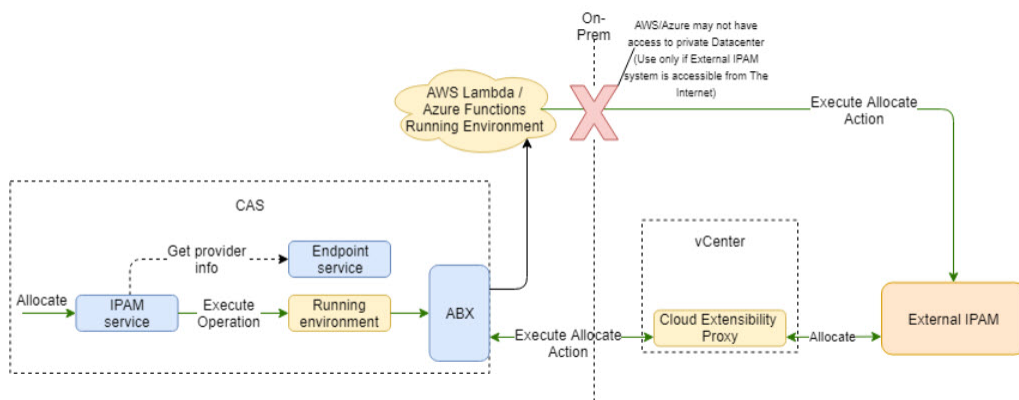
Если удалить облачную учетную запись NSX, описанное выше поведение будет таким же, но объекты сети будут помечены как *отсутствующие*, а не как *недействительные*.

## Использование IPAM SDK для создания внешнего пакета интеграции IPAM для определенного поставщика для vRealize Automation

Внешние поставщики IPAM и партнеры могут загружать и использовать IPAM SDK для создания пакета интеграции IPAM, который позволяет службе vRealize Automation поддерживать решения IPAM определенного поставщика.

Процесс сборки и развертывания настраиваемого пакета интеграции IPAM для vRealize Automation с помощью предоставленного комплекта IPAM SDK описан в документе [Создание и развертывание пакета интеграции IPAM определенного поставщика для VMware Cloud Assembly](#). Как описано в документе, актуальный комплект *VMware vRealize Automation Third-Party IPAM SDK* можно загрузить с веб-сайта [VMware code](#). Доступны следующие пакеты IPAM SDK.

- [VMware vRealize Automation Third-Party IPAM SDK 1.1.0](#)
- [VMware vRealize Automation Third-Party IPAM SDK 1.0.0](#)



Прежде чем приступить к созданию пакета интеграции IPAM для конкретного поставщика с помощью комплекта IPAM SDK, проверьте, не существует ли уже такой пакет для vRealize Automation.

Наличие пакета интеграции конкретного поставщика можно проверить на сайте поставщика IPAM или на [VMware Marketplace](#).

Несмотря на то, что пример Учебник «Настройка интеграции определенного внешнего поставщика IPAM для vRealize Automation» относится к определенному поставщику, в нем также содержатся полезные справочные сведения.

## Использование vRealize Automation с решением Azure VMware Solution

Ниже рассмотрена процедура настройки vRealize Automation для работы с гибридной облачной средой Microsoft Azure VMware Solution с самообслуживанием, позволяющая использовать внутри этой среды рабочие нагрузки vRealize Automation.

vRealize Automation поддерживает соединения с решением Azure VMware Solution (AVS) для перемещения и запуска рабочих нагрузок VMware в облачной среде Azure. Модуль AVS был разработан компанией «Майкрософт» для поддержки интерфейса в средах VMware.

Процедура использования AVS подробно описана в документации «Майкрософт». Документацию можно найти здесь:

- Azure VMware Solution — <https://docs.microsoft.com/en-us/azure/azure-vmware/>

Чтобы использовать AVS в vRealize Automation, необходимо настроить облачные учетные записи как для vCenter, так и для NSX-T. Процедуру настройки этих облачных учетных записей см. в следующих документах:

- Настройка облачной учетной записи vCenter — [Создание облачной учетной записи vCenter в vRealize Automation](#)
- Создание облачной учетной записи NSX-T — [Создание облачной учетной записи NSX-T в vRealize Automation](#)

Ниже описан процесс настройки среды для развертывания рабочих нагрузок vRealize Automation в AVS.

1. Установите и настройте Azure VMware Solution в соответствии с инструкциями поставщика и особенностями своей среды.
2. Создайте облачные учетные записи vCenter и NSX-T внутри развертывания vRealize Automation.

## Использование vRealize Automation с Google Cloud VMware Engine

Ниже рассмотрена процедура настройки vRealize Automation для работы с гибридной облачной средой Google Cloud VMware Solution с самообслуживанием, позволяющая использовать внутри этой среды рабочие нагрузки vRealize Automation.

vRealize Automation поддерживает соединения с Google Cloud VMware Engine (GCVE) для перемещения рабочих нагрузок VMware и их выполнения в среде Google Cloud. Модуль GCVE был разработан компанией Google для поддержки интерфейса в средах VMware.

Процедура использования GCVE подробно описана в документации Google. Документацию можно найти здесь:

- Google Cloud VMware Engine — <https://cloud.google.com/vmware-engine/docs>

Чтобы использовать GCVE с vRealize Automation, необходимо настроить облачные учетные записи как для vCenter, так и для NSX-T в vRealize Automation. Процедуру настройки этих облачных учетных записей см. в следующих документах:

- Настройка облачной учетной записи vCenter — [Создание облачной учетной записи vCenter в vRealize Automation](#)
- Создание облачной учетной записи NSX-T — [Создание облачной учетной записи NSX-T в vRealize Automation](#)

Ниже описан процесс настройки среды для развертывания рабочих нагрузок vRealize Automation в GCVE.

1. Установите и настройте Google Cloud VMware Engine в соответствии с инструкциями поставщика и особенностями своей среды.
2. Создайте облачные учетные записи vCenter и NSX-T внутри развертывания vRealize Automation.

## Использование vRealize Automation с решением Oracle Cloud VMware Solution

Ниже рассмотрена процедура настройки vRealize Automation для работы с гибридной облачной средой Oracle Cloud VMware Solution с самообслуживанием, позволяющая использовать внутри этой среды рабочие нагрузки vRealize Automation.

vRealize Automation поддерживает соединение с Oracle Cloud VMware Solution (OCVS) для перемещения и запуска рабочих нагрузок VMware в среде Oracle Cloud. Модуль OCVS был разработан компанией Oracle для поддержки интерфейса в средах VMware.

Процедура использования OCVS подробно описана в документации Oracle. Документацию можно найти [здесь](#):

- Oracle Cloud VMware Solution — <https://docs.oracle.com/en-us/iaas/Content/VMware/Concepts/ocvsoverview.htm>

Чтобы использовать OCVS, необходимо настроить облачные учетные записи как для vCenter, так и для NSX-T. Процедуру настройки этих облачных учетных записей см. в следующих документах:

- Настройка облачной учетной записи vCenter — [Создание облачной учетной записи vCenter в vRealize Automation](#)
- Создание облачной учетной записи NSX-T — [Создание облачной учетной записи NSX-T в vRealize Automation](#)

Ниже описан общий процесс настройки среды для развертывания рабочих нагрузок vRealize Automation в OCVS.

1. Установите и настройте Oracle Cloud VMware Solution в соответствии с инструкциями поставщика и особенностями своей среды.
2. Создайте облачные учетные записи vCenter и NSX-T внутри развертывания vRealize Automation.

## Использование vRealize Automation с VMware Cloud on Dell EMC

Ниже рассмотрена процедура настройки vRealize Automation для работы с гибридной облачной средой VMware Cloud on Dell EMC с самообслуживанием, позволяющая использовать внутри этой среды рабочие нагрузки vRealize Automation.

vRealize Automation поддерживает соединение с VMware Cloud on Dell EMC для перемещения и запуска рабочих нагрузок VMware.

Дополнительные сведения см. в документации по VMware Cloud on Dell EMC в <https://docs.vmware.com/ru/VMware-Cloud-on-Dell-EMC/index.html>.

Для использования vRealize Automation с VMware Cloud on Dell EMC необходимо настроить облачную учетную запись vCenter. Процедуру настройки этой облачной учетной записи см. в следующих документах.

- Настройка облачной учетной записи vCenter — [Создание облачной учетной записи vCenter в vRealize Automation](#)

Ниже описан процесс настройки среды для развертывания рабочих нагрузок vRealize Automation в VMware Cloud on Dell EMC.

1. Установите и настройте VMware Cloud on Dell EMC в соответствии с инструкциями поставщика и особенностями своей среды.
2. Создайте облачную учетную запись vCenter внутри развертывания vRealize Automation.

# Создание инфраструктуры ресурсов Cloud Assembly

# 4

В инфраструктуре ресурсов Cloud Assembly регионы облачной учетной записи определяются в виде зон, в которых можно развертывать облачные шаблоны и их рабочие нагрузки.

Кроме того, инфраструктура ресурсов позволяет создавать общие сопоставления образов и размеров компьютеров, а также профили, которые определяют возможности сети и хранилищ данных в областях учетных записей облачной службы или центрах обработки данных.

В эту главу входят следующие разделы:

- Добавление облачных зон для определения целевых областей размещения или центров обработки данных в службе Cloud Assembly
- Добавление сопоставлений конфигураций ресурсов в vRealize Automation для задания стандартного объема ресурсов компьютеров
- Добавление сопоставления образов в vRealize Automation для доступа к общим операционным системам
- Как добавлять профили сети в vRealize Automation
- Как добавлять профили хранилища Cloud Assembly с учетом различных требований
- Использование карт ценообразования в vRealize Automation
- Как использовать теги для управления ресурсами и развертываниями Cloud Assembly
- Как работать с ресурсами в vRealize Automation
- Настройка ресурсов арендатора от нескольких поставщиков с помощью vRealize Automation

## Добавление облачных зон для определения целевых областей размещения или центров обработки данных в службе Cloud Assembly

Облачная зона Cloud Assembly — это набор ресурсов в рамках облачной учетной записи определенного типа, например AWS или vSphere.

Облачные зоны, заданные в том или ином регионе учетной записи, определяют размещение рабочих нагрузок при развертывании облачных шаблонов. Каждая облачная зона связана с каким-либо проектом Cloud Assembly.

Выберите пункт **Инфраструктура > Настроить > Облачные зоны** и нажмите кнопку **Добавить новую зону**.

## Дополнительные сведения об облачных зонах Cloud Assembly

Облачные зоны Cloud Assembly — это разделы вычислительных ресурсов, которые подходят к типу облачной учетной записи, например, AWS или vSphere.

Облачные зоны зависят от региона. Их необходимо назначить проекту. Между облачными зонами и проектами устанавливается отношение «многие ко многим». Cloud Assembly поддерживает развертывание в самых популярных общедоступных облаках, включая Azure, AWS и GCP, а также в среде vSphere. См. раздел [Добавление учетных записей облачной службы в Cloud Assembly](#).

Управлять размещением в облачных зонах также можно с помощью параметров политик размещения, тегов возможностей и тегов вычислительных ресурсов.

### ■ Политика размещения

Политика размещения определяет процесс выбора узлов для развертываний в указанной облачной зоне.

- **default** — вычислительные ресурсы распределяются между кластерами и компьютерами узлов на основе доступности. Например, все компьютеры в конкретном развертывании подготавливаются на первом применимом узле.
- **binpack** — вычислительные ресурсы размещаются на наиболее загруженном узле, который имеет достаточно ресурсов для выполнения указанных вычислительных операций.
- **spread** — предоставляет вычислительные ресурсы на уровне развертывания для кластера или узла с минимальным количеством виртуальных машин. Для vSphere виртуальные машины между узлами распределяет компонент Distributed Resource Scheduler (DRS). Например, все запрошенные компьютеры в развертывании размещаются в одном и том же кластере, но в следующем развертывании можно выбрать другой кластер vSphere, в зависимости от текущей нагрузки.

Предположим, у вас есть следующая конфигурация.

- Кластер DRS 1 с пятью виртуальными машинами
- Кластер DRS 2 с девятью виртуальными машинами
- Кластер DRS 3 с шестью виртуальными машинами

Если запросить кластер из трех виртуальных машин и выбирать политику Spread, они все должны быть размещены в кластере 1. Теперь в кластере 1 восемь виртуальных машин, а нагрузки в кластерах 2 и 3 остаются без изменений (9 и 6).

Затем, если запросить дополнительно две виртуальные машины, они помещаются в кластер DRS 3, в котором теперь восемь виртуальных машин. Нагрузка в кластерах 1 и 3 остается без изменений (8 и 9).

Если две облачные зоны соответствуют всем критериям, необходимым для подготовки, логика размещения выбирает зону с более высоким приоритетом.

## ■ Теги возможностей

Схемы элементов содержат теги ограничений, которые помогают определить размещение для развертывания. В ходе развертывания теги ограничений, содержащиеся в схеме элементов, сопоставляются с соответствующими тегами возможностей в облачных зонах и вычислительных ресурсах, на основании чего определяются облачные зоны, доступные для размещения ресурсов виртуальных машин.

## ■ Вычислительные ресурсы

Можно просматривать вычислительные ресурсы, доступные для подготовки рабочих нагрузок, таких как зоны доступности AWS и кластеры vCenter, для этой облачной зоны и управлять ими.

---

**Примечание** Начиная с выпуска vRealize Automation 8.3 облачные зоны больше не могут совместно использовать вычислительные ресурсы. Устаревшие облачные зоны, в которых используются общие вычислительные ресурсы, по-прежнему поддерживаются, но пользователям будет предложено обновить их в соответствии с текущими стандартами.

Облачные зоны, которые создаются автоматически в ходе создания облачной учетной записи, связываются с базовыми вычислительными ресурсами после сбора данных.

---

Если кластер вычислительных ресурсов vCenter поддерживает DRS, то облачная зона отображает только кластер в списке вычислительных ресурсов и не отображает дочерние узлы. Если кластер вычислительных ресурсов vCenter не поддерживает DRS, то облачная зона отображает только автономные узлы ESXi, если они присутствуют.

Добавьте вычислительные ресурсы в соответствии с облачной зоной. На вкладке «Вычислительные ресурсы» имеется механизм фильтрации, который позволяет управлять включением вычислительных ресурсов в облачные зоны. Первоначально выбран фильтр «Включить все вычислительные ресурсы», а в приведенном ниже списке указаны доступные вычислительные ресурсы, (все они могут использоваться в развертываниях). Существует два дополнительных варианта добавления вычислительных ресурсов в облачную зону.

- «Выбрать вычислительные ресурсы вручную». Выберите этот вариант, чтобы вручную указать вычислительные ресурсы в списке ниже. После выбора ресурсов нажмите «Добавить вычислительный ресурс», чтобы добавить ресурсы в зону. Выбранные ресурсы доступны для использования в развертываниях.
- Динамически включать вычислительные ресурсы по тегам. Выберите этот вариант, если требуется добавлять вычислительные ресурсы в зону или исключать их из нее на основе тегов. Все вычислительные ресурсы отображаются до тех пор, пока не будут добавлены теги, соответствующие существующим тегам вычислительных ресурсов. После добавления одного тега или нескольких вычислительных ресурсов с тегами, соответствующими фильтру, включаются в зону и становятся доступны для использования в развертываниях, а те, которые не соответствуют фильтру, исключаются.

Для любого варианта выбора вычислительных ресурсов, чтобы удалить один или несколько ресурсов, отображаемых на странице, выберите поле справа и нажмите «Удалить».



Теги вычислительных ресурсов обеспечивают дополнительные возможности по управлению размещением. С помощью тегов можно отфильтровать доступные вычислительные ресурсы так, чтобы отображались только те ресурсы, которые соответствуют одному или нескольким заданным тегам (см. следующие примеры).

- В вычислительных ресурсах нет тегов, фильтрация не применена.

**New Cloud Zone**

Summary **Compute** Projects

All compute resources listed apply to this cloud zone. Use the filter to add or remove resources from the list.

**Filter tags** Enter tags to filter resources ⓘ

⌵ TAGS

<input type="checkbox"/>	Name	Account / region	Type	Tags
<input type="checkbox"/>	us-east-1a	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1b	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1c	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1d	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1e	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1f	Amazon / us-east-1	Availability Zone	

6 computes

- Два вычислительных ресурса содержат один и тот же тег, фильтрация не применена.

**New Cloud Zone**

Summary **Compute** Projects

All compute resources listed apply to this cloud zone. Use the filter to add or remove resources from the list.

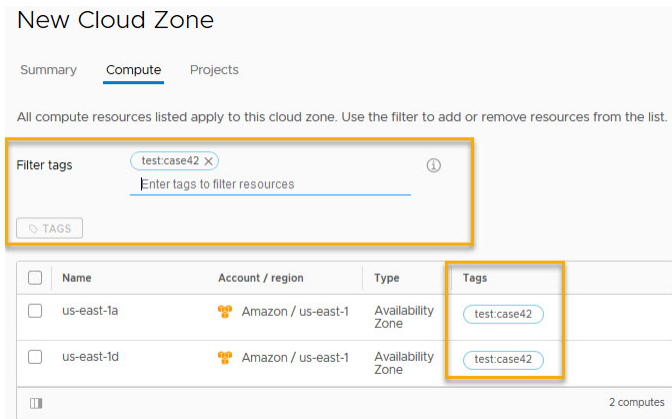
**Filter tags** Enter tags to filter resources ⓘ

⌵ TAGS

<input type="checkbox"/>	Name	Account / region	Type	Tags
<input type="checkbox"/>	us-east-1a	Amazon / us-east-1	Availability Zone	test:case42
<input type="checkbox"/>	us-east-1b	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1c	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1d	Amazon / us-east-1	Availability Zone	test:case42
<input type="checkbox"/>	us-east-1e	Amazon / us-east-1	Availability Zone	
<input type="checkbox"/>	us-east-1f	Amazon / us-east-1	Availability Zone	

6 computes

- Два вычислительных ресурса содержат один и тот же тег, применена фильтрация по данному тегу.



## ■ Проекты

Можно просмотреть список проектов, которые поддерживают предоставление рабочей нагрузки для данной облачной зоны.

После того как облачная зона создана, ее конфигурацию можно проверить.

## Панель управления ключевыми сведениями

Если у вас есть привязанное приложение vRealize Operations Manager, которое настроено для работы с vRealize Automation, вы можете использовать панель управления **Ключевые сведения** в облачной зоне. Панель управления отображает сведения о загрузке ресурсов и развертываний для облачной зоны vSphere или VMware Cloud on AWS при условии, что облачные учетные записи настроены в vRealize Automation и vRealize Operations Manager и отслеживаются в vRealize Operations Manager. Дополнительные сведения о панели управления **Ключевые сведения** см. в разделе [Управление ресурсами и оптимизация развертывания с помощью показателей vRealize Operations Manager в vRealize Automation](#).

## Добавление сопоставлений конфигураций ресурсов в vRealize Automation для задания стандартного объема ресурсов компьютеров

Сопоставление конфигураций ресурсов в vRealize Automation — это определение объемов ресурсов целевого развертывания для конкретной учетной записи облачной службы/региона с использованием естественного языка.

В сопоставлениях конфигураций выражаются размеры развертываний, которые целесообразно применять в вашей среде. Примеры конфигураций: *малая* с 1 ЦП и 2 ГБ памяти и *большая* с 2 ЦП и 8 ГБ памяти для учетной записи vCenter в именovanном центре обработки данных; «t2.nano» для учетной записи Amazon Web Services в именovanном регионе.

Перейдите в раздел **Управление арендаторами > Сопоставления конфигураций ресурсов** или **Инфраструктура > Сопоставления конфигураций ресурсов** и нажмите **Создать сопоставление конфигураций ресурсов**.

## Дополнительные сведения о сопоставлении конфигураций ресурсов в vRealize Automation

В сопоставлении конфигурации ресурсов объединяется набор параметров целевого развертывания для конкретной учетной записи облачной службы/региона в vRealize Automation с использованием имен на естественном языке.

Сопоставление конфигураций ресурсов — это именованное сопоставление, которое содержит конфигурации одинакового объема ресурсов в различных регионах учетной записи. Например, сопоставление конфигурации ресурсов с именем `standard_small` может содержать конфигурацию одинакового объема ресурсов (например, 1 ЦП, 2 ГБ ОЗУ) для некоторых или всех доступных учетных записей/регионов в проекте. При создании облачного шаблона необходимо выбрать доступную конфигурацию ресурсов, которая соответствует вашим потребностям.

Отсортируйте сопоставления конфигураций ресурсов для проекта по задачам развертывания.

Для упрощения создания облачного шаблона при добавлении новой облачной учетной записи можно выбрать параметр предварительной настройки. При выборе параметра предварительной настройки следует выбрать наиболее популярные в организации сопоставления конфигураций и образов для указанного региона.

С учетом сопоставления образов в облачных шаблонах, содержащих ресурсы vSphere, если для облачной зоны vSphere не определены сопоставления конфигураций ресурсов, можно настроить неограниченный объем памяти и ЦП, используя параметры, заданные для vSphere в облачном шаблоне. Если для облачной зоны vSphere определены сопоставления конфигураций ресурсов, такое сопоставление служит ограничением для конфигураций, предназначенных для vSphere, в облачном шаблоне.

## Добавление сопоставления образов в vRealize Automation для доступа к общим операционным системам

Сопоставление образов vRealize Automation — это определение операционных систем целевого развертывания для конкретной учетной записи облачной службы/региона с использованием естественного языка.

Выберите **Управление арендаторами > Сопоставления образов** и нажмите **Создать сопоставление образов**.

## Create Image Mapping

Account / region \*

Image name \*

Image \*

Constraints

Tenant \*

Cloud configuration

1
---

## Дополнительные сведения о сопоставлении образов в vRealize Automation

В сопоставлении образов объединяется набор заданных спецификаций целевых операционных систем для конкретной облачной учетной записи или ее региона в vRealize Automation с использованием имен на естественном языке.

Образы используются в учетных записях поставщиков облачных сред (например, Microsoft Azure и Amazon Web Services) для создания набора условий целевого развертывания, в том числе ОС и связанных параметров конфигурации. В средах на основе vCenter и NSX, в том числе VMware Cloud on AWS, используется аналогичный механизм объединения для определения набора условий развертывания ОС. При создании, развертывании и повторном использовании облачного шаблона необходимо выбрать доступный образ, который соответствует вашим требованиям.

Отсортируйте сопоставления образов для проекта по одинаковым параметрам операционных систем, правилам расстановки тегов и функциональным задачам развертывания.

Для упрощения создания облачного шаблона при добавлении новой облачной учетной записи можно выбрать параметр предварительной настройки. При выборе параметра предварительной настройки следует выбрать наиболее популярные в организации сопоставления конфигураций и образов для указанного региона.

При добавлении сведений об образе в облачный шаблон в разделе `properties` компонента компьютера используется запись `image` или `imageRef`. Например, если необходимо выполнить клонирование из моментального снимка, используйте свойство `imageRef`.

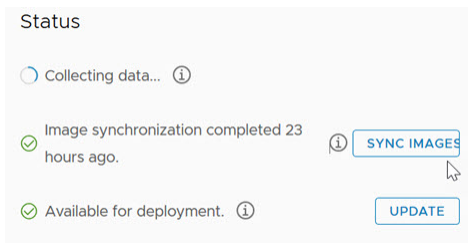
Примеры записей `image` и `imageRef` в коде облачного шаблона см. в разделе [Глава 6 Проектирование развертываний Cloud Assembly](#).

Чтобы назначить разрешение на доступ к библиотеке содержимого, администратор должен предоставить пользователю глобальное разрешение. Дополнительные сведения см. в разделе [Иерархическое наследование разрешений для библиотек содержимого](#) в документе *Администрирование виртуальных машин vSphere* в документации по VMware vSphere.

## Синхронизация образов для облачной учетной записи или ее региона

Благодаря синхронизации образов можно гарантировать, что образы, добавляемые или удаляемые для заданной облачной учетной записи или ее региона на странице **Инфраструктура > Настроить > Сопоставление образов**, являются актуальными.

1. Откройте соответствующую **облачную учетную запись или ее регион**, выбрав **Инфраструктура > Подключения > Облачные учетные записи**. Выберите существующую облачную учетную запись или ее регион.
2. Нажмите кнопку **Синхронизировать изображения** и иницилируйте действие.



3. После выполнения действия выберите **Инфраструктура > Настроить > Сопоставление образов**. Определите новое или измените существующее сопоставление образов и выберите облачную учетную запись или ее регион из шага 1.
4. Щелкните значок синхронизации образов на странице **Сопоставление образов**.



5. Настройте параметры сопоставлений образов для указанной облачной учетной записи или ее региона на странице **Сопоставление образов**.

## Просмотр сведений о файлах OVF

В объекты облачного шаблона Cloud Assembly можно включить спецификации OVF, например компоненты компьютера vCenter и сопоставления образов. Если образ содержит файл OVF, то его содержимое можно обнаружить, не открывая сам файл. Наведите указатель мыши на файл OVF для отображения сведений о нем, в том числе его имени и расположения. Дополнительные сведения о формате файла OVF см. в разделе [vcenter ovf: property](#). Для просмотра сведений о файле OVF сопоставление образов должно находиться на веб-сервере.



Дополнительную информацию о просмотре сведений о файле OVF с помощью ссылки OVF в поле сопоставления см. в статье [Облачный шаблон на основе файла OVA](#).

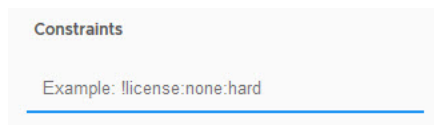
## Использование общедоступных и недавних образов из галереи образов Microsoft Azure

При создании сопоставлений образов для Microsoft Azure можно выбирать образы из общедоступной галереи Azure в подписке. Сбор и предоставление сведений об образах, доступных в раскрывающемся меню, выполняются в зависимости от выбранного региона.

Несмотря на то что общедоступные галереи образов можно использовать в рамках нескольких подписок, они не могут отображаться в раскрывающемся меню сопоставления образов в различных подписках. В списке сопоставлений образов представлены только образы для конкретной подписки. Чтобы использовать образ из галереи в другой подписке, укажите ИД образа в сопоставлении образов и используйте это сопоставление в облачном шаблоне.

## Использование ограничений и тегов для уточнения выбора образа

Для дальнейшего уточнения выбора образа в облачном шаблоне можно добавить одно ограничение или несколько, чтобы задать ограничения на основе тегов по типу развертываемого образа. Приведенным примером **ограничений**, который отображается при создании или изменении конфигурации сопоставления образов, является `!license:none:hard`. В этом примере показано ограничение на основе тегов: образ можно использовать только в том случае, если тег `license:none` *отсутствует* в облачном шаблоне. При добавлении тегов, таких как `license:88` и `license:92`, указанный образ можно использовать только в том случае, если в облачном шаблоне *присутствуют* теги `license:88` и `license:92`.



## Использование сценария настройки облачной среды для управления развертыванием

Сценарий настройки облачной среды можно использовать в сопоставлении образов, облачном шаблоне или сразу в обоих случаях для определения настраиваемых характеристик ОС, которые будут применяться в развертывании Cloud Assembly. Например, в зависимости от того, где развертывается облачный шаблон — в общедоступном или частном облаке, к образу можно применить необходимые разрешения для пользователя, разрешения для ОС или иные условия. Сценарий настройки облачной среды соответствует формату `cloud-init` для образов на основе Linux или формату `cloudbase-init` для образов на основе Windows. Cloud Assembly поддерживает средство `cloud-init` для систем Linux и средство `cloudbase-init` для Windows.

Для компьютеров под управлением Windows можно использовать любой формат сценария настройки облачной среды, поддерживаемый средством `cloudbase-init`.

Для ресурса компьютера в следующем примере кода облачного шаблона используется образ, содержащий сценарий настройки облачной среды, содержимое которого отображается в записи `image`.

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: MyUbuntu16
      https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ami-ubuntu-16.04-1.10.3-00-15269239.ova
      cloudConfig: |
        ssh_pwauth: yes
        chpasswd:
          list: |
            ${input.username}:${input.password}
          expire: false
        users:
          - default
          - name: ${input.username}
            lock_passwd: false
            sudo: ['ALL=(ALL) NOPASSWD:ALL']
            groups: [wheel, sudo, admin]
            shell: '/bin/bash'
        runcmd:
          - echo "Defaults:${input.username} !requiretty" >> /etc/sudoers.d/${input.username}
```

Оценка динамических свойств работает при использовании `cloudConfig` непосредственно в облачном шаблоне, но не поддерживается для `cloudConfig` в сопоставлении образов.

В коде облачного шаблона настройка `image` используется для создания ссылки на образ, определенный как сопоставление образов. Настройка `imageRef` используется для определения шаблона, содержащего моментальный снимок (для связанных клонов), шаблона образа или файла OVF шаблона библиотеки содержимого.

## Что происходит, когда сопоставление образов и облачный шаблон содержат сценарий настройки облачной среды

Если в облачном шаблоне, содержащем сценарий настройки облачной среды, используется сопоставление образов, содержащее другой сценарий настройки облачной среды, оба сценария объединяются. Действие слияния сначала обрабатывает содержимое сценария сопоставления образов, а затем содержимое сценария облачного шаблона с учетом того, представлены ли эти сценарии в формате `#cloud-config`.

- Для сценариев в формате `#cloud-config` при слиянии содержимое каждого модуля (например, `runcmd`, `users` и `write_files`) объединяется следующим образом.
  - Для модулей, в которых содержимое является списком, объединяются списки команд из сопоставления образов и из облачного шаблона, за исключением идентичных команд в обоих списках.

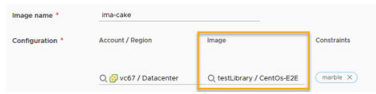
- Для модулей, в которых содержимое является словарем, объединяются команды, а результатом является комбинация обоих словарей. Если в обоих словарях имеется одинаковый ключ, то ключ из словаря сценария сопоставления образов сохраняется, а ключ из словаря сценария облачного шаблона игнорируется.
- Для модулей, в которых содержимое является строкой, значения содержимого из сценария сопоставления образов сохраняются, а значения содержимого из сценария облачного шаблона игнорируются.
- Для сценариев, которые имеют формат, отличный от `#cloud-config`, или в том случае, если один сценарий представлен в формате `#cloud-config`, а другой нет, оба сценария объединяются таким образом, чтобы сначала выполнялся сценарий сопоставления образов, а сценарий облачного шаблона запускался по окончании его выполнения.

Дополнительные сведения см. в разделе [Слияние разделов пользовательских данных](#).

## Добавление образа из библиотеки содержимого vCenter

Если локальная библиотека содержимого или библиотека содержимого издателя находится в системе vCenter, которой управляет ваша организация vRealize Automation, образы шаблонов библиотеки содержимого отображаются в раскрывающемся меню образов. Перечисленные образы включают в себя образы шаблонов OVF и виртуальных машин в локальных библиотеках содержимого vCenter или библиотеках содержимого издателя. Образы в библиотеках содержимого подписчика в раскрывающемся меню не отображаются. Шаблон, на основе которого клонирована ВМ, отображается в разделе сведений о компьютере в пользовательском интерфейсе развертываний компьютера.





**Примечание** Если управление библиотекой содержимого издателя vCenter обеспечивается службой vRealize Automation, то сведения об издателе отображаются в таблице выбора сопоставления образов в следующем формате: *publisher\_content\_library\_name / content\_item\_name*

Чтобы назначить разрешение на доступ к библиотеке содержимого, администратор должен предоставить пользователю глобальное разрешение. Дополнительные сведения см. в разделе [Иерархическое наследование разрешений для библиотек содержимого](#) в документе *Администрирование виртуальных машин vSphere* в [документации по VMware vSphere](#).

Если управление библиотекой содержимого издателя vCenter не обеспечивается службой vRealize Automation, сведения о подписчике отображаются в таблице выбора сопоставления образов в следующем формате: *subscriber\_content\_library\_name / content\_item\_name*

Например, в следующем сценарии в списке сопоставления образов vRealize Automation отображаются только элементы библиотеки содержимого подписчика:

- Для vCenter с именем VC-1 в VC есть библиотека содержимого подписчика, а облачная учетная запись создается в службе vRealize Automation, связанной с VC-1.
- Для vCenter с именем VC-2 в VC есть библиотека содержимого издателя, на которую подписана библиотека содержимого подписчика VC-1. Тем не менее в vRealize Automation нет облачной учетной записи, связанной с VC-2.

Так как VC-1 связан с облачной учетной записью vRealize Automation, библиотека содержимого подписчика доступна в vRealize Automation. Ее содержимое собирается и отображается в списке сопоставления образов vRealize Automation. Однако, поскольку VC-2 не связан с облачной учетной записью, vRealize Automation неизвестно о библиотеке содержимого издателя. Чтобы элементы библиотеки содержимого издателя отображались в списке сопоставления образов, необходимо связать облачную учетную запись с экземпляром vCenter VC-2.

При развертывании облачного шаблона, содержащего сопоставление образов шаблона ВМ, служба vRealize Automation пытается получить доступ к сопоставленному образу в библиотеке содержимого, которая находится ближе всего к хранилищу данных, а затем в библиотеке, которая находится ближе всего к узлу того компьютера, который нужно подготовить. Это может быть локальная библиотека содержимого, а также библиотека содержимого издателя или подписчика.

При развертывании облачного шаблона, который содержит сопоставление образов шаблона OVF, доступ к образам в формате OVF указывается в строке сопоставления образов, если образ находится в локальной библиотеке содержимого или у локального подписчика указанной удаленной библиотеке содержимого издателя.

Дополнительные сведения о создании и использовании библиотек содержимого vCenter см. в разделах [Использование библиотек содержимого](#) в [документации по продукту vSphere](#), а также в статье блога [Использование библиотек содержимого в vRealize Automation 8 и vRealize Automation Cloud](#).

## Дополнительные сведения о настройке и использовании сценариев настройки облачной среды

Дополнительные сведения о работе со сценариями настройки облачной среды в облачных шаблонах см. в разделе [Инициализация компьютера в Cloud Assembly](#).

См. также статьи в блоге VMware [Настройка vSphere с помощью Cloud-init при использовании vRealize Automation 8 или vRealize Automation Cloud](#) и [Настройка развертываний Cloud Assembly с помощью Cloud-init](#).

## Как добавлять профили сети в vRealize Automation

Профиль сети vRealize Automation описывает особенности сети, которую нужно развернуть.

Например, может потребоваться возможность передачи данных не только внутри сети, но и через Интернет.

Сети и их профили зависят от облачной среды.

Выберите **Инфраструктура > Настроить > Профили сети** и щелкните **Создать профиль сети**.

## Дополнительные сведения о профилях сетей в vRealize Automation

Профиль сети определяет группу сетей и их параметры, доступные для учетной записи облачной службы в определенном регионе или центре обработки данных в vRealize Automation.

Профиль сети обычно позволяет поддерживать целевую среду развертывания, например небольшую тестовую среду, в которой у существующей сети есть только исходящий доступ, или крупную производственную среду с балансировкой нагрузки, для которой требуется набор политик безопасности. Профиль сети следует рассматривать как набор характеристик сети для заданной рабочей нагрузки.

### Что содержит профиль сети

Профиль сети содержит специальную информацию о типе и регионе именованной облачной учетной записи в vRealize Automation, включая следующие параметры.

- Именованная облачная учетная запись или ее регион и необязательные теги возможностей для профиля сети.
- Именованные существующие сети и их параметры.
- Политики сети, определяющие параметры по требованию и другие особенности профиля сети.
- Необязательное включение существующих подсистем балансировки нагрузки.
- Необязательное включение существующих групп безопасности.

Функциональность управления IP-адресами сети определяется на основе профиля сети.

Теги возможностей профиля сети сопоставляются с тегами ограничений в облачных шаблонах для контроля выбора сети. Кроме того, все теги, назначенные сетям, данные о которых собираются профилем сети, также сопоставляются с тегами в облачном шаблоне, что позволяет управлять выбором сети после развертывания облачного шаблона.

Теги возможностей не являются обязательными. Теги возможностей применяются ко всем сетям в профиле сети, но только в том случае, когда сети используются в рамках данного профиля сети. Для профилей сети, которые не содержат теги возможностей, сопоставление тегов выполняется только для тегов сети. Параметры сети и безопасности, определенные в профиле соответствующей сети, применяются при развертывании облачного шаблона.

При использовании статического IP-адреса диапазоном адресов управляет vRealize Automation. Для DHCP начальным и конечным IP-адресами управляет независимый сервер DHCP, а не vRealize Automation. При использовании DHCP или смешанного выделения сетевых адресов для значения использования сети устанавливается 0. Диапазон адресов, выделяемых для обслуживаемой по требованию сети, основывается на CIDR и размере подсети, указанном в профиле сети. Для поддержки как статического, так и динамического назначения в развертывании выделяемый диапазон делится на два отдельных диапазона — один предназначен для статического выделения, а другой для динамического.

## Сети

Сети, также называемые подсетями, представляют собой логические подразделения IP-сети. Сеть объединяет в себе облачную учетную запись, IP-адрес или диапазон IP-адресов, а также теги сети, с помощью которых можно управлять процессом подготовки развертывания облачного шаблона. Сетевые параметры в профиле определяют способ обмена данными между компьютерами в развертывании по IP-протоколу на уровне 3. Сети могут содержать теги.

В профиле сети можно добавлять сети, изменять характеристики сетей, используемых профилем сети. Сети также можно удалять из профиля сети.

При добавлении сети в профиль сети можно выбрать доступные сети из отфильтрованного списка сетей vSphere и NSX. Если тип сети поддерживается для типа облачной учетной записи, его можно добавить в профиль сети.

В развертывании на основе VCF сегменты сети NSX создаются локально в сети NSX-T и не создаются как глобальные сети.

### ■ Сетевой домен или транспортная зона

Сетевой домен или транспортная зона — это распределенный виртуальный коммутатор (dvSwitch) для распределенных групп портов (dvPortGroup) в vSphere vNetwork. *Транспортная зона* — это понятие, существующее в NSX, которое похоже на *dvSwitch* или *dvPortGroup*.

При использовании облачной учетной записи NSX элемент на странице называется **транспортной зоной**, в противном случае это **сетевой домен**.

Для стандартных коммутаторов сетевой домен или транспортная зона — это то же самое, что и коммутатор. Сетевой домен или транспортная зона определяет границы подсетей в vCenter.

Транспортная зона определяет узлы, доступные логическому коммутатору NSX. Она может охватывать один или несколько кластеров vSphere. Транспортные зоны определяют, какие кластеры и виртуальные машины могут использовать конкретную сеть. Подсети, которые относятся к одной и той же транспортной зоне NSX, могут использоваться для одних и тех же узлов компьютера.

### ■ Домен

Представляет собой имя домена компьютера. Имя домена передается в спецификацию настройки компьютера vSphere.

- **IPv4 CIDR** и шлюз по умолчанию **IPv4**

Компоненты компьютера vSphere в облачном шаблоне поддерживают назначение IP-адресов IPv4, IPv6 и двухстековых IP-адресов для сетевых интерфейсов. Например, 192.168.100.14/24 представляет адрес IPv4 192.168.100.14 и связанный с ним префикс маршрутизации 192.168.100.0 или аналогично маску подсети 255.255.255.0, которая имеет 24 начальных единичных бита. Блок IPv4 192.168.100.0/22 представляет 1024 IP-адреса, от 192.168.100.0 до 192.168.103.255.

- **IPv6 CIDR** и шлюз по умолчанию **IPv6**

Компоненты компьютера vSphere в облачном шаблоне поддерживают назначение IP-адресов IPv4, IPv6 и двухстековых IP-адресов для сетевых интерфейсов. Например, блок IPv6 2001:db8::/48 представляет блок адресов IPv6 от 2001:db8:0:0:0:0:0:0 до 2001:db8:0:ffff:ffff:ffff:ffff:ffff.

Для сетей, обслуживаемых по требованию, формат IPv6 не поддерживается.

- **DNS-серверы и домены поиска DNS**

- **Поддержка общедоступного IP-адреса**

Выберите этот параметр, чтобы пометить сеть как общедоступную. Сетевые компоненты в облачном шаблоне со свойством `network type: public` сопоставляются с сетями, помеченными как общедоступные. Последующее сопоставление выполняется во время развертывания облачного шаблона для выбора сети.

- **По умолчанию для зоны**

Выберите этот параметр, чтобы пометить сеть как значение по умолчанию для облачной зоны. При развертывании облачных шаблонов предпочтение отдается сетям по умолчанию.

- **Исходный**

Определяет источник сети.

- **Теги**

Указывает один или несколько тегов, назначенных сети. Теги не являются обязательными. Сопоставление тегов определяет доступность сетей для развертывания облачных шаблонов.

Теги сети существуют в самом сетевом элементе, независимо от профиля сети. Теги сети применяются к каждому экземпляру сети, к которому они добавлены, и ко всем профилям сетей, содержащим эту сеть. Экземпляры сети могут содержаться в любом количестве профилей сетей. Независимо от местонахождения профиля сети, тег сети связан с данной сетью везде, где эта сеть используется.

При развертывании облачного шаблона теги ограничений в сетевых компонентах облачного шаблона сопоставляются с тегами сети, в том числе с тегами возможностей в профиле сети. Для профилей сетей, содержащих теги возможностей, такие теги применяются ко всем сетям, которые доступны для такого профиля. Параметры сети и безопасности, определенные в профиле соответствующей сети, применяются при развертывании облачного шаблона.

## Политики сети

Профили сети позволяют определить подсети для существующих сетевых доменов, которые содержат статические, DHCP или смешанные IP-адреса DHCP. На вкладке **Политики сети** можно определить подсети и указать параметры IP-адресов.

При использовании NSX-V, NSX-T или VMware Cloud on AWS настройки политики сети применяются, если для облачного шаблона требуется свойство `networkType: outbound` или `networkType: private` либо если для сети NSX требуется свойство `networkType: routed`.

В зависимости от связанной облачной учетной записи политики сетей можно использовать, чтобы задать параметры для типов сетей `outbound`, `private` и `routed`, а также для групп безопасности по требованию. Политики сети можно также использовать для контроля сетей `existing` при наличии подсистемы балансировки нагрузки, связанной с данной сетью.

Исходящие сети разрешают односторонний доступ к вышестоящим сетям. В частных сетях любой доступ извне не разрешается. В маршрутизируемых сетях разрешается «восточный»/«западный» трафик между маршрутизируемыми сетями. В качестве базовых или вышестоящих сетей в этом профиле используются существующие и общедоступные сети.

Параметры для следующих вариантов выбора по требованию описаны в справке на экране **Профили сети** и рассмотрены ниже.

- **Не создавать сеть по требованию или группу безопасности по требованию**

Этот параметр можно использовать при указании типа сети `existing` или `public`. Облачные шаблоны, для которых требуется сеть `outbound`, `private` или `routed`, не сопоставляются с этим профилем.

- **Создать сеть по требованию**

Этот параметр можно использовать при указании типа сети `outbound`, `private` или `routed`.

Amazon Web Services, Microsoft Azure, NSX, vSphere и VMware Cloud on AWS поддерживают этот параметр.

- **Создать группу безопасности по требованию**

Этот параметр можно использовать при указании типа сети `outbound` или `private`.

Если тип сети `outbound` или `private`, для соответствующих облачных шаблонов создается новая группа безопасности.

Amazon Web Services, Microsoft Azure, NSX и VMware Cloud on AWS поддерживают этот параметр.

Параметры политики сети могут зависеть от типа облачной учетной записи. Эти параметры описаны в указателе на экране и рассмотрены ниже.

- **Сетевой домен или транспортная зона**

Сетевой домен или транспортная зона — это распределенный виртуальный коммутатор (dvSwitch) для распределенных групп портов (dvPortGroup) в vSphere vNetwork. *Транспортная зона* — это понятие, существующее в NSX, которое похоже на *dvSwitch* или *dvPortGroup*.

При использовании облачной учетной записи NSX элемент на странице называется **транспортной зоной**, в противном случае это **сетевой домен**.

Для стандартных коммутаторов сетевой домен или транспортная зона — это то же самое, что и коммутатор. Сетевой домен или транспортная зона определяет границы подсетей в vCenter.

Транспортная зона определяет узлы, доступные логическому коммутатору NSX. Она может охватывать один или несколько кластеров vSphere. Транспортные зоны определяют, какие кластеры и виртуальные машины могут использовать конкретную сеть. Подсети, которые относятся к одной и той же транспортной зоне NSX, могут использоваться для одних и тех же узлов компьютера.

#### ■ **Внешняя подсеть**

Сети по требованию с исходящим доступом нужна внешняя подсеть с исходящим доступом. Внешняя подсеть используется для предоставления исходящего доступа, если это требование содержится в облачном шаблоне. Она не влияет на размещение сети. Например, внешняя подсеть не влияет на размещение частной сети.

#### ■ **CIDR**

Запись CIDR является компактным представлением IP-адреса и его связанного префикса маршрутизации. Значение CIDR указывает диапазон сетевых адресов, который должен использоваться во время подготовки для создания подсетей. Этот параметр CIDR на вкладке **Политики сети** поддерживает запись IPv4, которая оканчивается на /nn, и содержит значения от 0 до 32.

#### ■ **Размер подсети**

Этот параметр определяет размер обслуживаемой по требованию сети с помощью записи IPv4 для каждой изолированной сети в развертывании, в котором используется этот профиль сети. Для управления внешними или внутренними IP-адресами доступна настройка размера подсети.

Для сетей, обслуживаемых по требованию, формат IPv6 не поддерживается.

#### ■ **Распределенный логический маршрутизатор**

При использовании облачной учетной записи NSX-V для маршрутизируемой сети по требованию необходимо указать распределенную логическую сеть.

Распределенный логический маршрутизатор (DLR) используется для маршрутизации «восточного/западного» трафика между маршрутизируемыми сетями по требованию в NSX-V. Этот параметр отображается только в том случае, если значение учетной записи или ее региона для профиля сети связано с облачной учетной записью NSX-V.

#### ■ **Назначение диапазона IP-адресов**

Этот параметр доступен для облачных учетных записей, которые поддерживают NSX или VMware Cloud on AWS, в том числе vSphere.

Настройка диапазона IP-адресов доступна при использовании существующей сети с точкой интеграции внешней системы IPAM.

Чтобы указать тип назначения диапазона IP-адресов для сети развертывания, можно выбрать один из следующих параметров.

#### ■ **Статический и DHCP**

Это значение по умолчанию, которое рекомендуется к применению. Этот смешанный вариант использует выделенные параметры **CIDR** и **диапазона подсетей** для настройки пула серверов DHCP таким образом, чтобы половину выделяемого адресного пространства поддерживать при помощи метода DHCP (динамического), а другую половину пространства IP-адресов — при помощи статического метода. Используйте этот параметр, если для одних компьютеров, подключенных к сети по требованию, требуются назначенные статические IP-адреса, а для других компьютеров — динамические IP-адреса. Создаются два диапазона IP-адресов.

Этот вариант наиболее эффективен в развертываниях с компьютерами, подключенными к сети по требованию, где некоторым компьютерам назначаются статические IP-адреса, а другим — IP-адреса, динамически назначаемые сервером DHCP NSX, а также в развертываниях, где виртуальный IP-адрес подсистемы балансировки нагрузки является статическим.

#### ■ **DHCP (динамический)**

Этот вариант использует выделенную CIDR для настройки пула IP-адресов на сервере DHCP. Все IP-адреса для этой сети назначаются динамически. Для каждой выделенной CIDR создается отдельный диапазон IP-адресов.

#### ■ **Статический**

Этот вариант использует выделенную CIDR для статического выделения IP-адресов. Данный вариант следует использовать, если для этой сети не требуется настройка сервера DHCP. Для каждой выделенной CIDR создается отдельный диапазон IP-адресов.

#### ■ **Блоки IP-адресов**

Настройка блоков IP-адресов доступна при использовании обслуживаемой по требованию сети с точкой интеграции внешней системы IPAM.

С помощью параметра «Блок IP-адресов» в профиль сети можно добавить именованный блок или диапазон IP-адресов из интегрированного внешнего поставщика IPAM. Кроме того, из профиля сети можно удалить добавленный блок IP-адресов. Сведения о процедуре создания интеграции внешнего IPAM см. в [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#).

Внешний IPAM доступен для следующих типов облачных учетных записей или регионов.

- vSphere
- vSphere с NSX-T
- vSphere с NSX-V

#### ■ **Сетевые ресурсы — внешняя сеть**

Внешние сети также называются существующими сетями. Такие сети определяются на основе сбора данных и становятся доступными для выбора.

## ■ Сетевые ресурсы — логический маршрутизатор уровня 0

NSX-T использует логический маршрутизатор уровня 0 как шлюз к сетям, которые являются внешними по отношению к развертыванию NSX. Логический маршрутизатор уровня 0 настраивает исходящий доступ для сетей по требованию.

## ■ Сетевые ресурсы - пограничный кластер

Указанный пограничный кластер предоставляет услуги маршрутизации. Пограничный кластер используется для настройки исходящего доступа для сетей по требованию и подсистем балансировки нагрузки. Он определяет пограничный кластер или пул ресурсов, где должно быть развернуто пограничное устройство.

## ■ Сетевые ресурсы — пограничное хранилище данных

Указанное пограничное хранилище данных используется для подготовки пограничного устройства. Данный параметр применим только к решению NSX-V.

Теги можно использовать для указания сетей, которые доступны для облачного шаблона.

## Подсистемы балансировки нагрузки

В профиль сети можно добавлять подсистемы балансировки нагрузки. Список доступных подсистем балансировки нагрузки формируется на основе информации, полученной из облачной учетной записи источника.

Если тег для любой из подсистем балансировки нагрузки в профиле сети совпадает с тегом, который используется в компоненте подсистемы балансировки нагрузки в облачном шаблоне, эта подсистема учитывается во время развертывания. При развертывании облачного шаблона используются подсистемы балансировки нагрузки из соответствующего профиля сети.

Дополнительные сведения см. в [Использование параметров подсистемы балансировки нагрузки в профилях сетей в vRealize Automation](#) и [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Группы безопасности

После развертывания облачного шаблона группы безопасности в его профиле сети применяются к подготовленным сетевым адаптерам компьютеров. Для профиля сети, предназначенного для Amazon Web Services, группы безопасности в профиле сети доступны в том же сетевом домене (VPC), что и сети, указанные на вкладке «Сети». Если в сетевом профиле на вкладке «Сети» нет сетей, показываются все доступные группы безопасности.

Чтобы дополнительно определить параметры изоляции для сети `private` по требованию или `outbound`, можно использовать группу безопасности. Группы безопасности также применяются к сетям `existing`. Кроме того, можно назначить глобальные группы безопасности.

Перечисленные группы безопасности доступны на основании информации, полученной из облачной учетной записи источника, или добавляются как группа безопасности по требованию в облачный шаблон проекта. Дополнительные сведения см. в разделе [Ресурсы безопасности в vRealize Automation](#).



Группы безопасности применяются ко всем компьютерам в развертывании, подключенным к сети, которая соответствует профилю сети. Так как в облачном шаблоне может быть несколько сетей, каждая из которых соответствует отдельному профилю сети, для разных сетей можно использовать различные группы безопасности.

---

**Примечание** Помимо указания группы безопасности можно также выбрать сети NSX (по умолчанию) или сети vSphere либо оба варианта. При развертывании облачного шаблона vRealize Automation добавляет выделенную или указанную группу безопасности к сетевым адаптерам компьютера, подключенным к выделенной сети NSX. В группу безопасности NSX можно добавить только сетевые адаптеры компьютера, подключенные к сети NSX. Если сетевой адаптер компьютера подключен к сети vSphere, развертывание шаблона завершается сбоем.

---

Добавление тега в существующую группу безопасности позволяет использовать эту группу безопасности в компоненте `Cloud.SecurityGroup` облачного шаблона. Для использования в облачном шаблоне в группе безопасности должен быть хотя бы один тег. Дополнительные сведения см. в [Ресурсы безопасности в vRealize Automation](#) и [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Дополнительные сведения о профилях сетей, сетях, облачных шаблонах и тегах

Дополнительные сведения о сетях см. в разделе [Сетевые ресурсы в vRealize Automation](#).

Примеры кода компонента сети в облачном шаблоне см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

Примеры рабочих процессов автоматизации сети см. в разделе [Автоматизация сети с помощью Cloud Assembly и NSX](#).

Дополнительные сведения о тегах и стратегии расстановки тегов см. в разделе [Как использовать теги для управления ресурсами и развертываниями Cloud Assembly](#).

Дополнительные сведения об именовании сетевых адаптеров компьютера см. в разделе [Настройка имени контроллера сетевого интерфейса с помощью действий расширяемости](#).

## Использование параметров сети в профилях сетей и облачных шаблонах в vRealize Automation

Сети и профили сетей используются в vRealize Automation для определения поведения процесса подготовки ресурсов сети для развертываний.

Служба vRealize Automation позволяет определять профили сетей, применяемые в облаке. См. раздел [Дополнительные сведения о профилях сетей в vRealize Automation](#).

Параметры сети и настройки профиля сети позволяют контролировать метод использования IP-адресов сети в облачных шаблонах и развертываниях vRealize Automation.

## Поддержка IPv4 и IPv6 в сетях vRealize Automation

Сети vRealize Automation поддерживают одностековые сети IPv4 и IPv6 и двухстековые сети IPv4 и IPv6.

IPv6 поддерживается для существующих сетей vSphere и NSX.

IPv6 не поддерживается для подсистем балансировки нагрузки, сетей NSX по требованию или внешних сторонних поставщиков IPAM, таких как Infoblox.

## Поддержка внешних поставщиков IPAM

В дополнение к предоставляемой поддержке внутреннего IPAM можно использовать внешнего поставщика IPAM для динамического или статического выделения IP-адреса для сетей: в качестве диапазонов IP-адресов для существующих сетей в проектах облачных шаблонов и развертываниях и в качестве блоков IP-адресов для сетей по требованию в проектах облачных шаблонов и развертываниях.

Поддержка внешних поставщиков IPAM, например, Infoblox, доступна для специальных точек интеграции IPAM для поставщика, которые создаются с помощью меню **Инфраструктура > Подключения > Добавление интеграции > IPAM**.

Варианты для определения сведений об адресе внешнего поставщика IPAM доступны с помощью параметра **Добавление диапазона IP-адресов IPAM** на странице **Политики сети > Добавление диапазона IP-адресов IPAM**.

Сведения о процедуре создания внешней точки интеграции IPAM см. в разделе [Настройка интеграции внешнего поставщика IPAM в vRealize Automation](#). Пример создания точки интеграции IPAM для конкретного поставщика IPAM см. в разделе [Учебник «Настройка интеграции определенного внешнего поставщика IPAM для vRealize Automation»](#).

## Типы сетей

Сетевой компонент в облачном шаблоне может определяться одним из следующих типов `networkType`.

Тип сети	Определение
existing	<p>Выбор существующей сети, настроенной в среде базового поставщика облачных служб, например vCenter, Amazon Web Services и Microsoft Azure. Для сети по требованию <code>outbound</code> требуется существующая сеть.</p> <p>В существующей сети можно определить диапазон статических IP-адресов.</p>
public	<p>Для доступа к компьютерам в общедоступной сети можно использовать Интернет. ИТ-администратор определяет эти сети. Определение сети <code>public</code> идентично определению сети <code>existing</code> для сетей, которые разрешают передачу трафика из общедоступных сетей.</p>
private	<p>Тип сети по требованию.</p> <p>Ограничение сетевого трафика: только между ресурсами в развернутой сети. Предотвращает входящий и исходящий трафик. В NSX его можно сравнить с NAT по требованию, «один ко многим».</p>

Тип сети	Определение
outbound	<p>Тип сети по требованию.</p> <p>Ограничение сетевого трафика: только между вычислительными ресурсами в развертывании, а также односторонний исходящий сетевой трафик. В NSX его можно сравнить с NAT по требованию, «один ко многим», с внешним IP-адресом.</p>
routed	<p>Тип сети по требованию.</p> <p>Маршрутизируемые сети содержат пространство маршрутизируемых IP-адресов, разделенное по доступным подсетям, которые связаны друг с другом. Виртуальные машины, подготовленные с помощью маршрутизируемых сетей и имеющие один и тот же профиль маршрутизируемой сети, могут обмениваться данными друг с другом и с существующей сетью.</p> <p>Маршрутизируемые сети относятся к типу сети по требованию, который доступен для сетей NSX-V и NSX-T. В Microsoft Azure и Amazon Web Services такая возможность связи предоставляется по умолчанию.</p> <p>Сеть routed доступна только для спецификации облачного шаблона в компоненте сети Cloud.NSX.Network.</p>

Дополнительные сведения см. в разделе [Дополнительные сведения о сетевых ресурсах в облачных шаблонах vRealize Automation](#).

Примеры заполненных облачных шаблонов, содержащих данные компонентов сети, см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Примеры сценариев сети

При развертывании облачного шаблона, в котором используется следующая конфигурация профилей сетей, можно ожидать следующее поведение.

Тип сети или сценарий	Нет профилей сети, доступных для облачной зоны	Профили сети, доступные для облачной зоны
Нет сети	<p>Если сеть не задана в облачном шаблоне, то выбирается случайная сеть из того же региона подготовки ресурсов, к которому относится вычислительный ресурс.</p> <p>Предпочтение отдается сетям, которые помечены как используемые по умолчанию.</p> <p>Если в доступной области предоставления ресурсов нет сетей, предоставление ресурсов завершится ошибкой.</p>	<p>Сеть выбирается из подходящего профиля сети.</p> <p>Предпочтение отдается сетям, которые помечены как используемые по умолчанию.</p> <p>Если ни один из профилей сетей не соответствует критериям, то процесс предоставления завершится сбоем.</p>
Существующая сеть	<p>Если компонент сети в облачном шаблоне содержит теги ограничений, эти ограничения используются для фильтрации списка доступных сетей. Теги ограничений в компоненте сети, содержащемся в облачном шаблоне, сопоставляются с тегами сети и (при наличии) тегами ограничений профиля сети.</p> <p>В отфильтрованном списке сетей выбирается одна сеть из той же области предоставления ресурсов, что и вычислительный ресурс.</p> <p>Предпочтение отдается сетям, которые помечены как используемые по умолчанию.</p> <p>Если после фильтрации по ограничениям в области предоставления ресурсов не окажется подходящих сетей, предоставление ресурсов завершится ошибкой.</p>	<p>Сеть выбирается из подходящего профиля сети.</p> <p>Предпочтение отдается сетям, которые помечены как используемые по умолчанию.</p> <p>Если ни один из профилей сетей не соответствует критериям, то процесс предоставления завершится сбоем.</p> <p>Можно использовать ограничения сетей для фильтрации существующих сетей в профиле на основе предварительно назначенных им тегов.</p>
Общедоступная сеть	<p>Если в сети есть ограничения, то они используются для фильтрации списка доступных сетей с заданным атрибутом <code>supports public IP</code>.</p> <p>В отфильтрованном списке сетей случайным образом выбирается одна сеть из той же области предоставления ресурсов, что и вычислительный ресурс.</p> <p>Предпочтение отдается сетям, которые помечены как используемые по умолчанию.</p> <p>Если после фильтрации по ограничениям в области предоставления ресурсов не окажется подходящих сетей, предоставление ресурсов завершится ошибкой.</p>	<p>Сеть с атрибутом <code>supports public IP</code> выбирается из подходящего профиля сети.</p> <p>Предпочтение отдается сетям, которые помечены как используемые по умолчанию.</p> <p>Можно использовать ограничения сетей для фильтрации существующих сетей в профиле на основе предварительно назначенных им тегов.</p>

Тип сети или сценарий	Нет профилей сети, доступных для облачной зоны	Профили сети, доступные для облачной зоны
Частная сеть	Процесс подготовки завершается сбоем, поскольку для частных сетей требуются сведения из профиля сети.	На основе параметров в соответствующем профиле сети создается новая сеть или новая группа безопасности.  Теги ограничений сети можно использовать для фильтрации сетей и профилей сети.
Сеть исходящего трафика	Процесс подготовки завершается ошибкой, поскольку для сетей исходящего трафика требуются сведения из профиля сети.	На основе параметров в соответствующем профиле сети создается новая сеть или новая группа безопасности.  Теги ограничений сети можно использовать для фильтрации сетей и профилей сети.
Маршрутизируемая сеть по требованию	Процесс подготовки завершается ошибкой, поскольку для маршрутизируемых сетей требуются сведения из профиля сети.	В случае с NSX-V требуется выбрать настройки для распределенного логического маршрутизатора (Distributed Logical Router, DLR).  В случае с NSX-T и VMware Cloud on AWS требуется использовать аналогичные настройки по требованию (частная сеть и сеть исходящего трафика).
Пример использования Wordpress с имеющимися или общедоступными сетями	Предоставление ресурсов происходит в соответствии с описанием процесса для существующей сети или общедоступной сети.	См. описание поведения при работе с существующими сетями и общедоступными сетями выше.  См. раздел Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly».

Тип сети или сценарий	Нет профилей сети, доступных для облачной зоны	Профили сети, доступные для облачной зоны
Пример использования Wordpress с имеющимися и общедоступными сетями, а также частными сетями и сетями исходящего трафика	Процесс подготовки завершается ошибкой, поскольку для сети требуются сведения из профиля сети.	См. описания процесса работы с частной сетью или сетью исходящего трафика выше.  См. раздел Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly».
Пример использования Wordpress с подсистемой балансировки нагрузки	Процесс подготовки завершается ошибкой, поскольку для подсистемы балансировки нагрузки требуются сведения из профиля сети.  Процесс подготовки возможен при наличии существующих подсистем балансировки нагрузки.	Создается новая подсистема балансировки нагрузки на основе конфигурации профиля сети.  Можно указать существующую подсистему балансировки нагрузки, активированную в профиле сети.  Процесс подготовки завершается ошибкой, если запрос на существующую подсистему балансировки нагрузки отправлен, но она не соответствует ограничениям в профиле сети.  См. раздел Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly».

## Использование параметров группы безопасности в профилях сетей и проектах облачных шаблонов в vRealize Automation

Параметры группы безопасности можно определить и изменить в профилях сетей и проектах облачных шаблонов.

Возможности группы безопасности можно использовать по-разному.

- Существующая группа безопасности, указанная в профиле сети

В профиль сети можно добавить существующую группу безопасности. Если в проекте облачного шаблона используется этот профиль сети, компьютеры, входящие в него, объединяются в группу безопасности. Для этого метода не требуется добавлять ресурс группы безопасности в проект облачного шаблона. Кроме того, в этой конфигурации можно использовать подсистему балансировки нагрузки. Дополнительные сведения см. в [Дополнительные сведения о ресурсах подсистем балансировки нагрузки в облачных шаблонах vRealize Automation](#).

- Компонент группы безопасности, связанный с ресурсом компьютера в проекте облачного шаблона

Ресурс группы безопасности можно перетащить в проект облачного шаблона и привязать к сетевому адаптеру компьютера с помощью тегов ограничений для существующего ресурса группы безопасности в проекте облачного шаблона и для существующей группы безопасности в ресурсе, полученном в результате сбора данных. Эту связь можно также создать путем объединения объектов с помощью соединительной линии на холсте проекта облачного шаблона (данный способ похож на соединение сетей с компьютерами на холсте проекта).

При перетаскивании ресурса группы безопасности на холст проекта облачного шаблона он может иметь тип `existing` или `new`. Если это тип группы безопасности `existing`, необходимо добавить предлагаемое значение ограничения тега. Если это тип группы безопасности `new`, можно настроить правила брандмауэра.

- Существующая группа безопасности, выделенная с ограничениями тегов и связанная с сетевым адаптером компьютера в облачном шаблоне

Например, ресурс группы безопасности можно связать с сетевым адаптером компьютера (в ресурсе компьютера) в проекте облачного шаблона путем сопоставления тегов между двумя ресурсами.

В качестве примера для NSX-T, когда теги указаны в конечной точке-источнике, можно использовать теги NSX-T, указанные в приложении NSX-T. Затем можно использовать тег NSX-T, указанный в качестве ограничения для сетевого ресурса в проекте облачного шаблона, в котором сетевой ресурс подключен к сетевому адаптеру компьютера. Теги NSX-T позволяют динамически группировать компьютеры с помощью предварительно определенного NSX-T тега, который получается в режиме сбора данных из конечной точки-источника NSX-T. При создании тега NSX-T в NSX-T используйте логический порт.

- Правила брандмауэра в ресурсе группы безопасности по требованию в проекте облачного шаблона

В проекте облачного шаблона можно добавить правила брандмауэра в группу безопасности по требованию.

Сведения о доступных правилах брандмауэра см. в разделе [Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation](#).

## Подробнее

Сведения об определении групп безопасности в профилях сети см. в разделе [Дополнительные сведения о профилях сетей в vRealize Automation](#).

Сведения о просмотре и изменении параметров групп безопасности на страницах ресурсов инфраструктуры см. в разделе [Ресурсы безопасности в vRealize Automation](#).

Сведения об определении групп безопасности в проектах облачных шаблонов см. в разделе [Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation](#).

Примеры ресурсов группы безопасности в проектах облачных шаблонов см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Использование параметров подсистемы балансировки нагрузки в профилях сетей в vRealize Automation

Параметры подсистемы балансировки нагрузки можно настроить в конфигурации профиля сети.

Существующую подсистему балансировки нагрузки можно добавить в профиль сети на вкладке **Подсистема балансировки нагрузки**.

Подсистему балансировки нагрузки можно добавить в проект облачного шаблона, связав ее с профилем сети, содержащим одну подсистему балансировки нагрузки или несколько, либо напрямую с помощью ресурса подсистемы балансировки нагрузки на холсте проекта облачного шаблона или в коде проекта.

## Примеры для включения виртуального IP-адреса подсистемы балансировки нагрузки на основе использования группы безопасности в профиле сети

Существуют группы безопасности двух типов, которые можно использовать в профиле сети: существующая группа безопасности, выбираемая на вкладке **Группы безопасности**, и группа безопасности по требованию, создаваемая с помощью политики изоляции на вкладке **Политики сети**.

Если виртуальный IP-адрес подсистемы балансировки нагрузки связан с группой безопасности на основе параметров профиля сети, то конфигурацию группы безопасности предоставляет профиль сети.

В следующей таблице приведены примеры сценариев.

Ресурсы, связанные с топологией проекта облачного шаблона	Конфигурация профиля сети	Членство в группе безопасности
Одиночная подсистема балансировки нагрузки с виртуальным IP-адресом в частной сети и компьютер в той же частной сети.	В выбранном профиле сети используется политика изоляции, определенная как группа безопасности по требованию.	Сетевой адаптер компьютера и виртуальный IP-адрес подсистемы балансировки нагрузки добавляются в группу безопасности изоляции.
Одиночная подсистема балансировки нагрузки с виртуальным IP-адресом в частной сети и компьютер в той же частной сети.	В выбранном профиле сети используются существующая группа безопасности и политика изоляции, определенная как группа безопасности по требованию.	Сетевой адаптер компьютера и виртуальный IP-адрес подсистемы балансировки нагрузки добавляются в группу безопасности изоляции и существующую группу безопасности.
Двойная подсистема балансировки нагрузки с виртуальным IP-адресом в общедоступной сети и компьютер в частной сети.	В выбранном профиле сети используются существующая группа безопасности и политика изоляции, определенная как группа безопасности по требованию.	Сетевой адаптер компьютера и виртуальный IP-адрес подсистемы балансировки нагрузки добавляются в группу безопасности изоляции и существующую группу безопасности.
Двойная подсистема балансировки нагрузки с виртуальным IP-адресом в общедоступной сети и компьютер в частной сети.	В выбранном профиле сети используется существующая группа безопасности.	Сетевой адаптер компьютера и виртуальный IP-адрес подсистемы балансировки нагрузки добавляются в существующую группу безопасности.
Двойная подсистема балансировки нагрузки и виртуальный IP-адрес находятся в сети 1, а компьютер — в сети 2.	Два профиля сети: <ul style="list-style-type: none"> <li>■ Профиль сети 1: использует существующую группу безопасности 1.</li> <li>■ Профиль сети 2: использует существующую группу безопасности 2.</li> </ul>	Подсистема балансировки нагрузки находится в профиле сети 1, а компьютер — в профиле сети 2. Виртуальный IP-адрес подсистемы балансировки нагрузки добавляется в группу безопасности 1, а сетевой адаптер компьютера — в группу безопасности 2.

## Подробнее

Сведения о добавлении ресурсов подсистемы балансировки нагрузки в проект облачного шаблона см. в разделе [Дополнительные сведения о ресурсах подсистем балансировки нагрузки в облачных шаблонах vRealize Automation](#).



Примеры проектов облачных шаблонов, содержащих подсистемы балансировки нагрузки, см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Настройка поддержки сети по требованию в профиле сети для интеграции внешнего поставщика IPAM в vRealize Automation

В профиле сети можно настроить поддержку блоков IP-адресов для сети по требованию, если профиль сети используется в облачном шаблоне vRealize Automation, в котором применяется интеграция внешнего поставщика IPAM.

Используя существующую интеграцию определенного внешнего поставщика IPAM, можно предоставить сеть по требованию для создания новой сети в системе этого поставщика.

Используя этот процесс, можно настроить блок IP-адресов, а не указывать родительский CIDR (как это происходит при использовании внутреннего IPAM в vRealize Automation). Блок IP-адресов используется во время предоставления сети по требованию для сегментирования новой сети. Данные о блоках IP-адресов поступают от внешнего поставщика IPAM, если интеграция поддерживает сеть по требованию. Например, при использовании интеграции Infoblox IPAM блоки IP-адресов представляют собой контейнеры сети Infoblox.

Если в облачном шаблоне используется профиль сети по требованию и интеграция внешнего IPAM, то при развертывании данного шаблона происходят следующие события.

- У внешнего поставщика IPAM создается сеть.
- Сеть также создается в vRealize Automation и отражает новую конфигурацию сети от поставщика IPAM, включая такие параметры, как свойства CIDR и шлюза.
- IP-адрес для развернутой виртуальной машины извлекается из только что созданной сети.

В этом примере сети по требованию выполняется настройка профиля сети, чтобы при развертывании облачного шаблона можно было подготовить компьютер для сети по требованию в vSphere, используя Infoblox в качестве внешнего поставщика IPAM.

Дополнительные сведения см. в разделе [Настройка поддержки существующей сети в профиле сети для интеграции внешнего поставщика IPAM в vRealize Automation](#). Оба примера настройки сети соответствуют полному рабочему процессу для поставщика для интеграции внешнего IPAM в [Учебник «Настройка VMware Cloud on AWS для vRealize Automation»](#).

### Необходимые условия

Следующие предварительные требования относятся к пользователю, который создает или редактирует профиль сети, при этом профиль сети применяется при развертывании облачного шаблона, который содержит интеграцию поставщика IPAM. Дополнительные сведения о точках интеграции определенного поставщика IPAM см. в разделе [Настройка интеграции внешнего поставщика IPAM в vRealize Automation](#).

Данная последовательность шагов показана в контексте рабочего процесса интеграции поставщика IPAM. См. раздел [Учебник «Настройка интеграции определенного внешнего поставщика IPAM для vRealize Automation»](#).

- Убедитесь в наличии учетных данных администратора облака. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Убедитесь, что у вас есть учетная запись с внешним поставщиком IPAM, например [Infoblox](#) или [Bluecat](#), и корректные учетные данные для доступа к учетной записи организации с помощью поставщика IPAM. В данном примере рабочего процесса поставщиком IPAM является Infoblox.
- Убедитесь, что у вас есть точка интеграции поставщика IPAM и что пакет IPAM, используемый для создания интеграции IPAM, поддерживает сети по требованию. См. раздел [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#).

Несмотря на то, что пакет IPAM Infoblox поддерживает сети по требованию, если используется интеграция другого внешнего поставщика IPAM, убедитесь, что его пакет интеграции IPAM поддерживает такие сети.

#### Процедура

1. Чтобы настроить профиль сети выберите **Инфраструктура > Настроить > Профили сети**.
2. Щелкните **Создать профиль сети**.
3. Перейдите на вкладку **Сводка** и настройте следующие параметры.
  - Укажите облачную учетную запись или регион vSphere, например **vSphere-IPAM-OnDemandA/Datacenter**.  
  
В этом примере предполагается использование облачной учетной записи vSphere, которая не связана с облачной учетной записью NSX.
  - Назначьте профилю сети имя, например **Infoblox-OnDemandNP**.
  - Добавьте тег возможностей для профиля сети, например **infoblox\_ondemandA**.  
  
Запишите значение тега возможности, так как его также необходимо использовать в качестве тега ограничения облачного шаблона, чтобы создать связь с профилем сети, которая должна применяться при подготовке облачного шаблона.
4. Перейдите на вкладку **Политики сети** и настройте следующие параметры.
  - В раскрывающемся меню **Политика изоляции** выберите **Сети по требованию**.  
  
Этот вариант позволяет использовать блоки IP-адресов внешнего IPAM. В зависимости от облачной учетной записи, появляются новые варианты. Например, при использовании облачной учетной записи vSphere, связанной с облачной учетной записью NSX, появляются следующие варианты.
    - Транспортная зона

- Логический маршрутизатор уровня 0
- Пограничный кластер

В этом примере облачная учетная запись vSphere не связана с NSX, поэтому появляется пункт меню **Сетевой домен**.

- Оставьте параметр **Сетевой домен** пустым.

5. Щелкните **Внешний** в качестве **источника** управления адресами.
6. Щелкните **Добавить блок IP-адресов**. Появляется страница **Добавление блока IP-адресов для IPAM**.
7. В меню **Поставщик** на странице **Добавление блока IP-адресов для IPAM** выберите существующую интеграцию внешнего IPAM. Например, выберите точку интеграции *Infoblox Integration* из [Добавление точки интеграции внешнего поставщика IPAM Infoblox в vRealize Automation](#) в примере рабочего процесса.
8. В меню **Адресные пространства** выберите один из доступных и перечисленных блоков IP-адресов, например **10.23.118.0/24** и добавьте его.  
  
Если поставщик IPAM поддерживает адресные пространства, появляется меню **Адресные пространства**. Для интеграции Infoblox адресные пространства отображаются в виде представлений сети Infoblox.
9. Выберите **Размер подсети**, например **/29 (-6 IP-адресов)**.
10. Щелкните **Создать**.

## Результаты

Создается профиль сети, который можно использовать для подготовки сети по требованию с помощью указанной интеграции внешнего IPAM. В следующем примере облачного шаблона показан компьютер, который будет развернут в сети, определенной на основе нового профиля сети.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          assignment: static
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: private
```

```
constraints:
  - tag: infoblox_ondemandA
```

**Примечание** После развертывания облачного шаблона первая доступная сеть в указанном блоке IP-адресов извлекается и обрабатывается как CIDR сети. Если в облачном шаблоне используется сеть NSX, можно настроить CIDR для сети вручную, используя свойство сети `networkCidr`, как показано ниже, и переопределить параметры блоков IP-адресов и размера подсети, которые указаны в соответствующем профиле сети.

```
Cloud_Network_1:
  type: Cloud.Network
  properties:
    networkCidr: 10.10.0.0/16
```

## Настройка поддержки существующей сети в профиле сети для интеграции внешнего поставщика IPAM в vRealize Automation

Можно настроить в профиле сети поддержку диапазонов IP-адресов для существующей сети, если профиль сети используется в схеме элементов vRealize Automation, в которой применяется интеграция внешнего поставщика IPAM.

В разделе [Настройка сети и профиля сети для использования внешнего поставщика IPAM в существующей сети vRealize Automation](#) приведен пример с использованием образца рабочего процесса, относящегося к конкретному поставщику. Полный рабочий процесс для интеграции определенного внешнего поставщика IPAM см. в разделе [Учебник «Настройка VMware Cloud on AWS для vRealize Automation»](#).

Дополнительные сведения см. в разделе [Настройка поддержки сети по требованию в профиле сети для интеграции внешнего поставщика IPAM в vRealize Automation](#).

## Как добавлять профили хранилища Cloud Assembly с учетом различных требований

Профиль хранилища Cloud Assembly описывает тип хранилища, которое нужно развернуть.

Для хранилища данных обычно назначается профиль в соответствии с характеристиками, например уровнем обслуживания или затратами, производительностью, или назначением, например резервное копирование.

Выберите **Инфраструктура > Настроить > Профили хранилища** и щелкните **Создать профиль хранилища**.

## Дополнительные сведения о профилях хранилища в vRealize Automation

В регионе облачной учетной записи содержатся профили хранилища, которые позволяют администратору облачных систем определять хранилище для региона в vRealize Automation.

## Функции профиля хранилища

Профили хранилища включают в себя настройки дисков и средство для идентификации типа хранилища с помощью тегов возможностей. Впоследствии теги сопоставляются с ограничениями запросов службы подготовки для создания нужного хранилища во время развертывания.

Профили хранилища распределены по облачным регионам. Одна учетная запись облачной службы может включать несколько регионов, в каждом из которых может быть несколько профилей хранилища.

Возможно размещение, независимое от поставщика. Например, у вас может быть три разные учетные записи поставщика и регион в каждой из них. Каждый регион включает в себя профиль хранилища с возможностью, помеченной тегом *fast*. Во время подготовки запрос, содержащий тег жесткого ограничения *fast*, ищет соответствующую возможность *fast*, независимо от того, какое облако поставщика предоставляет ресурсы. Затем найденный соответствующий объект применяет параметры из связанного профиля хранилища при создании развертываемого элемента хранилища.

---

**Примечание** Другое облачное хранилище может иметь иные характеристики производительности, но для администратора, который его добавил, оно все равно будет относиться к категории *fast*.

---

Теги возможностей, добавляемые в профили хранилища, не должны определять фактические целевые ресурсы. Вместо этого они описывают типы хранилища. Дополнительные сведения о фактических ресурсах см. в разделе [Ресурсы хранилища в vRealize Automation](#).

## Тип подготовки по умолчанию

Тип подготовки профиля хранилища определяет только поведение по умолчанию. Эта настройка может не влиять на размещение и может быть переопределена свойством в облачном шаблоне.

Например, можно задать профиль хранилища для экономного выделения ресурсов. В большинстве случаев в результате запросов по умолчанию будут созданы хранилища экономного выделения ресурсов. Тем не менее, если для свойства `provisioningType` облачного шаблона установлено значение `eager-zero`, облачный шаблон переопределяет установленное по умолчанию экономное выделение ресурсов.

---

**Примечание** Если требуется точное управление, рекомендуется добавить теги возможностей и ограничений, помеченные для нужного типа подготовки.

---

Для типа подготовки по умолчанию свойство облачного шаблона переопределяет профиль хранилища по умолчанию, а профиль хранилища по умолчанию переопределяет значение по умолчанию из политики хранилища vCenter.

## Выделение дисков компьютерам

В проекте с несколькими облачными зонами, которые относятся к разным облачным учетным записям, диск размещается там же, где и компьютер, даже если не подключен к нему. Это позволяет объединять ресурсы и избегать сбоев, если позже потребуется присоединить диск.

Например, такой проект функционировать не будет. Облачный шаблон пытается выделить диск с помощью ограничений местоположения, но вместо этого развертывание выдает ошибку `No matching placement`.

Если нужно разместить диск в другой облачной учетной записи, используйте отдельное развертывание.

```
resources:
  Machine1:
    type: Cloud.vSphere.Machine
    properties:
      image: ubuntu
      flavor: small
      constraints:
        - tag: 'location:siteA'
  Disk1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
      constraints:
        - tag: 'location:siteB'
```

## Диски первого класса и стандартные диски

Используя параметр **Тип диска** на странице профиля хранилища или API-интерфейс службы vRealize Automation, можно создать профиль хранилища для поддержки хранилища на основе дисков первого класса (FCD) или стандартных дисков. Параметр диска первого класса позволяет создать профиль хранилища vSphere.

### ■ Диск первого класса

Диски первого класса могут существовать независимо от виртуальной машины vSphere. Для диска первого класса также существуют функции управления жизненным циклом, которые могут работать независимо от виртуальной машины. Диски первого класса доступны для vSphere 6.7 с обновлением 2 и более поздних версий и в настоящее время реализованы в vRealize Automation в виде функции, доступной только через API-интерфейс.

Дополнительные сведения о хранилище типа FCD, в том числе о возможностях, доступных через API-интерфейс в vRealize Automation, и ссылки на документацию по API-интерфейсу см. в разделе [Что можно делать с дисковым накопителем первого класса в vRealize Automation](#).

### ■ Стандартный диск

Хранилище на основе стандартного диска создается и управляется как интегрированный компонент виртуальной машины.

Сведения о стандартном дисковом хранилище см. в разделах [Что можно делать со стандартным дисковым накопителем в vRealize Automation](#) и [Что можно делать с постоянным дисковым накопителем в vRealize Automation](#).

## Шифрование диска на стороне сервера Azure

Для ресурсов Azure, если выбран вариант поддержки шифрования в профиле хранилища на основе управляемых дисков, также необходимо выбрать шифрование диска со связанным ключом. Доступные режим шифрования и ключи соответствуют наборам средств шифрования дисков, настроенным в Azure для данного расположения.

Microsoft Azure

Search resources, services, and docs (G+)

Home >

## Disk Encryption Sets

+ Add Manage view Refresh Export to CSV Open query Assign tags Feedback

Filter for any field... Subscription == R&D Resource group == all Location == all Add filter

Showing 1 to 100 of 305 records.

Name	Resource group	Location	Key
MyDES	DiskEncryptionSets	West US	WestUSKey...
MyDES1	DiskEncryptionSets	West US	WestUSKey...
MyDES10	DiskEncryptionSets	West US	WestUSKey...
MyDES100	DiskEncryptionSets	West US	WestUSKey...
MyDES101	DiskEncryptionSets	West US	WestUSKey...

Account / region \* AzureAcc / West US

Name \* SP-with-des

Description

Storage type \* Managed disks

Disk type \* Standard HDD

OS disk caching \* Read only

Data disk caching \* Read only

Supports encryption ☒

Encryption set Search for encryption set

Capability tags

CREATE CANCEL

MyDES

WestUSKeyForDisk

MyDES1

WestUSKeyForDisk

MyDES10

WestUSKeyForDisk

MyDES100

WestUSKeyForDisk

MyDES101

WestUSKeyForDisk

EncryptDiskWestUS

EncryptDiskWestUS

EncryptDiskWestUS

EncryptDiskWestUS

EncryptDiskWestUS

## Использование карт ценообразования в vRealize Automation

С помощью карт ценообразования Cloud Assembly администраторы облачных систем могут определять и назначать политику ценообразования для контроля отдельных развертываний в денежном выражении с целью управления ресурсами.

---

**Примечание** Для использования ценообразования в мультитенантных средах необходимо иметь отдельный экземпляр vRealize Operations Manager для каждого арендатора vRealize Automation.

---

В картах ценообразования определяются ставки для политики ценообразования. Затем политику ценообразования можно назначить конкретным проектам, чтобы определить общую цену. После создания конечной точки vRealize Operations Manager или CloudHealth становится доступна предварительно определенная «карта ставки по умолчанию» с ценой, равной значению цены на вкладке **Инфраструктура > Карты ценообразования**. Можно создать карты ценообразования, которые относятся только к проектам или облачным зонам. По умолчанию к проектам применяются все новые карты ценообразования.

---

**Примечание** Если изменить значение параметра **Все карты ценообразования применяются к**, все существующие назначения карт ценообразования будут удалены. Кроме того, если конечная точка vRealize Operations Manager удалена из Cloud Assembly, то все карты ценообразования и назначения также будут удалены.

---

Цена развертывания в динамике отображается на карте развертывания и в проекте в виде цены с начала месяца, которая сбрасывается до нуля в начале каждого месяца. В сведениях о развертывании содержатся данные о структуре затрат на компоненты. Эта информация, предоставляемая на уровне развертывания, предназначена администратору облачных систем. Она также может помочь рядовым участникам оценить потенциальное влияние их работы на бюджет и долгосрочную разработку.

Можно выбрать вариант отображения сведений о ценообразовании пользователям в Cloud Assembly и Service Broker, нажав кнопку **Показать сведения о ценообразовании**. Если этот параметр отключен, сведения о ценообразовании будут скрыты от пользователей Cloud Assembly и Service Broker.

### Расчет цены

Первоначальная цена, отображаемая на уровне развертывания как для вычислительных ресурсов, так и ресурсов хранилища, основывается на отраслевых эталонных значениях, а затем рассчитывается в динамике. Ставка применяется к узлам, а ставки для ЦП и памяти рассчитываются службой. Сервер пересчитывает значение цены каждые 6 часов.

Новые политики, назначения и авансовые цены начисляются во время следующего цикла сбора данных. По умолчанию цикл сбора данных выполняется каждые 5 минут. Обновление новых политик или внесение изменений в проекты и развертывания может занимать до 6 часов.



## Оценка цены для всех развертываний и проектов

Перед развертыванием элемента каталога можно использовать авансовую цену в качестве ориентировочной цены развертывания. Чтобы просмотреть цену в Cloud Assembly, необходимо иметь конечную точку интеграции vRealize Operations Manager, в которую включено ценообразование и где предварительно задана валюта.

### Daily Price Estimate



Guest OS and one time prices are excluded in this estimate.



price-service-f309c00

\$0.54



Cloud\_vSphere\_Machine\_1

\$0.53

Compute

\$0.39

Storage

\$0.03

Additional charges

\$0.11



Cloud\_vSphere\_Disk\_1

\$0.01

Storage

\$0.01

CLOSE

Для оценки авансовой цены размер загрузочного диска для каждой виртуальной машины всегда составляет 8 ГБ.

Авансовая цена развертывания — это ежедневная оценка цены на основе выделения ресурса для заданного элемента каталога перед его развертыванием. После развертывания элемента каталога можно просмотреть цену с начала месяца в виде совокупной авансовой цены на вкладках **Развертывание** и **Инфраструктура** > **Проекты**. Авансовая цена поддерживается для ресурсов частного облака, таких как компьютер vSphere и диск vSphere, элементы каталога Cloud Assembly и независимые от облачной среды элементы, в которых vCenter настроен для частного облака.

**Примечание** Авансовая цена не поддерживается для ресурсов общедоступного облака, а также ресурсов частного облака, не связанных с компьютером или диском vSphere.

Чтобы оценить затраты на развертывание, выберите элемент в каталоге и щелкните **Запрос > Рассчитать**. Если цена является приемлемой, нажмите **Отправить**.

Карты ценообразования проекта можно использовать для расчета общей цены всех проектов.

Чтобы оценить затраты на проект, на странице «Инфраструктура > Карты ценообразования» рядом с параметром **Все карты ценообразования применяются к** нажмите **Изменить** и выберите **Проекты**.

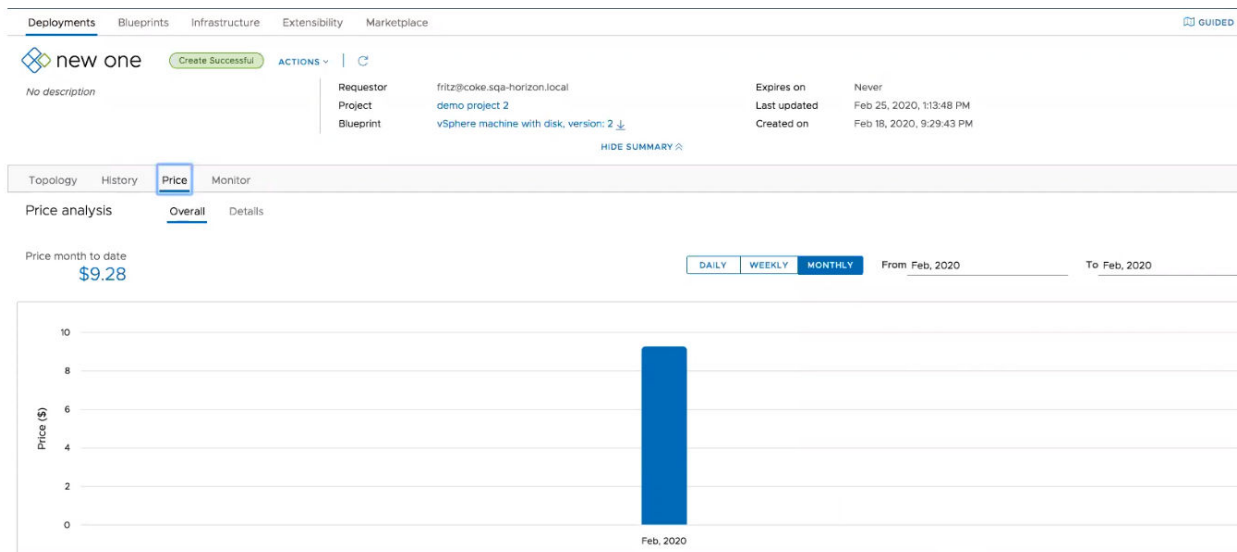
Если изменить значение параметра **Все карты ценообразования применяются к**, все существующие назначения карт ценообразования будут удалены. Создание карт ценообразования и назначений на основании затрат.

## Создание карт ценообразования для vSphere и VMC

Карту ценообразования можно создать и назначить проектам или облачным зонам в зависимости от стратегии ценообразования, которую определяет администратор облачных систем для развертываний частных облаков.

Для настройки карт ценообразования используются параметры, выбранные пользователем. После настройки карты ценообразования ее можно назначить одному или нескольким проектам и облачным зонам, которые определяются стратегией ценообразования.

Сервер ценообразования можно обновить вручную в любое время на странице «Конечная точка vROps»: **Инфраструктура > Интеграции > Конечная точка vROps > .** В разделе «Серверы vCenter» щелкните **Синхронизация**. При обновлении сервера ценообразования вручную с помощью функции **Синхронизация** пересчет цены выполняется для всех проектов в организации. В зависимости от количества проектов, существующих в организации, данный процесс может быть довольно интенсивным и занимать много времени.



После создания и назначения карты ценообразования можно просмотреть журнал цен для развертываний и проектов. Чтобы просмотреть журнал цен, перейдите к развертыванию и нажмите **Цена**. Анализ цен позволяет получить обзор и подробные сведения о цене развертывания, а также цену с начала месяца. Можно изменить графическое представление, чтобы отображались ежедневные, еженедельные или ежемесячные значения цены развертывания. Кроме того, для журнала цен можно указать точный диапазон дат или месяц.

Чтобы посмотреть разбивку цен по компонентам затрат, нажмите **Сведения**.

Цены определяются типами компонентов с расчетом затрат.

Таблица 4-1. Типы компонентов с расчетом затрат

Тип компонента схемы элементов	Имя службы/тип объекта	Тип ресурса схемы элементов	Комментарии
Независимый от облачной среды	компьютер	Cloud.Machine	Если на независимом от облака компьютере настроена среда vSphere, можно просмотреть затраты на развертывание.
	Диск	Cloud.Volume	Если к виртуальной машине, на которой настроена среда vSphere, подключен независимый диск, можно просмотреть затраты на развертывание.
vSphere	Компьютер vSphere	Cloud.vSphere.Machine	Развертывается с помощью специальной схемы элементов для данной облачной среды.
	Диск vSphere	Cloud.vSphere.Disk	Развертывается с использованием специальной схемы элементов для облачной среды, прикрепленной к виртуальной машине.
VMware Managed Cloud (VMC)	Компьютер vSphere	Cloud.vSphere.Machine	VMC поддерживает только карты ценообразования на основе ставки (карты ценообразования на основе затрат не поддерживаются).
	Диск vSphere	Cloud.vSphere.Disk	

### Необходимые условия

Перед созданием или назначением карт ценообразования необходимо настроить и включить ценообразование, а также настроить валюту в vRealize Operations для взаимодействия с vRealize Automation . При настройке взаимодействия vRealize Operations с vRealize Automation убедитесь, что в обоих приложениях указан один и тот же часовой пояс. Чтобы настроить часовой пояс в vRealize Operations, включите протокол SSH, войдите на каждый узел vRealize Operations, отредактируйте файл `$ALIVE_Base/user/conf/analytics/advanced.properties` и добавьте `timeZoneUsedInMeteringCalculation =<time zone>`.

Для использования ценообразования в мультитенантных средах необходимо иметь отдельный экземпляр vROps для каждого арендатора vRA.

Перед настройкой карт ценообразования необходимо настроить конечную точку vRealize Operations. Чтобы настроить конечную точку vRealize Operations выберите **Инфраструктура > Подключения > Интеграции > Добавить интеграцию**.

**Примечание** В случае добавления нескольких конечных точек vRealize Operations они не должны отслеживать один и тот же экземпляр vCenter.

## Процедура

1. Выберите **Инфраструктура > Карты ценообразования > Создать карту ценообразования**.
2. На вкладке «Сводка» введите имя и описание карты ценообразования. После определения политики на вкладке «Ценообразование» в таблицу «Обзор» добавляются ставки карт ценообразования.

---

**Примечание** Валюта определяется значениями, выбранными в vRealize Operations.

---

3. Необязательно. Установите флажок **По умолчанию для неназначенных проектов?**, чтобы назначить эту карту ценообразования всем неназначенным проектам по умолчанию.

#### 4. Нажмите **Ценообразование** и настройте параметры политики ценообразования.

Таблица 4-2. Конфигурация политики ценообразования

Параметр	Описание
Базовая оплата	<p>Введите имя и описание политики. Выберите «на основе затрат» или «на основе ставки».</p> <ul style="list-style-type: none"> <li>■ <b>Затраты.</b> Затраты определяются в vRealize Operations. Если установлен этот флажок, требуется указать множитель. Например, если выбрать множитель 1,1, затраты умножаются на 1,1, что приводит к увеличению рассчитанных затрат на 10%. Формула цены с использованием затрат: <math>\text{затраты} \times \text{множитель} = \text{цена}</math></li> <li>■ <b>Ставка.</b> Если установлен этот флажок, для определения затрат необходимо использовать абсолютные значения. Формула цены с использованием ставки: <math>\text{ставка} = \text{цена}</math>. Выберите интервал ставок в раскрывающемся списке, чтобы указать, как будет начисляться эта ставка.</li> </ul> <p>В разделе «Базовая оплата» определяются затраты или ставки на ЦП, память, хранилище, а также дополнительные затраты.</p>
Гостевые ОС	<p>Для определения затрат для гостевой ОС нажмите <b>Добавление оплаты</b>.</p> <p>Введите имя гостевой ОС и определите метод начисления и базовую ставку.</p> <ul style="list-style-type: none"> <li>■ <b>Повторяется.</b> Укажите базовую ставку и определите период повторения в качестве периода начисления. Требуется ввести абсолютное значение ставки, которое будет добавляться к общей цене.</li> <li>■ <b>Один раз.</b> Определите оплату в виде одноразовой базовой ставки. Необходимо указать абсолютное значение, которое добавляется в качестве одноразовой цены.</li> <li>■ <b>Коэффициент ставки.</b> Требуется множитель, который применяется к выбранной категории затрат. Например, выбран параметр «Оплата ЦП» и множитель 2. Ресурсы ЦП гостевой ОС оцениваются в два раза дороже, чем по стандартной стоимости.</li> </ul> <p>Для добавления нескольких гостевых ОС с разными ставками нажмите <b>Добавление оплаты</b> и настройте дополнительную политику оплаты.</p> <hr/> <p><b>Примечание</b> На странице сводки авансовая оплата для гостевых ОС не отображается, даже если она является частью политики.</p>

Таблица 4-2. Конфигурация политики ценообразования (продолжение)

Параметр	Описание
Теги	<p>Чтобы определить оплату на основе тега, нажмите <b>Добавление оплаты</b>.</p> <p>Выберите имя тега и определите метод начисления и базовую ставку.</p> <ul style="list-style-type: none"> <li>■ Повторяется. Укажите базовую ставку и определите период повторения в качестве периода начисления. Требуется ввести абсолютное значение ставки, которое будет добавляться к общей цене.</li> <li>■ Один раз. Определите оплату в виде одноразовой базовой ставки. Необходимо указать абсолютное значение, которое добавляется в качестве одноразовой цены.</li> <li>■ Коэффициент ставки. Требуется множитель, который применяется к выбранной категории затрат.</li> </ul> <p>Выберите способ оплаты по тегу на основе состояния «включено».</p> <p>Чтобы добавить несколько тегов с разными ставками, нажмите <b>Добавление оплаты</b> и настройте дополнительную политику оплаты.</p> <hr/> <p><b>Примечание</b> Дополнительные оплачиваемые позиции в составе окончательной рассчитываемой цены включают в себя теги виртуальных машин, но не включают теги дисков и сетей.</p>
Настраиваемые свойства	<p>Для определения затрат на настраиваемое свойство нажмите <b>Добавление оплаты</b>.</p> <p>Введите имя и значение свойства и определите метод начисления и базовую ставку.</p> <ul style="list-style-type: none"> <li>■ Повторяется. Укажите базовую ставку и определите период повторения в качестве периода начисления. Требуется ввести абсолютное значение ставки, которое будет добавляться к общей цене.</li> <li>■ Один раз. Определите оплату в виде одноразовой базовой ставки. Необходимо указать абсолютное значение, которое добавляется в качестве одноразовой цены.</li> <li>■ Коэффициент ставки. Требуется множитель, который применяется к выбранной категории затрат.</li> </ul> <p>Выберите способ оплаты за настраиваемое свойство на основе состояния «включено».</p> <p>Чтобы добавить несколько настраиваемых свойств с разными ставками, нажмите <b>Добавление оплаты</b> и настройте дополнительную политику оплаты.</p>
Общая оплата	<p>Определите все дополнительные суммы оплаты, которые нужно добавить в политику ценообразования. Можно добавить как одноразовую, так и повторяющуюся оплату.</p>

В сведениях об ориентировочной цене элемента каталога или на вкладке «Сводка» одноразовая оплата не отображается. Отображается только оценка ежедневной цены для заданного элемента каталога.

5. Перейдите на вкладку **Назначения** и нажмите кнопку **Назначить проекты**. Выберите один или несколько проектов, которым нужно назначить карту ценообразования.

---

**Примечание** Карты ценообразования по умолчанию применяются к проектам. На вкладке **Инфраструктура > Карты ценообразования** можно выбрать вариант применения карт ценообразования к облачным зонам. Если выбраны облачные зоны, нажмите **Назначение облачных зон** на вкладке «Назначения».

---

6. Нажмите **Создать**, чтобы сохранить и создать политику ценообразования.

#### Результаты

Новая политика ценообразования появится на странице «Карты ценообразования». Чтобы просмотреть или изменить сведения о политике и конфигурацию, нажмите **Открыть**.

## Как использовать теги для управления ресурсами и развертываниями Cloud Assembly

Теги — это важнейшие компоненты Cloud Assembly, обеспечивающие размещение развертываний с учетом возможностей и ограничений. Для оптимального применения Cloud Assembly необходимо хорошо понимать теги и эффективно их использовать.

По сути, теги — это метки, добавляемые к элементам Cloud Assembly. На практике можно создать любые теги, которые будут соответствовать требованиям организации и среды. Теги выполняют более широкие функции, чем метки, так как они определяют, как и где Cloud Assembly использует ресурсы и инфраструктуру для сборки развертываемых служб. Теги также позволяют управлять внутренними элементами Cloud Assembly.

### Структура тегов

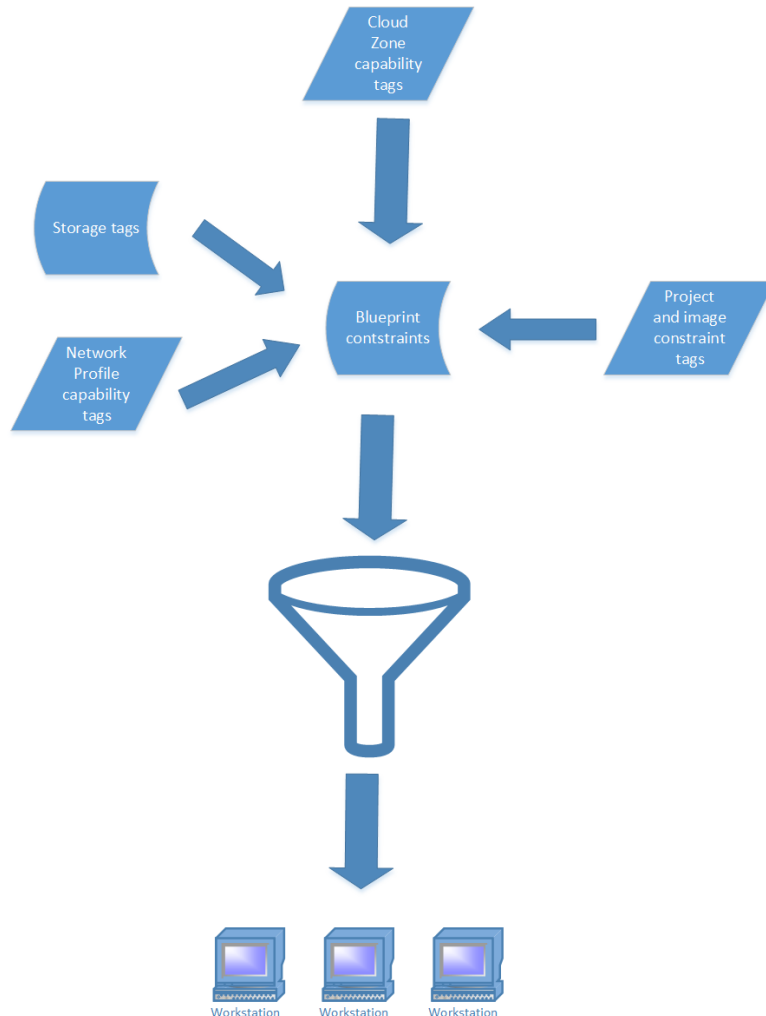
Структурно теги должны соответствовать правилам пар `name:value`, но во всем остальном их структура имеет достаточно произвольную форму. В Cloud Assembly все теги имеют одинаковый вид, а их возможности определяются контекстом.

Например, теги ресурсов инфраструктуры, прежде всего, являются тегами возможностей, так как Cloud Assembly использует их для сопоставления ресурсов с развертываниями. Кроме того, они также идентифицируют ресурсы.

### Функция тегов

Теги предназначены прежде всего для указания возможностей и ограничений, которые используются в Cloud Assembly для определения развертываний. Контекст определяет функцию тегов. Теги, размещенные в облачных зонах, профилях сети и хранилища, а также в отдельных ресурсах инфраструктуры, используются как теги возможностей и определяют функции элементов инфраструктуры, используемых в развертываниях. Теги, размещенные в облачных шаблонах, действуют как ограничения, определяющие ресурсы для развертываний. Кроме того, администраторы облачных систем могут применять теги ограничений для управления проектами. Эти теги ограничений добавляются к другим ограничениям, указанным в облачных шаблонах.

Во время предоставления Cloud Assembly сопоставляет эти возможности с ограничениями, также представленными в виде тегов, в облачных шаблонах для определения конфигурации развертывания. Функции возможностей и ограничений на основе тегов служат базой для настройки развертывания в Cloud Assembly. Например, можно использовать теги, чтобы сделать инфраструктуру доступной только для ресурсов PCI в определенном регионе.



Вторичная функция тегов — упрощение поиска и идентификации элементов хранилища и сети, а также других ресурсов инфраструктуры.

Предположим, вы настраиваете облачные зоны и у вас множество вычислительных ресурсов. Если вычислительным ресурсам надлежащим образом присвоены теги, можно использовать функцию поиска на вкладке «Вычисления» на странице «Облачная зона», чтобы отфильтровать ресурсы, связанные с определенной облачной зоной.

Кроме того, на странице «Управление тегами Cloud Assembly» и страницах настройки ресурсов содержатся функции поиска, позволяющие находить элементы по именам тегов. Использование логических и доступных для восприятия тегов для этих элементов является ключом к упрощению функции поиска и идентификации.



Для получения дополнительной информации и примеров использования тегов ознакомьтесь со следующим видео в Youtube. <https://youtu.be/4zNQ33RyQio>

## Внешние теги

Cloud Assembly также может содержать внешние теги. Эти теги автоматически импортируются из учетных записей облачной службы, которые связаны с экземпляром Cloud Assembly. Эти теги могут импортироваться из vSphere, AWS, Azure или других внешних программных продуктов. После импорта эти теги будут доступны для использования точно так же, как теги, созданные пользователем.

## Управление тегами

На странице «Управление тегами» в Cloud Assembly можно отслеживать библиотеку тегов и управлять ею. Кроме того, на этой странице можно создавать теги. «Управление тегами» также является единственной страницей, на которой можно просматривать и идентифицировать внешние теги.

Key	Value
a	
AAA	sofiaaaa
aktag1	vafl
alex	kris
AppID	ABC
AppID	XYZ
applicationtier	tango-machine
Application Tier	tango-machine
Application Tier	
astoyanov-rp	
Atos-Tagging-Category	Atos-Storage-Tag
AutomaticCleanExpirationTime	2019-01-08T08:45:33.127Z

## Стратегия расстановки тегов

Чтобы свести к минимуму путаницу, перед созданием тегов в Cloud Assembly разработайте соответствующую стратегию и правила расстановки тегов, чтобы все пользователи, создающие и использующие теги, понимали, что они означают и как используются. См. раздел [Создание политики расстановки тегов](#).

## Создание политики расстановки тегов

Тщательное планирование и внедрение адекватной политики расстановки тегов, отвечающей ИТ-структуре и целям организации, позволяет повысить эффективность использования Cloud Assembly и избежать возможной путаницы.

При формулировании политики расстановки тегов можно руководствоваться определенными общими принципами, однако следует помнить, что эта политика должна соответствовать потребностям, структуре и целям конкретного развертывания.

## Практические рекомендации по расстановке тегов

Некоторые общие характеристики эффективной политики расстановки тегов

- Необходимо разработать и внедрить согласованную политику расстановки тегов, отвечающую структуре организации, и донести эту стратегию до всех заинтересованных пользователей. Стратегия должна отвечать потребностям развертываний. Она должна быть написана четким и простым языком, понятным всем заинтересованным пользователям.
- Имена и значения, используемые для тегов, должны быть простыми, четкими и осмысленными. Например, имена тегов для элементов хранилищ и сетей должны быть четкими и последовательными, чтобы пользователи могли сразу понять, что они выбирают, просматривая теги, назначенные для развернутого ресурса.
- При создании тегов можно использовать имена без значений, однако все же рекомендуется создать для каждого имени тега подходящее значение, поскольку это упрощает работу с тегами для других пользователей.
- Избегайте создания повторяющихся или лишних тегов. Например, создавайте теги для элементов хранилища, которые относятся только к хранению данных.

## Внедрение политики расстановки тегов

Составьте план базовой политики расстановки тегов, отражающий основные факторы, которые эта политика должна учитывать. В следующем списке приведены распространенные факторы, которые следует учитывать при планировании такой политики. Обратите внимание, что эти факторы приведены для наглядности и не являются определяющими. То, какие факторы следует учитывать, определяется особенностями деятельности конкретной организации. Итоговая политика должна отвечать конкретным сферам ее применения.

- Число отдельных сред, в которых выполняются развертывания. Как правило, создается набор тегов, каждый из которых служит для обозначения определенной среды.
- Структура вычислительных ресурсов и их использование для поддержки развертываний.
- Число различных областей и расположений, в которых выполняются развертывания. Как правило, создается набор тегов на уровне профиля, каждый из которых служит для обозначения определенной области или расположения.
- Число вариантов хранилищ, доступных для развертываний, и их характеристики. Каждому из этих вариантов и характеристик следует назначить уникальный тег.
- Разбейте доступные варианты сетевых подключений по категориям и создайте теги для учета всех применимых вариантов.

- Стандартные переменные развертываний. Например, число различных сред, в которых выполняются развертывания. Как правило, в большинстве организаций используется как минимум три среды: тестовая среда, среда разработки и производственная среда. Чтобы упростить настройку развертываний в одной или нескольких средах, требуется создать теги ограничений и теги возможностей в облачных зонах и обеспечить их согласованность.
- Необходимо обеспечить согласованность тегов для ресурсов сетей и хранилищ, чтобы они выглядели адекватно в контексте профилей сетей и хранилищ, в которых они используются. Теги ресурсов можно использовать для управления развертыванием ресурсов на более детальном уровне.
- Необходимо обеспечить согласованность тегов возможностей в облачных зонах и сетевых профилях, а также прочих тегов возможностей с тегами ограничений. Как правило, администратор сначала создает теги возможностей для облачных зон и профилей сети, а затем пользователи могут создать облачные шаблоны с ограничениями, соответствующими этим тегам.

Когда все факторы, играющие роль для организации, были приняты в расчет, можно приступить к планированию имен для тегов, отвечающих этим факторам. После этого создается общая структура политики, доступ к которой предоставляется всем пользователям, имеющим право создавать или редактировать теги.

Начать внедрение политики можно с индивидуальной расстановки тегов во всех ресурсах вычислительной инфраструктуры. Как отмечалось выше, для имен тегов, относящихся к конкретному ресурсу, следует использовать соответствующие категории. Например, ресурсам хранилища можно назначить такие теги, как tier1, tier2 и т. д. Кроме того, вычислительным ресурсам можно назначать теги в соответствии с используемой в них операционной системой: Windows, Linux и т. д.

Добавив теги ко всем ресурсам, можно приступить к разработке подхода по созданию тегов для облачных зон, хранилищ и профилей сетей, который будет наилучшим образом отвечать потребностям организации.

## Использование тегов возможностей в Cloud Assembly

В Cloud Assembly с помощью тегов возможностей можно определить возможности развертывания для компонентов инфраструктуры. Вместе с ограничениями они служат основой для организации размещения в vRealize Automation.

Теги возможностей можно создавать в вычислительных ресурсах, облачных зонах, образах и картах образов, а также в сетях и профилях сетей. Страницы для создания этих ресурсов содержат параметры для создания тегов возможностей. Кроме того, для создания тегов возможностей в Cloud Assembly можно использовать страницу «Управление тегами». Теги возможностей в облачных зонах и профилях сети влияют на все ресурсы в этих зонах или профилях. Теги возможностей в хранилище или сетевых компонентах влияют только на те компоненты, к которым они применяются.

Как правило, теги возможностей могут определять такие характеристики, как расположение вычислительного ресурса, тип адаптера сети или уровень ресурса хранилища. Кроме того, они могут определять расположение или тип среды, а также любые другие вопросы, касающиеся бизнеса. Как и в общей стратегии тегирования, теги возможностей необходимо организовать логически с учетом бизнес-потребностей.

Cloud Assembly сопоставляет в ходе развертывания теги возможностей в облачных зонах с ограничениями в облачных шаблонах. Таким образом, при создании и использовании тегов возможностей необходимо предусматривать и планировать создание соответствующих ограничений облачных шаблонов, чтобы сопоставление выполнялось.

Например, в разделе, посвященном облачным зонам, в [Часть 1. Настройка примера инфраструктуры Cloud Assembly](#) из документации описано создание тегов dev и test для облачных зон OurCo-AWS-US-East и OurCo-AWS-US-West. В данном учебнике эти теги указывают, что зона OurCo-AWS-US-East — это среда разработки, а зона OurCo-AWS-US-West — среда тестирования. Если создать аналогичные теги ограничений в облачных шаблонах, такие теги возможностей позволяют направлять развертывания в нужные среды.

## Наследование тегов

Cloud Assembly использует наследование тегов, чтобы теги в облачных учетных записях выборочно переносились в другой связанный ресурс. В частности, если создать теги в облачной учетной записи, они также появятся во всех профилях хранилища и вычислительных ресурсах, которые связаны с этой облачной учетной записью.

---

**Примечание** Поведение при распространении тегов не применяется к профилям хранилища. vRealize Automation не будет автоматически выбирать ограничение для профилей хранилища, поэтому пользователи должны вручную добавить требуемый тег ограничения, чтобы он был выбран и применен к профилям хранилища.

---

Следующий пример демонстрирует, как работает наследование тегов.

Вычислительные ресурсы

- Кластер 1 с тегом cluster-1
- Кластер 2 с тегом cluster-2
- Кластер 3 с тегом cluster-3

```
Vm resource:
  properties:
    constraints:
      - tag: 'cluster-01'
```

Профили хранилища

- Профиль 1 для кластера хранилища данных 1 с тегом storage-01
- Профиль 2 для кластера хранилища данных 2 с тегом storage-02
- Профиль 3 для кластера хранилища данных 3 с тегом storage-03

```
vm-resource:
  properties:
    storage:
      constraints:
        - tag: 'storage-01'
```

## Учетная запись облачной службы

Облачная учетная запись vSphere со всеми тремя тегами: cluster-1, cluster-2 и cluster-3

При консолидации тегов в профилях хранилища и вычислительных ресурсах служба Cloud Assembly также учитывает теги уровня облачной учетной записи. Таким образом, действующими тегами во всех профилях хранилища и вычислительных системах являются cluster-1, cluster-2 и cluster-3. Поэтому, когда какой-либо из этих тегов предоставляется, как показано в предыдущем примере, все профили хранилища и вычислительные ресурсы могут быть размещены в любом из вычислительных узлов.

Любой тег должен применяться только на уровне облачной учетной записи, если он подходит для всех подчиненных ресурсов вычисления и хранения. Это поможет избежать неожиданных результатов и лишних тегов.

## Использование тегов ограничений в службе Cloud Assembly

Теги, добавленные в проекты и облачные шаблоны, работают как теги ограничений, если они используются для сопоставления тегов возможностей в ресурсах инфраструктуры, профилях и облачных зонах. В случае облачных шаблонов служба Cloud Assembly использует функцию сопоставления для выделения ресурсов развертываниям.

Cloud Assembly позволяет использовать теги ограничений двумя основными способами. Во-первых, при настройке проектов и образов. Теги можно использовать в качестве ограничений, чтобы связать ресурсы с проектом или образом. Во-вторых, при работе с облачными шаблонами, где теги, указанные в качестве ограничений, используются для выбора ресурсов для развертываний. Ограничения, применяемые в обоих случаях, объединяются в облачных шаблонах в набор требований к развертыванию, которые определяют ресурсы, доступные в нем.

### Использование тегов ограничений в проектах

При настройке ресурсов Cloud Assembly администраторы облачных систем могут применять теги ограничений для проектов. Это позволяет им внедрять управляющие ограничения непосредственно на уровне проекта. Все ограничения, добавленные на этом уровне, применяются ко всем облачным шаблонам, запрошенным для заданного проекта, а эти теги ограничений имеют более высокий приоритет, чем другие теги.

Если теги ограничений в проекте конфликтуют с тегами ограничений в облачном шаблоне, теги проекта получают более высокий приоритет, благодаря чему администратор облачных систем может принудительно внедрять управляющие правила. Например, если администраторы облачных систем создают в проекте тег `location:london`, а разработчик помещает в облачный шаблон тег `location:boston`, то первый тег получает более высокий приоритет, и развертывание ресурса выполняется в инфраструктуре, содержащей тег `location:london`.

Существуют три типа тегов ограничений, которые пользователи могут применять в проектах: «сеть», «хранилище» и «расширяемость». Для каждого типа тега можно применить любое нужное количество экземпляров. Ограничения проекта могут быть жесткими или мягкими. По умолчанию все ограничения являются жесткими. Жесткие ограничения позволяют внедрять строгие ограничения при развертывании.

Если одно или несколько жестких ограничений не выполняются, развертывание завершается ошибкой. Мягкие ограничения позволяют зафиксировать предпочтения, которые будут реализованы при наличии такой возможности, но если такой возможности не будет, это не приведет к неудачному завершению развертывания.

## Использование тегов ограничений в облачных шаблонах

В облачных шаблонах теги ограничений, соответствующие тегам возможностей, которые были определены администратором облака в ресурсах, облачных зонах и профилях сетей, добавляются к ресурсам в виде кода YAML. Существуют и другие, более сложные варианты внедрения тегов ограничений. Например, можно использовать переменную для заполнения одного или нескольких тегов по запросу. Таким образом можно задать один или несколько тегов при отправке запроса.

В коде YAML облачных шаблонов теги ограничений создаются с помощью метки `tag` в заголовке ограничения. К тегам ограничений, созданным в облачных шаблонах, добавляются теги ограничений из проектов.

Служба Cloud Assembly поддерживает простой строковый формат, что позволяет упростить использование ограничений в файлах YAML:

```
[!tag_key[:tag_value][:hard|soft]
```

По умолчанию в службе Cloud Assembly создается положительное ограничение с жестким применением. Хотя значение тега не является обязательным, его рекомендуется указывать, как и в других частях приложения.

В следующем примере использования WordPress с MySQL показаны теги ограничений YAML, задающие сведения о расположении вычислительных ресурсов.

```
name: "wordpressWithMySQL"
components:
  mysql:
    type: "Compute"
    data:
      name: "mysql"
      # ... skipped lines ...
  wordpress:
    type: "Compute"
    data:
      name: "wordpress"
      instanceType: small
      imageType: "ubuntu-server-1604"
      constraints:
        - tag: "!location:eu:hard"
        - tag: "location:us:soft"
        - tag: "!pci"
      # ... skipped lines ...
```

Дополнительные сведения о работе с облачными шаблонами см. в разделе [Часть 3. Проектирование и развертывание примера шаблона Cloud Assembly](#).

## Использование жестких и мягких ограничений в проектах и облачных шаблонах

Ограничения, используемые в проектах и облачных шаблонах, могут быть жесткими или мягкими. В приведенном выше фрагменте кода можно увидеть примеры как жестких, так и мягких ограничений. По умолчанию все ограничения являются жесткими. Жесткие ограничения позволяют внедрять строгие ограничения при развертывании. Если одно или несколько жестких ограничений не выполняются, развертывание завершается ошибкой. Мягкие ограничения позволяют зафиксировать предпочтения, которые будут реализованы при наличии такой возможности, но, если такой возможности не будет, это не приведет к сбою развертывания.

Если определенному типу ресурсов назначен набор из жестких и мягких ограничений, мягкие ограничения могут играть роль определяющих. Это значит, что если несколько ресурсов соответствуют определенному жесткому ограничению, то окончательный выбор ресурса, используемого в развертывании, определяется мягкими ограничениями.

Предположим, например, что создается жесткое ограничение для хранилища с тегом `location:boston`. Если в проекте нет хранилища, соответствующего этому ограничению, любое связанное с ним развертывание будет завершаться ошибкой.

## Стандартные теги

Cloud Assembly применяет стандартные теги к некоторым развертываниям для обеспечения анализа, мониторинга и объединения развернутых ресурсов.

Стандартные теги в Cloud Assembly не повторяются. В отличие от других тегов пользователи не работают с ними во время настройки развертывания, и никакие ограничения к ним не применяются. Эти теги применяются автоматически во время подготовки в развертываниях AWS, Azure и vSphere. Эти теги хранятся в виде настраиваемых свойств системы и добавляются в развертывания после подготовки.

Ниже приведен список стандартных тегов.

Таблица 4-3. Стандартные теги

Описание	Тег
Организация	<code>org:orgID</code>
Проект	<code>project:projectID</code>
Инициатор запроса	<code>requester:username</code>
Развертывание	<code>deployment:deploymentID</code>
Ссылка на облачный шаблон (если применимо)	<code>blueprint:blueprintID</code>
Имя компонента в схеме элементов	<code>blueprintResourceName:CloudMachine_1</code>
Ограничения на размещение: применяются в схеме элементов, параметрах запроса или с помощью ИТ-политики	<code>constraints:key:value:soft</code>
Учетная запись облачной службы	<code>cloudAccount:accountID</code>
Зона или профиль (если применимо)	<code>zone:zoneID, networkProfile:profileID, storageProfile:profileID</code>

## Как Cloud Assembly обрабатывает теги

В Cloud Assembly теги показывают возможности и ограничения, которые определяют, как и где выделяются ресурсы для подготовленных развертываний во время подготовки.

Для создания подготовленных развертываний в Cloud Assembly для обработки тегов используется определенная процедура и иерархия операций. Понимание основ данного процесса поможет эффективно реализовать теги для создания предсказуемых развертываний.

В следующем списке указаны высокоуровневые операции и последовательность, которые Cloud Assembly использует для обработки тегов и определения развертывания.

- Для фильтрации облачных зон применяются различные критерии, включая доступность и профили. В этой точке сопоставляются теги в профилях для области, к которой относится зона.
- Теги возможностей зоны и вычислительных ресурсов используются для фильтрации оставшихся облачных зон по жестким ограничениям.
- В массиве отфильтрованных зон для выбора облачной зоны используется приоритет. Если имеется несколько облачных зон с одинаковым приоритетом, они сортируются с помощью мягких ограничений, для чего применяется комбинация возможностей облачных зон и вычислительных ресурсов.
- После выбора облачной зоны выбирается узел с помощью различных фильтров, в том числе жестких и мягких ограничений, присутствующих в облачных шаблонах.

## Настройка простой структуры расстановки тегов

В этом разделе описываются основы и параметры логической стратегии расстановки тегов в Cloud Assembly. Эти примеры можно использовать в качестве отправной точки для фактического развертывания, или можно разработать другую стратегию, более соответствующую вашим требованиям.

Как правило, администратор облачных систем является главным пользователем, отвечающим за создание и поддержку тегов.

В этом разделе рассматривается сценарий использования WordPress, который приводится и в других разделах документации по Cloud Assembly; в нем показано, как добавлять теги в некоторые ключевые элементы. Кроме того, здесь описываются возможные альтернативы и расширенные варианты расстановки тегов, которые встречаются в сценарии использования WordPress.

Дополнительные сведения о сценарии использования WordPress см. в разделе [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).

В сценарии использования WordPress описываются способы размещения тегов в облачных зонах, хранилищах и профилях сетей. Эти профили похожи на структурированные пакеты ресурсов. Теги, размещаемые в профилях, применяются ко всем элементам в профиле. Кроме того, можно создать и разместить теги в ресурсах хранилища и отдельных сетевых элементах, а также в вычислительных ресурсах, но такие теги применяются только к тем ресурсам, в которых они размещены. При настройке тегов обычно лучше всего начинать с тегирования вычислительных ресурсов, а затем добавлять теги в профили и облачные зоны. Кроме того, с помощью этих тегов можно фильтровать список вычислительных ресурсов для облачной зоны.



Например, теги можно присваивать не только профилям хранилища, как показано в этом примере, но и отдельным политикам хранения, хранилищам данных и учетным записям хранилищ. Теги в этих ресурсах позволяют более точно контролировать развертывание ресурсов хранилища. Во время обработки данных при подготовке к развертыванию эти теги становятся следующим уровнем обработки после тегов профилей.

В качестве примера настройки типового сценария клиента можно разместить тег `region: eastern` в профиле сети. Этот тег будет применяться ко всем ресурсам в данном профиле. Затем можно разместить тег `networktype: pci` в сетевом ресурсе "pci" в рамках профиля. Облачный шаблон с ограничениями «eastern» и «pci» приведет к созданию развертываний, использующих эту сеть pci для региона eastern.

## Процедура

1. Отметьте тегами ресурсы вычислительной инфраструктуры в соответствии с логическими связями.

Крайне важно присваивать вычислительным ресурсам логически обусловленные теги, чтобы их можно было найти с помощью функции поиска на вкладке «Вычислительные ресурсы» на странице «Создание облачной зоны». С помощью этой функции поиска можно быстро отфильтровать вычислительные ресурсы, связанные с облачной зоной. Если тегировать хранилище и сети на уровне профиля, возможно, не потребуется тегировать отдельные ресурсы хранилища и сети.

- а) Для просмотра вычислительных ресурсов, импортированных для экземпляра Cloud Assembly, выберите **Ресурсы > Вычислительные ресурсы**.
- б) Выберите необходимые вычислительные ресурсы и щелкните **Теги**, чтобы добавить тег к ресурсу. При необходимости к каждому ресурсу можно добавить несколько тегов.
- в) Если необходимо, повторите предыдущий шаг для ресурсов хранилища и сети.

2. Создайте теги возможностей для облачной зоны и профиля сети.

Можно использовать одни и те же теги как для облачных зон, так и для профилей сети или создать уникальные теги для каждого элемента, если это имеет смысл для конкретной реализации.

В профилях сети теги можно присваивать профилю в целом или подсетям в профиле. Теги, примененные на уровне профиля, относятся ко всем компонентам (например, к подсетям) в этом профиле. Теги для подсетей применяются только к определенной подсети, которой они присвоены. Во время обработки теги на уровне профиля имеют более высокий приоритет, чем теги на уровне подсети.

Дополнительные сведения о добавлении тегов в облачные зоны и профили сети см. в разделах, посвященных облачным зонам и сети, в [Часть 1. Настройка примера инфраструктуры Cloud Assembly](#).

В этом примере создаются три простых тега, применяемых в документации по примеру использования для тегов облачной зоны и профиля сети в Cloud Assembly. Эти теги указывают среду для компонентов профиля.

- `zone:test`
- `zone:dev`
- `zone:prod`

### 3. Создайте теги профиля хранилища для компонентов хранилища.

Как правило, теги хранилища указывают уровень производительности элементов хранилища, например tier1 или tier2, или характер элементов хранилища, например pci.

Дополнительные сведения о добавлении тегов в профили хранилища см. в разделе, посвященном хранилищу, в [Часть 1. Настройка примера инфраструктуры Cloud Assembly](#).

- usage:general
- usage:fast

#### Результаты

После создания базовой структуры тегирования можно начать работу с ней, добавляя или изменяя теги соответствующим образом для уточнения и расширения возможностей тегирования.

## Как работать с ресурсами в vRealize Automation

Администратор облачных систем может просматривать ресурсы vRealize Automation, список которых можно получить путем сбора данных.

Администратор облачных систем может отмечать ресурсы с помощью тегов возможностей, которые влияют на место развертывания облачных шаблонов vRealize Automation.

В дополнение к представленным здесь представлениям управлять различными ресурсами также можно на вкладке «Ресурсы». См. раздел [Управление ресурсами в Cloud Assembly](#).

## Вычислительные ресурсы в vRealize Automation

Администратор облачных систем может просматривать вычислительные ресурсы, сведения о которых получены путем сбора данных.

Администратор облачных систем может применить теги непосредственно к ресурсам, чтобы отметить возможности для соответствующих целей в ходе подготовки vRealize Automation.

## Сетевые ресурсы в vRealize Automation

В vRealize Automation администраторы облачных систем могут просматривать и редактировать ресурсы сети, данные о которых были получены из облачных учетных записей и интеграций, сопоставленных с проектом.

После добавления облачной учетной записи в инфраструктуру Cloud Assembly, например при выборе пунктов меню **Инфраструктура > Подключения > Облачные учетные записи**, при сборе данных обнаруживаются данные о сети и безопасности для этой учетной записи. Эта информация доступна для использования в сетях, профилях сети и других определениях.

Сети — это компоненты доступного сетевого домена или транспортной зоны с учетом IP-адресов.

Пользователи Amazon Web Services или Microsoft Azure могут воспринимать сети в качестве подсетей.

Для отображения сведений о сетях в проекте откройте страницу **Инфраструктура > Ресурсы > Сети**.

На странице **Сети** в Cloud Assembly содержатся следующие сведения:

- Сети и подсистемы балансировки нагрузки, определенные извне в сетевом домене учетной записи облачной службы, например, в vCenter, NSX-T или Amazon Web Services.
- Сети и подсистемы балансировки нагрузки, развернутые администратором облачных систем.
- Диапазоны IP-адресов и другие характеристики сети, определенные или измененные администратором облачных систем.
- Диапазоны IP-адресов внешнего поставщика IPAM для определенного адресного пространства в рамках интеграции конкретного поставщика.

Дополнительные сведения о сетях см. в следующем разделе, в контекстной справке для указателей различных параметров на странице **Сети** и в [Дополнительные сведения о профилях сетей в vRealize Automation](#).

## Сети

Сети и их характеристики можно просматривать и изменять, например, добавлять теги или удалять поддержку общедоступного IP-адреса. Кроме того, можно управлять параметрами сети, такими как DNS, CIDR, шлюз и теги. Также можно определить новые диапазоны IP-адресов и управлять существующими диапазонами в сети.

Для существующих сетей можно изменить диапазон IP-адресов и параметры тегов, установив флажок сети и выбрав **Управление диапазонами IP-адресов** или **Теги**. Либо можно выбрать саму сеть, чтобы изменить сведения о ней.

Теги позволяют сопоставлять соответствующие сети и (при необходимости) профили сетей с компонентами сетей в облачных шаблонах. Теги сетей применяются к каждому экземпляру соответствующей сети, независимо от профилей сетей, в которых может содержаться эта сеть. Экземпляры сети могут содержаться в любом количестве профилей сетей. Независимо от местонахождения профиля сети, тег сети связан с данной сетью везде, где эта сеть используется. Сопоставление тегов сети с другими компонентами в облачном шаблоне выполняется после сопоставления этого шаблона с одним профилем сети или несколькими.

Что касается глобальных сетей, существующие и общедоступные сети поддерживаются для облачных учетных записей глобального и локального диспетчера NSX-T, а также для облачных учетных записей vCenter, связанных с локальными диспетчерами. Представление растянутых сетей в локальном диспетчере определяется в транспортной зоне. Транспортная зона — это концепция локального диспетчера NSX-T, определяющая диапазон сетей NSX-T для узлов и кластеров vCenter Server.

Cloud Assembly выполняет перечисление существующих и общедоступных сетей (собирает данные). Для создания глобальной сети можно добавить существующую или общедоступную сеть в глобальном диспетчере NSX-T. Затем эта глобальная сеть может использоваться всеми связанными локальными диспетчерами. Глобальные сети могут охватывать один, все или подмножество связанных локальных диспетчеров.

Компьютер можно подготовить в глобальной сети с помощью назначения статического IP-адреса. Протокол DHCP не поддерживается.

В глобальном диспетчере можно создать следующие типы глобальных сетей.

1. Сеть наложения — связана с локальным диспетчером нулевого уровня/первого уровня и автоматически растягивается, включая в себя все сайты, подключенные к локальному диспетчеру нулевого уровня/первого уровня. Для каждого локального диспетчера используется транспортная зона наложения по умолчанию.
2. Сеть VLAN — применяется к одному локальному диспетчеру, при этом транспортную зону можно выбрать вручную.

Глобальные сети перечислены на странице **Инфраструктура > Ресурсы** со всеми облачными учетными записями, к которым они применяются.

Для глобальных сетей поддерживаются следующие операции по регулярному обслуживанию.

- Перенастройка сети в определении облачного шаблона из глобальной в локальную и наоборот.
- Горизонтальное и вертикальное масштабирование компьютеров в глобальных сетях.

Дополнительные сведения об использовании сетей в облачных шаблонах см. в разделе [Дополнительные сведения о сетевых ресурсах в облачных шаблонах vRealize Automation](#).

Сведения об обновлении сетей vSphere в vRealize Automation после миграции NSX-T из N-VDS в C-VDS см. в [Обновление сетевых ресурсов в vRealize Automation после миграции с N-VDS на C-VDS в NSX-T](#).

## Диапазоны IP-адресов

С помощью диапазона IP-адресов можно определить начальный и конечный IP-адреса для определенной сети в организации или изменить их. Для перечисленных сетей можно отображать диапазоны IP-адресов и управлять ими. Если управление сетью выполняется внешним поставщиком IPAM, можно управлять диапазонами IP-адресов в сочетании со связанной точкой интеграции IPAM.

Чтобы добавить дополнительный диапазон IP-адресов в сеть нажмите **Создать диапазон IP-адресов**. Можно указать **диапазон внутренних IP-адресов** или (при наличии допустимой интеграции IPAM) **диапазон внешних IP-адресов**.

В диапазон IP-адресов нельзя включить шлюз по умолчанию. Диапазон IP-адресов подсети не может содержать значение шлюза подсети.

Если выполняется интеграция определенного внешнего поставщика IPAM, то для выбора диапазона IP-адресов из доступной точки интеграции внешнего IPAM можно использовать **диапазон внешних IP-адресов**. Этот процесс описан в контексте общего процесса интеграции внешнего IPAM в [Настройка сети и профиля сети для использования внешнего поставщика IPAM в существующей сети vRealize Automation](#).

---

**Примечание** При удалении диапазона IP-адресов от внешнего поставщика IPAM во внешнем приложении IPAM этот диапазон IP-адресов автоматически удаляется во время перечисления в vRealize Automation. Удаленный диапазон IP-адресов больше не отображается или не будет доступен для сетевой связи в vRealize Automation, что позволит избежать потери диапазонов IP-адресов.

---

vRealize Automation позволяет применять диапазон IP-адресов к нескольким сетям vSphere и NSX и управлять им. Поддержка общего диапазона IP-адресов обеспечивается как для внутреннего, так и для внешнего поставщика IPAM. Можно задать один диапазон IP-адресов в растянутой сети NSX, чтобы компьютеры в этой сети могли использовать IP-адреса, которые назначены с одного IP-адреса, даже если они развернуты в разных экземплярах vCenter.

## IP-адреса

Можно просмотреть IP-адреса, которые в настоящее время используются в вашей организации, и отобразить их статус, например `available` или `allocated`. Отображаемые IP-адреса — это IP-адреса, управление которыми выполняется в среде с помощью vRealize Automation, или IP-адреса, назначенные для развертываний, содержащих интеграцию внешнего поставщика IPAM. Управление IP-адресами, выделенными внешними поставщиками IPAM, осуществляется этими поставщиками.

Если управление сетью осуществляется внутренними средствами с помощью vRealize Automation, а не с помощью внешнего поставщика IPAM, можно также освободить IP-адреса.

При использовании внутренней службы управления IP-адресами и освобождении IP-адресов, например после удаления компьютера, который использовал IP-адреса, или нажатия кнопки **Освободить IP-адрес** для выбранной сети, применяется период ожидания. Это период с момента освобождения неиспользуемого IP-адреса до момента, когда этот IP-адрес становится доступным для использования. Период ожидания, или период ожидания освобождения, позволяет очистить кэш DNS. После этого IP-адреса можно выделить новому компьютеру. По умолчанию период ожидания освобождения IP-адреса — 30 минут. Период ожидания можно изменить, выбрав **Настройки** в правом верхнем углу страницы **Сети** и изменив значение в поле **Время ожидания освобождения**.

- В период ожидания освобождения соответствующие IP-адреса отображаются как освобожденные. По истечении периода ожидания освобождения они отображаются как доступные.
- Система проверяет наличие новых освобожденных IP-адресов каждые 5 минут, поэтому, даже если период ожидания освобождения — 1 минута, освобожденный IP-адрес может стать доступным через 1–6 минут в зависимости от того, когда была выполнена последняя проверка. Интервал проверки 5 минут применяется ко всем значениям, кроме 0.
- Если для времени ожидания освобождения установить значение 0, IP-адреса освобождаются и становятся доступными немедленно.
- Значение времени ожидания освобождения применяется ко всем облачным учетным записям в организации.

## Подсистемы балансировки нагрузки

В организации можно управлять сведениями о доступных подсистемах балансировки нагрузки для облачных учетных записей или их регионов. Для каждой доступной подсистемы балансировки нагрузки можно открыть и отобразить настроенные параметры. Кроме того, для подсистемы балансировки нагрузки можно добавлять и удалять теги.

Дополнительные сведения об использовании подсистем балансировки нагрузки в облачных шаблонах см. в разделе [Дополнительные сведения о ресурсах подсистем балансировки нагрузки в облачных шаблонах vRealize Automation](#).

## Сетевые домены

В списке сетевых доменов указаны связанные и неперекрывающиеся сети.

## Ресурсы безопасности в vRealize Automation

После добавления облачной учетной записи в Cloud Assembly процесс сбора данных обнаруживает сведения о сети и безопасности для этой учетной записи и делает полученную информацию доступной для использования в профилях сети и других параметрах.

Группы безопасности и правила брандмауэра поддерживают изоляцию сети. Сведения о группах безопасности извлекаются в результате сбора данных. Сведения о правилах брандмауэра не извлекаются в результате сбора данных.

Если выбрать пункты меню **Инфраструктура > Ресурсы > Безопасность**, можно просмотреть группы безопасности по требованию, созданные в проектах облачных шаблонов Cloud Assembly, а также существующие группы безопасности, созданные в исходных приложениях, таких как NSX-T и Amazon Web Services. Сведения о доступных группах безопасности поступают в результате процесса сбора данных.

Можно использовать тег для сопоставления интерфейса компьютера (сетевой адаптер) с группой безопасности в определении облачного шаблона или в профиле сети. Можно просмотреть доступные группы безопасности, а также добавить или удалить теги для выбранных групп безопасности. Разработчик облачного шаблона может назначить одну группу безопасности сетевому адаптеру компьютера или несколько, чтобы управлять обеспечением безопасности для развертывания.

В проекте облачного шаблона для параметра `securityGroupType` в ресурсе группы безопасности задано значение `existing` для существующей группы безопасности или `new` для группы безопасности по требованию.

## Существующие группы безопасности

Существующие группы безопасности отображаются в столбце **Источник** и указываются значением `Discovered`.

Существующие группы безопасности из базовой конечной точки облачной учетной записи, такие как приложения NSX-V, NSX-T или Amazon Web Services, доступны для использования.

Администратор облачных систем может назначить один тег или несколько существующей группе безопасности, чтобы ее можно было использовать в облачном шаблоне. Разработчик облачных шаблонов может использовать ресурс `Cloud.SecurityGroup` в проекте облачного шаблона для выделения существующей группы безопасности с помощью тегов ограничений. Для существующей группы безопасности необходимо указать по крайней мере один тег ограничения в ресурсе безопасности в проекте облачного шаблона.

Если существующую группу безопасности изменить непосредственно в исходном приложении, например в исходном приложении NSX, а не в Cloud Assembly, обновления не будут отображаться в Cloud Assembly до тех пор, пока не будет запущен процесс сбора данных и не будут получены данные о связанных облачной учетной записи или точке интеграции из Cloud Assembly. Сбор данных выполняется автоматически примерно каждые 10 минут.

Существующие группы безопасности поддерживаются для облачных учетных записей глобального и локального диспетчера NSX-T, а также для облачных учетных записей vCenter, связанных с локальными диспетчерами. Cloud Assembly выполняет перечисление существующих групп безопасности (собирает данные) и подключает их к сетевым интерфейсам компьютера (сетевые адаптеры). Для создания глобальной группы безопасности можно добавить существующую группу безопасности в глобальном диспетчере NSX-T. Затем эта глобальная группа безопасности может использоваться связанными локальными диспетчерами. Глобальные группы безопасности могут охватывать один, все или подмножество связанных локальных диспетчеров.

- Существующие глобальные группы безопасности поддерживаются и перечисляются для всех определенных регионов.
- Глобальные группы безопасности перечислены на странице **Инфраструктура > Ресурсы** вместе со всеми облачными учетными записями, к которым они применяются.
- Интерфейс компьютера (сетевой адаптер) можно связать с существующей глобальной группой глобальной безопасности непосредственно в облачном шаблоне или в выбранном профиле сети.
- Для глобальных групп безопасности поддерживаются следующие операции по регулярному обслуживанию.
  - Перенастройка группы безопасности в облачном шаблоне из глобальной в локальную и наоборот.
  - Горизонтальное и вертикальное масштабирование компьютеров, связанных с глобальными группами безопасности.

## Группы безопасности по требованию

Группы безопасности по требованию, созданные в Cloud Assembly либо в облачном шаблоне, либо в профиле сети, отображаются в столбце **Источник** и указываются значением *Managed by Cloud Assembly*. Группы безопасности по требованию, созданные в рамках профиля сети, классифицируются внутренним образом как группа безопасности изоляции с предварительно настроенными правилами брандмауэра и не добавляются в облачный шаблон в качестве ресурса группы безопасности. Группы безопасности по требованию, созданные в рамках проекта облачного шаблона, которые могут содержать явно выраженные правила брандмауэра, добавляются в виде части ресурса группы безопасности, которая классифицируется как *new*.

---

**Примечание** Правила брандмауэра можно создать для групп безопасности по требованию для NSX-V и NSX-T непосредственно в ресурсе группы безопасности в коде проекта облачного шаблона. В столбце **Применяется к** группы безопасности, которые классифицируются или управляются распределенным брандмауэром (DFW) в NSX, не указываются. Правила брандмауэра, которые применяются к приложениям, предназначены для «восточного»/«западного» трафика DFW. Некоторыми правилами брандмауэра можно управлять только в исходном приложении и их нельзя изменять в Cloud Assembly. Например, управление правилами для ethernet, аварийных ситуаций, инфраструктуры и среды осуществляется в NSX-T.

---

Группы безопасности по требованию в настоящее время не поддерживаются для облачных учетных записей глобального диспетчера NSX-T.

## Подробнее

Дополнительные сведения об использовании групп безопасности в профилях сетей см. в разделе [Дополнительные сведения о профилях сетей в vRealize Automation](#).

Сведения об определяющих правилах брандмауэра см. в разделе [Использование параметров группы безопасности в профилях сетей и проектах облачных шаблонов в vRealize Automation](#).

Дополнительные сведения об использовании групп безопасности в облачном шаблоне см. в разделе [Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation](#).

Примеры кода проекта облачного шаблона, содержащие группы безопасности, см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Ресурсы хранилища в vRealize Automation

Администратор облачных систем может работать с ресурсами хранилища и их возможностями, которые обнаруживаются vRealize Automation путем сбора данных из связанных учетных записей облачных служб.

Возможности ресурсов хранилища отражаются в тегах, которые обычно создаются в учетной записи исходной облачной службы. При этом администратор облачных систем может применять дополнительные теги в Cloud Assembly непосредственно к ресурсам хранилища. Дополнительные теги могут отражать конкретную возможность для сопоставления во время подготовки.

vRealize Automation поддерживает возможности стандартных дисков и дисков первого класса. Диски первого класса доступны только для vSphere.

- [Что можно делать со стандартным дисковым накопителем в vRealize Automation](#)
- [Что можно делать с дисковым накопителем первого класса в vRealize Automation](#)

Возможности ресурсов хранилища отображаются в рамках определения профиля хранилища Cloud Assembly. См. раздел [Дополнительные сведения о профилях хранилища в vRealize Automation](#).

Диски первого класса, о которых собраны данные, отображаются в представлении **Ресурсы > Ресурсы > Тома**.

## Дополнительные сведения о ресурсах в Cloud Assembly

Cloud Assembly может предоставлять дополнительную информацию о ресурсах, полученную в результате сбора данных, например карты ценообразования.

### Как сбор данных работает в vRealize Automation

После первоначального сбора данных последующие сеансы сбора данных о ресурсах выполняются автоматически каждые 10 минут. Интервал сбора данных не настраивается, а сам процесс сбора данных инициировать вручную нельзя.

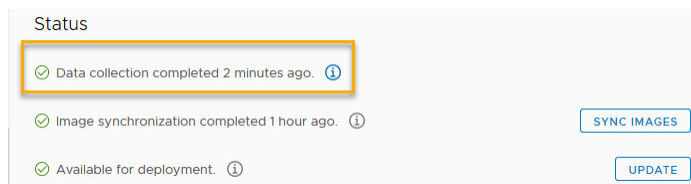


Сведения о сборе данных о ресурсах, а также о синхронизации образов для существующей облачной учетной записи можно найти в разделе «Состояние» на странице данной учетной записи. Для этого выберите **Инфраструктура > Подключения > Облачные учетные записи**, а затем нажмите **Открыть** в нужной существующей облачной учетной записи.

Можно открыть существующую облачную учетную запись и просмотреть версию связанной с ней конечной точки в разделе **Состояние** на странице данной учетной записи. Если связанная конечная точка была обновлена, то новая версия конечной точки будет определена во время сбора данных и отображена в разделе **Состояние** на странице облачной учетной записи.

### Сбор данных о ресурсах

Сбор данных происходит каждые 10 минут. В каждой облачной учетной записи отображается время завершения последнего сеанса сбора данных.

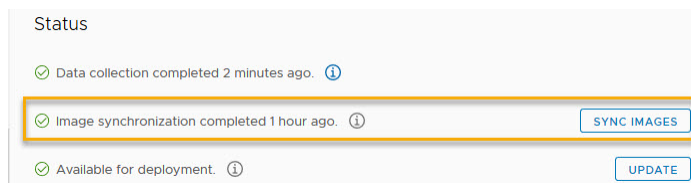


### Сбор данных об образах

Синхронизация образов выполняется каждые 24 часа. Процесс синхронизации образов можно инициировать для облачных учетных записей некоторых типов. Чтобы инициировать синхронизацию образов, откройте облачную учетную запись (**Инфраструктура > Облачные учетные записи**, затем выберите и откройте существующую облачную учетную запись) и нажмите **Синхронизировать образы**. Для облачных учетных записей NSX возможность синхронизации образов отсутствует.

**Примечание** Внутри системы образы классифицируются как общедоступные или частные. Общедоступные образы доступны всем и не зависят от конкретной облачной подписки или организации. Частные образы не являются общедоступными и относятся к конкретной подписке. Общедоступные и частные образы автоматически синхронизируются каждые 24 часа. На странице облачной учетной записи существует отдельный параметр, который позволяет запускать синхронизацию для частных образов.

На странице облачной учетной записи отображается время завершения последнего сеанса синхронизации образом.



Чтобы обеспечить отказоустойчивость и высокую доступность в развертываниях, каждая конечная точка центра обработки данных NSX-T представляет собой кластер из трех диспетчеров NSX. Дополнительные сведения см. в разделе [Создание облачной учетной записи NSX-T в vRealize Automation](#).

## Облачные учетные записи и планы внедрения

При создании облачной учетной записи со всех компьютеров, связанных с ней, собираются данные, после чего эти компьютеры отображаются на странице **Ресурсы > Ресурсы > Виртуальные машины**. Если в облачной учетной записи есть компьютеры, развернутые за пределами Cloud Assembly, можно воспользоваться планом подключения, чтобы служба Cloud Assembly могла управлять развертываниями этих компьютеров.

Дополнительные сведения о добавлении облачных учетных записей см. в разделе [Добавление учетных записей облачной службы в Cloud Assembly](#).

Сведения о внедрении неуправляемых компьютеров см. в разделе [Что такое планы внедрения в Cloud Assembly](#).

## Обновление сетевых ресурсов в vRealize Automation после миграции с N-VDS на C-VDS в NSX-T

После миграции NSX-T из виртуального распределенного коммутатора NSX (N-VDS) в конвергентную службу VDS (C-VDS) необходимо обновить затронутые сетевые ресурсы vSphere в vRealize Automation, чтобы продолжить использование этих ресурсов в новых и существующих облачных шаблонах и развертываниях.

После миграции с N-VDS на C-VDS ваши сети vSphere могут отсутствовать, хотя они присутствовали в профилях сети vRealize Automation. Чтобы предотвратить потерю таких сетей типа vSphere и продолжить их выделение в существующих и новых развертываниях, необходимо вручную обновить все перечисленные сети C-VDS в vRealize Automation Cloud Assembly.

---

**Примечание** Эта процедура предназначена для действий, необходимых в vRealize Automation для обновления сетей *vSphere* после миграции с N-VDS на C-VDS в NSX-T. После миграции с N-VDS на C-VDS никакие действия в vRealize Automation в сетях *NSX* не требуются. Для *NSX* вмешательство пользователя вручную после миграции с N-VDS на C-VDS также не требуется.

---

Несмотря на то что администратор NSX-T может перенести NSX-T в типы сети VDS (N-VDS) в конвергентные типы сетей VDS (C-VDS) в NSX, данное действие окажет влияние на существующие сетевые ресурсы vSphere в vRealize Automation. Администратор vRealize Automation может выполнять действия после миграции, чтобы согласовать эти ресурсы в vRealize Automation и связанные изменения в NSX-T и vCenter Server. Следует отметить, что C-VDS или просто VDS также обозначается в других местах как виртуальный распределенный коммутатор vSphere 7 (VDS).

Дополнительные сведения о конвергентной системе NSX-T VDS см. в статье базы знаний VMware [NSX-T в VDS \(79872\)](#) и [VMware Cloud on AWS \(VMConAWS\)](#) и миграция [VMware Cloud on Dell EMC из N-VDS в VDS \(82487\)](#).

---

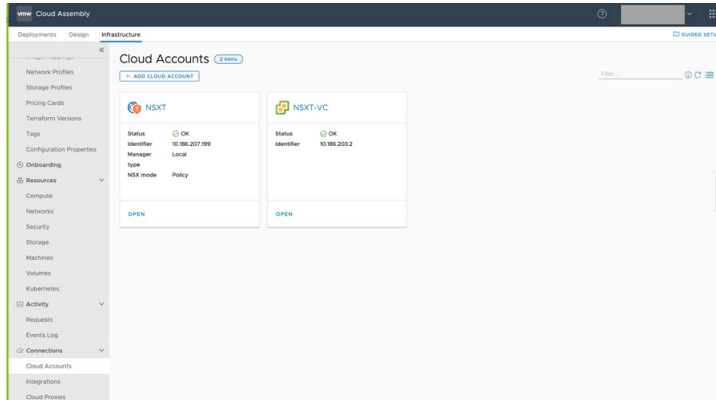
**Примечание** В примере такого сценария представлены действия, необходимые для согласования ресурсов в среде vRealize Automation после миграции с N-VDS на C-VDS. Этот пример и процедуру можно использовать в vRealize Automation 8.5 и более поздних версиях для согласования изменений, внесенных в vCenter Server после миграции из N-VDS в C-VDS в NSX-T.

---

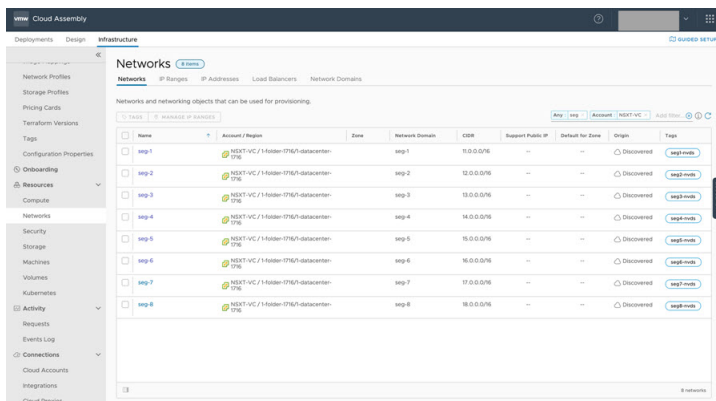
## Пример: ресурсы vRealize Automation до миграции

В этом примере показаны образцы ресурсов NSX-T в примере среды vRealize Automation до миграции с N-VDS на C-VDS.

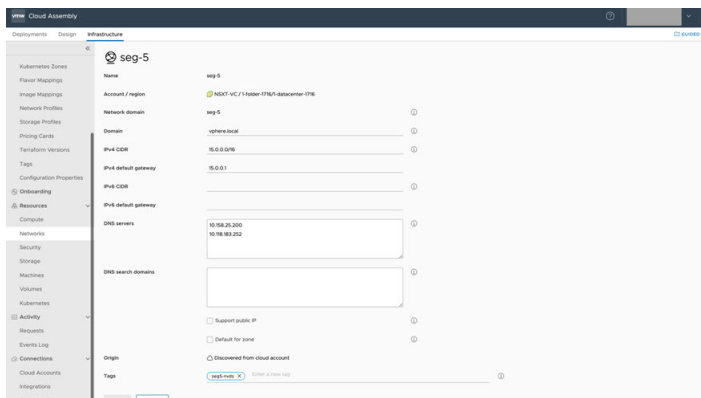
- В этом примере используются облачные учетные записи NSX-T и vCenter, как показано ниже.



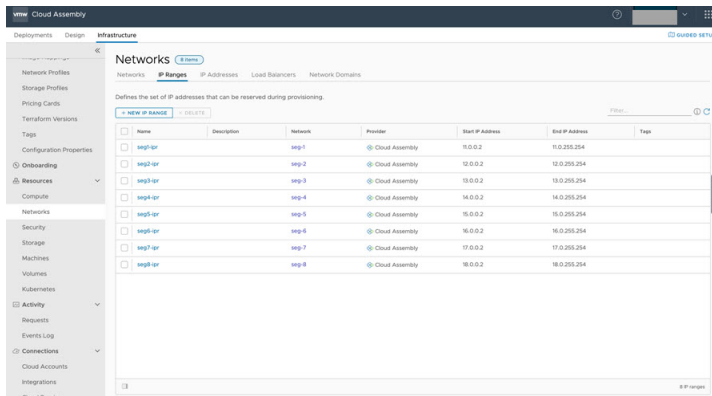
- В примере используется несколько сетей vSphere, как показано ниже.



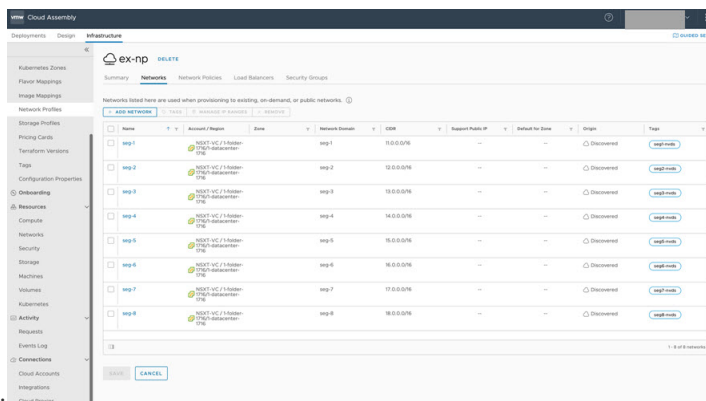
- В примере конфигурации сети используются параметры CIDR и DNS, как показано ниже.



- В примере также используются существующие диапазоны IP-адресов, как показано ниже.

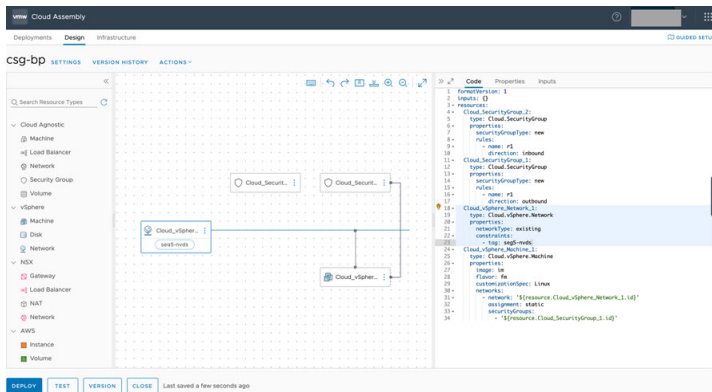


- В этом примере используется профиль сети (**ex-np**), который содержит несколько сетей N-VDS (N-VDS), в том числе **seg-5**, как показано

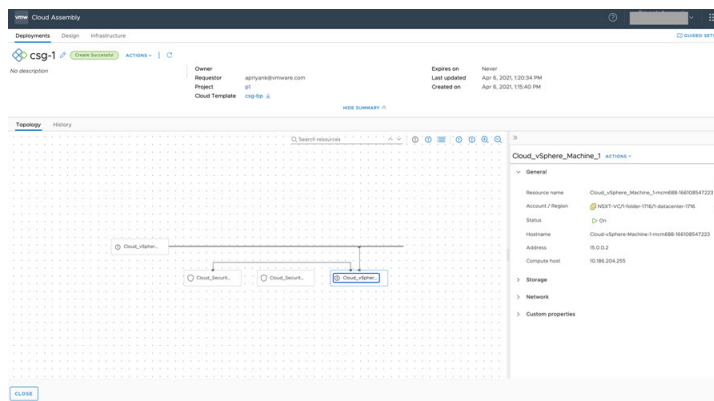


ниже.

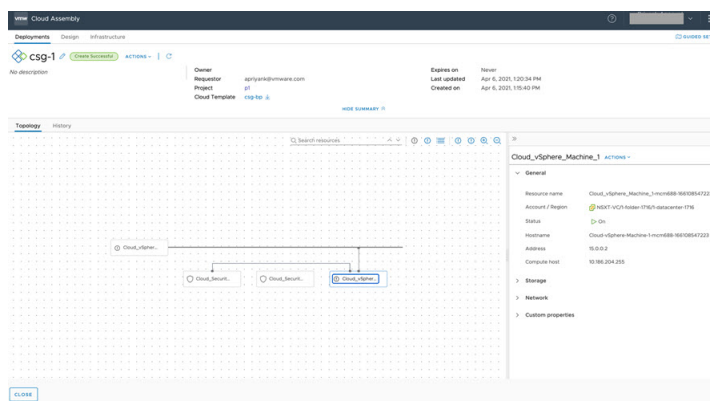
- В этом примере показан существующий компонент сети **seg5** со следующим образом синтаксиса облачного шаблона. Эта сеть отмечена как сеть N-VDS. В этом примере будут показаны необходимые обновления в сети **seg5** после миграции.



- В примере облачного шаблона создается развертывание, как показано ниже.



- Используемые в качестве примера IP-адреса компьютеров отображаются в образце развертывания, как показано ниже.

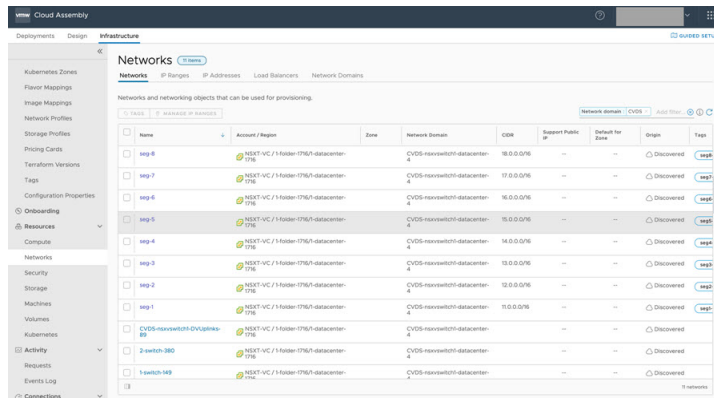


**Пример. Шаг 1 после миграции. Выполнение сбора данных после миграции с N-VDS на C-VDS и перечисление**

В разделе выше были использованы снимки экрана для демонстрации инфраструктуры, задействованной в образце среды vRealize Automation, а также выходной облачный шаблон и развертывание.

После выполнения миграции с N-VDS на C-VDS в NSX-T подождите не менее 10 минут, чтобы служба vRealize Automation начала выполнять периодический процесс сбора данных и перечисления для извлечения и отображения задействованных ресурсов в vRealize Automation.

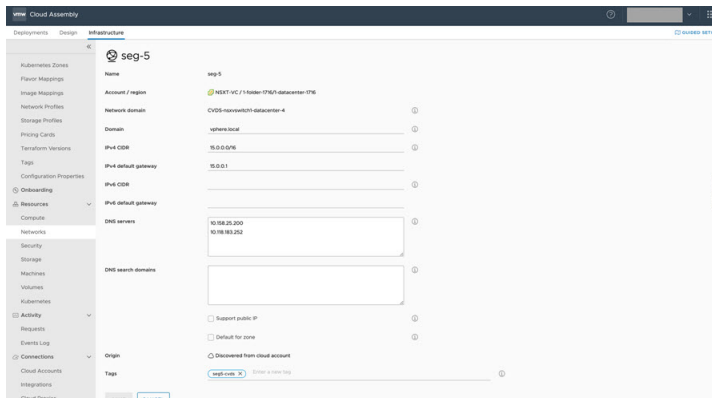
По окончании процесса сбора данных vRealize Automation перейдите в раздел **Инфраструктура > Сети**, чтобы увидеть доступные сети C-VDS и получить к ним доступ. Обратите внимание на сеть **seg5**, как показано ниже.



### Пример. Шаг 2 после миграции. Добавление предварительно определенных CIDR и DNS в перенесенные сети C-VDS

Измените перенесенную сеть C-VDS — добавьте сведения о CIDR и DNS, указанные в определении N-VDS до миграции, и измените порядок расстановки тегов для сети.

1. Добавьте сведения о CIDR и DNS, которые были заданы в определении N-VDS перед миграцией.
2. Добавьте новый тег для образца сегмента сети **seg-5** для C-VDS, например *seg5-cvds*.



Следует отметить, что исходная сеть **seg-5** для N-VDS была отмечена как *seg5-nvds*, как показано на предыдущих экранах. При перенастройке сети требуется изменить сведения о порядке расстановки тегов для ресурсов. Для vRealize Automation требуется, чтобы в облачный шаблон для сети C-VDS было добавлено другое имя тега, а не тег, который используется в исходной сети N-VDS. Измененный порядок расстановки тегов определяет изменение в облачном шаблоне при создании допустимого повторного развертывания.

### Пример. Шаг 3 после миграции. Добавление сведений об обновленном диапазоне IP-адресов

Чтобы изменить сведения о диапазоне IP-адресов сети, которые были указаны в определении N-VDS перед миграцией, можно использовать API-интерфейс командной строки или последовательность меню в vRealize Automation.

- Вариант 1. Для обновления данных о диапазоне IP-адресов используйте API-интерфейс, как показано на следующем примере экрана.

PATCH : `{{host}}/iaas/api/network-ip-ranges/{{subnet-range-id}}`

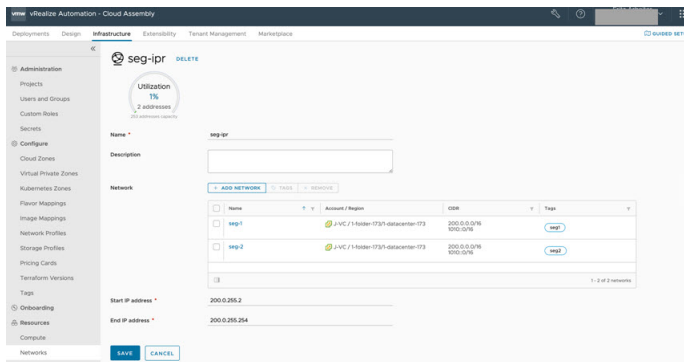
Headers :

- Authorization : Bearer `{{token}}`

Payload :

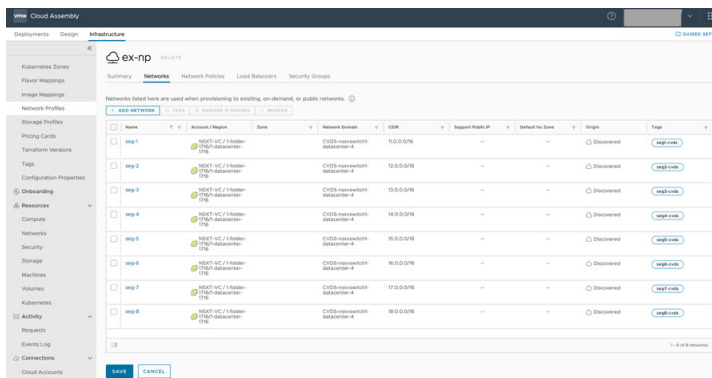
```
{
  "fabricNetworkIds": ["{{subnet-id}}"]
}
```

- Вариант 2. Для обновления данных о диапазоне IP-адресов используйте пользовательский интерфейс, как показано на следующем примере экрана.



Пример. Шаг 4 после миграции. Обновление профилей сети для восстановления отсутствующих сетей

После миграции сети N-VDS согласованы и удалены из vRealize Automation Cloud Assembly после сбора данных и перечисления. В затронутых профилях сетей (например, в **ex-np**) имеются недостающие сети. Чтобы восстановить недостающие сети, преобразуйте каждую сеть N-VDS в сеть C-VDS, как показано ниже.



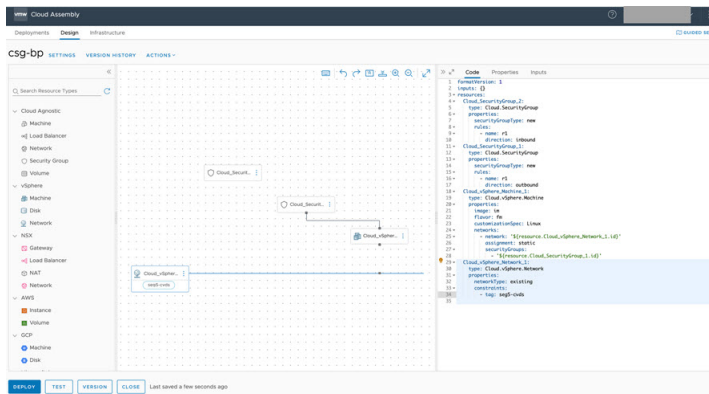
### Пример. Шаг 5 после миграции. Обновление ограничений сети в облачных шаблонах

Для существующих развертываний необходимо обновить ограничения сети в облачном шаблоне, чтобы они соответствовали новым сетям C-VDS в обновленных профилях сетей. Обновленные ограничения сети также необходимы для выполнения итеративных развертываний и перенастройки сетей из соответствующего представления vSphere N-VDS в представление vSphere C-VDS.

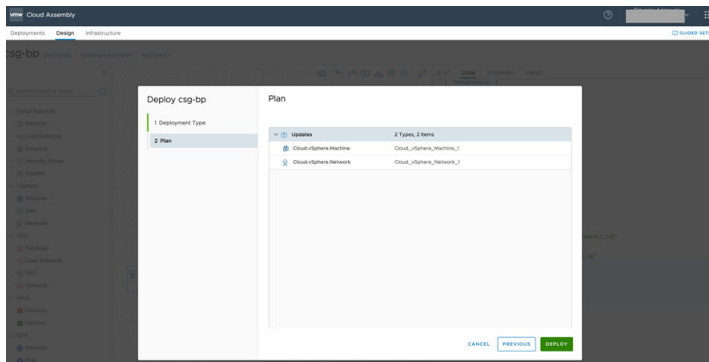
В новых развертываниях используются указанные ресурсы C-VDS, поэтому этот шаг не требуется.

Итеративные развертывания и перенастройка сети просто работают в соответствии с проектом.

1. В этом примере измените ограничения сети в облачном шаблоне со значения *seg5-nvds* на значение *seg5-cvds*, как показано ниже.

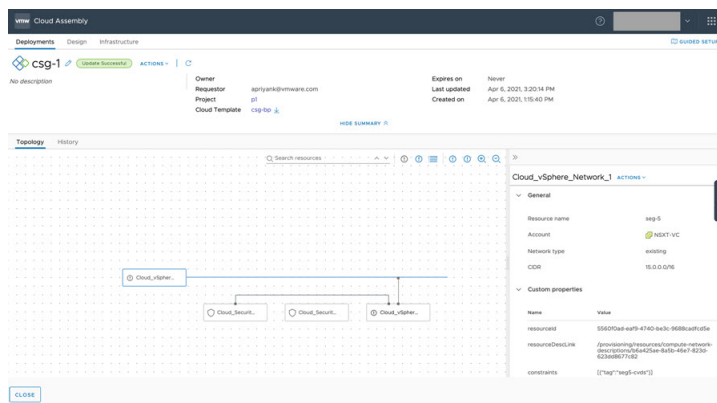


2. Выполните итеративное развертывание для перенастройки сети, как показано ниже.

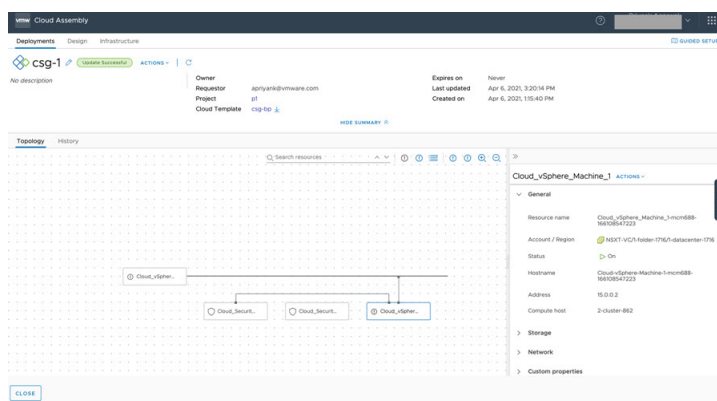


3. После успешного повторного развертывания обратите внимание, что настраиваемые свойства сети отображают обновленные ограничения, как показано ниже.





Если диапазон IP-адресов был обновлен ранее с помощью новых данных C-VDS, IP-адрес компьютера не будет корректно изменен в повторном развертывании, как показано ниже.



## Использование панели управления «Ключевые сведения» для мониторинга объема ресурсов и уведомления владельцев проектов в vRealize Automation

Администраторы облачных систем могут отслеживать ресурсы инфраструктуры и оптимизацию развертываний в каждой облачной зоне, а также управлять ими. Благодаря наглядному представлению данных в режиме реального времени и просмотру предлагаемых действий для поддерживаемых ресурсов вы можете помогать владельцам проектов в упреждающем режиме управлять емкостью ресурсов и оптимизировать их развертывание.

Значения показателей для ресурсов и развертываний в облачных зонах, связанных с управляемыми проектами, можно анализировать на панели управления **Ключевые сведения**. Информацию, полученную из vRealize Automation и интегрированного приложения vRealize Operations Manager, можно использовать, чтобы изменить объем памяти, количество ЦП и т. д., либо поделиться ею с участниками своей группы, чтобы они могли внести необходимые изменения на основе этой информации.

Панель управления «Ключевые сведения» позволяет связываться с некоторыми или всеми владельцами проектов, которые имеют развертывания в облачной зоне, где содержатся освобождаемые ресурсы. На этой панели отображается объем освобождаемых ресурсов для проектов и развертываний.

Владельцы проектов могут посмотреть уведомление о своих развертываниях на странице **Оповещения**. Такое уведомление содержит уникальное имя, а также имя каждого развертывания (и ссылку на него), которое можно оптимизировать.

Панель управления **Ключевые сведения** доступна для облачных зон vSphere и VMware Cloud on AWS при условии, что облачные учетные записи настроены в vRealize Automation и vRealize Operations Manager и отслеживаются в vRealize Operations Manager.

#### Необходимые условия

- См. раздел [Управление ресурсами и оптимизация развертывания с помощью показателей vRealize Operations Manager в vRealize Automation](#).
- У вас должны быть необходимые учетные данные администратора облачных систем vRealize Automation и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Вам должна быть назначена роль пользователя vRealize Automation с правами администратора облачных систем. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Настройте интеграцию vRealize Automation с vRealize Operations Manager.
- Настройте адаптер vRealize Automation в vRealize Operations Manager.

#### Сведения о vRealize Operations Manager и собранных показателях объема ресурсов

vRealize Operations Manager собирает показатели объема для ресурсов инфраструктуры, которые используются вами и поддерживаемыми вами группами в vRealize Automation. Благодаря интеграции vRealize Automation с vRealize Operations Manager данные показателей vRealize Operations Manager становятся доступными и отображаются для каждого управляемого проекта на панели управления **Ключевые сведения** в каждой облачной зоне.

Данные проекта направляются на панель управления vRealize Automation из интегрированного приложения vRealize Operations Manager. На панели управления «Ключевые сведения» отображается следующая информация.

- Процент использования ЦП от доступного объема ресурсов
- Процент использования памяти от доступного объема ресурсов
- Процент использования хранилища от доступного объема ресурсов
- Журнал расчетных требований по ЦП и памяти, а также прогнозируемые требования
- Возможность связываться с владельцами некоторых или всех развертываний в облачной зоне, которые можно оптимизировать путем освобождения ресурсов, например, посредством изменения размера или удаления компьютеров. Данные об оптимизации рассчитываются по дням.

Панель управления «Ключевые сведения» доступна только для ресурсов vSphere.

Мини-приложение тенденций показывает вычислительные компоненты облачной зоны (например, кластеры и узлы), их тактовые частоты (в ГГц) и объем используемой памяти (в ГБ).

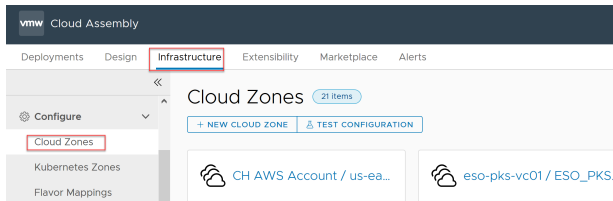
Сведения о ролях, необходимых для использования оповещений, можно найти в разделе [Настраиваемые роли пользователей в vRealize Automation](#).

Дополнительные сведения см. в разделе [Управление ресурсами и оптимизация развертывания с помощью показателей vRealize Operations Manager в vRealize Automation](#).

## Процедура

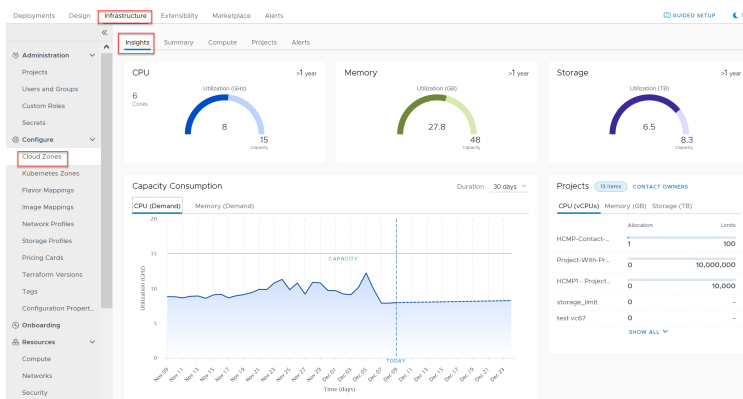
Откройте облачную зону, чтобы узнать показатели ее ресурсов. При необходимости можно получить сведения о развертываниях проектов, которые можно оптимизировать. Сбор данных и их отправка выполняются связанным приложением vRealize Operations Manager.

1. В Cloud Assembly выберите **Инфраструктура > Настроить > Облачные зоны**, а затем облачную зону.

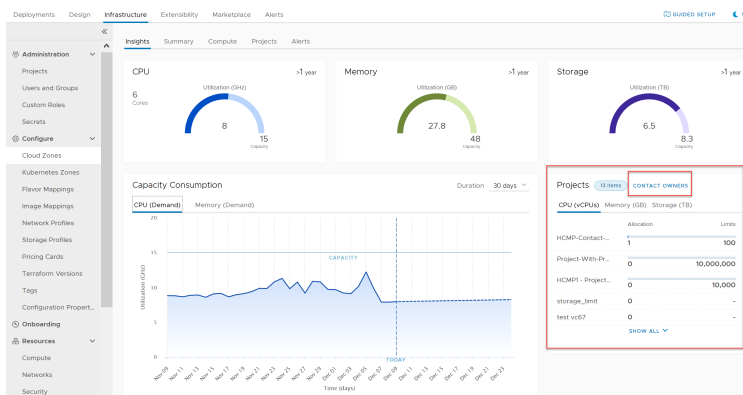


2. Перейдите на вкладку **Ключевые сведения** и проверьте данные на панели управления ключевых сведений.

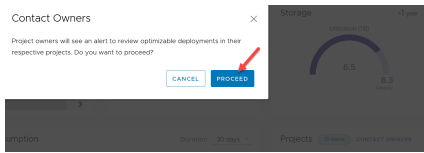
В следующем примере отображаются сведения о ЦП, памяти и емкости хранилища для ресурсов, которые используются проектами в облачной зоне.



3. Чтобы информировать владельца проекта о любых развертываниях, которые можно оптимизировать, нажмите **Связь с владельцем** в разделе **Проекты**. Уведомления отображаются на вкладке **Оповещения**.

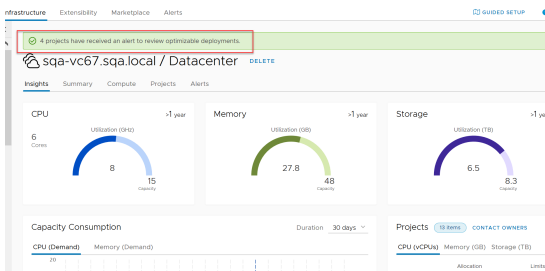


4. Чтобы получить сведения об оптимизации для каждого развертывания для проекта, нажмите **Продолжить**.

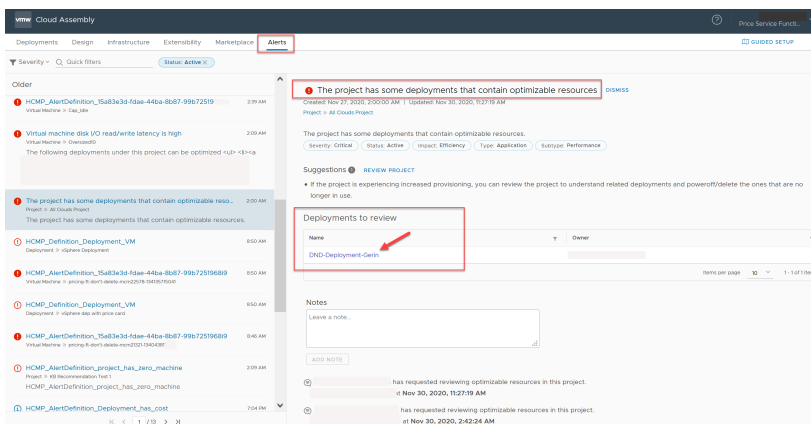


Если проект содержит развертывания, которые можно оптимизировать, эта информация передается владельцу проекта на вкладке **Cloud Assembly Оповещения**.

5. Появляется сообщение, где указывается количество развертываний, которые можно оптимизировать.



Уведомление об этих ресурсах и развертываниях доступно владельцу проекта в Cloud Assembly на вкладке **Оповещения**. В этом примере уведомление содержит имя каждого развертывания (и ссылку на него), которое можно оптимизировать, как показано в следующем примере:



## Дальнейшие действия

Информацию, приведенную на панели **Ключевые сведения**, можно использовать для внесения необходимых изменений в управляемые ресурсы. Откройте страницу **Оповещения**, где отображаются дополнительная информация, предлагаемые действия и ссылки на развертывания, которые можно оптимизировать. См. раздел [Использование оповещений для управления объемом, производительностью и доступностью ресурсов в vRealize Automation](#).

## Использование оповещений для управления объемом, производительностью и доступностью ресурсов в vRealize Automation

Администратору облачных систем необходимо получать информацию о проблемах с объемом, производительностью или доступностью ресурсов vRealize Automation, чтобы реагировать на них до того, как пользователи начнут испытывать нехватку ресурсов.

Можно настроить, какие оповещения будут отображаться в соответствующем приложении vRealize Operations Manager. Оповещения доступны только для объектов ресурсов vSphere и VMware Cloud on AWS. Используйте данные из оповещений, чтобы изменять управляемые ресурсы и развертывания, или поделитесь этой информацией с коллегами, чтобы они изменили свои объекты.

---

**Примечание** Сведения о проверке развертываний проектов и работе с теми развертываниями, которые необходимо оптимизировать, см. в разделе [Использование оповещений для оптимизации развертываний в vRealize Automation](#).

---

В настоящее время оповещения доступны только для объектов ресурсов vSphere и VMware Cloud on AWS. Вкладка **Оповещения** доступна только в том случае, если настроен доступ к vRealize Operations Manager.

Пороговые значения оповещений vRealize Automation задаются в vRealize Operations Manager. Параметры некоторых оповещений vRealize Automation определены заранее. Оповещения и уведомления также настраиваются в vRealize Operations Manager. Дополнительные сведения о создании определений оповещений и настройке уведомлений см. в [документации по продукту vRealize Operations Manager](#).

### Необходимые условия

- См. раздел [Управление ресурсами и оптимизация развертывания с помощью показателей vRealize Operations Manager в vRealize Automation](#).
- У вас должны быть необходимые учетные данные администратора облачных систем vRealize Automation и включен доступ по протоколу HTTPS к порту 443. См. раздел [Учетные данные, необходимые для работы с облачными учетными записями в vRealize Automation](#).
- Вам должна быть назначена роль пользователя vRealize Automation с правами администратора облачных систем. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Настройте интеграцию vRealize Automation с vRealize Operations Manager.
- Настройте адаптер vRealize Automation в vRealize Operations Manager.
- Настройте роли, необходимые для управления оповещениями. См. раздел [Настраиваемые роли пользователей в vRealize Automation](#).

Роли могут иметь следующие возможности.

- Администраторы облачных систем могут управлять оповещениями облачной зоны.
- Администраторы проектов могут управлять оповещениями в проектах.
- Администраторы Service Broker могут управлять оповещениями о развертывании.

## Сведения о vRealize Operations Manager и оповещениях о состоянии ресурсов

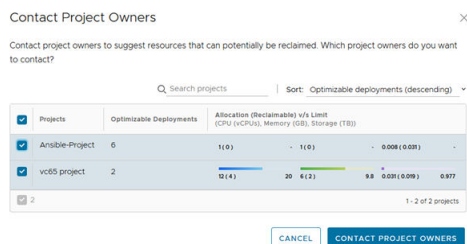
vRealize Operations Manager собирает сведения о работоспособности, использовании и других показателях для ресурсов инфраструктуры и развертываний, которыми вы управляете в vRealize Automation. Интеграция vRealize Automation с vRealize Operations Manager позволяет получить доступ к контролируемым данным в vRealize Automation на вкладке **Оповещения** в главном меню Cloud Assembly.

Данные оповещений, поступающие из vRealize Operations Manager, включают в себя рекомендации по пороговым значениям работоспособности и рисков для облачных шаблонов, развертываний, организаций и проектов. Они также содержат сведения о развертываниях, которые можно оптимизировать, в зависимости от владельца, выполняющего действие на вкладке **Ключевые сведения** облачной зоны. См. раздел [Использование панели управления «Ключевые сведения» для мониторинга объема ресурсов и уведомления владельцев проектов в vRealize Automation](#).

Подробные данные оповещений для каждого развертывания включают в себя следующую информацию:

- Имя проекта
- Имя развертывания (и ссылка на развертывание), содержащего ресурсы, которые можно оптимизировать
- Предлагаемые действия
- Потенциальная экономия затрат за счет реорганизации и оптимизации
- Общее количество виртуальных ЦП, используемых в развертывании
- Общий объем ОЗУ, используемый в развертывании
- Общий объем хранилища, используемый в развертывании
- Виртуальные машины в развертывании, которые рекомендуется реорганизовать и оптимизировать, в том числе имя ресурса, простаивающие компьютеры, выключенные компьютеры, слишком крупные или недостаточно крупные компьютеры, компьютеры с низким коэффициентом использования и моментальные снимки компьютеров

С помощью параметра **Свяжитесь с владельцами проектов** на панели «Ключевые сведения» облачной зоны можно получить сводку о всех проектах, у которых есть освобождаемые ресурсы (ЦП, память и хранилище) в облачной зоне и которые отправляют оповещения некоторым или всем владельцам проектов.



## Процедура

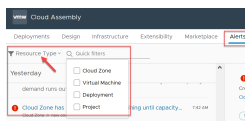
Для отображения пороговых значений оповещений об управляемых ресурсах можно использовать параметры фильтрации на странице **Оповещения**. Данные оповещений предоставляются связанным приложением vRealize Operations Manager. Для каждого оповещения предлагаются действия.

Чтобы открыть и оптимизировать нужное развертывание, его также можно выбрать в разделе **Развертывания для просмотра**. См. раздел [Использование оповещений для оптимизации развертываний в vRealize Automation](#).

1. В службе Cloud Assembly перейдите на вкладку **Оповещения** в главном меню.

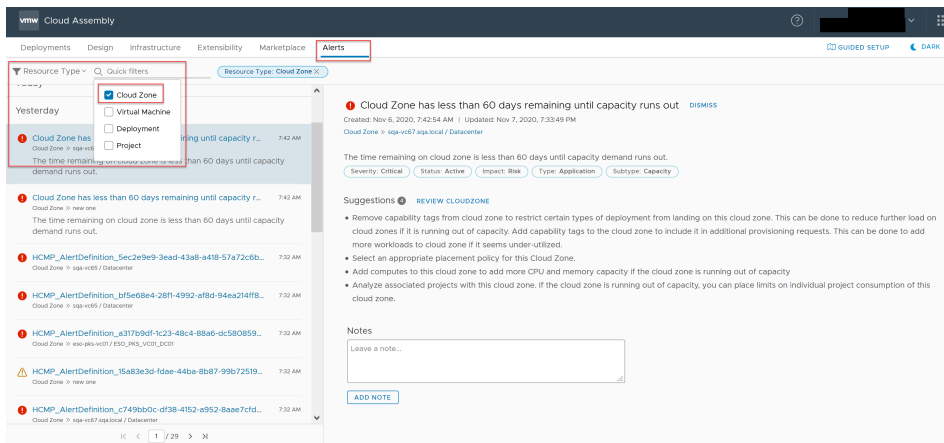


2. Для выбора режима отображения оповещений поэкспериментируйте с доступными фильтрами. Например, в раскрывающемся меню фильтров выберите параметр **Ресурсы**.



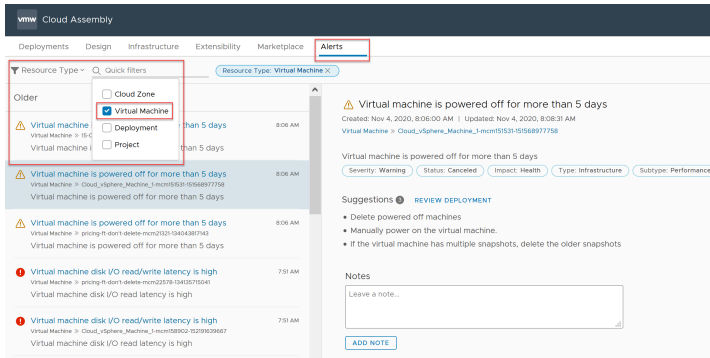
3. Для отображения оповещений и рекомендуемых действий для этих оповещений воспользуйтесь параметрами быстрой фильтрации на панели селектора.

- Отображение оповещений о ресурсах облачной зоны.



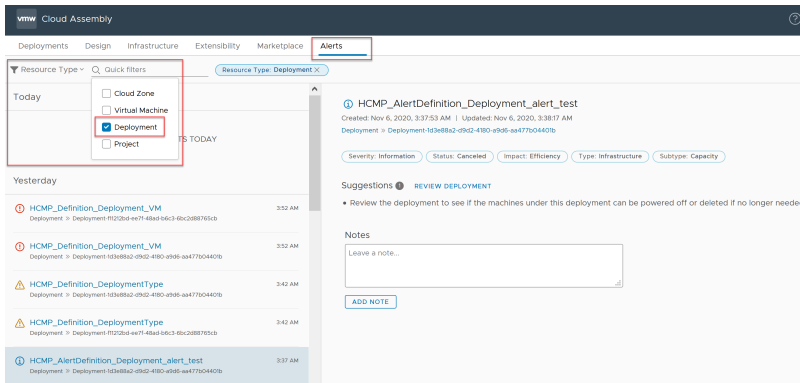
vRealize Operations Manager может отслеживать оставшееся время, оставшийся объем ресурсов, ресурсы, доступные для освобождения, и т. д.

- Отображение оповещений о ресурсах виртуальных машин.



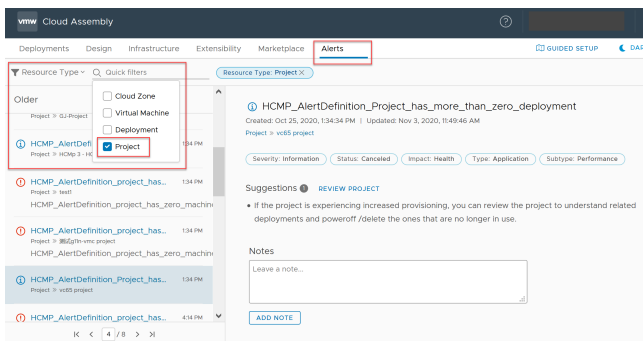
Большинство оповещений о виртуальных машинах связано с состоянием (вкл. или выкл.), задержками и т. д.

- Отображение оповещений о ресурсах развертывания.



Оповещения о развертывании относятся к высвобождаемым ресурсам и выбору объема инициализируемых ресурсов.

- Отображение оповещений о ресурсах проекта.



Оповещения о проекте относятся к высвобождаемым ресурсам и ограничениям на выделение ресурсов.

4. Ознакомьтесь с другими типами фильтров и возможностями быстрой фильтрации, чтобы эффективнее управлять списком оповещений.

- Используйте быстрые фильтры **Влияние** для фильтрации оповещений по работоспособности, риску и эффективности.



- Используйте быстрые фильтры **Важность** для фильтрации критических оповещений, срочных оповещений, предупреждений и информационных сообщений.
  - Используйте быстрые фильтры **Статус** для фильтрации активных, отмененных и отклоненных оповещений.
  - Используйте фильтры **Подтип** для фильтрации по доступности, производительности и объему ресурсов.
  - Используйте быстрые фильтры **Тип** для фильтрации оповещений, относящихся к приложениям, оборудованию, инфраструктуре, хранилищу и сети.
5. В зависимости от содержания оповещений и предложений, можно принимать любые необходимые меры.

### Дальнейшие действия

Сведения о других доступных действиях см. в разделе [Использование оповещений для оптимизации развертываний в vRealize Automation](#).

В проектах, которыми вы управляете, можно также отобразить **ключевые сведения** о ресурсах, связанных с облачной зоной. Дополнительные сведения об использовании в vRealize Automation **ключевых сведений**, предоставленных vRealize Operations Manager, см. в разделе [Использование панели управления «Ключевые сведения» для мониторинга объема ресурсов и уведомления владельцев проектов в vRealize Automation](#).

## Использование оповещений для оптимизации развертываний в vRealize Automation

Администратор облачных систем или владелец проекта может отслеживать ресурсы компьютеров и управлять ими. Это позволяет максимально оптимизировать ресурсы на основе данных, получаемых из vRealize Operations Manager и отображаемых в vRealize Automation.

Если служба vRealize Automation подключена к vRealize Operations Manager, можно получить доступ к накапливаемым сведениям о ресурсах в управляемых проектах. Благодаря оповещениям и ключевым сведениям пользователь получает информацию о различных проблемах в проектах, которыми он управляет. Они представляют собой удобное средство отправки предложений об оптимизации и вспомогательных данных, полученных из vRealize Operations Manager, владельцам проектов удобным и эффективным способом, не выходя из приложения vRealize Automation. Например, для каждого развертывания в облачной зоне может отображаться освобождаемый объем ресурсов и возможная экономия затрат. Если облачная зона содержит несколько развертываний, которые можно оптимизировать, пользователь может уведомить некоторых или всех владельцев проектов и развертываний.

Оповещения об оптимизации развертывания можно создать на панели управления «Ключевые сведения». См. раздел [Использование панели управления «Ключевые сведения» для мониторинга объема ресурсов и уведомления владельцев проектов в vRealize Automation](#). Можно связаться с владельцами проектов, чтобы они смогли открыть именованное развертывание, подлежащее оптимизации, используя ссылку

на странице **Оповещения**. Владелец проектов также может открывать развертывания напрямую и использовать вкладку **Оптимизация** для выполнения доступных задач оптимизации. К действиям, которые может выполнять владелец проекта, относятся освобождение ресурсов путем удаления второстепенных развертываний и прекращение дальнейшей подготовки ресурсов в пределах облачной зоны.

---

**Примечание** Сведения о других возможных действиях по исправлению ресурсов см. в [Использование оповещений для управления объемом, производительностью и доступностью ресурсов в vRealize Automation](#).

---

### Необходимые условия

Сведения о необходимых учетных данных и настройке для доступа к данным vRealize Operations Manager в vRealize Automation см. в [Использование оповещений для управления объемом, производительностью и доступностью ресурсов в vRealize Automation](#).

Сведения о запросе оповещений владельцев проекта об оптимизируемых развертываниях см. в [Использование панели управления «Ключевые сведения» для мониторинга объема ресурсов и уведомления владельцев проектов в vRealize Automation](#).

### О программе

Каждое развертывание содержит вкладку **Оптимизация**. Доступны следующие параметры оптимизации.

- Компьютеры, размер которых можно изменить. Позволяет отображать сведения и действия для слишком крупных или недостаточно крупных компьютеров в развертывании, а также потенциальную экономию затрат благодаря оптимизации.
- Компьютеры с низким коэффициентом использования. Позволяет отображать сведения и действия для простаивающих или выключенных компьютеров в развертывании, а также потенциальную экономию затрат благодаря оптимизации.
- Моментальные снимки компьютеров. Позволяет отображать сведения и действия для моментальных снимков компьютеров, если компьютеры в развертывании содержат моментальные снимки, а также потенциальную экономию затрат благодаря оптимизации.

Администратор может уведомлять владельцев проектов о наличии развертываний, которые можно оптимизировать. Уведомления отображаются на вкладке **Оповещения** в Cloud Assembly.

Вкладка **Оповещения** доступна только в том случае, если настроен доступ к vRealize Operations Manager. Владелец проектов может открывать и оптимизировать свои развертывания с учетом оповещений.

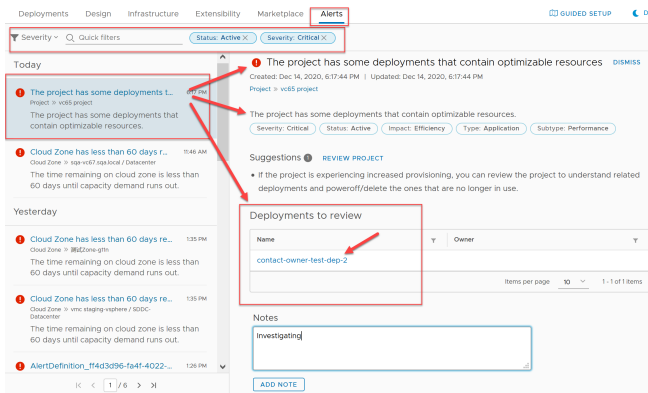
## Процедура

Для отображения пороговых значений оповещений об управляемых ресурсах можно использовать параметры фильтрации на странице **Оповещения**. Данные оповещений предоставляются связанным приложением vRealize Operations Manager. Для каждого оповещения предлагаются действия. В этом примере владелец проекта открывает свое развертывание, используя ссылку в оповещении. На вкладке **Оптимизация** развертывания отображаются доступные параметры компьютера, которые можно оптимизировать.

1. Если вы владелец проекта или администратор, перейдите на вкладку **Оповещения** в главном меню.



2. Найдите оповещение с информацией о развертывании, которое можно оптимизировать, и щелкните имя данного развертывания в разделе **Развертывания для просмотра**, чтобы его открыть и отобразить вкладку **Оптимизация** для него.



3. После открытия развертывания перейдите на вкладку **Оптимизация**.



4. Если есть компьютеры с низким коэффициентом использования, выполните необходимые действия для простаивающих, а также выключенных компьютеров. Развертывание, размер которого является недостаточным, можно выключить или удалить.
5. Если существуют компьютеры, размер которых можно оптимизировать, выполните необходимые действия для всех слишком крупных и недостаточно крупных компьютеров в развертывании.
6. Если один или несколько компьютеров в развертывании содержат моментальный снимок, каждый такой снимок можно удалить или экспортировать.
7. По окончании процедуры убедитесь, что развертывание оптимизировано оптимальным образом, и закройте его.

## Дальнейшие действия

Сведения о других доступных действиях см. в разделе [Использование оповещений для управления объемом, производительностью и доступностью ресурсов в vRealize Automation](#).

В проектах, которыми вы управляете, можно также отобразить **ключевые сведения** о ресурсах, связанных с облачной зоной. Дополнительные сведения об использовании в vRealize Automation **ключевых сведений**, предоставленных vRealize Operations Manager, см. в разделе [Использование панели управления «Ключевые сведения» для мониторинга объема ресурсов и уведомления владельцев проектов в vRealize Automation](#).

## Что можно делать со стандартным дисковым накопителем в vRealize Automation

Стандартные диски могут быть постоянными или непостоянными.

vRealize Automation поддерживает две категории дисков хранилища: стандартный диск и диск первого класса. Диск первого класса доступен только для vSphere.

### ■ vSphere

vSphere поддерживает зависимые (по умолчанию), независимые постоянные и независимые непостоянные стандартные диски. Дополнительные сведения см. в разделе [Что можно делать с постоянным дисковым накопителем в vRealize Automation](#).

При удалении виртуальной машины ее зависимые и независимые непостоянные диски также удаляются.

При удалении виртуальной машины ее независимые постоянные диски не удаляются.

Можно создать моментальный снимок зависимых и независимых непостоянных дисков. Моментальный снимок независимого постоянного диска создать нельзя.

### ■ Amazon Web Services (AWS) EBS

Можно присоединить том EBS к экземпляру вычислительного ресурса AWS или отсоединить том EBS от экземпляра вычислительного ресурса AWS.

При удалении виртуальной машины связанный с ней том EBS отсоединяется, но не удаляется.

### ■ Microsoft Azure VHD

Присоединенные диски всегда являются постоянными.

При удалении виртуальной машины необходимо указать, нужно ли удалить подключенные к ней диски хранилища.

### ■ Google Cloud Platform (GCP)

Присоединенные диски всегда являются постоянными.

Постоянные диски располагаются независимо от экземпляров виртуальной машины, поэтому постоянные диски можно отсоединить или переместить, чтобы сохранить данные даже после удаления экземпляров.

При удалении виртуальной машины связанный с ней диск отсоединяется, но не удаляется.

Дополнительные сведения см. в разделе [Дополнительные сведения о профилях хранилища в vRealize Automation](#).

### Что можно делать с постоянным дисковым накопителем в vRealize Automation

На дисках с сохранением состояния обеспечивается сохранение ценных данных от случайного удаления.

В облачном шаблоне внутри тома можно добавить свойство `persistent: true`, чтобы обеспечить сохранение диска в случае удаления Cloud Assembly или Service Broker. Диски с сохранением состояния не удаляются в случае удаления развертывания, удаления в рамках регулярного обслуживания или удаления диска.

Благодаря этому диски с сохранением состояния могут оставаться в инфраструктуре даже после удаления развертывания или диска. Для удаления таких дисков можно использовать следующие методы.

- Передайте явным образом флаг очистки в качестве параметра запроса с помощью API-интерфейса DELETE.
- Удалите необходимые диски прямо из облачной конечной точки.

Следует отметить, что для удаления таких дисков в Cloud Assembly и Service Broker нет интерфейса пользователя.

### Что можно делать с дисковым накопителем первого класса в vRealize Automation

Диск первого класса (FCD) обеспечивает управление жизненным циклом хранилища на виртуальных дисках в виде диска как услуги или в качестве дискового хранилища EBS, которое позволяет создавать диски и управлять ими независимо от виртуальных машин vSphere.

vRealize Automation поддерживает две категории дисков хранилища: стандартный диск и диск первого класса. Диски первого класса поддерживаются только для vSphere. В настоящее время vRealize Automation предоставляет функциональные возможности диска первого класса только в рамках API-интерфейса.

Диск первого класса обладает собственными возможностями управления жизненным циклом, которые реализуются независимо от ВМ. Одним из отличий диска первого класса от независимого постоянного диска является то, что диск первого класса можно использовать для создания моментальных снимков и управления ими независимо от ВМ.

Для поддержки возможностей дисков первого класса или стандартных дисков можно создать новый профиль хранилища vRealize Automation. См. разделы [Дополнительные сведения о профилях хранилища в vRealize Automation](#) и [Ресурсы хранилища в vRealize Automation](#).

Кроме того, для поддержки дисков первого класса vSphere можно добавить элемент диска первого класса `Cloud.vSphere.Disk` в облачные шаблоны и развертывания vRealize Automation. Диски первого класса, о которых собраны данные, отображаются на странице **Ресурсы > Ресурсы > Тома**.

В vCenter диски первого класса также называются *улучшенными виртуальными дисками (IVD)* или *управляемыми виртуальными дисками*.

## Возможности

Благодаря возможностям API-интерфейса vRealize Automation можно выполнять следующие операции.

- Создание, перечисление и удаление диска первого класса.
- Изменение размера диска первого класса.
- Присоединение и отсоединение диска первого класса.
- Создание моментальных снимков дисков первого класса и управление ими.
- Преобразование существующего стандартного диска в диск первого класса

Не поддерживаются следующие сценарии.

- Подготовка виртуальных машин на основе моментальных снимков в кластере хранилища данных.
- Владение пользователями и арендаторами блоков хранилища на основе устройств, а также предоставление к ним общего доступа.
- Создание и восстановление моментальных снимков ВМ.
- Присоединение хранилища к нескольким виртуальным машинам и кластерам.

Дополнительные сведения об API-интерфейсе применительно к созданию диска первого класса (FCD) и управлению им с помощью API-интерфейса vRealize Automation, включая определение профиля хранилища для использования возможностей диска первого класса, см. на сайте [code.vmware.com](https://code.vmware.com) в разделе [Что такое API-интерфейсы в vRealize Automation Cloud, и как их использовать](#) или в следующих ресурсах.

- Документация по API-интерфейсу для FCD доступна в разделе [Диск первого класса \(FCD\)](#) в [руководстве по программированию Virtual Disk Development Kit](#).
- Ссылки на документацию по примерам использования API-интерфейса для FCD в vRealize Automation см. на [странице документации по API-интерфейсу в vRealize Automation](#) для версии vRealize Automation.

## Рекомендации и ограничения

В настоящее время применительно к дискам первого класса существуют следующие замечания и ограничения.

- Диск первого класса доступен только для виртуальных машин vSphere.
- Для использования дисков первого класса требуется vSphere 6.7 с пакетом обновления 2 или более поздняя версия.
- Не поддерживается предоставление дисков первого класса в кластерах хранилищ данных.
- Для дисков первого класса функция добавления тома к нескольким узлам не поддерживается.
- Размер дисков первого класса с моментальными снимками изменить нельзя.
- Диски первого класса с моментальными снимками удалить нельзя.
- Иерархия моментальных снимков дисков первого класса может быть создана только с помощью параметра API-интерфейса `createdAt`.

- Минимальная версия оборудования виртуальной машины, необходимая для подключения диска первого класса, — vmx-13 (совместимая с ESX 6.5).

## Настройка ресурсов арендатора от нескольких поставщиков с помощью vRealize Automation

В средах с несколькими арендаторами клиенты могут управлять выделением ресурсов для каждого арендатора отдельно с помощью виртуальных частных зон.

В vRealize Automation 8.x клиенты могут настраивать среды с несколькими арендаторами, используя VMware Lifecycle Manager и Workspace ONE Access. Благодаря этим решениям пользователи могут настраивать множественную аренду, а также создавать и настраивать арендаторов. После настройки арендаторов администраторы поставщика могут создавать виртуальные частные зоны в Cloud Assembly, а затем назначать их арендаторам, используя функцию Cloud Assembly «Управление арендаторами».

Для множественной аренды требуется скоординировать и настроить три продукта VMware, как описано ниже.

- Workspace ONE Access обеспечивает поддержку инфраструктуры для множественной аренды и подключения к доменам Active Directory, которые позволяют управлять пользователями и группами в организациях-арендаторах.
- vRealize Suite Lifecycle Manager поддерживает создание и настройку арендаторов для поддерживаемых продуктов, таких как vRealize Automation. Кроме того, он предоставляет ряд возможностей управления сертификатами.
- vRealize Automation. Поставщики и пользователи заходят в vRealize Automation, чтобы получить доступ к арендаторам, в которых они создают развертывания и управляют ими.

Для настройки множественной аренды пользователям необходимо ознакомиться со всеми тремя этими продуктами и соответствующей документацией.

Дополнительные сведения о работе с vRealize Suite Lifecycle Manager и Workspace ONE Access см. в следующих разделах.

## Создание виртуальной частной зоны для vRealize Automation

Администраторы поставщика могут создать виртуальную частную зону (VPZ), чтобы выделять ресурсы инфраструктуры арендаторам в среде vRealize Automation с несколькими организациями. Администраторы также могут использовать VPZ для управления выделением ресурсов в развертываниях с одним арендатором.

Для выделения ресурсов, таких как образы, сети и хранилище, можно использовать виртуальные частные зоны (VPZ). Они выполняют функции облачных зон для каждого клиента, но в первую очередь они предназначены для использования с мультиарендными развертываниями. В любом проекте можно использовать только одну из этих функций: либо облачные зоны, либо VPZ. Кроме того, между VPZ и арендаторами существует взаимно однозначная связь. Это значит, что VPZ может быть назначена в один промежуток времени только одному арендатору.

---

**Примечание** Настройка сопоставления образов и конфигураций ресурсов для зоны VPZ выполняется на странице «Управление арендаторами».

---

VPZ можно создать с сетью NSX или без нее. При создании зоны без NSX в конечных точках vSphere существуют предельные значения, которые относятся к функциям, связанным с NSX.

- Безопасность (группы, брандмауэр)
- Сетевые компоненты (NAT)

#### Необходимые условия

- Включите и настройте множественную аренду в развертывании vRealize Automation, используя VMware Life Cycle Manager и VMware Workspace ONE Access.
- Создайте администраторов арендаторов в соответствии с конфигурацией арендаторов.
- Если необходимо использовать NSX, создайте соответствующую облачную учетную запись NSX в организации поставщика.

#### Процедура

1. Выберите **Инфраструктура > Настроить > Виртуальные частные зоны**.

На странице VPZ отображаются все существующие зоны. На ней также можно создавать зоны.



## 2. Нажмите **Создать виртуальную частную зону**.

### New Virtual Private Zone

В левой части страницы можно выбрать 4 элемента, которые позволяют настроить сводные данные и компоненты инфраструктуры для зоны.

## 3. Введите сводную информацию для новой зоны.

- а) Введите имя и описание.
- б) Выберите учетную запись, к которой применяется зона.
- в) Выберите политику размещения.

Политика размещения определяет процесс выбора узлов для развертываний в указанной облачной зоне.

- «По умолчанию». Обеспечивает распределение вычислительных ресурсов между кластерами и узлами случайным образом. Данный вариант работает на уровне отдельных компьютеров. Например, все компьютеры в заданном развертывании распределяются случайным образом по доступным кластерам и узлам, которые удовлетворяют требованиям.
- **binpack** — вычислительные ресурсы размещаются на наиболее загруженном узле, который имеет достаточно ресурсов для выполнения указанных вычислительных операций.
- **spread** — вычислительные ресурсы на уровне развертывания подготавливаются для кластера или узла с минимальным количеством виртуальных машин. Для vSphere виртуальные машины между узлами распределяет компонент Distributed Resource Scheduler (DRS). Например, все запрошенные компьютеры в развертывании размещаются в одном и том же кластере, но для следующего развертывания может быть выбран другой кластер vSphere в зависимости от существующей нагрузки.

#### 4. Выберите вычислительный ресурс для зоны.

Добавьте вычислительные ресурсы в соответствии с облачной зоной. Первоначально выбран фильтр «Включить все вычислительные ресурсы», а в приведенном ниже списке указаны все доступные вычислительные ресурсы, и они выделены в соответствующую зону. Существует два дополнительных варианта добавления вычислительных ресурсов в облачную зону.

- «Выбрать вычислительные ресурсы вручную». Выберите этот вариант, если хотите вручную выбрать вычислительные ресурсы в списке ниже. После выбора ресурсов нажмите «Добавить вычислительный ресурс», чтобы добавить ресурсы в зону.
- «Динамически включать вычислительные ресурсы по тегам». Выберите этот вариант, если необходимо выбирать вычислительные ресурсы для добавления в зону на основе тегов. Пока не будут добавлены соответствующие теги, отображаются все вычислительные ресурсы. Можно выбрать либо ввести один или несколько тегов в варианте «Включить вычислительные ресурсы с этими тегами».

При любом варианте выбора вычислительных ресурсов, чтобы удалить один или несколько вычислительных ресурсов, отображаемых на странице, выберите поле справа и нажмите «Удалить».

#### 5. Введите или выберите нужные теги.

#### 6. Выберите в меню слева хранилище, политику хранилища и другие настройки хранилища для зоны.

#### 7. В меню слева выберите сеть и определите сети, а также политику сети, которая будет использоваться в этой зоне. Кроме того, можно настроить подсистемы балансировки нагрузки и группы безопасности для выбранных политик сети.

Сеть	<ul style="list-style-type: none"> <li>■ Все существующие сети, связанные с этой зоной VPZ, появляются в таблице на вкладке «Сети».</li> <li>■ Щелкните <b>Добавить сеть</b>, чтобы просмотреть все сети, связанные с выбранным регионом. Добавление сети для использования в этой зоне.</li> <li>■ Выберите сеть и щелкните <b>Теги</b>, чтобы добавить один или несколько тегов в указанную сеть.</li> <li>■ Выберите <b>Управление диапазонами IP-адресов</b> чтобы указать диапазон IP-адресов, в рамках которого пользователи могут получить доступ к этой сети.</li> <li>■ Если применимо, перейдите на вкладку «Политики сети» и выберите политику изоляции.</li> </ul>
Политики сети	<p>Если настроено, выберите политику сети, которая будет использоваться с этой зоной для применения политики изоляции к исходящим и частным сетям.</p> <ul style="list-style-type: none"> <li>■ При необходимости выберите политику изоляции.</li> <li>■ При необходимости выберите логический маршрутизатор уровня 0 и пограничный кластер.</li> </ul>

Подсистемы балансировки нагрузки	Щелкните <b>Добавить подсистему балансировки нагрузки</b> , чтобы настроить подсистемы балансировки нагрузки для облачных учетных записей или регионов.
Группы безопасности	Щелкните <b>Добавить группу безопасности</b> , чтобы использовать группы безопасности для применения правил брандмауэра к подготовленным компьютерам.

## Результаты

Создается виртуальная частная зона с указанными ресурсами.

## Следующие шаги

Администраторы облачных систем могут связать VPZ с проектом.

1. В Cloud Assembly выберите **Администрирование > Проекты**.
2. Перейдите на вкладку «Предоставление».
3. Щелкните **Добавить зону** и выберите «Добавить виртуальную частную зону».
4. Выберите нужную VPZ в списке.
5. Можно настроить приоритет предоставления и предельные значения количества экземпляров, объема доступной памяти и количества доступных ЦП.
6. Нажмите кнопку **Добавить**.

## Управление конфигурацией виртуальной частной зоны для арендаторов vRealize Automation

Администраторы поставщика могут управлять виртуальными частными зонами (VPZ) в Cloud Assembly, контролируя выделение ресурсов инфраструктуры для каждого арендатора. На странице «Управление арендаторами» администраторы могут просматривать сведения об арендаторах и зонах VPZ, а также включать или отключать зоны VPZ для арендаторов.

По умолчанию виртуальные частные зоны не выделяются ни одному из арендаторов. Необходимо выделить зоны VPZ на этой странице, чтобы использовать их для арендаторов.

При первоначальном создании зоны VPZ по умолчанию включены. Включенная VPZ готова к выделению и использованию с указанным арендатором. Если зоны VPZ отключены, они не могут быть использованы для подготовки или выделения арендатору. Зона VPZ может быть выделена для арендатора, но отключена.

Когда администратор поставщика переходит на страницу «Управление арендаторами», на ней отображаются все доступные арендаторы и администратор может выбрать одного из них. После выбора арендатора на этой странице отображаются зоны VPZ, выделенные в данный момент для этого арендатора (при наличии). Администратор может использовать эту страницу для выделения зон VPZ выбранному арендатору.

После выделения зоны VPZ администраторы арендаторов могут добавлять ее в проекты, и затем пользователи арендатора могут использовать ее для подготовки. После выделения зоны VPZ одному арендатору ее можно назначить другому арендатору.

После включения VPZ она готова к использованию в указанном арендаторе. Администраторы поставщика могут отключить зоны VPZ, чтобы упростить обслуживание или перенастройку арендаторов, а также могут отправить пользователям уведомление об отключении. Если VPZ необходимо сделать недоступной для арендатора на постоянной основе, для нее можно отменить выделение. Если выделение существующей VPZ в арендаторе отменено по какой-либо причине, ее нельзя будет использовать для создания развертываний из этого арендатора.

#### Необходимые условия

- Настройте множественную аренду и создайте виртуальные частные зоны в соответствии со своим развертыванием.
- Настройте глобальные сопоставления образов и конфигураций ресурсов для зоны VPZ и арендаторов, используя соответствующее меню в левой части страницы «Управление арендаторами» в Cloud Assembly. См. раздел [Создание глобального сопоставления образов и конфигураций ресурсов для арендаторов vRealize Automation](#).

Эти глобальные назначения можно переопределить прямо сейчас или позже с помощью меню в верхней части страницы «Управление арендаторами». См. раздел [Настройка сопоставлений образов и конфигураций ресурсов для конкретного арендатора в vRealize Automation](#).

#### Процедура

1. В Cloud Assembly выберите «Управление арендаторами».

На странице «Управление арендаторами» отображаются все арендаторы, настроенные для организации администратора в представлении «Карточки».

2. Щелкните арендатора, чтобы его выбрать.
3. Перейдите на вкладку «Управление инфраструктурой», чтобы просмотреть все зоны VPZ, выделенные для этого арендатора.
4. Выберите **Выделить виртуальную частную зону**, чтобы открыть диалоговое окно, в котором отображаются все зоны, не выделенные для арендаторов. Выделение зоны для арендатора
5. Выберите одну зону или несколько в диалоговом окне и щелкните **Выделить для арендатора**.

#### Следующие шаги

После выделения зон VPZ администраторы арендатора могут назначать их проектам.

Администраторы поставщика могут использовать карточки арендаторов для мониторинга и управления состоянием зон VPZ.

- Если арендатора необходимо отключить, щелкните **Отключить** на карточке данного арендатора.
- Чтобы включить арендатора, щелкните **Включить** на карточке арендатора.
- Если необходимо отменить выделение арендатора, нажмите **Освободить** на карточке данного арендатора.

## Создание глобального сопоставления образов и конфигураций ресурсов для арендаторов vRealize Automation

Администраторы поставщиков могут выбрать или создать глобальные сопоставления образов и конфигураций ресурсов, которые могут быть назначены арендаторам vRealize Automation.

Глобальное сопоставление образов и конфигураций ресурсов позволяет быстро настроить сопоставления, применяемые к нескольким арендаторам. Такие сопоставления также можно быстро обновить. На странице управления арендаторами также можно создать сопоставления образов и конфигураций ресурсов для конкретного арендатора, которые могут переопределять конфигурации по умолчанию.

---

**Примечание** Сопоставления образов и конфигураций ресурсов, настроенные на странице управления арендаторами, применяются только к арендаторам в соответствии с настройками, но не применяются к организациям поставщиков в целом.

---

### Необходимые условия

### Процедура

1. В Cloud Assembly выберите «Управление арендаторами».

На странице «Управление арендаторами» отображаются все арендаторы, настроенные для организации администратора в представлении «Карточки».

2. Выберите сопоставление образов в меню слева на странице «Управление арендаторами».

На странице «Сопоставление образов» отображаются все образы, настроенные для арендаторов в Cloud Assembly, и указывается, являются ли сопоставления глобальными или они связаны только с конкретным арендатором.

## Create Image Mapping

Account / region \*

Q Search for regions

Image Name \*

Image \*

Q Search for images

Constraints

Example: !license:none:hard

Scope \*

Q All tenants

### Cloud Configuration

1	
---	--

CANCEL

CREATE

3. Нажмите **Добавить сопоставление образов**, чтобы добавить сопоставление, которое можно будет использовать с арендаторами.
- а) Выберите учетную запись или регион, к которым будет применяться сопоставление образов.
  - б) Введите имя для сопоставления образов и выберите конкретный экземпляр или версию образа, к которому оно относится.

- в) Введите нужные теги ограничений.
  - г) Выберите область для сопоставления образов. Для области можно указать значение «Все арендаторы» или «Глобальная», либо можно выбрать конкретного арендатора, к которому будет применяться сопоставление образов.
4. При необходимости можно использовать сценарий настройки облачной среды, чтобы определить настраиваемые характеристики ОС для развертываний.
- Например, в зависимости от того, где развертывается облачный шаблон — в общедоступном или частном облаке, к образу можно применить необходимые разрешения для пользователя, разрешения для ОС или иные условия. Сценарий настройки облачной среды соответствует формату `cloud-init` для образов на основе Linux или формату `cloudbase-init` для образов на основе Windows. Дополнительные сведения см. в [Дополнительные сведения о сопоставлении образов в vRealize Automation](#).
5. Нажмите кнопку **Создать**, чтобы создать сопоставление образов.
6. Нажмите **Добавить сопоставление конфигураций ресурсов**, чтобы добавить конфигурацию ресурсов, которую можно будет использовать с арендаторами.

**Create Flavor Mapping** [X]

**Account / region \*** [Q Search for regions]

**Name \*** [\_\_\_\_\_]

**Value \*** [Q Search for flavors]

**Scope \*** [Q All tenants]

[CANCEL] [CREATE]

- а) Выберите учетную запись или регион, к которому будет применяться сопоставление конфигураций ресурсов.
- б) Введите имя для создаваемого сопоставления конфигураций ресурсов.

- в) Выберите параметры размера для создаваемого сопоставления конфигураций ресурсов.

Для данной конфигурации ресурсов можно указать количество процессоров и объем памяти.

- г) Выберите область для сопоставления конфигураций ресурсов. Для области можно указать значение «Все арендаторы» или «Глобальная», либо можно выбрать конкретного арендатора, к которому будет применяться сопоставление конфигураций ресурсов. Вариант «Все арендаторы» применяется ко всем арендаторам в организации администратора поставщика.

7. Нажмите кнопку **Создать**, чтобы создать сопоставление конфигураций ресурсов.

### Результаты

После создания глобальных сопоставлений они будут отображаться на вкладке «Сопоставление конфигураций ресурсов» или «Сопоставление арендаторов» на странице управления арендаторами для соответствующих арендаторов.

### Следующие шаги

На этой странице можно изменить или удалить глобальные сопоставления образов и конфигураций ресурсов. Чтобы изменить сопоставление, выберите его и внесите необходимые изменения.

## Настройка сопоставлений образов и конфигураций ресурсов для конкретного арендатора в vRealize Automation

Cloud Assembly позволяет настроить глобальные сопоставления образов и конфигураций ресурсов, доступные всем виртуальным частным зонам (VPZ) в организации. Кроме того, можно переопределить глобальные параметры и настроить сопоставления образов и конфигураций ресурсов для конкретного арендатора в соответствии с развертываниями.

Как правило, администратор облачных систем настраивает глобальные сопоставления образов и конфигураций ресурсов, используя панель ссылок слева на странице управления арендаторами, и эти сопоставления применяются сразу ко всем арендаторам. В некоторых случаях может потребоваться создание и настройка сопоставлений образов и конфигураций ресурсов для конкретных арендаторов. Это можно сделать на странице управления арендаторами.

Сопоставления образов и конфигураций ресурсов отображаются на соответствующих вкладках на странице управления арендаторами. Чтобы изменить любое существующее сопоставление образов и конфигураций ресурсов, щелкните его. Чтобы удалить сопоставление образов или конфигураций ресурсов, выберите его, а затем нажмите кнопку **Удалить**.

### Необходимые условия

- Включите мультитенантность и настройте арендаторов для развертывания.
- Создайте соответствующие зоны VPZ.

### Процедура

1. В главном меню Cloud Assembly нажмите «Управление арендаторами».



2. Выберите арендатора, для которого необходимо настроить сопоставление образов или конфигураций ресурсов.
3. Щелкните ссылку «Сопоставление образов» в верхней части страницы, а затем нажмите **Добавить сопоставление образов**.

Появляется диалоговое окно «Создание сопоставления образов».

4. Убедитесь, что указана правильная учетная запись или регион, и добавьте имя сопоставления в текстовом поле **Имя образа**.
5. Выберите в раскрывающемся меню **Образ** базовый образ компьютера, который будет использоваться.
6. При необходимости добавьте теги ограничений для используемого образа.
7. Выберите соответствующую **область** для образа.

- Установите переключатель «Доступно только для этого арендатора», чтобы это сопоставление образов было доступно для использования только выбранным арендатором.
- Установите переключатель «Общий доступ для арендаторов», чтобы это сопоставление образов было доступно для других арендаторов.

8. Нажмите кнопку **Создать**, чтобы сохранить настроенное сопоставление образов.
9. Щелкните ссылку «Сопоставление конфигураций ресурсов» в верхней части страницы, а затем нажмите кнопку **Добавить сопоставление конфигураций ресурсов**, чтобы создать сопоставление.

Появится диалоговое окно «Создание сопоставления конфигураций ресурсов».

10. Убедитесь, что указана правильная учетная запись или регион, и добавьте имя сопоставления в текстовом поле **Имя**.
11. Введите значения параметров процессора и памяти для конфигурации ресурсов в поле **Значение**.
12. Выберите соответствующую **область** для образа.
  - Установите переключатель «Доступно только для этого арендатора», чтобы это сопоставление образов было доступно для использования только выбранным арендатором.
  - Установите переключатель «Общий доступ для арендаторов», чтобы это сопоставление образов было доступно для других арендаторов.

13. Нажмите кнопку **Создать**, чтобы сохранить настроенное сопоставление конфигураций ресурсов.

## Результаты

Сопоставления образов и конфигураций ресурсов для конкретного арендатора настроены требуемым образом.

## Создание подписок с поддержкой расширяемости для поставщиков или арендаторов

Администраторы поставщиков и арендаторов могут создавать подписки с поддержкой расширяемости для доступа к рабочим процессам vRealize Orchestrator. Рабочие процессы vRealize Orchestrator

запускаются на основе событий, если существует подписка на некоторые темы событий, соответствующие определенному этапу жизненного цикла приложения.

Характеристики подписки с поддержкой расширяемости могут быть разными и зависят от того, была ли подписка создана администратором поставщика или администратором арендатора.

- Администратор арендатора может создать подписку, но не может указать область организации. Эта подписка будет активироваться только в случае событий, инициируемых арендатором.
- Администратор поставщика может создать подписку и указать область поставщика. Подписка будет иметь те же особенности, что и подписка арендатора или среда, не являющаяся мультитенантной. Она будет активироваться на основании событий, инициируемых поставщиком.
- Поставщик может создать подписку и указать область арендатора. Подписка активируется на основании событий, инициируемых любым арендатором. Она не активируется событиями, инициируемыми поставщиком.

Подписки запускают рабочие процессы vRealize Orchestrator на основании определенных событий. Они не вызывают действия с поддержкой расширяемости. В настоящее время для любой организации поставщика поддерживается только один экземпляр vRealize Orchestrator. Дополнительные сведения о событиях, темах событий и подписках см. в разделе [Расширяемость: терминология](#).

#### Необходимые условия

Настройте арендаторов и виртуальные частные зоны для своего развертывания.

#### Процедура

1. В vRealize Automation перейдите на страницу «Подписки» и нажмите **Создать подписку**.
2. Введите **имя** и **описание** для подписки.
3. Убедитесь, что переключатель «Включить подписку» активирован.

Эту кнопку можно оставить в состоянии «Выкл.», если вы не хотите, чтобы подписка немедленно становилась активной.

4. Если вы администратор поставщика, выберите соответствующую **область организации**.

В области организации имеются два варианта: поставщик или арендатор. Если выбран вариант «арендатор», областью проекта является любой проект и эту зависимость нельзя изменить. Если выбран вариант «поставщик», можно указать область проекта в нижней части страницы «Подписки».

5. Выберите **тему события**, на которую хотите подписаться.
6. Выберите один рабочий процесс или несколько.

## Результаты

Поставщики и арендаторы могут отслеживать возвращенные события для конкретного развертывания на странице «События» в Cloud Assembly. Отображаемые результаты зависят от роли и области организации.

- Если область организации — «поставщик», то поставщики будут видеть события на основе их действий в одной и той же организации поставщика.
- Если область организации — «арендатор», арендаторы будут видеть события, а поставщик не будет их видеть. События всегда происходят в организации издателя.

1. Выберите **Расширяемость > События** в Cloud Assembly.
2. В поле поиска на странице «События» введите идентификатор развертывания, для которого нужно просматривать события.

На этой странице отображаются события, соответствующие критериям поиска.

## Работа с устаревшими виртуальными частными зонами в новых версиях vRealize Automation

Параметры конфигурации для виртуальных частных зон (VPZ, Virtual Private Zone) в Cloud Assembly были изменены. В текущих версиях vRealize Automation можно обновлять и использовать устаревшие VPZ.

В vRealize Automation 8.2 пользователи настраивали сопоставления образов и конфигураций ресурсов в VPZ. В новых версиях vRealize Automation пользователи создают сопоставления образов и конфигураций ресурсов для каждого арендатора, что повышает эффективность и гибкость настройки, особенно в развертываниях с большим количеством арендаторов. Несмотря на то что устаревшие VPZ, созданные в vRealize Automation 8.2, перенести невозможно, существует несколько вариантов их использования с новыми версиями vRealize Automation.

Первый и самый гибкий вариант — удалить устаревшие сопоставления образов и конфигураций ресурсов из старых VPZ и повторно настроить их, используя новые сопоставления, созданные на странице управления арендаторами.

1. Выберите **Инфраструктура > Настроить > Виртуальные частные зоны**, чтобы открыть страницу VPZ.
2. Выберите «Сопоставление образов», чтобы просмотреть существующее сопоставление.
3. Выберите сопоставления и щелкните, чтобы удалить их.
4. Выберите «Сопоставление образов», чтобы просмотреть существующее сопоставление.
5. Выберите сопоставления и щелкните, чтобы удалить их.
6. Закройте страницу VPZ.
7. Выберите «Сопоставление арендатора» и создайте глобальное сопоставление для соответствующих арендаторов или сопоставление для определенного арендатора.

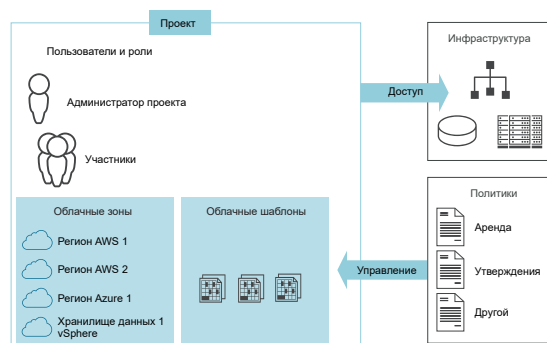
Альтернативный вариант — использование устаревших VPZ в существующей конфигурации с новыми версиями vRA. Устаревшие сопоставления образов и конфигураций ресурсов по-прежнему работают в соответствии с заданной конфигурацией, но их параметры настройки на странице VPZ доступны только для чтения. Эти варианты обеспечивают меньшую гибкость, чем первый вариант.

# Добавление проектов Cloud Assembly и управление ими

## 5

Проекты позволяют управлять доступом пользователей к облачным шаблонам Cloud Assembly и определять места развертывания шаблонов. Проекты используются для организации работы и управления действиями, которые могут выполнять пользователи, а также облачными зонами, в которых пользователи могут развертывать облачные шаблоны в рамках облачной инфраструктуры.

Администраторы облачных систем настраивают проекты, в которые они могут добавлять пользователей и облачные зоны. Любой пользователь, который создает и развертывает облачные шаблоны, должен быть членом по крайней мере одного проекта.



В эту главу входят следующие разделы:

- [Как добавить проект для группы разработчиков Cloud Assembly](#)
- [Дополнительные сведения о проектах Cloud Assembly](#)

## Как добавить проект для группы разработчиков Cloud Assembly

Необходимо создать проект и добавить в него участников и облачные зоны, чтобы участники проекта могли развертывать облачные шаблоны в связанных зонах. Администратор Cloud Assembly создает проект для группы разработчиков. После этого можно назначить отдельного администратора проекта, либо в этом качестве может выступать системный администратор.

При создании облачного шаблона сначала нужно выбрать проект, чтобы связать его с данным шаблоном. Проект должен быть создан до того, как будет создан облачный шаблон.

Убедитесь, что проекты поддерживают бизнес-потребности группы разработчиков.

- Предоставляет ли проект ресурсы, которые поддерживают цели группы? Пример выбора ресурсов инфраструктуры и параметров проекта для облачного шаблона см. в разделе [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).
- Какие развертывания требуются участникам проекта или ожидаются ими: общие или частные? Общие развертывания доступны всем участникам проекта на странице «Развертывания», а не только участникам, которые непосредственно занимаются развертыванием. Состояние общего доступа развертывания можно изменить в любое время.

Если развертывание доступно всем участникам проекта, они могут выполнять одно и то же действие по регулярному обслуживанию. Чтобы управлять способностью участников выполнять действия по регулярному обслуживанию, в Service Broker можно создать политики регулярного обслуживания. Политики применяются к развертываниям Cloud Assembly и Service Broker.

Дополнительные сведения о политиках регулярного обслуживания см. в разделе [Предоставление пользователям развертывания права на действия по регулярному обслуживанию с помощью политик](#).

Эта процедура основана на создании исходного проекта, который включает в себя только основные настройки. По мере того как команда разработчиков создает облачные шаблоны и выполняет их развертывание, можно изменять проект. Для повышения эффективности развертывания можно добавлять ограничения, настраиваемые свойства и другие возможности. См. статьи в [Дополнительные сведения о проектах Cloud Assembly](#).

#### Необходимые условия

- Убедитесь, что настроены облачные зоны. См. раздел [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).
- Убедитесь, что настроены сопоставления и профили для регионов, которые являются облачными зонами для этого проекта. См. раздел [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#).
- Убедитесь, что у вас есть необходимые разрешения для выполнения этой задачи. См. раздел [Что такое роли пользователей vRealize Automation](#).
- Определите, кто будет администратором проекта. Сведения о том, какие действия администратор проекта может выполнять в Cloud Assembly, см. в разделе [Что такое роли пользователей vRealize Automation](#).
- При добавлении групп Active Directory в проекты убедитесь, что эти группы настроены для организации. См. раздел [Изменение назначений ролей групп в vRealize Automation в Администрирование vRealize Automation](#). Если группы не синхронизированы, при попытке добавления в проект они будут недоступны.

#### Процедура

1. Выберите **Инфраструктура > Администрирование > Проекты**, а затем щелкните **Создать проект**.
2. Введите имя проекта.

### 3. Перейдите на вкладку **Пользователи**.

- а) Чтобы развертывания по участникам проекта были доступны только запрашивающему пользователю, выключите **Общий доступ к развертыванию**. Чтобы иметь возможность назначать владельцем развертывания другого участника проекта, включите **Общий доступ к развертыванию**.
- б) Добавьте пользователей с назначенными ролями.

### 4. Перейдите на вкладку **Подготовка** и добавьте одну или несколько облачных зон.

Добавьте облачные зоны и виртуальные частные зоны, содержащие ресурсы для облачных шаблонов, развернутых пользователями проекта.

Для каждой зоны можно задать приоритет, а также ограничить количество ресурсов, которые может использовать проект. К возможным ограничениям относятся количество экземпляров, объем памяти и количество ЦП. (Только для облачных зон vSphere.) Для развернутых ресурсов, основанных на шаблонах VM vSphere, можно настроить ограничения на уровне хранилища. Ограничения для хранилища оцениваются при запросе развертывания и при внесении изменений с помощью действий изменения размера диска, изменения размера загрузочного диска, удаления диска и обновления количества элементов. Данные ограничения для хранилища не применяются к другим типам ресурсов, таким как AWS, Microsoft Azure или Google Cloud Platform.

При добавлении каждой облачной зоны и применении предельных значений устанавливайте ограничения ресурсов проекта таким образом, чтобы участники могли развертывать свои облачные шаблоны.

Когда пользователи отправляют запрос на развертывание, выполняется оценка зон, чтобы определить, какие зоны имеют ресурсы для поддержки развертывания. Если развертывание поддерживается несколькими зонами, выполняется оценка приоритета, а рабочая нагрузка размещается в зоне с более высоким приоритетом, представляющим собой наименьшее целое число.

### 5. Если для развертывания рабочих нагрузок, запрашиваемых для этого проекта, требуется свыше двух часов, введите более высокое значение для параметра **Время ожидания**.

Значение по умолчанию составляет два часа.

### 6. Щелкните **Создать**.

### 7. Чтобы протестировать проект с облачными зонами, нажмите **Проверить конфигурацию** на странице «Проекты».

При моделировании выполняется стандартизованный тест предполагаемого развертывания для оценки ресурсов облачных зон проекта. Если тест завершится неудачно, можно просмотреть сведения и изменить настройки ресурсов.

### Следующие шаги

Начните работать с облачными шаблонами. См. раздел [Глава 6 Проектирование развертываний Cloud Assembly](#).

## Дополнительные сведения о проектах Cloud Assembly

Проекты представляют собой связующее звено между облачными шаблонами и ресурсами. Чем лучше вы будете понимать особенности и преимущества их использования, тем эффективнее будет процесс разработки и развертывания Cloud Assembly.

### Использование тегов проекта и настраиваемых свойств в Cloud Assembly

Если требования проекта не совпадают с содержанием облачных шаблонов Cloud Assembly, администратор может добавить ограничения по управлению или настраиваемые свойства на уровне проекта. Помимо тегов ограничений, для управления ресурсами можно добавить теги ресурсов, которые были добавлены в развернутые ресурсы в процессе подготовки.

#### Что такое теги ресурсов проекта?

Тег ресурсов проекта представляет собой стандартизированный идентифицирующий тег, который можно использовать для управления развернутыми ресурсами и обеспечения соответствия нормативным требованиям.

Теги ресурсов, определенные в проекте, добавляются ко всем ресурсам компонентов, развернутым в рамках данного проекта. Затем можно использовать стандартное тегирование для управления ресурсами с помощью других приложений, например отслеживать затраты, используя CloudHealth, а также, что важно, для обеспечения соответствия нормативным требованиям.

Например, для управления затратами администратор облачных систем хочет использовать приложение CloudHealth. Тег `costCenter:eu-cc-1234` добавляется в проект, предназначенный для разработки средства по управлению персоналом для Европейского союза. Если рабочая проекта выполняет развертывание из этого проекта, тег добавляется в развернутые ресурсы. Затем необходимо настроить средство расчета затрат, чтобы определить ресурсы, содержащие этот тег, и управлять ими. Другие проекты с другими центрами затрат будут иметь альтернативные значения для перехода с помощью ключа.

#### Что такое теги ограничений проекта?

Ограничение проекта действует как определение правил управления. Это тег `key:value`, определяющий ресурсы, которые потребляются или исключаются в рамках запроса на развертывание в облачных зонах проекта.

В процессе развертывания выполняется поиск тегов для сетей и хранилища, которые соответствуют ограничениям проекта, и развертывание на основе таких тегов.

Ограничение расширяемости используется, чтобы указать интегрированный экземпляр vRealize Orchestrator, который нужно использовать для рабочих процессов расширяемости.

При настройке ограничений проекта следует учитывать следующие форматы.

- **key:value** и **key:value:hard**. Этот тег следует использовать в любом формате, когда необходимо подготовить облачный шаблон для ресурсов с соответствующим тегом возможностей. Если соответствующий тег найти не удастся, процесс развертывания завершается сбоем. Например,



облачный шаблон, развертываемый участниками проекта, должен быть подготовлен в сети, соответствующей стандарту PCI. Следует использовать `security:pci`. Если в облачных зонах проекта сети не обнаружены, развертывание завершится сбоем, предотвращая создание незащищенных развертываний.

- **key:value:soft.** Этот тег следует использовать, если вы хотите применить совпадающий ресурс и продолжить выполнение процесса развертывания без сбоев и можете задействовать ресурсы с несовпадающим тегом. Например, если вы хотите, чтобы участники проекта развертывали облачные шаблоны в менее дорогом хранилище, и не хотите, чтобы доступность хранилища ограничивала возможности развертывания. Следует использовать `tier:silver:soft`. Если в облачных зонах проекта нет хранилища с тегом `tier:silver`, облачный шаблон будет развертываться в других ресурсах хранилища.
- **!key:value.** Этот тег можно использовать с ограничением `hard` (жесткий) или `soft` (мягкий), если необходимо исключить развертывания в ресурсах с совпадающим тегом.

Важно отметить, что теги ограничений проекта имеют более высокий приоритет, чем теги ограничений облачного шаблона, и переопределяют их во время развертывания. Если у вас есть облачный шаблон, в котором такая ситуация не должна происходить, можно использовать `failOnConstraintMergeConflict:true`. Например, в проекте есть сетевое ограничение `loc:london`, но облачный шаблон содержит параметр `loc:mumbai`. Требуется, чтобы приоритет не отдавался расположению, указанному в проекте, и развертывание завершалось сбоем с выводом сообщения о конфликте ограничений. Для этого необходимо добавить, например, следующее свойство.

```
constraints:
  - tag: 'loc:mumbai'
failOnConstraintMergeConflict:true
```

## Как можно использовать настраиваемые свойства проектов

Можно использовать настраиваемое свойство проекта для создания отчетов, запуска и заполнения параметров действий и рабочего процесса расширяемости, а также для переопределения свойств на уровне облачного шаблона.

Добавление настраиваемого свойства в развертывание позволяет использовать значение в пользовательском интерфейсе или извлекать его с помощью API-интерфейса для создания отчетов.

В функции расширяемости также может использоваться настраиваемое свойство для подписки на расширяемость. Дополнительные сведения о расширяемости см. в разделе [Увеличение и автоматизация жизненных циклов приложений с помощью действий расширяемости](#).

Облачный шаблон может содержать некоторое значение свойства, которое необходимо изменить для проекта. В качестве настраиваемого свойства можно указать альтернативное имя и значение.

Кроме того, значение свойства можно зашифровать, чтобы ни вы, ни ваши пользователи не могли увидеть значение, включенное в развертывание. Например, можно зашифровать пароль, который используется всеми пользователями в проекте, но не должен быть видимым. После шифрования значения и сохранения проекта данное значение нельзя будет ни дешифровать, ни заменить. Если снять флажок **Зашифровано**, значение будет удалено. Значение нужно будет ввести снова.

## Влияние политик размещения уровня проекта на выделение ресурсов в vRealize Automation

Администратор может определить политику размещения для проектов, в которых для использования в качестве целевой зоны развертывания подходит несколько облачных зон. Например, может быть проект, в котором необходимо развернуть облачные шаблоны в соответствии с установленным приоритетом. Или же может потребоваться сбалансировать ресурсы, развернутые в нескольких зонах, исходя из того, какая зона имеет наилучшее соотношение виртуальных машин и узлов.

### Факторы, которые необходимо учитывать при выделении

Для политики размещения по умолчанию или с распределением.

- Если у пользователя, который проводит развертывание, есть разрешение на управление облачными учетными записями, находящимися в режиме обслуживания, в процессе выделения может быть выбрана облачная учетная запись в режиме обслуживания, так как пользователю может потребоваться запустить тестовое развертывание перед закрытием окна обслуживания.
- Если у пользователя нет разрешения на управление облачными учетными записями, то облачные учетные записи, находящиеся в режиме обслуживания, будут исключены из процесса выделения.
- Узлы, которые находятся в режиме обслуживания, учитываются при определении соотношения распределения. Чтобы не учитывать узел, который находится в режиме обслуживания, при определении соотношения, необходимо задать для питания значение «выключено».

Для политики распределения.

- Соотношения определяются на основе узлов. Узлы могут быть автономными или являться частью кластера.
- Если автономный узел выключен, он не учитывается при определении соотношения.
- Если узел, который является частью кластера, выключен, состояние выключения не отражается в кластере и узел все равно учитывается при определении соотношения.

### Настройка политики размещения

При наличии в проекте нескольких облачных зон, которые одинаково подходят для использования в качестве целевой зоны развертывания, выбор этой зоны в рамках запроса на развертывание выполняется на основе настроек **Политики размещения**.

1. Выберите **Инфраструктура > Проекты** и создайте или выберите проект.
2. В проекте перейдите на вкладку **Подготовка**.

## 3. Выберите политику.

Политика размещения	Описание
По умолчанию	<p>Развертывание запрашиваемых ресурсов в первой облачной зоне, которая соответствует требованиям.</p> <p>Выберите «По умолчанию», если необходимо выполнить развертывание рабочих нагрузок в порядке приоритета и можно использовать все ресурсы на узле.</p> <p>Если выбран этот параметр, количество виртуальных машин и узлов не определяется</p>
Распределение	<p>Развертывание запрашиваемых ресурсов в облачной зоне с наименьшим количеством виртуальных машин на узлах.</p> <p>Выберите «Распределение», если необходимо распределить рабочие нагрузки между узлами, широко используя ресурсы на разных узлах.</p> <p>Если выбран этот параметр, для ресурсов облачных зон определяется и оценивается количество виртуальных машин и узлов.</p>

4. Нажмите **Сохранить**.

## Проверка применения политики

После настройки политики размещения уровня проекта можно просмотреть схему подготовки и узнать, где система планирует выполнить развертывание облачного шаблона.

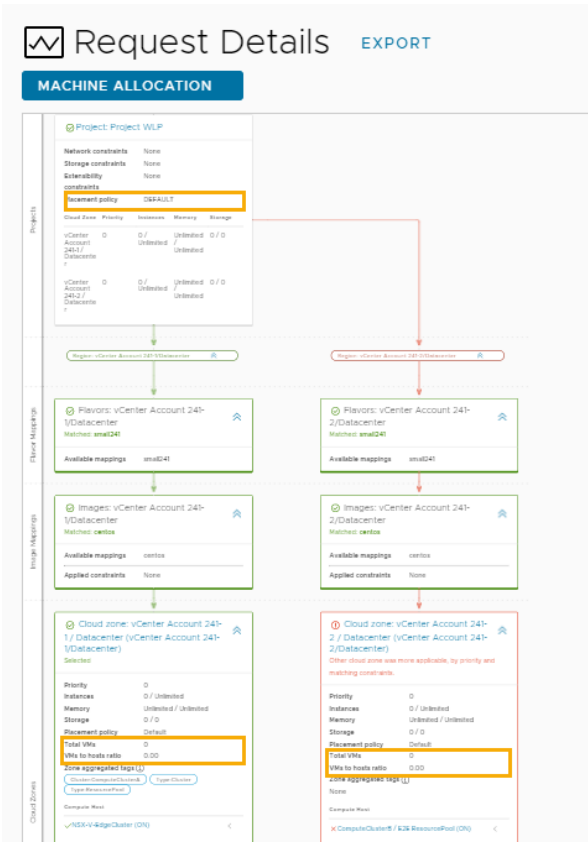
1. Выберите **Проект > Облачные шаблоны**, а затем выберите или настройте шаблон, в котором используется проект, для которого выбрана политика.
2. Щелкните **Тест**.
3. После успешного выполнения теста нажмите **Схема подготовки** в результатах тестирования.

4. Схема будет похожа на один из двух примеров.

Тип политики

Схема подготовки

По умолчанию



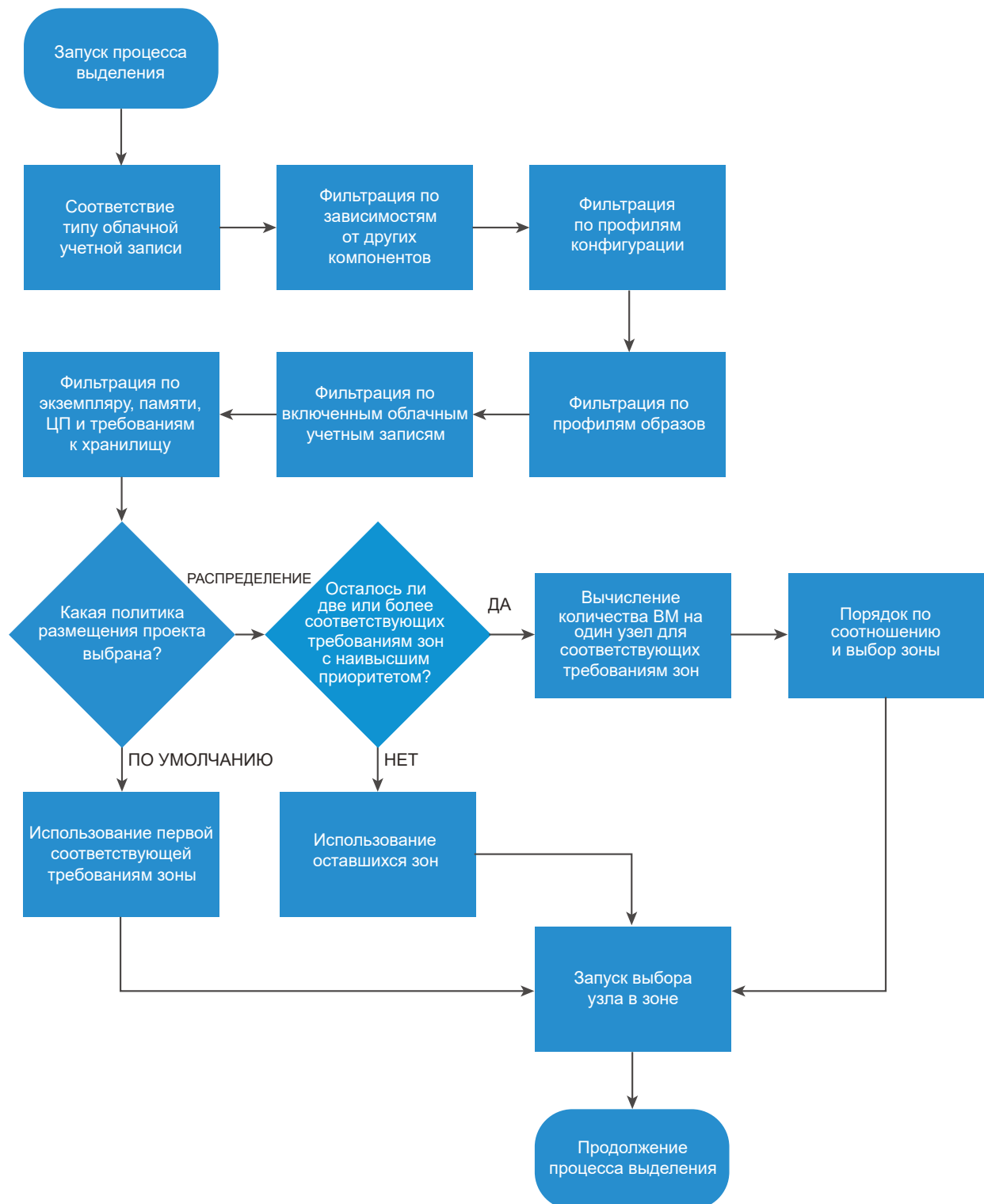
Распределение



5. Если все готово к развертыванию, вернитесь к облачному шаблону и щелкните **Развернуть**.

## Оценка политики размещения в процессе выделения ресурсов

Следующая схема позволяет понять, когда в процессе выделения ресурсов выполняется оценка политики, а также когда определяется целевая зона и узел.



## Что такое цены проекта в Cloud Assembly

Затраты, доступные в проектах Cloud Assembly, позволяют управлять затратами на ресурсы, связанными со всем проектом. Проект также включает в себя затраты на развертывание.

**Ansible-Project** DELETE

Summary Users Provisioning Kubernetes Provisioning **Price** Integrations

Price Analysis **\$10.81**  
Month to date (private cloud only)

Deployment Name	Description	Requestor	Created On	Expiring In	Price
AnsibleTower-Demo		skuradua@vmware.com	Jan 26, 2021	Never expires	\$3.07
Check-Delete		krishnaiah@vmware.com	Jan 18, 2021	Never expires	\$3.04
Ansible vSphere		skuradimutti@vmware.com	Jan 19, 2021	Never expires	\$3.01
WT with 2 machines		gijumxrv@vmware.com	Feb 14, 2021	Never expires	\$0.61
Create with templates		gijumxrv@vmware.com	Feb 14, 2021	Never expires	\$0.32
Ansible		skuradimutti@vmware.com	Jan 07, 2021	Never expires	\$0.31
Create with job templates		gijumxrv@vmware.com	Feb 14, 2021	Never expires	\$0.31

7 deployments

SAVE CANCEL

Сведения о затратах, которые отображаются для проекта и отдельных развертываний, отображаются после подготовки по крайней мере одного развертывания, связанного с проектом. Затраты рассчитываются и обновляются ежедневно, чтобы их можно было отслеживать в динамике. Первоначальные значения основаны на результатах отраслевых тестов производительности.

Администраторы облачных систем могут настраивать значения таким образом, чтобы они отражали фактические затраты.

Дополнительные сведения см. в разделе [Использование карт ценообразования в vRealize Automation](#).

## Какова роль проектов Cloud Assembly во время развертывания

Проекты позволяют контролировать доступ пользователей к облачным зонам и права собственности пользователей на подготовленные ресурсы. Независимо от того, кем вы являетесь — администратором облачной системы или разработчиком облачных шаблонов, необходимо понимать, как функционируют проекты во время развертывания, чтобы управлять развертываниями и устранять возникающие проблемы.

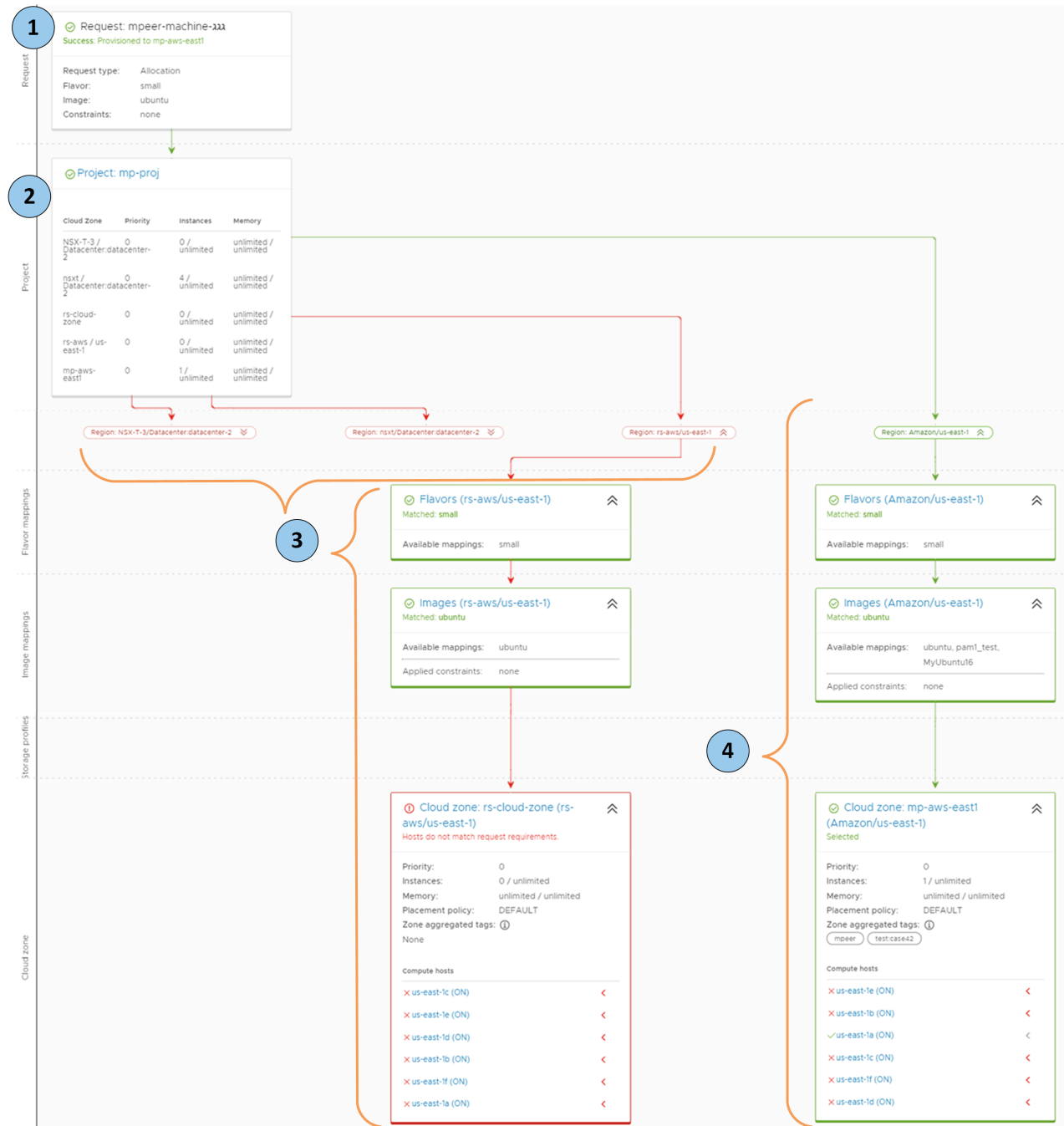
Администратор облачной системы, настраивающий проекты для различных рабочих групп, должен понимать, как определяется место развертывания компонентов облачного шаблона в проектах. Это поможет создавать проекты с необходимыми возможностями для разработчиков облачных шаблонов и устранять проблемы при сбое развертываний.

При создании облачного шаблона сначала свяжите его с проектом. Во время развертывания выполняется анализ требований к облачному шаблону с учетом облачных зон проекта, что позволяет найти оптимальное расположение для развертывания.

Описанный ниже рабочий процесс демонстрирует эту процедуру.

1. Вы отправляете запрос на развертывание облачного шаблона.

- Проект оценивает шаблон и требования к проекту, например конфигурацию ресурсов, образ и теги ограничений. Выполняется сравнение требований с облачными зонами проекта, чтобы найти зону, которая поддерживает эти требования.
- Данные зоны не имеют ресурсов для обеспечения запроса.
- Эта облачная зона соответствует требованиям к запросу, а шаблон развертывается в данном регионе облачной учетной записи.





# Проектирование развертываний Cloud Assembly

# 6

Для выполнения развертываний сначала создаются облачные шаблоны, которые ранее назывались схемами элементов. Это закодированные спецификации, определяющие параметры компьютеров, приложений и служб, создаваемых на основе облачных ресурсов при помощи Cloud Assembly.

## Принципы работы облачных шаблонов

Шаблоны могут быть ориентированы на конкретных поставщиков облачных решений или не зависеть от облака. Выбор подхода определяется облачными зонами, назначенными проекту. Узнайте у своего администратора облачных систем, какие ресурсы входят в ваши облачные зоны.

Создание шаблона Cloud Assembly — это процесс, выполняемый по схеме «инфраструктура как код». Начните с добавления ресурсов на холст проекта. Затем введите сведения с помощью редактора кода. Редактор кода позволяет ввести код напрямую или ввести значения в форму.

## Перед созданием облачного шаблона

Шаблон Cloud Assembly можно создать в любое время. Однако для его развертывания сначала необходимо [Глава 4 Создание инфраструктуры ресурсов Cloud Assembly](#) и [Глава 5 Добавление проектов Cloud Assembly и управление ими](#), включающий в себя эту инфраструктуру.

## Готовы проектировать?

Ознакомьтесь с навигацией слева или перейдите непосредственно к темам в следующей таблице.

Начало работы	Дополнительные сведения о проектах и компонентах облачных шаблонов		Дополнительные примеры
Начало работы с проектами Cloud Assembly	Входные данные пользователей в запросах vRealize Automation	Флаги ресурса Cloud Assembly для запросов	Документированный пример шаблона Cloud Assembly
Создание привязок и зависимостей между ресурсами в Cloud Assembly	Настраиваемое именование развернутых ресурсов в Cloud Assembly	Выражения Cloud Assembly	Примеры ресурсов vSphere в Cloud Assembly
Управление версиями шаблонов Cloud Assembly	Повторное использование группы свойств в Cloud Assembly	Секретные свойства Cloud Assembly	Дополнительные сведения о сетевых ресурсах в облачных шаблонах vRealize Automation

Начало работы	Дополнительные сведения о проектах и компонентах облачных шаблонов		Дополнительные примеры
<a href="#">Другие способы создания шаблонов Cloud Assembly</a>	<a href="#">Удаленный доступ к развертыванию Cloud Assembly</a>	<a href="#">Инициализация компьютера в Cloud Assembly</a>	<a href="#">Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation</a>
<a href="#">Помощь с завершением создания кода в Cloud Assembly</a>	<a href="#">Статические IP-адреса vSphere в Cloud Assembly</a>	<a href="#">Конфигурации Terraform в Cloud Assembly</a>	<a href="#">Дополнительные сведения о ресурсах подсистем балансировки нагрузки в облачных шаблонах vRealize Automation</a>
	<a href="#">Кластеры компьютеров и дисков в Cloud Assembly</a>	<a href="#">Размещение диска SCSI с помощью Cloud Assembly</a>	<a href="#">Примеры облачных шаблонов vCenter с конфигурацией Puppet</a>
	<a href="#">Настраиваемые типы ресурсов для облачных шаблонов Cloud Assembly</a>	<a href="#">Увеличение и автоматизация жизненных циклов приложений с помощью действий расширяемости</a>	

В эту главу входят следующие разделы:

- [Начало работы с проектами Cloud Assembly](#)
- [Помощь с завершением создания кода в Cloud Assembly](#)
- [Создание привязок и зависимостей между ресурсами в Cloud Assembly](#)
- [Управление версиями шаблонов Cloud Assembly](#)
- [Входные данные пользователей в запросах vRealize Automation](#)
- [Повторное использование группы свойств в Cloud Assembly](#)
- [Флаги ресурса Cloud Assembly для запросов](#)
- [Выражения Cloud Assembly](#)
- [Секретные свойства Cloud Assembly](#)
- [Удаленный доступ к развертыванию Cloud Assembly](#)
- [Размещение диска SCSI с помощью Cloud Assembly](#)
- [Инициализация компьютера в Cloud Assembly](#)
- [Кластеры компьютеров и дисков в Cloud Assembly](#)
- [Настраиваемое именование развернутых ресурсов в Cloud Assembly](#)
- [Добавление ресурса SaltStack Config в проектах Cloud Assembly](#)
- [Конфигурации Terraform в Cloud Assembly](#)
- [Настраиваемые типы ресурсов для облачных шаблонов Cloud Assembly](#)

- [Проекты Cloud Assembly, обеспечивающие подготовку к изменениям в рамках регулярного обслуживания](#)
- [Другие примеры кода Cloud Assembly](#)
- [Схема свойств ресурса vRealize Automation](#)
- [Другие способы создания шаблонов Cloud Assembly](#)
- [Увеличение и автоматизация жизненных циклов приложений с помощью действий расширяемости](#)

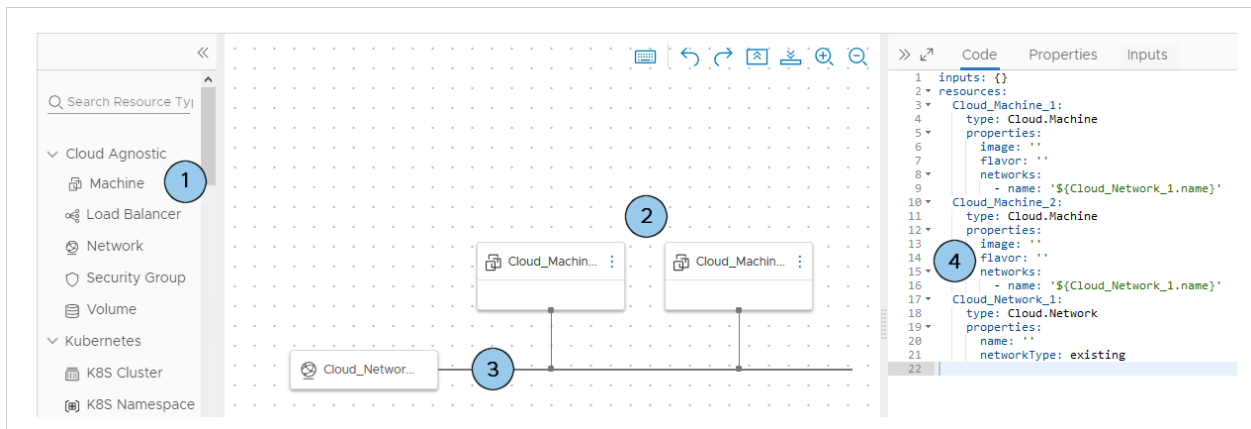
## Начало работы с проектами Cloud Assembly

На странице проекта можно создавать спецификации шаблонов Cloud Assembly для компьютеров и приложений, которые требуется подготовить к использованию.

### Использование страницы проекта

Чтобы создать облачный шаблон с нуля, перейдите в раздел **Проект > Облачные шаблоны**. Затем щелкните **Создать из > Пустой холст**.

1. Найдите ресурсы.
2. Перетащите ресурсы на холст.
3. Подключите ресурсы.
4. Чтобы настроить ресурсы, измените код облачного шаблона.



### Выбор ресурсов и добавление их на холст

Ресурсы отображаются в левой части страницы проекта. Их можно выбирать и перетаскивать.

Ресурсы, независимые от облачной среды	Независимые от облачной среды ресурсы можно развертывать в облачной среде любого поставщика. Во время подготовки в развертывании используются соответствующие ресурсы, относящиеся к определенной облачной среде. Например, если предполагается, что облачный шаблон должен быть развернут в облачных зонах AWS и vSphere, используйте ресурсы, независимые от облачной среды.
Ресурсы поставщика облачной среды	Ресурсы поставщика, например ресурсы, которые относятся к Amazon Web Services, Microsoft Azure, Google Cloud Platform или VMware vSphere, можно развертывать только в соответствующих облачных зонах AWS, Azure, GCP или vSphere.  Независимые от облачной среды ресурсы можно добавить в облачный шаблон, который содержит ресурсы, относящиеся к облачной среде определенного поставщика. Необходимо только учитывать, какие поставщики поддерживаются в облачных зонах проекта.
Ресурсы управления конфигурацией	Ресурсы управления конфигурацией зависят от интегрированных приложений. Например, ресурс Puppet может отслеживать другие ресурсы и обеспечивать их настройку.

## Подключение ресурсов

Для подключения ресурсов используйте графические элементы управления на холсте проекта в Cloud Assembly.

Для подключения ресурсы должны быть совместимыми. Например:

- Можно подключить подсистему балансировки нагрузки к кластеру компьютеров.
- Можно подключить компьютер к сети.
- Можно подключить внешнее хранилище к компьютеру.

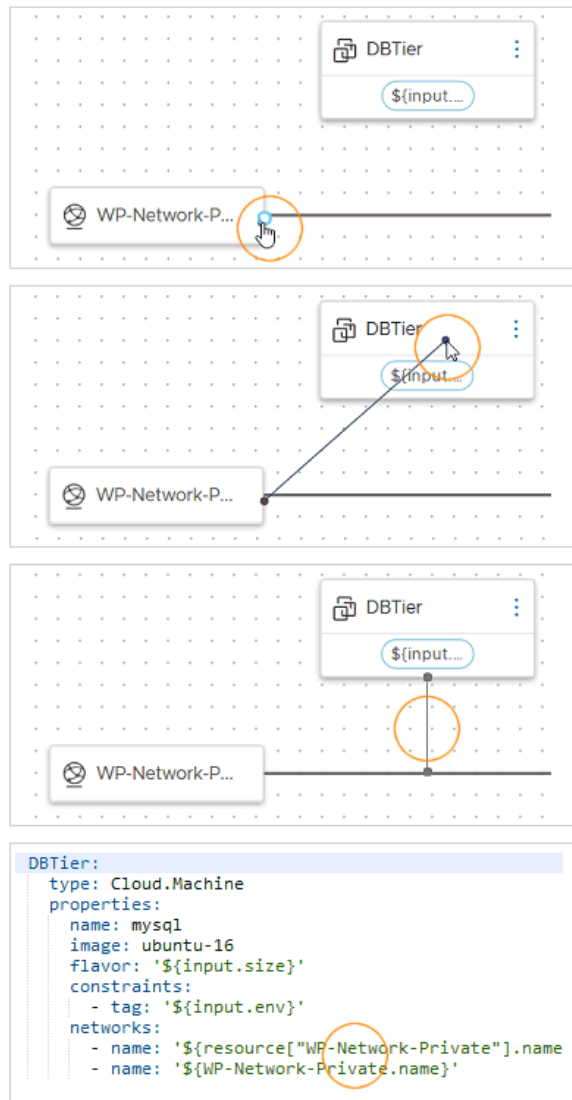
**Важно!** Для соединителя в виде сплошной линии требуется, чтобы два ресурса были развернуты в одной и той же облачной зоне. Если в ресурсы добавить конфликтующие ограничения, развертывание может завершиться сбоем.

Например, нельзя развернуть подключенные ресурсы, если теги ограничений определяют, что один из ресурсов должен быть размещен в зоне в us-west-1, а другой — в зоне в us-east-1.

Сплошные или пунктирные стрелки указывают только зависимость, но не подключение. Дополнительные сведения о зависимостях см. в разделе [Создание привязок и зависимостей между ресурсами в Cloud Assembly](#).

Чтобы подключить ресурс, наведите указатель на его кромку и дождитесь появления всплывающего значка подключения. Щелкните и перетащите этот значок к целевому ресурсу, а затем отпустите кнопку мыши.

В редакторе кода видно, что к коду целевого ресурса добавлен дополнительный код для исходного ресурса.



На рисунке компьютер SQL и частная сеть соединены друг с другом, поэтому они должны быть развернуты в одной и той же облачной зоне.

## Редактирование кода облачного шаблона

Редактор кода позволяет непосредственно вводить, вырезать, копировать и вставлять код. Если вам неудобно редактировать код, можно щелкнуть ресурс, который уже есть на холсте проекта, перейти на вкладку редактора кода **Свойства** и ввести значения там. Введенные значения свойств отображаются в коде так, как если бы они были введены напрямую.

The screenshot displays the 'WebTier' component configuration in the vRealize Automation Cloud Assembly interface. The left pane shows the JSON configuration, and the right pane shows the UI controls for the 'Properties' tab.

```

WebTier:
  type: Cloud.Machine
  properties:
    name: wordpress
    flavor: '${input.size}'
    image: ubuntu
    count: '${input.count}'
    constraints:
      - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    storage:
      disks:
        - capacityGb: '${input.archiveDiskSize}'
          name: ArchiveDisk
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all
    packages:
      - apache2
      - php
      - php-mysql
      - libapache2-mod-php
  
```

The 'Properties' tab on the right includes the following controls:

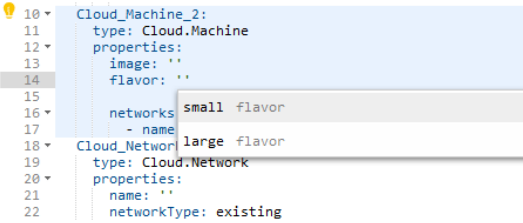
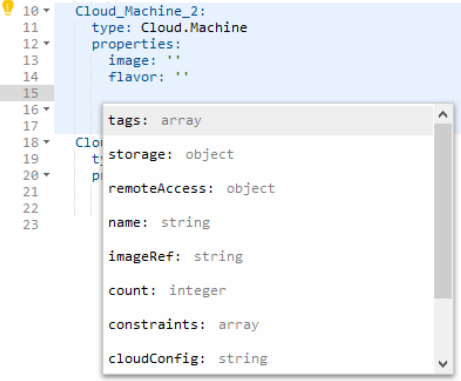
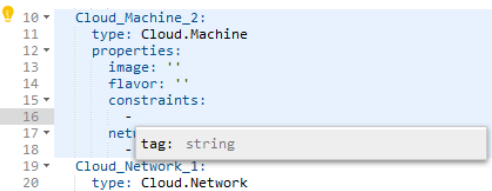
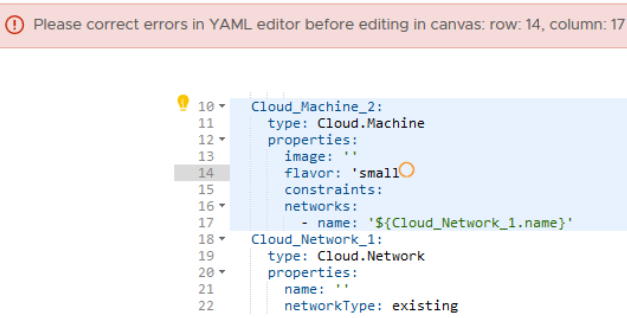
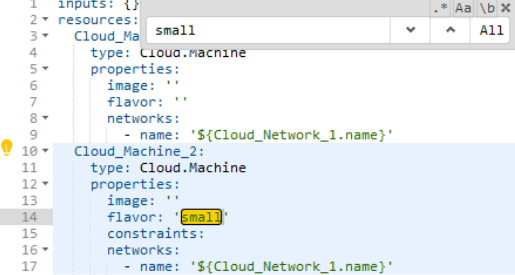
- Count:** Set to `"${input.count}"`.
- Image Type:** Set to `ubuntu`.
- Flavor \*:** Set to `${input.size}`.
- Storage:** Section for disk configuration.
- Constraints:** Includes a 'Tag' dropdown menu.
- Maximum Capacity of the disk in GB:** Set to `1`.
- Size of boot disk in GB:** Set to `1`.
- Networks:** Section for network configuration.

Следует отметить, что код можно скопировать из одного облачного шаблона в другой.

## Помощь с завершением создания кода в Cloud Assembly

При добавлении компонентов Cloud Assembly и их подключении на холсте можно создать только начальный код. Для выполнения их полной настройки код необходимо отредактировать.

Редактор кода позволяет ввести код напрямую или ввести значения свойств в форму. Чтобы упростить создание кода напрямую, в редактор Cloud Assembly добавлены функции обработки синтаксиса и проверки ошибок.

Подсказки в редакторе	Пример
Доступные значения	
Разрешенные свойства	
Дочерние свойства	
Синтаксические ошибки	
Сочетание клавиш CTRL+F для поиска	

Подсказки в редакторе	Пример
<p>Необязательные параметры</p> <ul style="list-style-type: none"> <li>+ name</li> <li>+ tags</li> <li>+ storage</li> <li>+ imageName</li> </ul>	<pre> 1 inputs: {} 2 resources: 3   Cloud_Machine_1: 4     type: Cloud.Machine 5     properties: 6       image: '' 7       flavor: '' 8       networks: 9         - name: '\${Cloud_Network_1.name}' 10  Cloud_Machine_2: 11    type: Cloud.Machine 12    properties: 13      image: '' 14      flavor: 'small' 15      constraints: 16      networks: 17        - name: '\${Cloud_Network_1.name}' </pre>

Справка по схеме

Для всех настраиваемых свойств также можно обращаться к схеме типов ресурсов vRealize Automation на сайте VMware {code}.

### cloudConfig

Type string

When provisioning an instance, machine cloud-init startup instructions from user data fields. Sample cloud config instructions:

```

#cloud-config
repo_update: true
repo_upgrade: all
packages:
- httpd
- db-server

runcmd:
- [ sh, -c, "amazon-linux-extras insta
- systemctl start httpd
- sudo systemctl enable httpd

```

```

DBTier:
type: Cloud.Machine
properties:
  name: mysql
  image: ubuntu-16
  flavor: '${input.size}'
  constraints:
    - tag: '${input.env}'
  networks:
    - name: '${resource["WP-Network-Private
    - name: '${WP-Network-Private.name}'
  remoteAccess:
    authentication: usernamePassword
    username: '${input.username}'
    password: '${input.userpassword}'
  cloudConfig:
    cloud-config
    repo_update: true
    repo_upgrade: all

    packages:
      - mysql-server

    runcmd:
      - sed -e '/bind-address/ s/^#/#/' -i
      - service mysql restart
      - mysql -e "GRANT ALL PRIVILEGES ON *.
      - mysql -e "FLUSH PRIVILEGES;"
  attachedDisks: []
  bTier:
  type: Cloud.Machine

```

## Создание привязок и зависимостей между ресурсами в Cloud Assembly

При развертывании шаблона Cloud Assembly для одного ресурса может предварительно потребоваться наличие другого.

**Важно!** Стрелки означают зависимость, а не подключение. Процедуру подключения ресурсов для их взаимодействия друг с другом см. в разделе [Начало работы с проектами Cloud Assembly](#).

### Явные зависимости

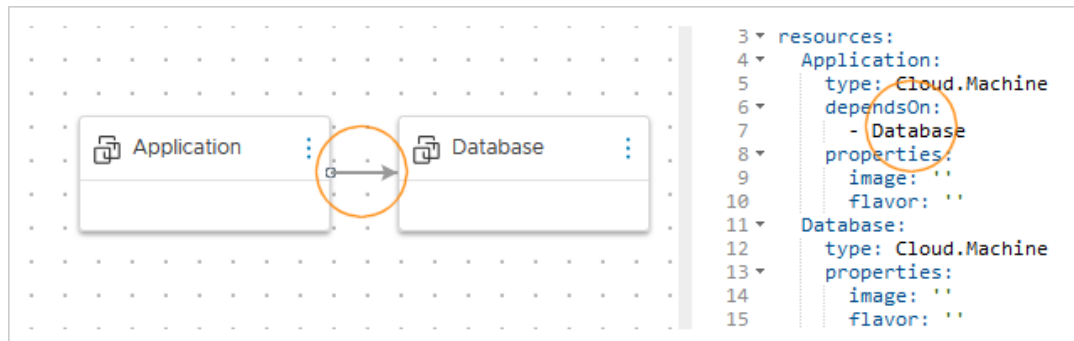
Иногда для одного ресурса необходимо, чтобы сначала был развернут другой ресурс. Например, для создания сервера приложений и настройки доступа к нему может потребоваться наличие сервера базы данных.



Явная зависимость устанавливает последовательность сборки во время развертывания, а также при вертикальном или горизонтальном масштабировании. Для добавления явной зависимости можно использовать графический холст проекта или редактор кода.

- На холсте проекта: нарисуйте соединение, которое начинается на зависимом ресурсе и заканчивается на ресурсе, который необходимо развернуть первым.
- В редакторе кода: добавьте свойство `dependsOn` к зависимому ресурсу и укажите ресурс, который необходимо развернуть первым.

Явная зависимость отображается на холсте сплошной стрелкой.



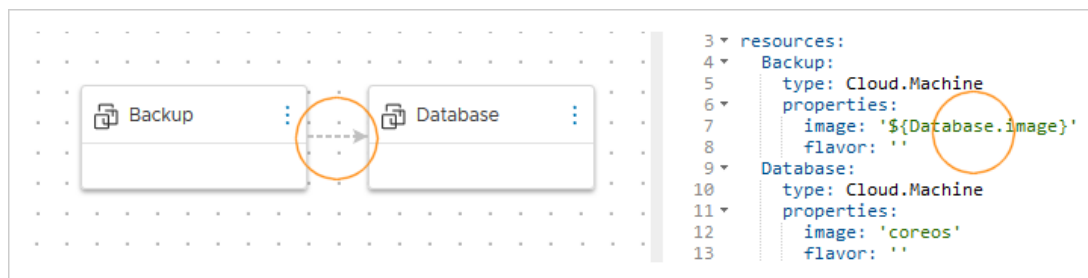
## Привязки свойств

Иногда для свойства ресурса требуется значение, существующее в свойстве другого ресурса. Например, для сервера резервного копирования может потребоваться образ операционной системы сервера базы данных, для которого выполняется резервное копирование, поэтому сервер базы данных должен быть создан раньше.

Привязка свойства, также называемая неявной зависимостью, контролирует последовательность сборки, ожидая, пока необходимое свойство станет доступно, после чего проводится развертывание зависимого ресурса. Для добавления привязки свойства следует использовать редактор кода.

- Измените зависимый ресурс, добавив свойство, которое указывает ресурс, и свойство, которое должно существовать раньше.

Привязка свойства отображается на холсте пунктирной стрелкой.



## Управление версиями шаблонов Cloud Assembly

Разработчик облачных шаблонов может использовать безопасную процедуру создания моментального снимка рабочего проекта, прежде чем вносить дальнейшие изменения.

Во время развертывания можно выбрать любую из версий для развертывания.

### Получение версии облачного шаблона

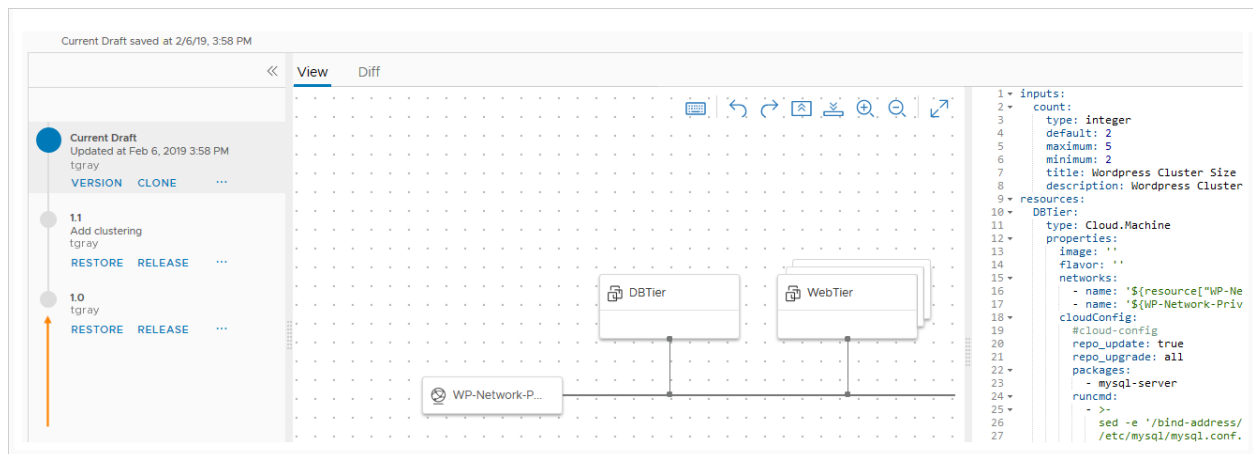
На странице проекта щелкните **Версия** и укажите имя.

Имя должно быть буквенно-цифровым, указываться без пробелов и содержать в качестве специальных символов только точки, дефисы и символы подчеркивания.

### Восстановление старой версии

На странице проекта щелкните **Журнал версий**.

В левой части окна выберите старую версию, чтобы проверить ее на холсте и в редакторе кода. При обнаружении нужной версии щелкните **Восстановить**. При восстановлении текущий черновик перезаписывается, но именованные версии не удаляются.



### Публикация версии в Service Broker

На странице проекта щелкните **Журнал версий**.

В левой части выберите версию и опубликуйте ее.

Пока текущему черновику не будет назначена версия, опубликовать его нельзя.

### Повторный импорт версии в Service Broker

Чтобы включить новую версию для пользователей каталога, повторно импортируйте ее.

В Service Broker выберите **Содержимое и политики > Источники содержимого**.

В списке источников щелкните источник проекта, который содержит облачный шаблон с новой опубликованной версией.

Щелкните **Сохранить и импортировать**.

## Сравнение версий облачного шаблона

При существовании нескольких изменений и версий может понадобиться определить различия между ними.

В Cloud Assembly в представлении «Журнал версий» выберите версию и щелкните **Разница**. Затем в раскрывающемся списке **Разница с** выберите другую версию для сравнения.

Следует отметить, что можно переключаться между просмотром различий в коде или отличий визуальной топологии.

Рис. 6-1. Различия в коде

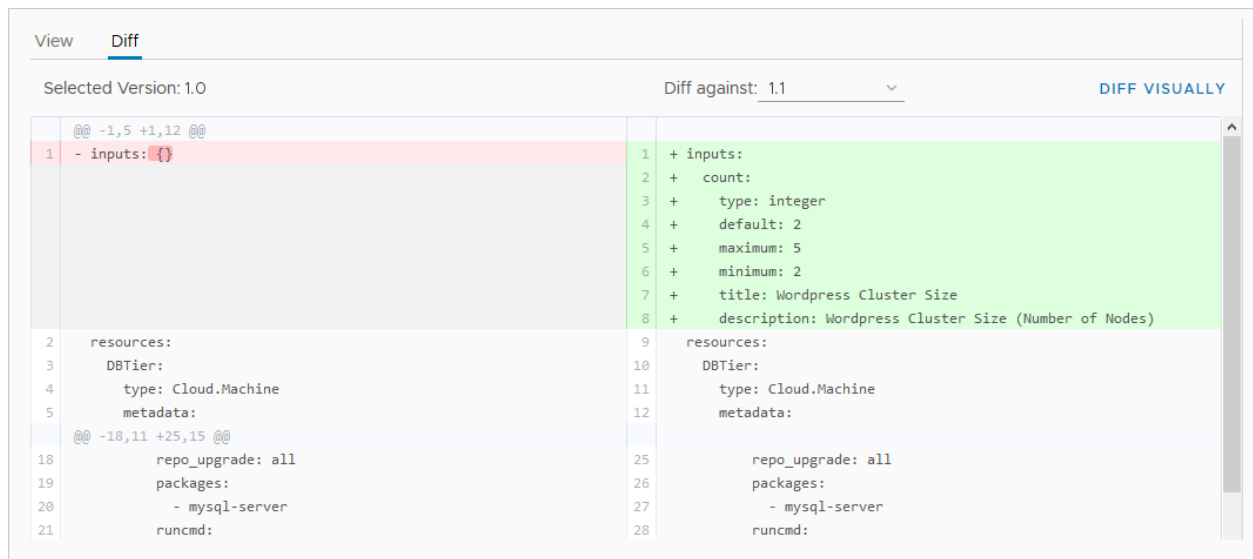
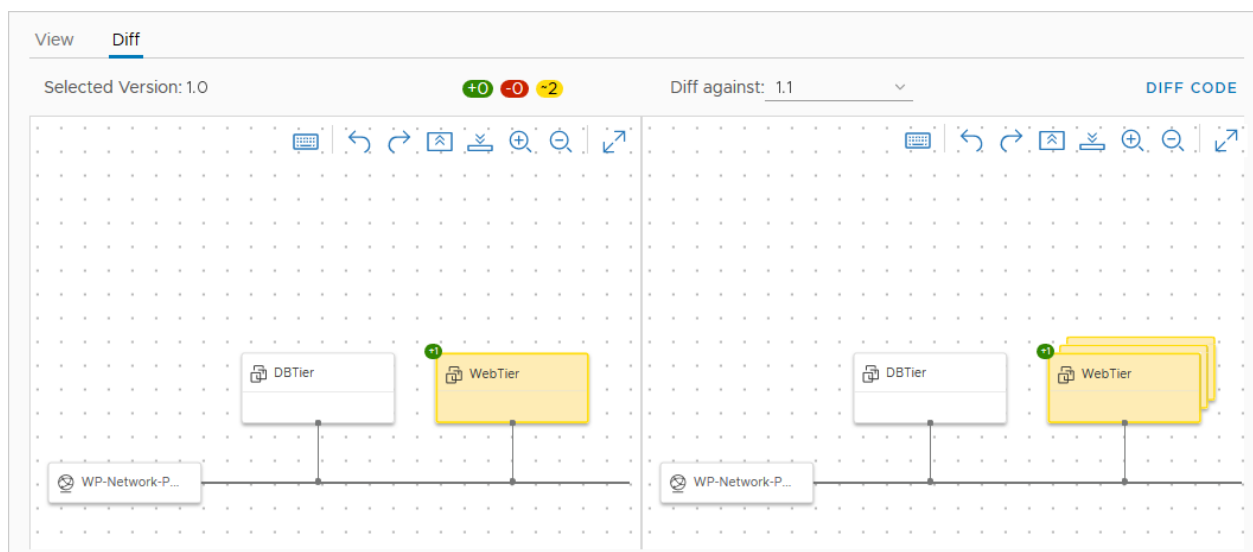


Рис. 6-2. Различия в визуальной топологии



## Клонирование облачного шаблона

Несмотря на то что данная процедура отличается от процедуры сохранения версии, на странице проекта при выборе Действия > **Клонировать** создается копия текущего шаблона для альтернативной разработки.

## Входные данные пользователей в запросах vRealize Automation

Разработчики облачных шаблонов могут использовать входные параметры, чтобы пользователи могли самостоятельно выбирать настройки при отправке запроса.

### Принципы работы входных данных

Теперь, когда пользователи вводят свои входные данные, разработчикам больше не нужно сохранять огромное количество незначительно различающихся облачных шаблонов. Кроме того, входные данные могут подготавливать шаблон для операций по регулярному обслуживанию. См. раздел [Использование входных данных облачного шаблона для обновлений при регулярном обслуживании vRealize Automation](#).

Следующие входные данные показывают, как можно создать один облачный шаблон для сервера базы данных MySQL, где пользователи смогут выполнять развертывание этого облачного шаблона в разных средах облачных ресурсов и каждый раз применять различный объем ресурсов и различные учетные данные.

### Добавление входных параметров

Добавьте раздел `inputs` в код шаблона, где задаются выбираемые значения.

В следующем примере в качестве выбираемых значений используется размер компьютера, операционная система и число кластерных серверов.

```
inputs:
  wp-size:
    type: string
```

```

enum:
  - small
  - medium
description: Size of Nodes
title: Node Size
wp-image:
  type: string
  enum:
    - coreos
    - ubuntu
  title: Select Image/OS
wp-count:
  type: integer
  default: 2
  maximum: 5
  minimum: 2
  title: Wordpress Cluster Size
  description: Wordpress Cluster Size (Number of nodes)

```

Если вам неудобно редактировать код, можно перейти на вкладку **Входные данные** в редакторе кода и там настроить параметры. В следующем примере показаны некоторые входные данные для базы данных MySQL, упомянутой выше.

The screenshot shows the 'Inputs' tab of the 'Cloud Template Inputs' editor. At the top, there are tabs for '>>', 'Code', 'Properties', and 'Inputs'. Below the tabs are buttons for '+ NEW', 'EDIT', and 'X DELETE'. A table lists the inputs:

<input type="checkbox"/>	Name	Title	Type	Default Value
<input type="checkbox"/>	size	Tier Machine Size	string	
<input type="checkbox"/>	username	Database Username	string	
<input type="checkbox"/>	userpassword	Database Password	string	****
<input type="checkbox"/>	databaseDiskSize	MySQL Data Disk Size	number	4

Below the table is a modal titled 'Edit Cloud Template Input: size'. It contains the following fields:

- Name \***: size
- Title**: Tier Machine Size
- Description**: Size of Nodes
- Type**: string (dropdown menu)
- Encrypted**: ☐

## Ссылки на входные параметры

Затем в разделе `resources` создайте ссылку на входной параметр, используя синтаксис `${input.СВОЙСТВО-ИМЯ}`.

Если имя свойства содержит пробелы, вместо использования нотации с точкой используйте квадратные скобки и двойные кавычки: `${input["ИМЯ СВОЙСТВА"]}`

**Важно!** Слово `input` может использоваться в коде облачных шаблонов только для указания входных параметров.

```
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      flavor: '${input.wp-size}'
      image: '${input.wp-image}'
      count: '${input.wp-count}'
```

## Необязательные входные данные

Входные данные, как правило, являются обязательными и отмечены звездочкой. Чтобы сделать входные данные необязательными, задайте пустое значение по умолчанию, как показано ниже.

```
owner:
  type: string
  minLength: 0
  maxLength: 30
  title: Owner Name
  description: Account Owner
  default: ''
```

Testing Basic

Environment: env.dev ⓘ

Tier Machine Size: small ⓘ

Owner Name ⓘ

Database Username: ouradmin

Database Password: .....

CANCEL TEST

## Список входных свойств

Свойство	Описание
const	Используется с oneOf. Реальное значение, связанное с понятным заголовком.
default	Предварительно заполненное значение для входных данных. Значение по умолчанию должно иметь допустимый тип. Не вводите слово в качестве значения по умолчанию там, где нужно ввести целое число.
description	Текст справки по входным данным для пользователя.
encrypted	Указывает, следует ли шифровать вводимые пользователем данные (true или false). К данным, которые обычно шифруются, относятся пароли. Также можно создать зашифрованные свойства, которые можно повторно использовать в нескольких облачных шаблонах. См. раздел <a href="#">Секретные свойства Cloud Assembly</a> .
enum	Раскрывающийся список разрешенных значений. Используйте следующий пример в качестве инструкции по формату. <pre>enum:   - value 1   - value 2</pre>
format	Задаёт ожидаемый формат входных данных. Например, формат (25/04/19) поддерживает ввод даты и времени. Позволяет использовать управляющий элемент выбора даты в настраиваемых формах Service Broker.
items	Объявляет элементы в массиве. Поддерживаемые значения: число, целое число, строка, логическое значение или объект.
maxItems	Максимальное количество выбираемых элементов в массиве.
maxLength	Максимальное разрешенное количество символов в строке. Например, чтобы ограничить размер поля 25 символами, введите maxLength: 25.
maximum	Наибольшее допустимое значение числа или целого числа.
minItems	Минимальное количество выбираемых элементов в массиве.
minLength	Минимальное разрешенное количество символов в строке.
minimum	Наименьшее допустимое значение для числа или целого числа.
oneOf	Позволяет отображать понятное имя (title) вместо менее понятного значения (const) в форме для ввода пользовательских данных. При задании значения по умолчанию следует указывать const, а не title. Допустимо для использования с данными следующих типов: строка, целое число и число.

Свойство	Описание
pattern	Допустимые символы для ввода строковых данных в виде регулярного выражения. Например, '[a-z]+' или '[a-zA-Z@#]+\$'.
properties	Объявляет блок свойств key:value для объектов.
readOnly	Используется только для пометки формы.
title	Используется с oneOf. Понятное имя для значения const. Этот заголовок отображается в форме для ввода пользовательских данных во время развертывания.
type	Тип данных: число, целое число, строка, логическое значение или объект.  <b>Важно!</b> Логическое значение добавляет пустой флажок в форму запроса. Если оставить поле нетронутым, для входных данных не будет задано значение «ложь».  Чтобы задать для входных данных значение «ложь», необходимо проверить, а затем очистить поле.
writeOnly	Заменяет введенные в форму символы звездочками. Нельзя использовать в сочетании с enum. Отображается в настраиваемых формах Service Broker как поле для ввода пароля.

## Дополнительные примеры

### Строка с перечислением

```
image:
  type: string
  title: Operating System
  description: The operating system version to use.
  enum:
    - ubuntu 16.04
    - ubuntu 18.04
  default: ubuntu 16.04

shell:
  type: string
  title: Default shell
  description: The default shell that will be configured for the created user.
  enum:
    - /bin/bash
    - /bin/sh
```



## Целое число с минимальным и максимальным значениями

```
count:
  type: integer
  title: Machine Count
  description: The number of machines that you want to deploy.
  maximum: 5
  minimum: 1
  default: 1
```

## Массив объектов

```
tags:
  type: array
  title: Tags
  description: Tags that you want applied to the machines.
  items:
    type: object
    properties:
      key:
        type: string
        title: Key
      value:
        type: string
        title: Value
```

## Строка с понятными именами

```
platform:
  type: string
  oneOf:
    - title: AWS
      const: platform:aws
    - title: Azure
      const: platform:azure
    - title: vSphere
      const: platform:vsphere
  default: platform:aws
```

## Строка с проверкой шаблона

```
username:
  type: string
  title: Username
  description: The name for the user that will be created when the machine is provisioned.
  pattern: ^[a-zA-Z]+$
```

## Строка как пароль

```
password:
  type: string
  title: Password
  description: The initial password that will be required to logon to the machine.
  Configured to reset on first login.
  encrypted: true
  writeOnly: true
```

## Строка как текстовая область

```
ssh_public_key:
  type: string
  title: SSH public key
  maxLength: 256
```

## Логическое значение

```
public_ip:
  type: boolean
  title: Assign public IP address
  description: Choose whether your machine should be internet facing.
  default: false
```

## Селектор даты и времени в календаре

```
leaseDate:
  type: string
  title: Lease Date
  format: date-time
```

# Действия vRealize Orchestrator в качестве входных данных

В шаблон Cloud Assembly можно включить действия vRealize Orchestrator в качестве входных данных облачного шаблона.

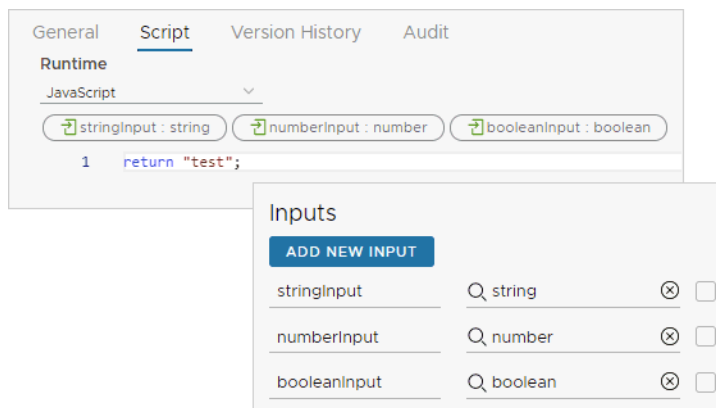
## Добавление действия vRealize Orchestrator во входные данные облачного шаблона

Чтобы использовать действия vRealize Orchestrator в качестве входных данных облачного шаблона, выполните следующие инструкции.

1. В экземпляре vRealize Orchestrator, встроенном в vRealize Automation, создайте нужное действие.

Действие vRealize Orchestrator должно включать только простейшие типы: строку, целое число, число и логическое значение. Типы vRealize Orchestrator не поддерживаются.

В этом простом примере действие vRealize Orchestrator получает три экземпляра входных данных и возвращает жестко заданную строку.



- В Cloud Assembly создайте или измените облачный шаблон.
- В редакторе кода перейдите на вкладку **Входные данные** и выберите **Новые входные данные облачного шаблона**.
- Чтобы добавить входные данные действия vRealize Orchestrator, щелкните тип и выберите **Константа**.

Отдельно добавьте входные данные каждого действия vRealize Orchestrator в качестве новых входных данных облачного шаблона.

The screenshot shows the 'New Cloud Template Input' form. It has the following fields and options:

- Name \***: numberInput
- Display Name**: Number for VRO
- Description**: (Empty text area)
- Type**: A row of buttons: STRING, INTEGER, **NUMBER** (selected), BOOLEAN, OBJECT, ARRAY. An orange arrow points to the 'NUMBER' button.
- Default value source**: Two radio buttons: **Constant** (selected), External source. An orange arrow points to the 'Constant' radio button.
- Default value**: (Empty text field)

- После добавления входных данных действия создайте новые входные данные облачного шаблона, щелкните **Внешний источник** и **Выбрать**.

New Cloud Template Input

Name \* vroAction

Display Name VRO Action

Description

Type

STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

Default value source ☐ Constant ☒ External source

Action Add an existing action

SELECT

- В разделе **Действие** найдите и выберите действие vRealize Orchestrator, которое создали, и щелкните **Сохранить**.

Add an existing action

Action \*

returnSimpleAction  
com.form.service.test

CANCEL SAVE

При разворачивании облачного шаблона настройки действия vRealize Orchestrator отображаются в форме входных данных для запрашивающего пользователя.

Values for VRO

String for VRO

VRO Action

Number for VRO

On-Off for VRO ☐

## Настраиваемые значения по умолчанию

Чтобы заполнить форму входных данных значениями по умолчанию, при добавлении действия vRealize Orchestrator качестве внешнего источника выполните одно из следующих действий.

- Вручную введите значение свойства по умолчанию.

Очистите поле **Привязка** и введите значение.

Add an existing action

Action \*

Action Parameters

Input  ☐ Bind

- Используйте другое значение свойства из входных данных, которые уже имеются в облачном шаблоне.

Нажмите **Привязка**, а затем в раскрывающемся меню выберите свойство.

Add an existing action

Action \*

Action Parameters

Input  ☒ Bind

Canvas fields

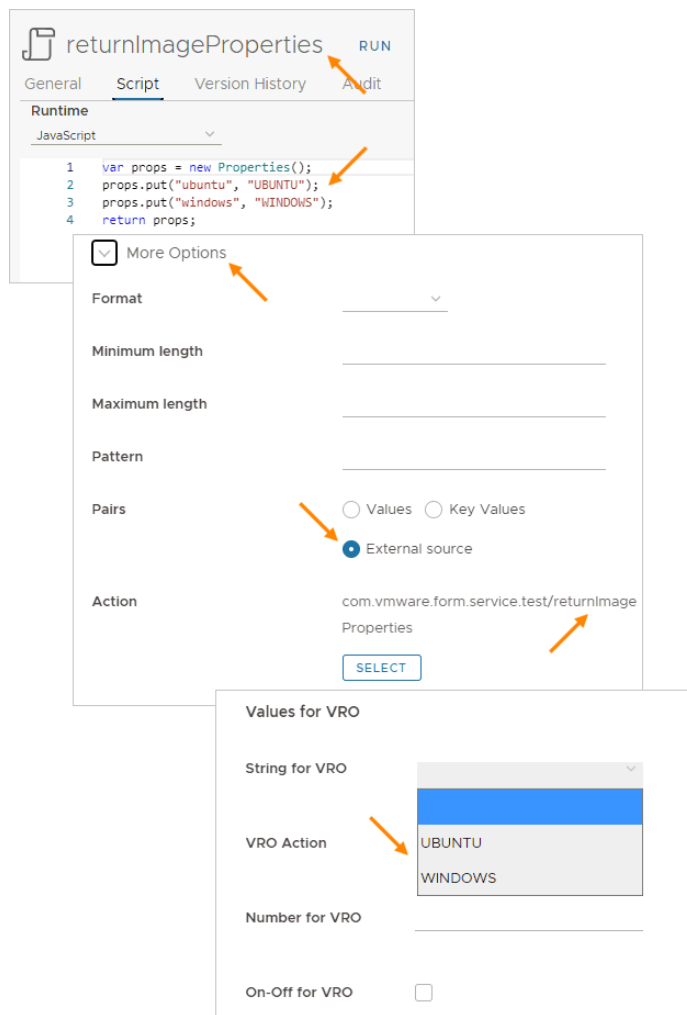
- Hard String
- Input String

## Выбор и добавление значений из перечня в vRealize Orchestrator

Чтобы создать в форме входных данных список для выбора из перечисленных значений на основе vRealize Orchestrator, при добавлении входных данных облачного шаблона выполните следующие действия.

1. В vRealize Orchestrator создайте действие сопоставления значений, которые должны быть в списке.
2. В Cloud Assembly при добавлении входных данных облачного шаблона разверните узел **Дополнительные параметры**.
3. В разделе **Пары** щелкните **Внешний источник**, нажмите **Выбрать** и добавьте созданное действие vRealize Orchestrator.

**Примечание** Если при добавлении свойства также создается значение по умолчанию, это значение должно в точности совпадать с одним из перечисленных значений из действия vRealize Orchestrator.



## Повторное использование группы свойств в Cloud Assembly

Если у вас есть свойства Cloud Assembly, которые всегда применяются вместе, их можно объединить в группе свойств.

Группу свойств можно быстро добавлять в разные проекты Cloud Assembly. Это будет быстрее, чем добавление одних и тех же свойств по одному. Кроме того, в данном случае возникает единое место для сохранения или изменения набора свойств, что обеспечивает их согласованное применение.

Создавать, обновлять или удалять группу свойств может только администратор Cloud Assembly. Администратор может предоставить доступ к группе свойств для всей организации или разрешить ее использовать только в рамках проекта.

---

**Осторожно!** Группа свойств может быть включена в разные облачные шаблоны, включая те, которые уже опубликованы в каталоге. Изменения группы свойств могут распространяться на других пользователей.

---

Существует два типа групп свойств.

- **Группы входных свойств в Cloud Assembly**

Группы входных свойств выполняют сбор и применяют согласованный набор свойств во время запроса пользователя. Группы входных свойств могут содержать записи, которые пользователь должен добавить или выбрать, либо значения только для чтения, необходимые в проекте.

Свойства, которые пользователь может изменять или выбрать, могут быть доступными для чтения или зашифрованными. Свойства «только для чтения» отображаются в форме запроса, но их нельзя изменить. Если значения «только для чтения» необходимо полностью скрыть, используйте группу свойств констант.

- **Группы свойств констант в Cloud Assembly**

Группы свойств констант применяют заранее известные свойства без запроса (автоматически). В сущности, группы свойств констант представляют собой невидимые метаданные. Они вводят значения в проекты Cloud Assembly таким образом, что пользователь, отправляющий запрос, не может прочесть эти значения или даже не знает об их существовании. Это, например, могут быть лицензионные ключи или учетные данные для учетной записи домена.

Cloud Assembly обрабатывает группы свойств таких типов по-разному. При создании группы свойств необходимо сначала указать ее тип: входные данные или константы. Группу свойств создать нельзя, а также нельзя преобразовать существующий набор свойств и соответствующую группу свойств из одного типа в другой.

## Группы входных свойств в Cloud Assembly

Группы входных свойств Cloud Assembly, как правило, содержат связанные параметры, которые пользователь может настраивать или выбирать. Они также могут включать в себя значения «только для чтения», необходимые при проектировании облачного шаблона.

### Создание группы входных свойств

1. Выберите **Проектирование > Группы свойств** и нажмите **Создать группу свойств**.
2. Выберите **Входные значения**.
3. Введите имя и описание новой группы свойств.

Имя	Имена групп свойств должны быть уникальными в пределах организации. Разрешено использовать только буквы, цифры и символы подчеркивания.
Отображаемое имя	Добавьте заголовок для группы свойств, который будет отображаться в форме запроса.
Описание	Поясните, для чего предназначен этот набор свойств.
Scope	<p>Укажите, может ли администратор предоставить общий доступ к группе свойств для всей организации. В противном случае к группе свойств будет возможен доступ только из одного проекта.</p> <p>Несмотря на то что свойства в группе всегда можно добавить или изменить, область является постоянной и не может быть позже изменена.</p>
Проект	Если для области указано ограничение «только для проекта», этот проект может получить доступ к группе свойств.

#### 4. Чтобы добавить свойство в группу, нажмите **Создать свойство**.

Панель для добавления нового свойства похожа на вкладку «Входные данные» в редакторе кода на странице проектирования Cloud Assembly.

Имя	Имя отдельного свойства в произвольном формате. Разрешено использовать только буквы, цифры и символы подчеркивания.
Отображаемое имя	Добавьте имя свойства, которое будет отображаться в форме запроса.
Тип	«Строка», «Целое число», «Число», «Логическое» значение (Истина или Ложь), «Объект» или «Массив».
Значение по умолчанию	<p>Предварительно заданное значение, которое отображается в форме запроса.</p> <p>Для всех типов, кроме логических, входные данные пользователя по умолчанию не являются обязательными. Чтобы проверить наличие всех входных данных, выполните одно из следующих действий.</p> <ul style="list-style-type: none"> <li>■ Задайте значение по умолчанию.</li> <li>■ Сделайте обязательными входные данные пользователя, добавив следующее свойство облачного шаблона в готовый код.</li> </ul> <pre>populateRequiredOnNonDefaultProperties: true</pre>
Зашифровано	Если выбран этот вариант, при вводе значения в форме запроса и в последующем развертывании оно маскируется. Зашифрованные свойства не могут принимать значение по умолчанию.



Только для чтения

Не подлежащее изменению, но видимое значение в форме запроса. Требуется значение по умолчанию.

Дополнительные параметры

Параметры, зависящие от типа свойства. Разверните раскрывающийся список, добавьте дополнительные параметры и нажмите **Создать**.

В следующем примере добавляемое свойство представляет образ операционной системы, а пользователь, отправляющий запрос, может выбрать одно из двух значений.

**Примечание** Операционные системы, показанные на рисунке в примере, должны входить в настроенную инфраструктуру Cloud Assembly.

**New Property**

**Name \*** image

**Display Name** Machine Image

**Description**

**Type** STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

**Default value** coreos

**Encrypted** ☐

**Read-only** ☐ ⓘ

▼ More Options

**Format** ▼

**Minimum length**

**Maximum length**

**Pattern**

**Pairs** ☒ Values ☐ Key Values

**Enum** Value

coreos -

ubuntu - +

- Добавьте в группу дополнительные свойства и нажмите кнопку **Сохранить** после завершения процедуры.

Properties 2 items

Add at least one property in order to create a property group

+ NEW PROPERTY

× DELETE

<input type="checkbox"/>	Name	Display Name	Type	Default Value
<input type="checkbox"/>	image	Machine Image	string	coreos
<input type="checkbox"/>	flavor	Machine Flavor	string	small

## Добавление группы свойств во входные данные облачного шаблона

Даже если есть длинный список входных данных свойств, чтобы указать все эти свойства в форме запроса, необходимо просто добавить группу свойств.

- На странице проектирования облачных шаблонов над областью редактирования, расположенной справа, перейдите на вкладку **Входные данные**.
- Нажмите **Создать входные данные облачного шаблона**.
- Введите имя и описание для группы свойств.

Имя	Введите значение, похожее на созданное ранее имя группы свойств.
Отображаемое имя	Введите заголовок, созданный ранее для всей группы свойств, который отображается в форме запроса.
Тип	Выберите <b>Объект</b> .
Тип объекта	Выберите <b>Группа свойств</b> .
Список групп свойств	Выберите нужную группу свойств. Отображаются только те группы свойств, которые были созданы и доступны для проекта. Следует отметить, что группы свойств констант не отображаются.

**New Cloud Template Input**

Name \*

Display Name

Description

Type

STRING INTEGER NUMBER BOOLEAN **OBJECT** ARRAY

Select Object Type ☐ Properties ☒ Property Groups

Select from the existing property groups

Search

Name	Description
<input checked="" type="radio"/> machine	

#### 4. Щелкните **Создать**.

Процесс создает код входных данных облачного шаблона, похожий на код в следующем примере.

```
inputs:
  pgmachine:
    type: object
    title: Machine Properties
    $ref: /ref/property-groups/machine
  pgrequester:
    type: object
    title: Requester Details
    $ref: /ref/property-groups/requesterDetails
```

Код также можно ввести непосредственно на странице проектирования Cloud Assembly, чтобы воспользоваться функцией автоматического запроса при вводе \$ref: /ref/p... в редакторе кода.

### Привязка ресурсов облачного шаблона к группе свойств

Чтобы использовать входные значения группы свойств, добавьте привязки для ресурса.

В зависимости от значений в группе свойств для них может потребоваться создание отдельной ссылки. Значения можно вводить по отдельности (по имени группы свойств и имени свойства).

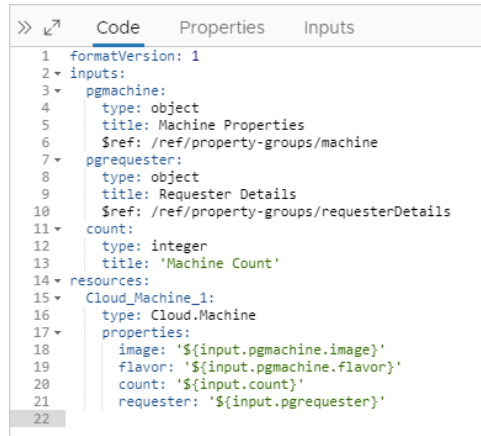
```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: '${input.pgmachine.image}'
      flavor: '${input.pgmachine.flavor}'
```

Кроме того, для быстрого добавления всего набора значений в ресурс можно использовать ссылку на саму группу свойств.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      requester: '${input.pgrequester}'
```

## Законченный код

После настройки входных данных и ресурсов готовый код будет выглядеть примерно так, как код в следующем примере.



```
>> Code Properties Inputs
1 formatVersion: 1
2 inputs:
3   pgmachine:
4     type: object
5     title: Machine Properties
6     $ref: /ref/property-groups/machine
7   pgrequester:
8     type: object
9     title: Requester Details
10    $ref: /ref/property-groups/requesterDetails
11 count:
12   type: integer
13   title: 'Machine Count'
14 resources:
15   Cloud_Machine_1:
16     type: Cloud.Machine
17     properties:
18       image: '${input.pgmachine.image}'
19       flavor: '${input.pgmachine.flavor}'
20       count: '${input.count}'
21       requester: '${input.pgrequester}'
22
```

При поступлении запроса на развертывание запрашивающему пользователю предлагается указать группы свойств.

### Deployment Inputs

Machine Properties

Machine Image

coreos

▼

Machine Flavor

small

▼

Requester Details

Email

Mobile

Internal account?

☐

PIN

Account Type

User

Machine Count \*

## Группы свойств в редакторе настраиваемых форм Service Broker

Группы входных свойств отображаются в интерфейсе настраиваемых форм Service Broker и доступны для настройки. Специальных рекомендаций по настройке групп свойств нет. Пользователям Service Broker даже не нужно знать, что источником записей является группа свойств, а не отдельно созданные свойства.

The screenshot shows the 'General' tab of a form in vRealize Automation Cloud Assembly. The form is set against a light gray grid background. At the top left, the 'General' tab is selected, and an 'ADD TAB' link is visible. The form contains several input fields and two grouped sections indicated by dashed orange borders. The fields are: 'Project' (with a dropdown arrow), 'Deployment Name', 'Machine Count', 'Machine Properties' (a group containing 'Machine Image' and 'Machine Flavor', both with dropdown arrows), 'Requester Details' (a group containing 'Email', 'Mobile', 'Internal account?' with a checkbox, 'PIN', and 'Account Type'), and 'Account Type'.

Дополнительные сведения см. в разделе [Настройка значка Service Broker и формы запроса](#).

## Действия vRealize Orchestrator в группе входных свойств

В группе входных свойств Cloud Assembly можно добавить динамическое взаимодействие с vRealize Orchestrator.

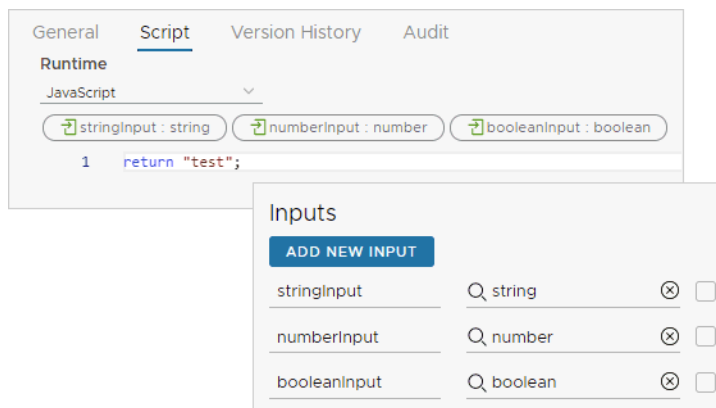
### Добавление действия vRealize Orchestrator в группу входных свойств

Выполните эти действия, чтобы добавить динамическое взаимодействие с vRealize Orchestrator в группу входных свойств.

1. В экземпляре vRealize Orchestrator, встроенном в vRealize Automation, создайте нужное действие.

Действие vRealize Orchestrator должно включать только простейшие типы: строку, целое число, число и логическое значение. Типы vRealize Orchestrator не поддерживаются.

В этом простом примере действие vRealize Orchestrator получает три экземпляра входных данных и возвращает жестко заданную строку.



2. В Cloud Assembly запустите процесс создания или изменения группы входных свойств. Если необходимо, изучите раздел [Группы входных свойств в Cloud Assembly](#).
3. Чтобы добавить входные данные действия vRealize Orchestrator в группу свойств, добавьте новые свойства, щелкните тип и нажмите **Константа**.

Добавьте входные данные действия vRealize Orchestrator отдельно.

4. После добавления входных данных добавьте новое свойство, щелкните тип, затем **Внешний источник** и нажмите **Выбрать**.

**New Property**

Name \* vroAction

Display Name VRO Action

Description

Type

STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

Default value source ☐ Constant ☒ External source

Action Add an existing action

SELECT

5. В разделе **Действие** найдите и выберите действие vRealize Orchestrator, которое создали, и щелкните **Сохранить**.

**Add an existing action**

Action \*

returnSimpleAction  
com.form.service.test

CANCEL SAVE

6. Сохраните группу свойств и добавьте ее в облачный шаблон. Если необходимо, изучите раздел [Группы входных свойств в Cloud Assembly](#).

При развертывании облачного шаблона группа свойств действия vRealize Orchestrator отображается в форме входных данных для запрашивающего пользователя.



Values for VRO

String for VRO

VRO Action test

Number for VRO

On-Off for VRO ☐

### Настраиваемые значения по умолчанию

Чтобы заполнить форму входных данных значениями по умолчанию, при добавлении действия vRealize Orchestrator качестве внешнего источника выполните одно из следующих действий.

- Вручную введите значение свойства по умолчанию.

Очистите поле **Привязка** и введите значение.

Add an existing action

Action \* com.form.service.test/Action

Action Parameters

Input Readme ☐ Bind

- Используйте другое значение свойства из той же группы свойств.

Нажмите **Привязка**, а затем в раскрывающемся меню выберите свойство.

Add an existing action

Action \* com.form.service.test/Action

Action Parameters

Input 

Canvas fields  
Hard String  
Input String

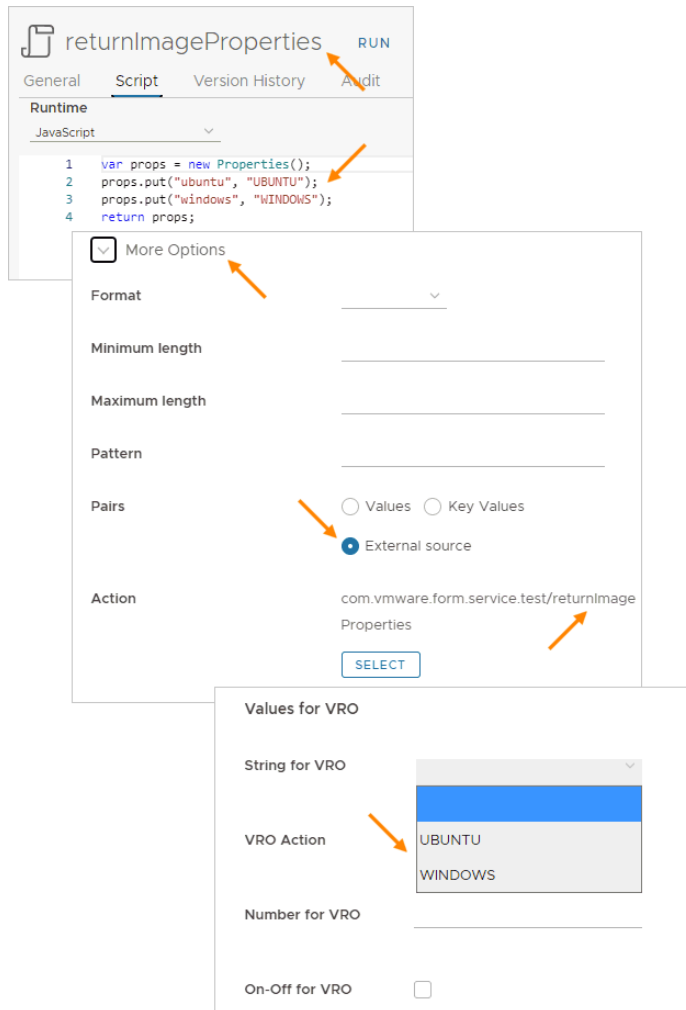
☒ Bind

## Выбор и добавление значений из перечня в vRealize Orchestrator

Чтобы создать в форме входных данных список для выбора из перечисленных значений на основе vRealize Orchestrator, при добавлении в группу свойств выполните следующие действия.

1. В vRealize Orchestrator создайте действие сопоставления значений, которые должны быть в списке.
2. В Cloud Assembly при добавлении свойства в группу откройте раздел **Дополнительные параметры**.
3. В разделе **Пары** щелкните **Внешний источник**, нажмите **Выбрать** и добавьте созданное действие vRealize Orchestrator.

**Примечание** Если при добавлении свойства также создается значение по умолчанию, это значение должно в точности совпадать с одним из перечисленных значений из действия vRealize Orchestrator.



## Группы свойств констант в Cloud Assembly

Константы Cloud Assembly позволяют автоматически применять к проектам известные пары «ключ — значение».

## Принцип работы констант

Ключ присутствует в коде облачного шаблона, а значение становится частью развертываний, которые основаны на этом облачном шаблоне. Для констант требуется привязка `propgroup` для ресурса.

Привязка `propgroup` используется только для групп свойств констант, но не для групп входных свойств.

## Свойства секретного ключа

Если планируется добавить свойство секретного ключа в группу свойств, создайте его, прежде чем продолжить. См. раздел [Секретные свойства Cloud Assembly](#).

## Создание группы свойств констант

1. Выберите **Проектирование > Группы свойств** и нажмите **Создать группу свойств**.
2. Выберите **Значения констант**.
3. Введите имя и описание новой группы свойств.

Имя	Имена групп свойств должны быть уникальными в пределах организации. Разрешено использовать только буквы, цифры и символы подчеркивания.
Отображаемое имя	Оставьте поле пустым. Заголовок в форме запроса не отображается.
Описание	Поясните, для чего предназначен этот набор констант.
Scope	<p>Укажите, может ли администратор предоставить общий доступ к группе свойств для всей организации. В противном случае к группе свойств будет возможен доступ только из одного проекта.</p> <p>Несмотря на то что свойства в группе всегда можно добавить или изменить, область является постоянной и не может быть позже изменена.</p> <p>Секретные ключи. Если планируется добавить секретное свойство в группу свойств, нужно использовать одну область проекта. Свойства секретных ключей сохраняются только на уровне проекта.</p>
Проект	Если для области указано ограничение «только для проекта», этот проект может получить доступ к группе свойств.

4. Чтобы добавить свойство константы в группу, нажмите **Создать свойство**.
5. Введите имя, которое будет выступать в качестве ключа, и описание.
6. Выберите тип свойства.
7. Введите необходимую константу и нажмите **Создать**.
  - Типы «строка», «целое число» и «число» вводятся напрямую.
  - Значение строки секретного ключа выбирается из списка свойств секретного ключа для проекта.
  - Для логического значения необходимо поставить флажок «истина».

- Для типов «Объект» или «Массив» замените `null` нужным значением.

**New Property**

Name \* `payerAccountNumber`

Description

Type

STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

Select Type ☒ Constant value ☐ Secret

Constant value `123456`

**New Property**

Name \* `payerAccountNumber`

Description

Type

STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

Select Type ☐ Constant value ☒ Secret

Search

Name	Description
<input checked="" type="radio"/> AccountNumber	
<input type="radio"/> password	
<input type="radio"/> RemoteAccessKey1	

7 secrets

- Добавьте в группу дополнительные константы и нажмите кнопку **Сохранить** после завершения процедуры.

**Properties** 3 items

Add at least one property in order to create a property group

+ NEW PROPERTY x DELETE

<input type="checkbox"/>	Name	Display Name	Type	Constant Value
<input type="checkbox"/>	<code>payerFederal</code>		boolean	true
<input type="checkbox"/>	<code>payerCostCenter</code>		integer	7890
<input type="checkbox"/>	<code>payerAccountNumber</code>		string	123456

## Привязка ресурсов облачного шаблона к группе свойств

Чтобы значения констант использовались в ресурсе без запроса, добавьте привязки `propgroup` к ресурсу.

Для быстрого добавления всего набора констант в ресурс, используйте ссылку на саму группу свойств.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      payerInfo: '${propgroup.payerDetails}'
```

Кроме того, отдельные константы можно добавить из группы свойств в выбранные части проекта.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      payerAccount: '${propgroup.payerDetails.payerAccountNumber}'
      payerCost: '${propgroup.payerDetails.payerCostCenter}'
      payerFed: '${propgroup.payerDetails.payerFederal}'
```

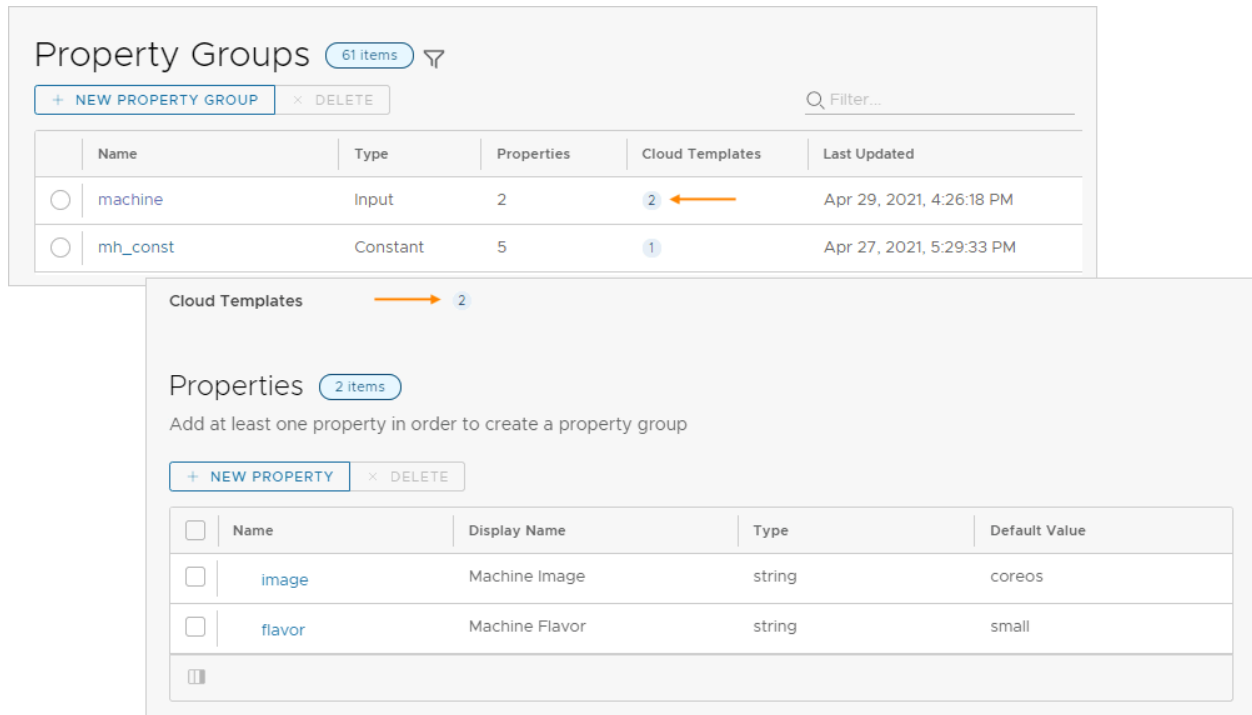
## Дополнительные сведения о группах свойств Cloud Assembly

Одну группу свойств Cloud Assembly можно включить в несколько облачных шаблонов. От этого зависит способ управления группами свойств.

### Изменение группы свойств

Изменения, внесенные в группу свойств Cloud Assembly, влияют на все облачные шаблоны, в которых она используется. Кроме того, после выпуска измененной версии облачного шаблона эти изменения будут относиться к пользователям каталога Service Broker.

На страницах списка групп свойств и изменения группы свойств указывается количество облачных шаблонов, содержащих группу свойств. Чтобы увидеть, на какой облачный шаблон будет влиять внесенное изменение, щелкните это численное значение.



Прежде чем изменить группу свойств, убедитесь, что изменение является приемлемым для всех пользователей, создающих или обновляющих развертывания на основе перечисленных облачных шаблонов.

## Удаление группы свойств

Удаление группы свойств может вызвать ошибки в каждом облачном шаблоне, в котором она используется.

Для полного удаления группы свойств необходимо вручную удалить ее из всех облачных шаблонов, которые ее содержат. Чтобы удалить группу свойств из облачного шаблона, откройте его на холсте проекта.

### ■ Группы входных свойств

На вкладке «Входные данные» выберите группу свойств и удалите ее. Для удаления связанной группы свойств в разделе `inputs` кода также можно использовать редактор кода.

### ■ Группы свойств констант

Для удаления связанной записи или записей `propgroup` в разделе `resources` кода используйте редактор кода.

**Примечание** Группу свойств нельзя удалить, если она включена в версированный облачный шаблон. Версированные облачные шаблоны доступны только для чтения.

## Флаги ресурса Cloud Assembly для запросов

В Cloud Assembly имеется несколько параметров облачных шаблонов, которые позволят настроить способ обработки ресурса во время запроса.

Настройки флага ресурса не являются частью схемы свойств объекта ресурса. Для заданного ресурса необходимо добавить параметры флага за пределами раздела свойств, как показано ниже.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    preventDelete: true
    properties:
      image: coreos
      flavor: small
      attachedDisks:
        - source: '${resource.Cloud_Volume_1.id}'
  Cloud_Volume_1:
    type: Cloud.Volume
    properties:
      capacityGb: 1
```

Флаг ресурса	Описание
allocatePerInstance	<p>Если для этого свойства установлено значение «истина», распределение ресурсов можно настроить для каждого компьютера в кластере. Если используется расширяемость, значение «истина» вызывает многократный запуск темы события расширяемости <code>compute.allocation.pre</code> при развертывании нескольких облачных компьютеров.</p> <p>Значение по умолчанию — «ложь». Если установлено это значение, в кластере выполняется равномерное распределение ресурсов, благодаря чему все компьютеры имеют одинаковую конфигурацию. Кроме того, действия по регулярному обслуживанию невозможно выполнять отдельно для каждого ресурса.</p> <p>Выделение для каждого экземпляра обеспечивает корректное применение конфигурации для отдельных компьютеров с помощью свойства <code>count.index</code>. Примеры кода см. в разделе <a href="#">Кластеры компьютеров и дисков в Cloud Assembly</a>.</p>
createBeforeDelete	<p>Для некоторых операций обновления требуется удалить существующий ресурс и создать новый. По умолчанию сначала выполняется удаление. При этом может возникнуть следующая ситуация: старый ресурс уже удален, а новый по какой-то причине еще не создан.</p> <p>Установите для этого флага значение <code>true</code>, если необходимо убедиться, что новый ресурс успешно создан, прежде чем удалить предыдущий.</p>
createTimeout	<p>Время ожидания в Cloud Assembly по умолчанию для выделения ресурсов, создания и планирования запросов равно 2 часам (2h). Кроме того, администратор проекта может установить произвольное время ожидания по умолчанию для таких запросов, применимое ко всему проекту.</p> <p>Этот флаг позволяет переопределять любые значения по умолчанию и устанавливать уникальное время ожидания для заданной операции с ресурсом. См. также <code>updateTimeout</code> и <code>deleteTimeout</code>.</p>

Флаг ресурса	Описание
deleteTimeout	<p>Время ожидания по умолчанию для запросов на удаление в Cloud Assembly равно 2 часам (2h). Кроме того, администратор проекта может установить произвольное время ожидания по умолчанию для запросов на удаление, применимое ко всему проекту.</p> <p>Этот флаг позволяет переопределять любые значения по умолчанию и устанавливать уникальное время ожидания для заданной операции удаления для ресурса. См. также updateTimeout и createTimeout.</p>
dependsOn	<p>Этот флаг определяет явную зависимость между ресурсами, при которой один ресурс уже должен существовать перед созданием следующего ресурса. Дополнительные сведения см. в разделе <a href="#">Создание привязок и зависимостей между ресурсами в Cloud Assembly</a>.</p>
dependsOnPreviousInstances	<p>Если для этого свойства установлено значение «истина», ресурсы кластера можно создавать последовательно. Значение по умолчанию — «ложь». Это позволяет одновременно создать все ресурсы в кластере.</p> <p>Например, последовательный процесс создания удобен для кластеров базы данных, где необходимо создать основной и вспомогательные узлы. Но для создания вспомогательного узла необходимы параметры конфигурации, которые подключают этот узел к существующему основному узлу.</p>
forceRecreate	<p>Не для всех операций обновления требуется удалить существующий ресурс и создать новый. Если нужно, чтобы операция обновления удалила старый ресурс и создала новый, независимо от того, задано ли это действие по умолчанию, установите для данного флага значение true.</p>
ignoreChanges	<p>Пользователи ресурса могут перенастроить его, изменив состояние ресурса после развертывания.</p> <p>Если необходимо выполнить обновление развертывания без переопределения измененного ресурса конфигурацией из облачного шаблона, установите для этого флага значение «истина».</p>
ignorePropertiesOnUpdate	<p>Пользователи ресурсов могут настраивать некоторые свойства, которые могут быть сброшены до исходного состояния облачного шаблона во время обновления.</p> <p>Для предотвращения сброса любых свойств во время обновления установите для этого флага значение true.</p>
preventDelete	<p>Если необходимо защитить созданный ресурс от случайного удаления во время обновлений, установите для этого флага значение true. Однако, если пользователь удалит развертывание, ресурс также будет удален.</p>



Флаг ресурса	Описание
recreatePropertiesOnUpdate	<p>Пользователи ресурса могут перенастроить свойства, изменив состояние ресурса после развертывания. Во время обновления ресурс может (или не может) быть создан повторно. Ресурсы, которые не создаются повторно, могут сохранить измененные состояния свойств.</p> <p>Если необходимо повторно создать все свойства, независимо от того, задано ли это действие по умолчанию при обновлении, установите для данного флага значение «истина».</p>
updateTimeout	<p>Время ожидания по умолчанию для запросов на обновление в Cloud Assembly равно 2 часам (2h). Кроме того, администратор проекта может установить произвольное время ожидания по умолчанию для запросов на обновление, применимое ко всему проекту.</p> <p>Этот флаг позволяет переопределять любые значения по умолчанию и устанавливать уникальное время ожидания для заданной операции обновления для ресурса. См. также <code>deleteTimeout</code> и <code>createTimeout</code>.</p>

## Выражения Cloud Assembly

Чтобы повысить гибкость кода облачных шаблонов в Cloud Assembly, в него можно добавлять выражения.

### Принципы работы выражений

В выражениях Cloud Assembly используется компонент `${expression}`, как показано в следующих примерах.

**Примечание** Выражения Cloud Assembly отличаются от регулярных выражений. См. [Синтаксис выражений Cloud Assembly](#) для Cloud Assembly.

Следующие примеры кода упрощены и содержат только важные строки. Полный вариант облачного шаблона приведен в конце.

### Примеры

Во время развертывания разрешите пользователю вставить зашифрованный ключ, необходимый для удаленного доступа:

```
inputs:
  sshKey:
    type: string
    maxLength: 500
resources:
  frontend:
    type: Cloud.Machine
```

```
properties:
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: '${input.sshKey}'
```

При разворачивании в среде VMware Cloud on AWS замените имя папки на *Workload* (обязательное значение):

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
  resources:
    frontend:
      type: Cloud.Machine
      properties:
        folderName: '${input.environment == "VMC" ? "Workload" : ""}'
```

В процессе разворачивания добавьте к машине тег *env* (все буквы строчные), соответствующий выбранной среде:

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
  resources:
    frontend:
      type: Cloud.Machine
      properties:
        constraints:
          - tag: '${"env:" + to_lower(input.environment)}'
```

Задайте количество машин в клиентском кластере: 1 — малый (small), 2 — большой (large). Обратите внимание, что большой кластер задается путем исключения:

```
inputs:
  envsize:
    type: string
    enum:
      - Small
      - Large
  resources:
```

```
frontend:
  type: Cloud.Machine
  properties:
    count: '${input.envsize == "Small" ? 1 : 2}'
```

Присоедините машины к одной и той же сети *По умолчанию (Default)* путем привязки к свойству, найденному в ресурсе сети:

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
      networks:
        - network: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - network: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing
```

Зашифруйте учетные данные доступа, отправленные в API:

```
resources:
  apitier:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        #cloud-config
        runcmd:
          - export apikey=${base64_encode(input.username:input.password)}
          - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com
```

Узнайте адрес машины API:

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        runcmd:
          - echo ${resource.apitier.networks[0].address}
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - network: '${resource.Cloud_Network_1.name}'
```

## Полный облачный шаблон

```

inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
  sshKey:
    type: string
    maxLength: 500
  envsize:
    type: string
    enum:
      - Small
      - Large
resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu
      flavor: medium
      count: '${input.envsize == "Small" ? 1 : 2}'
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'
      cloudConfig: |
        packages:
          - nginx
        runcmd:
          - echo ${resource.apitier.networks[0].address}
      constraints:
        - tag: '${"env:" + to_lower(input.environment)}'
      networks:
        - network: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu
      flavor: small
      cloudConfig: |
        #cloud-config
        runcmd:
          - export apikey=${base64_encode(input.username:input.password)}
          - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com
      remoteAccess:
        authentication: publicPrivateKey

```

```

      sshKey: '${input.sshKey}'
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'
    networks:
      - network: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'

```

## Синтаксис выражений Cloud Assembly

Синтаксис выражений — это совокупность всех возможностей работы с выражениями в шаблонах Cloud Assembly.

---

**Примечание** Выражения Cloud Assembly отличаются от регулярных выражений.

---

Представленный ниже синтаксис частично представлен в примерах, приведенных в разделе [Выражения Cloud Assembly](#).

### Литералы

Поддерживаются следующие литералы.

- Логическое значение: истина или ложь
- Целое число
- Число с плавающей запятой
- Строка

Обратная косая черта используется в качестве escape-символа для одинарных и двойных кавычек, а также для самой себя.

Escape-последовательность для " выглядит как \"

Escape-последовательность для ' выглядит как \'

Escape-последовательность для \ выглядит как \\

Escape-последовательность для кавычек нужна только в строках, заключенных в кавычки того же типа, как показано в следующем примере.

```
"I am a \"double quoted\" string inside \"double quotes\"."
```

- NULL

## Переменные сред

Имена сред:

- orgId
- projectId
- projectName
- deploymentId
- deploymentName
- blueprintId
- blueprintVersion
- blueprintName
- requestedBy (user)
- requestedAt (time)

Синтаксис

```
env.ENV_NAME
```

Пример

```
${env.blueprintId}
```

## Переменные ресурсов

Переменные ресурсов позволяют выполнять привязку к свойствам ресурсов из других ресурсов.

Синтаксис

```
resource.RESOURCE_NAME.PROPERTY_NAME
```

Имена ресурсов не могут содержать тире и точки. Символы подчеркивания разрешены.

Примеры:

- \${resource.db.id}
- \${resource.db.networks[0].address}
- \${resource.app.id} (Возвращает строковое значение для некластерных ресурсов, если значение количества не указано; возвращает массив для кластерных ресурсов.)
- \${resource.app[0].id} (возвращает первую запись для кластерных ресурсов)

## Переменные ресурсов self

Переменные ресурсов self разрешены только для ресурсов, которые поддерживают стадию выделения.

Переменные ресурсов self доступны только (или имеют только заданное значение) по окончании стадии выделения.

Синтаксис

```
self.property_name
```

### Пример

```
${self.address} (возвращает адрес, назначенный на стадии выделения)
```

Следует отметить, что для ресурса с именем `resource_x` `self.property_name` и `resource.resource_x.property_name` являются одинаковыми и считаются рекурсивными ссылками.

## Условия

### Синтаксис

- В качестве операторов равенства используются `==` и `!=`.
- В качестве реляционных операторов используются `<`, `>`, `<=` и `>=`.
- В качестве логических операторов используются `&&`, `||` и `!`.
- В условиях используется следующий шаблон:

*выражение условия ? выражение истины :выражение лжи*

### Примеры:

```
${input.count < 5 && input.size == 'small'}
```

```
${input.count < 2 ? "small":"large"}
```

## Индекс счетчика кластеров

### Синтаксис

```
count.index
```

### Примеры:

- Возвращает тип узла для кластерных ресурсов:  

```
${count.index == 0 ? "primary":"secondary"}
```
- Устанавливает размер каждого диска во время выделения:

```
inputs:
  disks:
    type: array
    minItems: 0
    maxItems: 12
    items:
      type: object
      properties:
        size:
          type: integer
          title: Size (GB)
          minSize: 1
          maxSize: 2048
resources:
  Cloud_vSphere_Disk_1:
```

```

type: Cloud.vSphere.Disk
allocatePerInstance: true
properties:
  capacityGb: '${input.disks[count.index].size}'
  count: '${length(input.disks)}'

```

- Дополнительные примеры см. в разделе [Кластеры компьютеров и дисков в Cloud Assembly](#).

## Арифметические операторы

### Синтаксис

В качестве операторов используются +, -, /, \* и %.

### Пример

```

${(input.count + 5) * 2}

```

## Объединение строк

### Синтаксис

`${'ABC' + 'DEF'}` эквивалентно ABCDEF.

## Операторы «[ ]» и «.»

Обработка операторов «[ ]» и «.» в выражениях унифицирована в соответствии с ECMAScript.

Таким образом, `expr.identifier` будет эквивалентно `expr["identifier"]`. Идентификатор используется для создания литерала, значение которого равняется идентификатору, а затем оператор «[ ]» используется с этим значением.

### Пример

```

${resource.app.networks[0].address}

```

Кроме того, если свойство содержит пробелы, в качестве разделителей используйте квадратные скобки и двойные кавычки вместо нотации с точкой.

Неправильно:

```

input.operating system

```

Правильно:

```

input["operating system"]

```

## Создание карты

### Синтаксис

```

${{'key1':'value1', 'key2':input.key2}}

```

## Создание массива

### Синтаксис



```
${['key1','key2']}
```

### Пример

```
${[1,2,3]}
```

## Функции

### Синтаксис

```
${function(arguments...)}
```

### Пример

```
${to_lower(resource.app.name)}
```

Таблица 6-1. Функции

Функция	Описание
<code>abs(number)</code>	Абсолютное значение числа
<code>avg(array)</code>	Возвращает среднее от всех значений из массива чисел
<code>base64_decode(string)</code>	Возвращает декодированное значение base64
<code>base64_encode(string)</code>	Возвращает значение в кодировке base64
<code>ceil(number)</code>	Возвращает наименьшее значение (ближайшее к отрицательной бесконечности), которое больше или равно аргументу и равно математическому целому значению
<code>contains(array, value)</code>	Проверяет массив на наличие значения
<code>contains(string, value)</code>	Проверяет строку на наличие значения
<code>digest(value, type)</code>	Возвращает выборку по значению с использованием поддерживаемого типа (md5, sha1, sha256, sha384, sha512)
<code>ends_with(subject, suffix)</code>	Проверяет, заканчивается ли строка темы (subject) строкой суффикса (suffix)
<code>filter_by(array, filter)</code>	Возвращает только записи массива, для которых успешно выполняется операция фильтрации <code>filter_by([1,2,3,4], x =&gt; x &gt;= 2 &amp;&amp; x &lt;= 3)</code> возвращает <code>[2, 3]</code> <code>filter_by({'key1':1, 'key2':2}, (k,v) =&gt; v != 1)</code> возвращает <code>[{"key2": 2}]</code>
<code>floor(number)</code>	Возвращает наибольшее значение (ближайшее к положительной бесконечности), которое меньше аргумента или равно ему и равно математическому целому значению
<code>format(format, values...)</code>	Возвращает отформатированную строку с использованием формата и значений <a href="#">форматировщика класса Java</a> .
<code>from_json(string)</code>	Разбирает строку json
<code>join(array, delim)</code>	Присоединяет массив строк с разделителем и возвращает строку
<code>json_path(value, path)</code>	Сравнивает путь со значением с использованием <a href="#">XPath for JSON</a> .
<code>keys(map)</code>	Возвращает ключи сопоставления

Таблица 6-1. Функции (продолжение)

Функция	Описание
<code>length(array)</code>	Возвращает длину массива
<code>length(string)</code>	Возвращает длину строки
<code>map_by(array, operation)</code>	<p>Возвращает каждую запись массива с примененной для ней операцией</p> <pre>map_by([1,2], x =&gt; x * 10)</pre> <p>возвращает [10, 20]</p> <pre>map_by([1,2], x =&gt; to_string(x))</pre> <p>возвращает ["1", "2"]</p> <pre>map_by({'key1':1, 'key2':2}, (k,v) =&gt; {k:v*10})</pre> <p>возвращает [{"key1":10}, {"key2":20}]</p>
<code>map_to_object(array, keyname)</code>	<p>Возвращает массив пар «ключ:значение» для указанного имени ключа в сочетании со значениями из другого массива</p> <pre>map_to_object(resource.Disk[*].id, "source")</pre> <p>возвращает массив пар «ключ:значение», содержащий поле ключа, которое называется «source», вместе со строками идентификатора диска.</p> <p>Обратите внимание:</p> <pre>map_by(resource.Disk[*].id, id =&gt; {'source':id})</pre> <p>возвращает тот же результат</p>
<code>matches(string, regex)</code>	Проверяет, совпадает ли строка с регулярным выражением
<code>max(array)</code>	Возвращает максимальное значение из массива чисел
<code>merge(map, map)</code>	Возвращает объединенной сопоставление
<code>min(array)</code>	Возвращает минимальное значение из массива чисел
<code>not_null(array)</code>	Возвращает первую запись, которая не является пустой (null)
<code>now()</code>	Возвращает текущее время в формате ISO-8601
<code>range(start, stop)</code>	Возвращает последовательность чисел с шагом 1, которая начинается с начального числа и заканчивается непосредственно перед конечным числом
<code>replace(string, target, replacement)</code>	Заменяет строку, содержащую целевой элемент <code>string</code> , строкой <code>target</code>
<code>reverse(array)</code>	Обращает записи в массиве
<code>slice(array, begin, end)</code>	Возвращает фрагмент массива с начального ( <code>begin</code> ) до конечного ( <code>end</code> ) индекса
<code>split(string, delim)</code>	Разбивает строку с помощью разделителя и возвращает массив строк
<code>starts_with(subject, prefix)</code>	Проверяет, начинается ли строка темы ( <code>subject</code> ) со строки префикса ( <code>prefix</code> )
<code>substring(string, begin, end)</code>	Возвращает подстроку строки с начального ( <code>begin</code> ) до конечного ( <code>end</code> ) индекса
<code>sum(array)</code>	Возвращает сумму всех значений из массива чисел
<code>to_json(value)</code>	Сериализирует значение как строку json

Таблица 6-1. Функции (продолжение)

Функция	Описание
to_lower(str)	Переводит все символы в строке в нижний регистр
to_number(string)	Разбирает строку как число
to_string(value)	Возвращает строковое представление значения
to_upper(str)	Переводит все символы в строке в верхний регистр
trim(string)	Удаляет начальные и конечные пробелы
url_encode(string)	Выполняет кодировку строки с использованием спецификации кодировки url
uuid()	Возвращает случайно сгенерированный универсальный уникальный идентификатор
values(map)	Возвращает значения сопоставления

## Устранение неполадок

В языке YAML двоеточие и пробел («: ») используются в качестве разделителя между ключом и значением в парах «ключ-значение». Синтаксис выражений зависит от YAML, поэтому использование в выражении пробела после двоеточия иногда приводит к сбоям.

Например, в следующем выражении наличие пробела между "win" : и "lin" приводит к сбою.

```
${contains(input.image,"(Windows)") == true ? "win" : "lin"}
```

В работающем выражении пропущен пробел.

```
${contains(input.image,"(Windows)") == true ? "win" : "lin"}
```

Если обработка выражения по-прежнему завершается сбоем, попробуйте заключить всё выражение в одинарные кавычки, как показано ниже.

```
ezOS: '${contains(input.image,"(Windows)") == true ? "win" : "lin"}'
```

## Секретные свойства Cloud Assembly

Секретное свойство Cloud Assembly — это многократно используемое зашифрованное значение, которое пользователи проектов могут добавлять в свои облачные шаблоны.

Типичными примерами секретных свойств являются ключи и учетные данные для безопасного доступа. После создания и сохранения значение секретного свойства будет нельзя расшифровать или прочитать.

## Создание секретного свойства

1. Войдите в Cloud Assembly, используя права роли администратора проекта.
2. Выберите **Инфраструктура > Администрирование > Секретные элементы** и нажмите **Создать секретный элемент**.
3. Выберите проект.

- Введите уникальное имя свойства для секретного элемента без пробелов и специальных символов.

Имя является видимым идентификатором для секретного элемента.

- Введите секретное значение.

При вводе значение оно по умолчанию всегда скрыто, что позволяет обеспечить защиту, если экран доступен нескольким пользователям.

При необходимости можно щелкнуть значок глаза, чтобы показать и проверить значение. После сохранения секретное значение шифруется в базе данных и снова увидеть его будет нельзя.

- При желании можно ввести более подробное описание секретного свойства.

- Щелкните **Создать**.

## Добавление секретного свойства в облачный шаблон

Пользователи проектов могут добавить секретное свойство в качестве привязки в код облачного шаблона.

Следует отметить, что при наборе символов `'${secret.` появится список секретных элементов выбора, созданных для проекта.

```
type: Cloud.Machine
properties:
  name: ourvm
  image: mint20
  flavor: small
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: '${secret.ourPublicKey}'
    username: root
```

Сведения о добавлении секретного свойства в конфигурацию Terraform см. в [Использование секретного свойства Cloud Assembly в конфигурации Terraform](#).

## Удаленный доступ к разворачиванию Cloud Assembly

Для организации удаленного доступа к компьютеру, который был развернут системой Cloud Assembly, перед разворачиванием необходимо добавить в облачный шаблон свойства для этого компьютера.

Для удаленного доступа можно настроить один из следующих параметров проверки подлинности.

**Примечание** В тех случаях, когда нужно скопировать ключи, в облачном шаблоне также можно создать раздел `cloudConfig`, чтобы автоматически копировать ключи во время подготовки. Более подробные сведения здесь не приводятся, но в разделе [Инициализация компьютера в Cloud Assembly](#) можно найти общую информацию о `cloudConfig`.

## Создание пары ключей во время подготовки

Если для проверки подлинности в режиме удаленного доступа нет собственной пары открытого и закрытого ключей, Cloud Assembly может создать такую пару ключей.

В качестве примера используйте следующий код.

1. В Cloud Assembly перед подготовкой добавьте свойства `remoteAccess` в облачный шаблон, как показано в примере.

Имя пользователя не является обязательным. Если его опустить, система создаст случайный идентификатор в качестве имени пользователя.

Пример

```
type: Cloud.Machine
properties:
  name: our-vm2
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: generatedPublicPrivateKey
    username: testuser
```

2. В Cloud Assembly подготовьте компьютер на основе его облачного шаблона и переведите его в состояние «Запущено».

Процесс подготовки создает ключи.

3. Найдите имя ключа в разделе свойств **Ресурсы > Развертывания > Топология**.
4. Для доступа к командной строке предоставленного компьютера используйте интерфейс поставщика облачных служб, например клиент vSphere.
5. Предоставьте закрытому ключу разрешение на чтение.

```
chmod 600 key-name
```

6. Перейдите в развертывание Cloud Assembly, выберите компьютер, а затем **Действия > Получить закрытый ключ**.

7. Скопируйте файл закрытого ключа на локальный компьютер.

Стандартный путь к локальному файлу — `/home/username/.ssh/key-name`.

8. Откройте удаленный сеанс SSH и подключитесь к подготовленному компьютеру.

```
ssh -i key-name user-name@machine-ip
```

## Указание собственной пары открытого и закрытого ключей

Многие компании для проверки подлинности создают и распространяют собственные пары открытого и закрытого ключей.

В качестве примера используйте следующий код.

1. В локальной среде получите или создайте пару открытого и закрытого ключей.

А сейчас просто создайте ключи и сохраните их локально.

2. В Cloud Assembly перед подготовкой добавьте свойства `remoteAccess` в облачный шаблон, как показано в примере.

`sshKey` содержит длинную алфавитно-цифровую строку, которая находится в файле открытого ключа `key-name.pub`.

Имя пользователя не является обязательным и создается для входа в систему. Если его опустить, система создаст случайный идентификатор в качестве имени пользователя.

### Пример

```
type: Cloud.Machine
properties:
  name: our-vm1
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: ssh-rsa Iq+5aQgBP3ZNT4o1baP5Ii+dstIcowRRkyobbfpA1mj9tslf
    qGxvU66PX9IeZax5hZvNWFgJw6ag+Z1zndOLhVdVoW49f274/mIRild7Uuw...
    username: testuser
```

3. В Cloud Assembly подготовьте компьютер на основе его облачного шаблона и переведите его в состояние «Запущено».
4. Используя клиент поставщика облачных сред, получите доступ к подготовленному компьютеру.
5. Добавьте файл открытого ключа в домашнюю папку на компьютере. Используйте ключ, указанный в `remoteAccess.sshKey`.
6. Убедитесь, что на вашем локальном компьютере есть соответствующий файл закрытого ключа.

Обычно это `/home/username/.ssh/key-name` без расширения. `pub`.

7. Откройте удаленный сеанс SSH и подключитесь к подготовленному компьютеру.

```
ssh -i key-name user-name@machine-ip
```

## Указание пары ключей AWS

Если добавить имя пары ключей AWS в облачный шаблон, можно получить удаленный доступ к компьютеру, который Cloud Assembly развертывает в AWS.

Следует иметь в виду, что пары ключей AWS относятся к конкретному региону. Если рабочие нагрузки подготавливаются для `us-east-1`, то пара ключей должна существовать в этой области.

В качестве примера используйте следующий код. Этот параметр работает только для облачных зон AWS.

```
type: Cloud.Machine
properties:
  image: Ubuntu
  flavor: small
  remoteAccess:
    authentication: keyPairName
    keyPair: cas-test
constraints:
  - tag: 'cloud:aws'
```

## Ввод имени пользователя и пароля

Если добавить имя пользователя и пароль в облачный шаблон, можно организовать простой удаленный доступ к компьютеру, который развертывает служба Cloud Assembly.

Несмотря на то что такой вариант является менее защищенным, удаленный вход в систему с помощью имени пользователя и пароля может быть достаточным для такого случая. Следует отметить, что некоторые поставщики или конфигурации облачных сред могут не поддерживать этот менее защищенный вариант.

1. В Cloud Assembly перед подготовкой добавьте свойства `remoteAccess` в облачный шаблон, как показано в примере.

Задайте имя пользователя и пароль в учетной записи, по которой будет выполняться вход.

Пример

```
type: Cloud.Machine
properties:
  name: our-vm3
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: usernamePassword
    username: testuser
    password: admin123
```

2. В Cloud Assembly подготовьте компьютер на основе его облачного шаблона и переведите его в состояние «Запущено».
3. Перейдите в интерфейс поставщика облачных сред и получите доступ к подготовленному компьютеру.
4. Создайте учетную запись на подготовленном компьютере или включите ее.
5. На локальном компьютере откройте удаленный сеанс по IP-адресу или полному доменному имени (FQDN) подготовленного компьютера, а затем войдите в систему обычным образом, используя имя пользователя и пароль.

## Размещение диска SCSI с помощью Cloud Assembly

Для управления диском SCSI необходимо указать и запомнить его SCSI-контроллер и логический том (LUN). Для жесткого диска vSphere можно использовать Cloud Assembly для назначения обоих значений в облачном шаблоне.

Возможность использования различных SCSI-контроллеров имеет большое значение для производительности и требуется для некоторых типов развертываний, таких как Oracle Real Application Clusters (RAC).

### Свойства диска: SCSI-контроллер и LUN

Чтобы назначить контроллер SCSI и LUN, добавьте следующие свойства облачного шаблона:

```
SCSIController
```

```
unitNumber
```

Можно не использовать эти свойства. В этом случае назначаются элементы по умолчанию. Cloud Assembly больше не развертывает диски SCSI произвольным образом, так как это затрудняет управление ими.

Контроллеры и диски SCSI нумеруются по порядку, начиная с 0. Каждый SCSI-контроллер может поддерживать диски SCSI с номерами томов 0–15.

### Вариант 1. Указание контроллера SCSI и тома

Можно полностью задать оба свойства, как показано в следующем примере. В таком случае назначаемые контроллер SCSI и том совпадают с вводимыми значениями.

```
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      cpuCount: 1
      totalMemoryMB: 1024
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
        - source: '${resource.Cloud_vSphere_Disk_2.id}'
        - source: '${resource.Cloud_vSphere_Disk_3.id}'
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
      SCSIController: SCSI_Controller_2
      unitNumber: 0
  Cloud_vSphere_Disk_2:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
      SCSIController: SCSI_Controller_2
      unitNumber: 1
```



```
Cloud_vSphere_Disk_3:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1
    SCSIController: SCSI_Controller_3
    unitNumber: 4
```

## Вариант 2. Настройка только SCSI-контроллера

Можно указать SCSI-контроллер и не указывать номер тома. В этом случае назначаемый SCSI-контроллер совпадает с вводимым значением. В качестве номера тома устанавливается первый доступный номер тома для этого контроллера.

```
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      cpuCount: 1
      totalMemoryMB: 1024
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
        - source: '${resource.Cloud_vSphere_Disk_2.id}'
        - source: '${resource.Cloud_vSphere_Disk_3.id}'
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
      SCSIController: SCSI_Controller_0
  Cloud_vSphere_Disk_2:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
      SCSIController: SCSI_Controller_0
  Cloud_vSphere_Disk_3:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
      SCSIController: SCSI_Controller_1
```

## Вариант 3. Пропуск обоих свойств

Можно не указывать SCSI-контроллер и номер тома. В этом случае назначаются первый доступный SCSI-контроллер и первый доступный номер тома для этого контроллера.

```
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      cpuCount: 1
      totalMemoryMB: 1024
      attachedDisks:
```

```

- source: '${resource.Cloud_vSphere_Disk_1.id}'
- source: '${resource.Cloud_vSphere_Disk_2.id}'
- source: '${resource.Cloud_vSphere_Disk_3.id}'
Cloud_vSphere_Disk_1:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1
Cloud_vSphere_Disk_2:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1
Cloud_vSphere_Disk_3:
  type: Cloud.vSphere.Disk
  properties:
    capacityGb: 1

```

## Невозможный вариант. Только LUN

Нельзя указать только номер тома, не указав SCSI-контроллер. В этом случае может появиться развертывание, в котором несколько SCSI-контроллеров имеют диск с одинаковым номером, но при управлении нельзя будет узнать, какой диск к чему относится.

## Использование входных данных для настройки SCSI-контроллера и LUN

Чтобы сделать проект более динамическим, используйте входные данные. Тогда пользователь сможет указать SCSI-контроллер и номер тома во время запроса или обновления.

```

inputs:
  diskProperties:
    type: array
    minItems: 1
    maxItems: 10
    items:
      type: object
      properties:
        size:
          type: integer
        SCSIController:
          type: string
          title: SCSI Controller
          enum:
            - SCSI_Controller_0
            - SCSI_Controller_1
            - SCSI_Controller_2
            - SCSI_Controller_3
        unitNumber:
          type: integer
          title: Unit Number

resources:
  app:
    type: Cloud.vSphere.Machine

```

```

allocatePerInstance: true
properties:
  flavor: small
  image: centos
  attachedDisks: '${map_to_object(slice(resource.disk[*].id, 0, 4), 'source')}'
disk:
  type: Cloud.vSphere.Disk
  allocatePerInstance: true
  properties:
    capacityGb: '${input.diskProperties[count.index].size}'
    SCSIController: '${input.diskProperties[count.index].SCSIController}'
    unitNumber: '${input.diskProperties[count.index].unitNumber}'
    count: ${length(input.diskProperties)}

```

size	1
SCSI Controller	SCSI_Controller_0
Unit Number	2

CANCEL APPLY

## Инициализация компьютера в Cloud Assembly

Инициализацию компьютеров в Cloud Assembly можно выполнять непосредственно с помощью команд или (в случае развертывания в облачных зонах на основе vSphere) с помощью спецификаций настройки.

### Принципы работы команд и спецификаций настройки

#### ■ Команды

Раздел `cloudConfig` в коде облачного шаблона содержит команды, которые можно запускать.

#### ■ Спецификации настройки

Одно из свойств в коде облачного шаблона ссылается на спецификацию настройки vSphere по имени.

### Команды и спецификации настройки не всегда можно использовать совместно

При развертывании в vSphere в случае использования `cloudConfig` и одновременной инициализации спецификаций настройки необходимо соблюдать осторожность. Их совместимость не подтверждена формально, и их совместное использование может привести к непредвиденным или нежелательным результатам.

Пример взаимодействия команд и спецификаций настройки см. в [Статические IP-адреса vSphere в Cloud Assembly](#).

## Спецификации настройки vSphere в шаблонах Cloud Assembly

При развертывании в облачных зонах на основе vSphere в Cloud Assembly во время развертывания спецификации настройки могут применять параметры гостевой операционной системы.

### Включение спецификации настройки

Спецификация настройки должна существовать в vSphere в целевом расположении, где выполняется развертывание.

Отредактируйте напрямую код облачного шаблона. Следующий пример указывает на спецификацию настройки `cloud-assembly-linux` для узла WordPress в среде vSphere.

```
resources:
  WebTier:
    type: Cloud.vSphere.Machine
    properties:
      name: wordpress
      cpuCount: 2
      totalMemoryMB: 1024
      imageRef: 'Template: ubuntu-18.04'
      customizationSpec: 'cloud-assembly-linux'
      folderName: '/Datacenters/Datacenter/vm/deployments'
```

### Что нужно использовать — спецификации настройки или команды cloudConfig

Если нужно, чтобы процесс предоставления ресурсов соответствовал операциям, выполняемым в данный момент в среде vSphere, то лучше будет продолжать использование спецификаций настройки. Однако для реализации гибридной подготовки ресурсов и подготовки ресурсов в нескольких облачных системах больше подойдет настройка с использованием команд инициализации cloudConfig.

Дополнительные сведения о разделах cloudConfig в облачных шаблонах см. в разделе [Команды настройки в шаблонах Cloud Assembly](#).

### Команды и спецификации настройки не всегда можно использовать совместно

При развертывании в vSphere в случае использования встроенной команды cloudConfig и одновременной инициализации спецификаций настройки необходимо соблюдать осторожность. Их совместимость не подтверждена формально, и их совместное использование может привести к непредвиденным или нежелательным результатам.

Пример взаимодействия команд и спецификаций настройки см. в [Статические IP-адреса vSphere в Cloud Assembly](#).

## Команды настройки в шаблонах Cloud Assembly

В код шаблона Cloud Assembly можно добавить раздел cloudConfig, который будет содержать команды инициализации компьютера, выполняемые во время развертывания.

## Форматы команд cloudConfig

- В ОС Linux команды инициализации соответствуют открытому стандарту [cloud-init](#).
- В ОС Windows для поддержки команд инициализации используется служба [Cloudbase-init](#).

В командах [cloud-init](#) (Linux) и [Cloudbase-init](#) (Windows) используется разный синтаксис. Раздел cloudConfig для одной операционной системы не будет работать в образе компьютера другой операционной системы.

## Функции команд cloudConfig

Команды инициализации используются для автоматизации применения данных или параметров во время создания экземпляра. С их помощью можно настраивать пользователей, разрешения, устанавливаемые решения и любые другие операции на основе выполнения команд. Примеры

- Настройка имени узла
- Создание и настройка закрытых ключей SSH
- Установка пакетов

## Куда можно добавлять команды cloudConfig

Раздел cloudConfig можно добавить в код облачного шаблона, но при настройке инфраструктуры данный раздел также можно заранее добавить в образ компьютера. В этом случае выполняется одинаковая инициализация всех облачных шаблонов, ссылающихся на исходный образ.

Команды инициализации могут содержаться как в облачном шаблоне, так и в сопоставлении образа. Во время развертывания происходит объединение команд, и в службе Cloud Assembly запускаются консолидированные команды.

Если команда присутствует и в схеме элементов, и в сопоставлении образа, но ее параметры различаются, то выполняется только команда из сопоставления образа.

Дополнительные сведения см. в разделе [Дополнительные сведения о сопоставлении образов в vRealize Automation](#).

## Примеры команд cloudConfig

Следующий пример раздела cloudConfig взят из кода облачного шаблона для сервера MySQL на базе Linux (этот код используется для [Создание базового облачного шаблона](#)).

---

**Примечание** Чтобы обеспечить правильную интерпретацию команд, всегда указывайте символ канала cloudConfig: |, как показано ниже.

---

```
cloudConfig: |
  #cloud-config
  repo_update: true
  repo_upgrade: all
  packages:
    - apache2
    - php
```

```

- php-mysql
- libapache2-mod-php
- php-mcrypt
- mysql-client
runcmd:
- mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget
https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
- i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
- mysql -u root -pmysqlpassword -h ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
- mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
- sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER',
'root' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i
-e s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD',
'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '$
{DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-config.php
- service apache2 reload

```

Если сценарий cloud-init имеет непредвиденное поведение, проверьте выходные данные на консоли в /var/log/cloud-init-output.log в ходе устранения неполадок. Дополнительные сведения о cloud-init см. в [документации по cloud-init](#).

## Команды и спецификации настройки не всегда можно использовать совместно

При развертывании в vSphere в случае использования встроенной команды cloudConfig и одновременной инициализации спецификаций настройки необходимо соблюдать осторожность. Их совместимость не подтверждена формально, и их совместное использование может привести к непредвиденным или нежелательным результатам.

Пример взаимодействия команд и спецификаций настройки см. в [Статические IP-адреса vSphere в Cloud Assembly](#).

## Шаблоны vSphere для инициализации в Cloud Assembly

Если шаблон Cloud Assembly обеспечивает развертывание образа на основе шаблона vSphere, необходимо заранее настроить шаблон vSphere для поддержки cloud-init.

Чтобы настроить vSphere для поддержки cloud-init, выполните следующие действия.

1. Установите cloud-init на виртуальную машину, которая станет шаблоном.

Например, используйте yum для установки cloud-init на CentOS или apt-get для установки на Ubuntu.

2. Переведите CD-ROM виртуальной машины в транзитный режим.

CD/DVD drive 1 *	Client Device ▾
Status	<input type="checkbox"/> Connect At Power On
CD/DVD Media	To connect, power on the VM and select the media from the VM Hardware panel on Summary tab
Device Mode	→ Passthrough CD-ROM ▾

- Запустите `cloud-init clean` из командной строки гостевой операционной системы.

---

**Примечание** По завершении работы `cloud-init clean` больше не изменяйте виртуальную машину.

---

- Выключите виртуальную машину и преобразуйте ее в шаблон.

## Статические IP-адреса vSphere в Cloud Assembly

При развертывании в vSphere в Cloud Assembly можно назначить статический IP-адрес, но нужно быть очень осторожным, чтобы не создать конфликты между командами инициализации `cloudConfig` и спецификациями настройки.

### Примеры проектов

В следующих проектах показано безопасное применение статического IP-адреса без каких-либо конфликтов между командами инициализации облачного шаблона и спецификациями настройки. Все они содержат параметр сети `assignment: static`.

Проектирование	Пример кода облачного шаблона
<p>Назначение статического IP-адреса компьютеру Linux без кода cloud-init</p>	<pre>resources:   wpnet:     type: Cloud.Network     properties:       name: wpnet       networkType: public       constraints:         - tag: sqa   DBTier:     type: Cloud.vSphere.Machine     properties:       flavor: small       image: linux-template       networks:         - name: '\${wpnet.name}'           assignment: static           network: '\${resource.wpnet.id}'</pre>
<p>Назначьте статический IP-адрес компьютеру Linux с кодом cloud-init, который не содержит команды назначения сети.</p> <p><b>ПРИМЕЧАНИЕ.</b> Спецификация настройки vSphere применяется независимо от того, задано ли для свойства customizeGuestOs значение «истина» или данное свойство вообще не используется.</p>	<p>Пример Ubuntu</p> <pre>resources:   wpnet:     type: Cloud.Network     properties:       name: wpnet       networkType: public       constraints:         - tag: sqa   DBTier:     type: Cloud.vSphere.Machine     properties:       flavor: small       image: ubuntu-template       customizeGuestOs: true       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               root:Pa\$\$w0rd           expire: false         write_files:           - path: /tmpFile.txt             content:                 \${resource.wpnet.dns}       runcmd:         - hostnamectl set-hostname --pretty \$         {self.resourceName}         - touch /etc/cloud/cloud-init.disabled       networks:         - name: '\${wpnet.name}'           assignment: static           network: '\${resource.wpnet.id}'</pre> <p>Пример CentOS</p> <pre>resources:   wpnet:     type: Cloud.Network     properties:</pre>



**Проектирование****Пример кода облачного шаблона**

```
name: wpnet
networkType: public
constraints:
  - tag: sqa
DBTier:
type: Cloud.vSphere.Machine
properties:
  flavor: small
  image: centos-template
  customizeGuestOs: true
  cloudConfig: |
    #cloud-config
    write_files:
      - path: /test.txt
        content: |
          deploying in power off.
          then rebooting.
networks:
  - name: '${wpnet.name}'
    assignment: static
    network: '${resource.wpnet.id}'
```

Проектирование	Пример кода облачного шаблона
<p>Назначьте статический IP-адрес компьютеру Linux с кодом cloud-init, который содержит команды назначения сети.</p> <p>Для свойства <code>customizeGuestOs</code> необходимо указать значение «ложь».</p>	<p>Пример Ubuntu</p> <pre> resources:   wpnet:     type: Cloud.Network     properties:       name: wpnet       networkType: public       constraints:         - tag: sqa   DBTier:     type: Cloud.vSphere.Machine     properties:       flavor: small       image: ubuntu-template       customizeGuestOs: false       cloudConfig:           #cloud-config         write_files:           - path: /etc/netplan/99-installer-             config.yaml             content:                 network:                 version: 2                 renderer: networkd                 ethernet:                   ens160:                     addresses: - \${resource.DBTier.networks[0].address}/\${   {resource.wpnet.prefixLength}                     gateway4: \$   {resource.wpnet.gateway}                     nameservers:                       search: \$   {resource.wpnet.dnsSearchDomains}                       addresses: \${resource.wpnet.dns}           runcmd:             - netplan apply             - hostnamectl set-hostname --pretty \$   {self.resourceName}             - touch /etc/cloud/cloud-init.disabled           networks:             - name: '\${wpnet.name}'               assignment: static               network: '\${resource.wpnet.id}' </pre> <p>Пример CentOS</p> <pre> resources:   wpnet:     type: Cloud.Network     properties:       name: wpnet       networkType: public       constraints:         - tag: sqa   DBTier:     type: Cloud.vSphere.Machine     properties:       flavor: small       image: centos-template </pre>

**Проектирование****Пример кода облачного шаблона**

```

    customizeGuestOs: false
    cloudConfig: |
      #cloud-config
      ssh_pwauth: yes
      chpasswd:
        list: |
          root:VMware1!
        expire: false
      runcmd:
        - nmcli con add type
      ethernet con-name 'custom ens192'
      ifname ens192 ip4 ${self.networks[0].address}/
        ${resource.wpnet.prefixLength} gw4 $
        {resource.wpnet.gateway}
        - nmcli con mod 'custom ens192' ipv4.dns "$
        {join(resource.wpnet.dns, ' ')}"
        - nmcli con mod 'custom ens192' ipv4.dns-
        search "${join(resource.wpnet.dnsSearchDomains, ',')}"
        - nmcli con down 'System ens192' ; nmcli
      con up 'custom ens192'
        - nmcli con del 'System ens192'
        - hostnamectl set-hostname --static `dig -x
        ${self.networks[0].address} +short | cut -d "." -f 1`
        - hostnamectl set-hostname --pretty $
        {self.resourceName}
        - touch /etc/cloud/cloud-init.disabled
      networks:
        - name: '${wpnet.name}'
          assignment: static
          network: '${resource.wpnet.id}'

```

При выполнении развертывания на основе указанного образа назначьте статический IP-адрес компьютеру Linux с кодом cloud-init, который содержит команды назначения сети.

Для свойства customizeGuestOs необходимо указать значение «ложь».

Кроме того, облачный шаблон не должен содержать свойство ovfProperties, которое блокирует настройку.

```

resources:
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: public
      constraints:
        - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small

imageRef: 'https://cloud-images.ubuntu.com/releases/focal/release/ubuntu-20.04-server-cloudimg-amd64.ova'
customizeGuestOs: false
cloudConfig: |
  #cloud-config
  ssh_pwauth: yes
  chpasswd:
    list: |
      root:Pa$$w0rd
      ubuntu:Pa$$w0rd
    expire: false
  write_files:
    - path: /etc/netplan/99-netcfg-vrac.yaml
      content: |
        network:
          version: 2
          renderer: networkd

```

Проектирование	Пример кода облачного шаблона
	<pre> ethernets:   ens192:     dhcp4: no     dhcp6: no     addresses:       - \${resource.DBTier.networks[0].address}/\${         {resource.wpnet.prefixLength}         gateway4: \$         {resource.wpnet.gateway}         nameservers:           search: \$           {resource.wpnet.dnsSearchDomains}           addresses: \${resource.wpnet.dns}       runcmd:         - netplan apply         - hostnamectl set-hostname --pretty \$         {self.resourceName}         - touch /etc/cloud/cloud-init.disabled       networks:         - name: '\${wpnet.name}'           assignment: static           network: '\${resource.wpnet.id}' </pre>

## Проекты, которые не будут работать или могут привести к нежелательным результатам

- Код cloud-init не содержит команды назначения сети, а свойству customizeGuestOs присвоено значение «ложь».

Для настройки параметров сети нет ни команд инициализации, ни спецификации настройки.

- В коде cloud-init нет команд назначения сети, а для свойства ovfProperties задано значение.

Команды инициализации отсутствуют, но свойство ovfProperties заблокировало спецификацию настройки.

- Код cloud-init содержит команды назначения сети, а свойство customizeGuestOs отсутствует или имеет значение «истина».

Применение спецификации настройки приводит к конфликту с командами инициализации.

## Другие решения для cloud-init и спецификаций настройки

При развертывании в vSphere также можно настроить образ преодоления конфликтов между cloud-init и спецификацией настройки. Дополнительные сведения см. в следующем внешнем репозитории.

- [Сценарии подготовки образа vSphere](#)

## Отложенное развертывание в Cloud Assembly

Для перехода к развертыванию Cloud Assembly может потребоваться полностью инициализировать виртуальную машину.

Например, развертывание компьютера, на котором продолжается установка пакетов и запуск веб-сервера, может привести к ситуации, когда нетерпеливый пользователь будет пытаться получить доступ к приложению до того, как оно станет доступным.

При использовании этого компонента необходимо учитывать следующие факторы.

- Данный компонент использует модуль `cloud-init phone_home` и доступен при развертывании компьютеров Linux.
- Модуль `phone home` недоступен для Windows из-за ограничений `Cloudbase-init`.
- Модуль `phone home` может влиять на порядок развертывания, как явная зависимость, но отличается большей гибкостью применительно к параметрам времени и обработки.

См. раздел [Создание привязок и зависимостей между ресурсами в Cloud Assembly](#).

- Для модуля `phone home` в облачном шаблоне требуется раздел `cloudConfig`.
- Ваш творческий подход также играет важную роль. В командах инициализации может указываться предварительно заданное время ожидания следующей операции, которое можно использовать в сочетании с `phone home`.
- Модуль `phone home` на базе облачного шаблона не будет работать, если шаблон компьютера уже содержит настройки модуля `phone_home`.
- Компьютер должен иметь исходящий коммуникационный доступ для обратной связи с Cloud Assembly.

Чтобы отложить развертывание в Cloud Assembly, добавьте раздел `cloudConfigSettings` в облачный шаблон.

```
cloudConfigSettings:
  phoneHomeShouldWait: true
  phoneHomeTimeoutSeconds: 600
  phoneHomeFailOnTimeout: true
```

Свойство	Описание
<code>phoneHomeShouldWait</code>	Следует ли ожидать инициализации: истина или ложь.
<code>phoneHomeTimeoutSeconds</code>	Момент принятия решения, следует ли продолжить развертывание, несмотря на то, что инициализация все еще выполняется. По умолчанию установлено значение 10 минут.
<code>phoneHomeFailOnTimeout</code>	Следует ли продолжить развертывание после превышения времени ожидания: истина или ложь. Следует отметить, что даже в случае продолжения развертывание может завершиться сбоем по разным причинам.

## Гостевая настройка Windows в Cloud Assembly

Чтобы служба Cloud Assembly автоматически инициализировала компьютер Windows при развертывании, подготовьте образ, который поддерживает `Cloudbase-Init`, а затем облачный шаблон, содержащий соответствующие команды.

Процесс создания образа зависит от поставщика облачной среды. Приведенный здесь пример предназначен для vSphere.

## Образ Windows Cloud Assembly для vSphere

Чтобы Cloud Assembly удалось инициализировать компьютер Windows, развернутый в vSphere, в основе образа должен лежать шаблон vSphere с установленной и настроенной службой Cloudbase-Init.

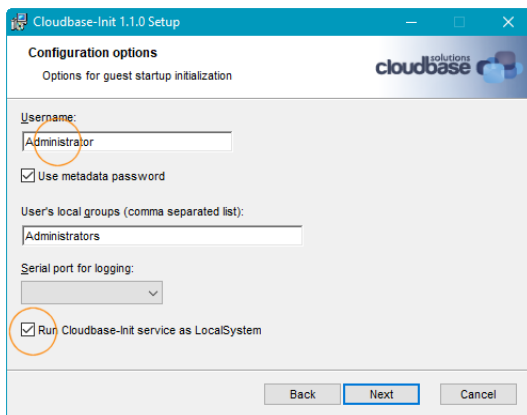
### Создание образа

1. Для создания и включения виртуальной машины Windows используйте vSphere.
2. Войдите в Windows на виртуальной машине.
3. Загрузите Cloudbase-Init.

<https://cloudbase.it/cloudbase-init/#download>

4. Запустите файл setup.msi для Cloudbase-Init.

Во время установки введите **Администратор** в качестве имени пользователя и выберите вариант запуска LocalSystem.



Для других параметров настройки можно оставить значения по умолчанию.

5. Разрешите установку, но не закрывайте последнюю страницу «Завершено» в мастере установки.

---

**Важно!** Не закрывайте последнюю страницу мастера установки.

---

6. При открытой странице «Завершено» мастера установки перейдите в Windows в папку установки Cloudbase-Init и откройте в текстовом редакторе следующий файл.

```
conf\cloudbase-init-unattend.conf
```

7. Установите для параметра `metadata_services` значение `OvfService`, как показано ниже. Добавьте параметр, если он еще не существует.

```
metadata_services=cloudbaseinit.metadata.services.ovfservice.OvfService
```

8. Сохраните и закройте `cloudbase-init-unattend.conf`

9. В той же папке откройте следующий файл в текстовом редакторе.

```
conf\cloudbase-init.conf
```

10. Настройте `first_logon_behaviour`, `metadata_services` и `plugins`, как показано ниже. Добавьте параметры, если они еще не существуют.

```
first_logon_behaviour=always
. . .
metadata_services=cloudbaseinit.metadata.services.ovfservice.OvfService
. . .
plugins=cloudbaseinit.plugins.windows.createuser.CreateUserPlugin,cloudbaseinit.plugins.win
dows.setuserpassword.SetUserPasswordPlugin,cloudbaseinit.plugins.common.sshpublickeys.SetUs
erSSHPublicKeysPlugin,cloudbaseinit.plugins.common.userdata.UserDataPlugin
. . .
```

11. Сохраните и закройте `cloudbase-init.conf`
12. На странице «Завершено» мастера установки выберите необходимые параметры, чтобы запустить программу Sysprep и завершить работу по окончании выполнения этой программы, а затем нажмите кнопку **Готово**.

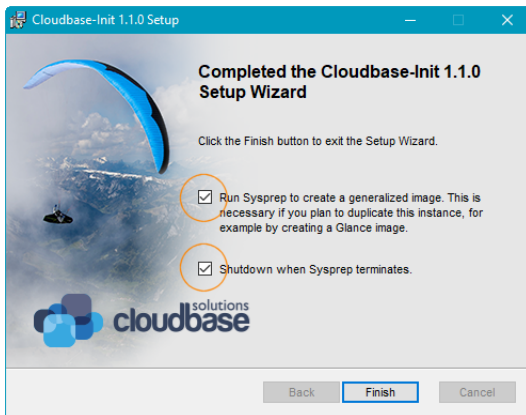
---

**Примечание** Специалисты VMware сталкивались с ситуациями, когда запуск программы Sysprep препятствовал нормальному развертыванию образа.

В ходе развертывания служба Cloud Assembly применяет динамически создаваемую спецификацию настройки, которая отключает сетевой интерфейс. Состояние ожидания Sysprep в образе может привести к сбою спецификации настройки и отключению развертывания.

Если вы подозреваете, что это происходит в вашей среде, попробуйте не активировать параметры Sysprep при создании образа.

---



13. После завершения работы виртуальной машины используйте vSphere, чтобы преобразовать ее в шаблон.

### Дополнительные сведения

В следующей таблице подробно описываются записи конфигурации, созданные во время настройки.

Параметр конфигурации	Назначение
Username, CreateUserPlugin и SetUserPasswordPlugin	После завершения работы программы Sysprep при первом запуске используется параметр CreateUserPlugin, чтобы создать учетную запись с именем пользователя «Администратор» и пустым паролем. Параметр SetUserPasswordPlugin позволяет команде Cloudbase-Init заменить пустой пароль на пароль для удаленного доступа, который будет добавлен в облачный шаблон.
Действия при первом входе в систему	Этот параметр предлагает пользователю изменить пароль при первом входе в систему.
Службы метаданных	Если задано только значение OvfService, Cloudbase-Init не будет искать другие службы метаданных, которые не поддерживаются в vCenter. Это уменьшит объем информации в файлах журнала, так как в противном случае в журналах будут создаваться записи об ошибках при поиске таких служб.
Подключаемые модули	Если указаны только подключаемые модули с возможностями, которые поддерживаются значением OvfService, количество записей в журналах также уменьшается. Cloudbase-Init выполняет подключаемые модули в указанном порядке.
Запустить как LocalSystem	Этот параметр поддерживает любые расширенные команды инициализации, которые могут потребовать запуска Cloudbase-Init в рамках выделенной учетной записи администратора.

## Команды Cloudbase-Init для Windows в Cloud Assembly

Чтобы выполнять инициализацию компьютеров Windows во время развертывания, добавьте команды Cloudbase-Init в код шаблона Cloud Assembly.

Здесь приводится пример для vSphere; для других поставщиков облачной среды используются похожие параметры.

### Необходимые условия

- Создание инфраструктуры. В Cloud Assembly добавьте облачную учетную запись vSphere и связанную облачную зону.
- Добавьте сопоставления конфигураций ресурсов и образов, а затем профили сети и хранилища.  
Сопоставление образов в инфраструктуре должно указывать на шаблон Windows, созданный для поддержки Cloudbase-Init. См. раздел [Образ Windows Cloud Assembly для vSphere](#).  
Если шаблон отсутствует в списке, перейдите в раздел «Облачные учетные записи» и синхронизируйте образы. В противном случае будет выполнена автоматическая синхронизация (выполняется через каждые 24 часа).
- Добавьте проект и пользователей. Убедитесь, что пользователи могут выполнять предоставление ресурсов в данной облачной зоне.

Дополнительные сведения о создании инфраструктуры и проектов см. в [Учебник «Настройка и тестирование инфраструктуры и развертываний с несколькими облачными средами в Cloud Assembly»](#).



## Процедура

1. В Cloud Assembly перейдите на вкладку **Проект** и создайте новый облачный шаблон.
2. Добавьте раздел `cloudConfig` с нужными командами Cloudbase-init.

Следующие примеры команд позволяют создать новый файл на диске C: в ОС Windows и задать имя узла.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: cloudbase-init-win-2016
      flavor: small
      remoteAccess:
        authentication: usernamePassword
        username: Administrator
        password: Password1234@$
      cloudConfig: |
        #cloud-config
        write_files:
          content: Cloudbase-Init test
          path: C:\test.txt
        set_hostname: testname
```

Дополнительные сведения см. в разделе [документации по Cloudbase-init](#).

3. Добавьте свойства `remoteAccess`, чтобы настроить компьютер для начального входа в Windows.

Как было указано при создании шаблона, служба метаданных выбирает учетные данные для входа и предоставляет их службам CreateUserPlugin и SetUserPasswordPlugin. Следует отметить, что пароль должен соответствовать требованиям Windows.

4. В Cloud Assembly протестируйте облачный шаблон и разверните его.
5. После развертывания войдите на новый компьютер Windows, используя Windows RDP и учетные данные в шаблоне, и проверьте настройки.

В примере выше необходимо найти файл `C:\test.txt` и проверить имя узла в свойствах системы.

## Кластеры компьютеров и дисков в Cloud Assembly

С помощью проектов шаблона Cloud Assembly можно развернуть кластер компьютеров и присоединить кластер дисков.

Чтобы развернуть кластеры компьютеров и дисков, используйте в облачных шаблонах `allocatePerInstance` [Флаги ресурса Cloud Assembly для запросов](#), а также [Синтаксис выражений Cloud Assembly](#) `count.index` и `map_to_object`.

Следующие примеры кода облачного шаблона могут служить в качестве инструкций для проектов по развертыванию кластеров.

## Два компьютера, совместно использующие кластер дисков

```
resources:
  app0:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, 0,2), "source")}'
  app1:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, 2,4), "source")}'
  disk:
    type: Cloud.Volume
    allocatePerInstance: true
    properties:
      count: 4
      capacityGb: 5
```

## Переменное количество компьютеров, содержащих по одному диску

```
inputs:
  count:
    type: integer
    default: 2
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      count: '${input.count}'
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, count.index, count.index +
1), "source")}'
  disk:
    type: Cloud.Volume
    allocatePerInstance: true
    properties:
      count: '${input.count}'
      capacityGb: 5
```

## Переменное количество компьютеров, содержащих по два диска

```
inputs:
  count:
    type: integer
    default: 2
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      count: ${input.count}
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, 2*count.index,
2*(count.index + 1)), "source")}'
    disk:
      type: Cloud.Volume
      allocatePerInstance: true
      properties:
        count: ${2*input.count}
        capacityGb: 5
```

## Задание размера диска во время запроса

```
inputs:
  disksize:
    type: array
    minItems: 2
    maxItems: 2
    items:
      type: object
      properties:
        size:
          type: integer
resources:
  app:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      flavor: small
      image: ubuntu
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, 0, 2), 'source')}'
    disk:
      type: Cloud.Volume
      allocatePerInstance: true
      properties:
        count: 2
        capacityGb: ${input.disksize[count.index].size}
```

## Настраиваемое именование развернутых ресурсов в Cloud Assembly

Администратор облачных систем или проектов имеет предварительно заданное соглашение об именовании ресурсов в среде. Необходимо, чтобы развернутый ресурс следовал данному соглашению без вмешательства пользователя. Можно создать шаблон именования для всех развертываний из проекта Cloud Assembly.

Например, в соответствии с соглашением об именовании узлов для ресурса добавляется префикс *projectname-sitecode-costcenter-whereDeployed-identifier*. Настройте шаблон настраиваемого наименования для компьютеров для каждого проекта. Некоторые переменные шаблона извлекаются из системы после развертывания, а другие основаны на настраиваемых свойствах проекта. Настраиваемый шаблон именования для вышеуказанного префикса выглядит примерно следующим образом.

```
${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}
```

Идентификатор, введенный в шаблоне в виде `${#####}`, содержит шесть цифр. Идентификатор — это счетчик, гарантирующий уникальность. Счетчик относится ко всем элементам организации. Его значение увеличивается во всех проектах, а не только в текущем. При наличии нескольких проектов для развертываний в текущем проекте не предполагается последовательность от 000123 до 000124. Можно ожидать приращение от 000123 до 000127.

Все имена ресурсов должны быть уникальными. Для обеспечения уникальности используйте свойство приращения чисел. Приращение чисел обеспечивается для всех развертываний, в том числе тех, именование которых осуществляется Cloud Assembly. Так как система становится все более надежной, а также ко многим типам ресурсов применяются настраиваемые имена, нумерование может казаться произвольным, но значения все равно уникальны. При запуске тестового развертывания числовые значения также увеличиваются.

Следующий список содержит примеры ресурсов, к которым применяются настраиваемые имена. Список не является исчерпывающим.

Таблица 6-2. Примеры ресурсов, к которым применяются настраиваемые имена

Группа ресурсов	Типы ресурсов
Виртуальные машины	<ul style="list-style-type: none"> <li>■ Cloud.Machine</li> <li>■ Cloud.vSphere.Machine</li> <li>■ Cloud.AWS.EC2.Instance</li> <li>■ Cloud.GCP.Machine</li> <li>■ Cloud.Azure.Machine</li> </ul>
Подсистемы балансировки нагрузки	<ul style="list-style-type: none"> <li>■ Cloud.LoadBalancer</li> <li>■ Cloud.NSX.LoadBalancer</li> </ul>
Сети	<ul style="list-style-type: none"> <li>■ Cloud.Network</li> <li>■ Cloud.vSphere.Network</li> <li>■ Cloud.NSX.Network</li> </ul>
Группы безопасности	<ul style="list-style-type: none"> <li>■ Cloud.SecurityGroup</li> </ul>

Таблица 6-2. Примеры ресурсов, к которым применяются настраиваемые имена (продолжение)

Группа ресурсов	Типы ресурсов
Диски	<ul style="list-style-type: none"> <li>■ Cloud.Volume</li> <li>■ Cloud.vSphere.Disk</li> <li>■ Cloud.AWS.Volume</li> <li>■ Cloud.GCP.Disk</li> <li>■ Cloud.Azure.Disk</li> </ul>
NSX	<ul style="list-style-type: none"> <li>■ Cloud.NSX.Gateway</li> <li>■ Cloud.NSX.NAT</li> </ul>
Microsoft Azure	<ul style="list-style-type: none"> <li>■ Cloud.Azure.ResourceGroup</li> </ul>

Помимо приведенных здесь примеров, также можно добавить имя пользователя, используемый образ, другие встроенные параметры и простые строки. При создании шаблона система выводит подсказки по возможным параметрам.

Следует отметить, что некоторые используемые здесь значения приведены только в качестве примера. Их нельзя перенести в вашу среду одно к одному. Подумайте о том, где можно внести необходимые изменения и как экстраполировать примеры значений на актуальные параметры, чтобы облачная инфраструктура и управление развертыванием соответствовали вашим требованиям.

#### Необходимые условия

- Для этого требуется знакомство с соглашением об именовании, которое будет использоваться для развертываний из проекта.
- В этой процедуре предполагается, что у вас есть простой облачный шаблон, используемый для проверки именования настраиваемых префиксов узлов, или его можно будет создать.

#### Процедура

1. Выберите **Инфраструктура > Проекты**.
2. Выберите существующий проект или создайте новый.
3. На вкладке **Предоставление** перейдите в раздел «Настраиваемые свойства» и создайте свойства для кода сайта и значений источника расходов.

Здесь можно заменить отображаемые значения на значения, которые относятся к вашей среде.

Custom Properties

Specify the custom properties that should be added to all requests in this project. ⓘ

Define custom properties	Name	Value	
	siteCode	BGL	-
	costCenter	IT-research	- +

Custom Naming

Specify the naming template to be used for machines provisioned in this project.

Template

`${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}` ⓘ

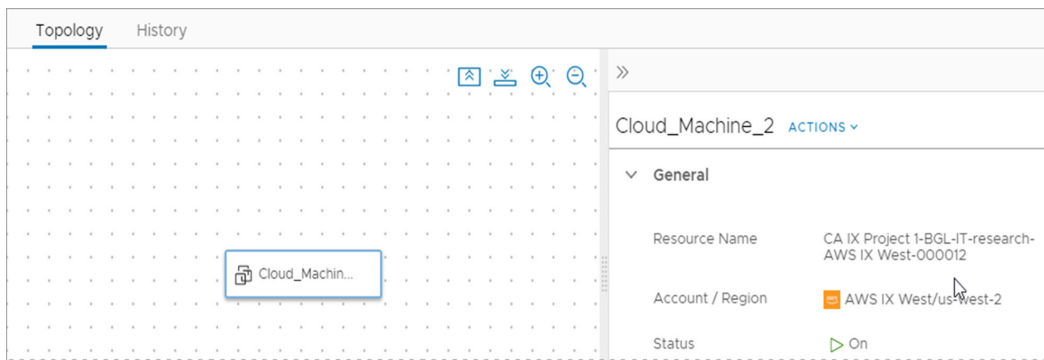
- а) Создайте настраиваемое свойство с именем **siteCode** и значением **BGL**.
  - б) Добавьте еще одно настраиваемое свойство с именем **costCenter** и значением **IT-research**.
4. Перейдите в раздел «Настраиваемое именование» и добавьте следующий шаблон.

```
${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}
```

Можно скопировать в строке, но если это ваш первый шаблон именования, рассмотрите возможность использования пояснительного текста и быстрого выбора в ходе создания шаблона.

5. Разверните облачный шаблон, связанный с проектом, и убедитесь, что настраиваемое имя применяется к ресурсу.
  - а) Перейдите на вкладку **Проектирование**, а затем щелкните облачный шаблон, связанный с проектом.
  - б) Разверните облачный шаблон.
 

Откроется страница **Развертывания**, где отображается выполняемое развертывание.
  - в) По окончании развертывания щелкните его имя.
  - г) На вкладке **Топология** обратите внимание на то, что настраиваемое имя — это имя ресурса на правой панели.



6. Если тестовый облачный шаблон развернут для проверки соглашения об именовании, развертывание можно удалить.

## Следующие шаги

Создайте шаблоны настраиваемых именованных для других проектов.

# Добавление ресурса SaltStack Config в проектах Cloud Assembly

Если служба SaltStack Config интегрирована с vRealize Automation, можно применить ресурс SaltStack Config, чтобы установить служебные серверы на виртуальных машинах в развертываниях. После развертывания служебного сервера для управления ресурсами можно использовать мощные возможности службы SaltStack Config по управлению конфигурацией, устранению дрейфа и управлению состоянием.

Служебные серверы — это агенты, на которых выполняется служба salt-minion. Служба подписывается на задания, опубликованные главным сервером Salt, на котором выполняется служба salt-master. Когда определенное задание применяется к служебному серверу, он его выполняет.

Ресурс SaltStack Config можно использовать для развертывания служебных узлов и применения файлов состояния в ходе развертывания компьютеров Linux и Windows. Для добавления или обновления служебных серверов и файлов состояния в существующих развертываниях можно выполнить действие по регулярному обслуживанию **Применение конфигурации Salt**. В этом действии используется свойство saltConfiguration. Дополнительные сведения о действии по регулярному обслуживанию см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

Если для развертывания служебных серверов и файлов состояния в рамках действия по регулярному обслуживанию использовалось свойство saltConfiguration, попробуйте обновить свои облачные шаблоны, чтобы использовать ресурс SaltStack Config. Свойство saltConfiguration будет признано устаревшим в будущем выпуске и будет заменено на ресурс SaltStack Config вместе с альтернативным действием по регулярному обслуживанию.

---

**Примечание** Свойство saltConfiguration и ресурс SaltStack Config поддерживаются в одном и том же облачном шаблоне, но не для одного и того же ресурса.

Например, можно создать облачный шаблон с двумя компьютерами. Первый компьютер присоединен к ресурсу SaltStack Config. Второй компьютер не присоединен к ресурсу SaltStack Config, и к нему также не применена конфигурация Salt. После развертывания облачного шаблона для применения конфигурации Salt операцию по регулярному обслуживанию можно выполнить только для второго компьютера. Действие по регулярному обслуживанию для компьютера с ресурсом SaltStack Config будет отключено.

---

## Перед началом работы

1. Убедитесь, что вы установили SaltStack Config и настроили интеграцию. См. раздел [Создание интеграции SaltStack Config в vRealize Automation](#).

2. В SaltStack Config убедитесь, что преобразование полных доменных имен от служебного сервера к главному серверу работает.

- а) Чтобы проверить полное доменное имя на главном сервере Salt в SaltStack Config, выберите **Служебные серверы > Все служебные серверы**.
- б) Отфильтруйте столбец **ИД служебного сервера** для значения **saltmaster**.
- в) Щелкните **saltmaster** для просмотра сведений.
- г) Убедитесь в правильности значения полного доменного имени.

3. Если выполняется развертывание служебных серверов на компьютере Linux, убедитесь, что для образов в службе vSphere, которые планируется развернуть с помощью служебного сервера Salt, включены возможности SSH. Протокол SSH используется для удаленного доступа к компьютеру и развертывания служебного сервера.

4. Сведения о развертывании служебных серверов на компьютере Windows см. в разделе [Развертывание служебных серверов с помощью API-интерфейса в среде Windows](#).

5. Убедитесь, что развертываемым компьютерам можно назначить IP-адреса.

Для SaltStack Config требуется, чтобы компьютеры имели IP-адреса. Используйте IP-адреса общедоступного диапазона IP CIDR для программно-определяемого ЦОД, где находится главный сервер Salt.

6. Прежде чем добавить свойства ресурса SaltStack Config, убедитесь, что облачный шаблон, к которому добавляется служебный сервер, может быть развернут.

7. Убедитесь в наличии следующих ролей службы:

- а) администратор Cloud Assembly;
- б) пользователь Cloud Assembly;
- в) администратор Service Broker;

Эти роли службы необходимы для использования ресурса SaltStack Config.

## Добавление ресурса SaltStack Config в облачный шаблон

Разработчик облачных шаблонов может добавлять в код YAML свойства, обеспечивающие установку служебного сервера SaltStack Config при развертывании шаблона.

К базовым свойствам, добавляемым в шаблон, относятся удаленный доступ для компьютера, который необходимо развернуть, и свойства конфигурации для ресурса SaltStack Config. В процедуру включены только выбранные свойства. Код YAML содержит другие свойства ресурса SaltStack Config, которые не используются в этом примере. Дополнительные сведения см. на схеме.

В этом примере показано, как добавить имя пользователя и пароль для свойств удаленного доступа, однако можно настроить секретное свойство и добавить его в шаблон. См. пример в разделе [Секретные свойства Cloud Assembly](#).



## Процедура

1. В Cloud Assembly выберите **Проектирование > Облачные шаблоны**.
2. Откройте существующий шаблон.
3. Найдите ресурс **SaltStack Config** и перетащите его на холст.
4. Присоедините ресурс **SaltStack Config** к компьютеру, на котором будет установлен служебный сервер.
5. На панели кода добавьте свойства к ресурсу Cloud\_SaltStack\_1.

Не требуется добавлять все возможные свойства. Значения, используемые в этом примере, описаны в таблице.

```
Cloud_SaltStack_1:
  type: Cloud.SaltStack
  properties:
    masterId: saltstack_enterprise_installer
    hosts:
      - ${resource.Cloud_vSphere_Machine_1.id}
    saltEnvironment: sse
    stateFiles:
      - /doe.sls
    variables:
      user: joe
```

Описание свойств Cloud\_SaltStack\_1, используемых в этом примере

Свойство	Описание
masterId	В примере схемы masterId имеет значение saltstack_enterprise_installer. Возможно, у вас есть идентификаторы главного сервера (masterID), заданные в SaltStack Config в разделе <b>Администрирование &gt; Ключи главного сервера</b> .
Узлы	<p>Значение hosts — это идентификатор компьютера или кластера компьютеров, на которых необходимо установить служебный сервер. По умолчанию имя компьютера передается в качестве идентификатора служебного сервера в SaltStack Config.</p> <p>Рекомендуется выбирать имена компьютеров размером не более 15 символов, особенно если развертывание служебных серверов происходит в ОС Windows. Windows не позволяет использовать имена узлов, длина которых превышает 15 символов.</p> <p>Чтобы определить настраиваемое соглашение об именовании для компьютеров, которые необходимо развернуть, см. раздел <a href="#">Настраиваемое именование развернутых ресурсов в Cloud Assembly</a>.</p>

Свойство	Описание
saltEnvironment	В этом примере <code>sse</code> представляет собой расположение для файлов состояния. Ваш и файлы состояний могут находиться в других папках на файловом сервере в SaltStack Config в разделе <b>Config &gt; Файловый сервер</b> .
stateFiles	В этом примере значение <code>doe.sls</code> представляет собой файл состояния, указанный в каталоге файлового сервера и заданный как <code>saltEnvironment</code> .
variables	<code>variables</code> (переменные) — это значения, которые используются в файле состояния. В этом примере <code>doe.sls</code> принимает значение <code>user</code> .

6. Добавьте свойства `remoteAccess` к компьютеру, на котором размещается служебный сервер Salt.

Значение ключа `authentication` должно быть `usernamePassword` или `generatedPublicPrivateKey`. `publicPrivateKey` не поддерживается.

```
remoteAccess:
  authentication: usernamePassword
  username: adminUser
  password: adminPassword
```

7. Убедитесь, что код YAML содержит свойства, аналогичные свойствам в примере ниже.

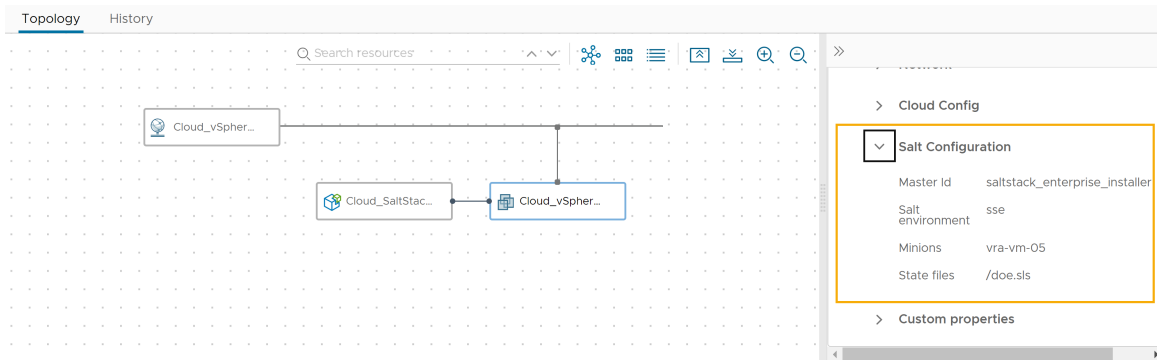
```
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: ubuntu
      flavor: small
      remoteAccess:
        authentication: usernamePassword
        username: adminUser
        password: adminPassword
  Cloud_SaltStack_1:
    type: Cloud.SaltStack
    properties:
      masterId: saltstack_enterprise_installer
      hosts:
        - ${resource.Cloud_vSphere_Machine_1.id}
      saltEnvironment: sse
      stateFiles:
        - /doe.sls
      variables:
        user: joe
```

8. Протестируйте облачный шаблон и разверните его.

Если развертывание служебного сервера выполнено неудачно, см. раздел [Устранение неполадок в развертываниях служебных серверов](#).

## 9. Проверьте свойства конфигурации Salt для развернутого компьютера.

- Выберите **Развертывания > Развертывания** и откройте сведения о развертывании.
- На вкладке **Топология** щелкните компьютер и разверните свойства на правой панели.



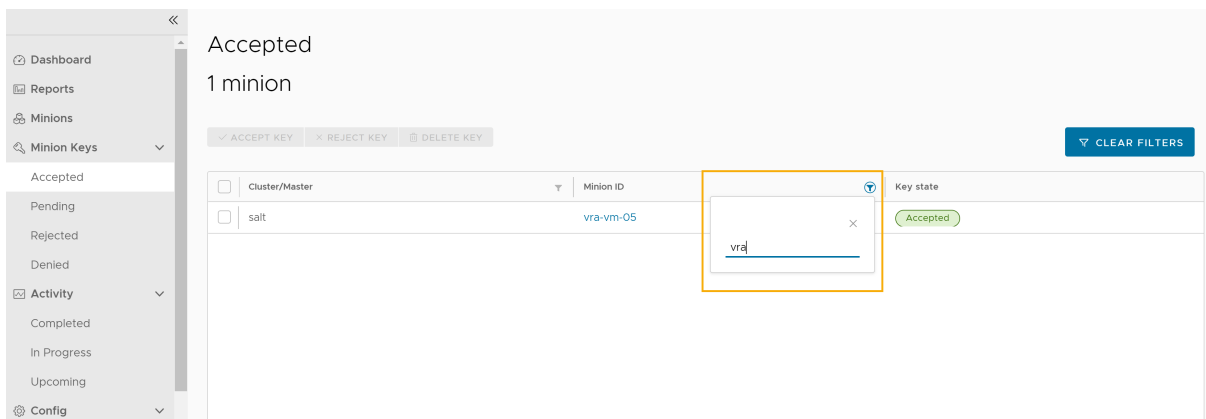
## Проверка служебного сервера в SaltStack Config

После установки служебного сервера на виртуальной машине необходимо найти этот служебный сервер и выполнить любые задания или команды на ресурсе.

### Процедура

- Чтобы открыть SaltStack Config, щелкните меню приложений в правом верхнем углу и выберите пункт **Консоль облачных служб**.
- Щелкните плитку службы **SaltStack Config**.
- В SaltStack Config разверните **Ключи служебного сервера** и нажмите **Принято**.
- В столбце **ИД служебного сервера** щелкните значок фильтра и введите имя служебного сервера.

По умолчанию имя служебного сервера — имя узла виртуальной машины. В этом примере ИД служебного сервера — vra-vm-05.



- Для просмотра сведений щелкните имя служебного сервера.

На служебном сервере можно выполнять задания и команды. Например, получение статистики использования дисков. Это задание возвращает статистику использования дисков для служебного сервера.

vra-vm-05

Presence: Present

Key state: Accepted

Master: salt

Targets: All Minions , Linux , Ubuntu

IPv4: 10.196.194.192, 127.0.0.1

OS: Ubuntu16.04

Salt Version: 3002.7

⚡ RUN JOB

RUN COMMAND

Grains

Activity

biosreleasedate	12/12/2018
biosversion	6.00
> cpu_flags	--
cpu_model	Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz
cpuarch	x86_64
cwd	/
> disks	--

## Устранение неполадок в развертываниях служебных серверов

Ознакомьтесь с распространенными ошибками пользователей при развертывании служебных серверов Salt с помощью ресурса SaltStack Config или свойства `saltConfiguration`.

### Отложенный запуск узла

Если службы Windows или Linux на узле не готовы после развертывания облачного шаблона, в Cloud Assembly может возникнуть ошибка «Сбой при развертывании служебного сервера и/или выполнении файла состояния».

Чтобы устранить эту ошибку, обновите подключаемый модуль Master до последней устойчивой версии. После обновления можно включить параметр конфигурации в `/etc/salt/master.d/raas.conf`, который предоставляет службам Windows и Linux время для активации до развертывания служебного сервера Salt.

После обновления до последней версии подключаемого модуля Master выполните следующие действия, чтобы отложить запуск узла.

1. Перейдите на вкладку **Журнал** на странице сведений о развертывании.
2. Если отображается сообщение об ошибке «Сбой при развертывании служебного сервера и/или выполнении файла состояния», скопируйте идентификатор задания (JID) и откройте SaltStack Config.
3. В SaltStack Config выберите **Действие > Выполнено**, чтобы открыть выполненные задания.
4. В столбце **JID** нажмите значок фильтра и введите значение JID.
5. Щелкните JID, чтобы просмотреть страницу с результатами заданий.

6. Перейдите на вкладку **Необработанные данные**, чтобы увидеть необработанные выходные данные для задания.

#### Windows

Если последняя строка в необработанных выходных данных для задания содержит сообщение «Не удалось подключиться к узлу: время ожидания истекло», необходимо добавить в `/etc/salt/master.d/raas.conf` следующий параметр конфигурации, чтобы отложить запуск на 180 секунд.

```
sseapi_win_minion_deploy_delay: 180
```

#### Linux

Если последняя строка в необработанных выходных данных для задания содержит сообщение «Удаленный узел недоступен при использовании указанных учетных данных», необходимо добавить в `/etc/salt/master.d/raas.conf` следующий параметр конфигурации, чтобы отложить запуск на 90 секунд.

```
sseapi_linux_minion_deploy_delay: 90
```

7. Перезапустите службу Salt Master:

```
systemctl restart salt-master
```

8. Повторно разверните облачный шаблон.

Если при развертывании произошла ошибка, можно увеличить значение параметра задержки и повторно развернуть шаблон.

### Следующие шаги

Сведения о возможностях SaltStack Config по управлению ресурсами см. в документации по [SaltStack Config](#).

## Конфигурации Terraform в Cloud Assembly

Конфигурации Terraform можно внедрять в облачные шаблоны Cloud Assembly как ресурсы.

### Подготовка среды выполнения Cloud Assembly Terraform

Для проектов, которые включают в себя конфигурации Terraform, требуется доступ к среде выполнения Terraform, интегрируемой с локальной версией Cloud Assembly.

### Добавление среды выполнения Terraform

Среда выполнения состоит из кластера Kubernetes, выполняющего команды Terraform CLI для осуществления запрошенных операций. Кроме того, среда выполнения собирает журналы и возвращает результаты команд Terraform CLI.

Для локальной версии vRealize Automation требуется, чтобы пользователи настроили собственный кластер Kubernetes в среде выполнения Terraform. Для каждой организации поддерживается только одна среда выполнения Terraform. Все развертывания Terraform для этой организации используют одну и ту же среду выполнения.

1. Убедитесь, что у вас есть кластер Kubernetes, в котором можно запустить интерфейс командной строки Terraform.
  - Все пользователи могут предоставлять файл kubeconfig для запуска интерфейса командной строки Terraform в неуправляемом кластере Kubernetes.

- Пользователи лицензии уровня Enterprise могут запускать интерфейс командной строки Terraform в кластере Kubernetes под управлением vRealize Automation.

В Cloud Assembly перейдите в раздел **Инфраструктура > Ресурсы > Kubernetes** и подтвердите, что у вас есть кластер Kubernetes. Если у вас нет такого кластера, см. инструкции по его добавлению в [Работа с Kubernetes в Cloud Assembly](#).

2. Если кластер Kubernetes только что добавлен или изменен, дождитесь завершения сбора данных.  
Процесс сбора данных извлекает список пространств имен и другие сведения и может занимать до 5 минут в зависимости от поставщика.
3. После завершения сбора данных выберите **Инфраструктура > Подключения > Интеграции > Добавить интеграцию**, а затем карточку **Среда выполнения Terraform**.
4. Введите значения параметров.

Рис. 6-3. Пример интеграции среды выполнения Terraform

## New Integration

Name \*

OurOrg TF Runtime

Description

### Terraform Runtime Integration

Runtime type \*

☒ Managed kubernetes cluster
 ☐ External kubeconfig

Kubernetes cluster \*

OurK8Cluster

Kubernetes namespace \*

OurK8Namespace

### Runtime Container Settings

Image

projects.registry.vmware.com/vra/terraform:latest

CPU request (Millicores)

250

CPU limit (Millicores)

250

Memory request (MB)

512

Memory limit (MB)

512

VALIDATE

Настройка	Описание
Имя	Назначьте для интеграции среды выполнения уникальное имя.
Описание	Укажите, для чего предназначена интеграция.
Интеграция среды выполнения Terraform:	
Тип среды выполнения (только для уровня Enterprise)	Пользователи лицензии уровня Enterprise могут выбрать, запускать ли интерфейс командной строки Terraform в кластере Kubernetes под управлением vRealize Automation или в неуправляемом кластере.
Kubernetes kubeconfig (все пользователи)	<p>Для неуправляемого кластера Kubernetes: вставьте все содержимое файла kubeconfig кластер, предназначенное для внешнего кластера.</p> <p>Сведения об использовании внешней среды выполнения Kubernetes с прокси-сервером см. в разделе <a href="#">Добавление поддержки прокси-сервера</a>.</p> <p>Этот параметр доступен для всех пользователей.</p>

Настройка	Описание
Кластер Kubernetes (только для уровня Enterprise)	<p>Для Kubernetes под управлением vRealize Automation выберите кластер, в котором будет запускаться интерфейс командной строки Terraform.</p> <p>Кластер и соответствующий файл kubeconfig должны быть доступны. Для проверки доступа к kubeconfig используйте метод GET в каталоге <code>/cmx/api/resources/k8s/clusters/{clusterId}/kube-config</code>.</p> <p>Этот параметр доступен только для лицензий уровня Enterprise.</p>
Пространство имен Kubernetes	Выберите пространство имен, которое будет использоваться в кластере для создания модулей, которые запускают Terraform CLI.
Параметры контейнера среды выполнения:	
Изображение	<p>Введите путь к образу контейнера версии Terraform, которую необходимо запустить.</p> <p><b>Примечание</b> Кнопка «Проверить» не позволяет проверить наличие образа контейнера.</p>
Требование к ЦП	Укажите объем ЦП для выполнения контейнеров. Значение по умолчанию — 250 millicore.
Ограничение ЦП	Укажите максимально допустимый объем ЦП для выполнения контейнеров. Значение по умолчанию — 250 millicore.
Требование к памяти	Укажите объем памяти для выполнения контейнеров. Значение по умолчанию — 512 МБ.
Ограничение памяти	Укажите максимально допустимый объем памяти для выполнения контейнеров. Значение по умолчанию — 512 МБ.

5. Щелкните **ПРОВЕРИТЬ** и при необходимости измените значения параметров.

6. Нажмите кнопку **ДОБАВИТЬ**.

Настройки параметров кэшируются. После добавления интеграции можно изменить такие параметры, как кластер или пространство имен, но для обнаружения изменений и запуска Terraform CLI с новыми параметрами может потребоваться до 5 минут.

## Устранение неполадок среды выполнения Terraform

Некоторые проблемы при развертывании конфигурации Terraform могут быть связаны с интеграцией среды выполнения.



Проблема	Причина	Разрешение
Не удастся выполнить проверку, выводится сообщение о недопустимости пространства имен.	Кластер был изменен, но оставлено предыдущее пространство имен в пользовательском интерфейсе.	После изменения выбранного кластера всегда снова выбирайте пространство имен.
Раскрывающийся список пространств имен пуст или не содержит новые добавленные пространства имен.	Сбор данных для кластера не завершен. Сбор данных занимает до 5 минут после ввода или изменения кластера и до 10 минут при вводе или изменении пространства имен.	В случае нового кластера с существующими пространствами имен подождите 5 минут до завершения сбора данных. В случае нового пространства имен в существующем кластере подождите 10 минут до завершения сбора данных. Если проблема сохраняется, удалите кластер и снова добавьте его в разделе <b>Инфраструктура &gt; Ресурсы &gt; Kubernetes</b> .
Контейнеры Terraform CLI создаются в предыдущем кластере, предыдущем пространстве имен или с предыдущими настройками среды выполнения даже после обновления учетной записи интеграции.	Клиент API-интерфейса Kubernetes, используемый системой vRealize Automation, кэшируется в течение 5 минут.	Чтобы изменения вступили в силу, может потребоваться до 5 минут.
При выполнении операции проверки или развертывания Terraform произошла ошибка; сообщается, что Kubeconfig недоступен.	Иногда эти ошибки возникают из-за того, что кластер недоступен из vRealize Automation. В других случаях учетные данные пользователя, маркеры или сертификаты являются недействительными.	Ошибка Kubeconfig может возникать по различным причинам. Для устранения неполадок может потребоваться обращение в службу технической поддержки.

## Добавление поддержки прокси-сервера

Чтобы кластер внешней среды выполнения Kubernetes мог подключаться через прокси-сервер, выполните следующие действия.

1. Войдите на сервер внешнего кластера Kubernetes.
2. Создайте пустую папку.
3. В новой папке добавьте в новый файл с именем Dockerfile следующие строки.

```
FROM projects.registry.vmware.com/vra/terraform:latest as final
ENV https_proxy=protocol://username:password@proxy_host:proxy_port
ENV http_proxy=protocol://username:password@proxy_host:proxy_port
ENV no_proxy=.local,.localdomain,localhost
```

4. Измените значения заполнителя так, чтобы переменные среды `https_proxy` и `http_proxy` содержали параметры прокси-сервера, используемого для доступа к Интернету.

Параметр *protocol* будет иметь значение `http` или `https`, в зависимости от того, какой протокол использует ваш прокси-сервер, и может отличаться от имени переменной среды `https_proxy` или `http_proxy`.

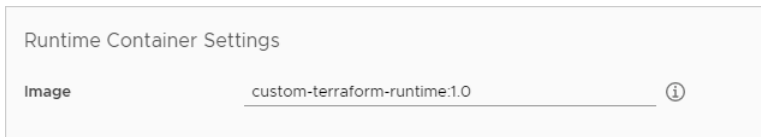
5. Сохраните файл Dockerfile и закройте его.

- Из пустой папки выполните следующую команду. В зависимости от прав своей учетной записи, возможно, потребуется выполнить команду в режиме `sudo`.

```
docker build --file Dockerfile --tag custom-terraform-runtime:1.0 .
```

Эта команда создает локальный образ Docker `custom-terraform-runtime:1.0`.

- В Cloud Assembly в разделе **Инфраструктура > Подключения > Интеграции** перейдите к своей интеграции среды выполнения Terraform.
- Создайте или измените параметры контейнера в среде выполнения для использования образа `custom-terraform-runtime:1.0`:



## Среда выполнения Terraform в Cloud Assembly без доступа к Интернету

Пользователи службы Cloud Assembly, которым необходимо проектировать и запускать интеграции Terraform без подключения к Интернету, могут настроить среду выполнения, используя данный пример.

**Примечание** Чтобы получить источник для создания образа, при настройке выполняется кратковременное подключение к Интернету. Если временное подключение невозможно, выполните эти действия вне отключенного сайта.

Этот процесс предполагает, что у вас есть [собственный реестр Docker](#) и вы можете получить доступ к его репозиториям без подключения к Интернету.

### Создание настраиваемого образа контейнера

- Создайте образ настраиваемого контейнера, который содержит двоичные файлы подключаемого модуля поставщика Terraform.

В следующем файле `Dockerfile` показан пример создания настраиваемого образа с поставщиком Terraform GCP.

Для загрузки базового образа `projects.registry.vmware.com/vra/terraform:latest` в файле `Dockerfile` требуется доступ через Интернет к реестру VMware Harbor на сайте `projects.registry.vmware.com`.

Настройки брандмауэра или прокси-сервера могут вызвать сбой сборки образа. Может потребоваться разрешить доступ к сайту `releases.hashicorp.com` для загрузки двоичных файлов подключаемого модуля поставщика Terraform. Тем не менее в качестве альтернативы можно использовать частный реестр для предоставления двоичных файлов подключаемого модуля.

```
FROM projects.registry.vmware.com/vra/terraform:latest as final


# Create provider plug-in directory
ARG plugins=/tmp/terraform.d/plugin-cache/linux_amd64
RUN mkdir -m 777 -p $plugins
```

```
# Download and unzip all required provider plug-ins from hashicorp to provider directory
RUN cd $plugins \
    && wget -q https://releases.hashicorp.com/terraform-provider-google/3.58.0/terraform-
provider-google_3.58.0_linux_amd64.zip \
    && unzip *.zip \
    && rm *.zip

# For "terraform init" configure terraform CLI to use provider plug-in directory and not
download from internet
ENV TF_CLI_ARGS_init="-plugin-dir=$plugins -get-plugins=false"
```

2. Выполните сборку, поставьте теги и отправьте настраиваемый образ контейнера в свой репозиторий Docker на отключенном сайте.
3. В Cloud Assembly на отключенном сайте в разделе **Инфраструктура > Подключения > Интеграции** перейдите к интеграции среды выполнения Terraform.
4. Создайте или отредактируйте параметры контейнера в среде выполнения, чтобы добавить репозиторий для настраиваемого образа контейнера. Созданный настраиваемый образ контейнера в примере имеет имя `registry.ourcompany.com/project1/image1:latest`.

Runtime Container Settings

Image	<u>registry.ourcompany.com/project1/image1:latest</u>	
-------	---	---

### Локальное размещение интерфейса командной строки Terraform

1. Загрузите двоичные файлы интерфейса командной строки Terraform.
2. Отправьте двоичные файлы интерфейса командной строки Terraform на свой локальный веб-сервер или FTP-сервер.
3. В службе Cloud Assembly выберите **Инфраструктура > Настроить > Версии Terraform**.
4. Создайте или отредактируйте версию Terraform, чтобы она содержала URL-адрес двоичных файлов интерфейса командной строки Terraform, размещенных на вашем локальном сервере.
5. Если для локального веб-сервера или FTP-сервера требуется проверка подлинности, выберите **Базовая проверка подлинности** и введите имя пользователя и пароль для доступа к серверу.

Чтобы изменить тип проверки подлинности, требуется роль администратора облака в Cloud Assembly.

0.12.29 [DELETE](#)

**Version \*** 0.12.29 ⓘ

**Description**

**Enabled** ☒ ⓘ

**URL \*** http://host1.ourcompany.com:8080/tf/0.12.29/terraform\_0.12.29\_linux\_amd64.zip ⓘ

**Authentication type \*** ☒ No authentication ☐ Basic authentication ⓘ

**SHA256 Checksum \*** 872245d9c6302b24dc0d98a1e010aef1e4ef60865a2d1f60102c8ad03e9d5a1d ⓘ

## Проектирование и развертывание конфигураций Terraform

При наличии среды выполнения можно добавить файлы конфигурации в Git, спроектировать для них облачные шаблоны и развернуть.

Сведения о начале работы см. в [Подготовка Cloud Assembly для конфигураций Terraform](#).

### Устранение неполадок

В ходе развертывания откройте развертывание в Cloud Assembly. На вкладке "Журнал" найдите события Terraform и нажмите **Показать журналы** справа. Если локальный поставщик Terraform работает, в журнале появляются следующие сообщения.

```
Initializing provider plugins
```

```
Terraform has been successfully initialized
```

Чтобы получить более надежный журнал, можно вручную отредактировать код облачного шаблона и добавить `TF_LOG: DEBUG`, как показано в примере ниже.

```
resources:
  terraform:
    type: Cloud.Terraform.Configuration
    properties:
      providers:
        - name: google
          # List of available cloud zones: gcp/us-west1
          cloudZone: gcp/us-west1
      environment:
        # Configure terraform CLI debug log settings
        TF_LOG: DEBUG
    terraformVersion: 0.12.29
    configurationSource:
      repositoryId: fc569ef7-f013-4489-9673-6909a2791071
      commitId: 3e00279a843a6711f7857929144164ef399c7421
      sourceDirectory: gcp-simple
```

## Создание собственного базового образа

VMware иногда обновляет базовый образ на `projects.registry.vmware.com/vra/terraform:latest`. Он может устарел и может содержать уязвимости.

Чтобы создать собственный базовый образ, используйте следующий файл Dockerfile.

```
FROM alpine:latest as final
RUN apk add --no-cache git wget curl openssh
```

## Подготовка Cloud Assembly для конфигураций Terraform

Перед добавлением конфигурации Terraform в шаблон Cloud Assembly настройте репозиторий управления версиями и интегрируйте его.

1. [Необходимые условия](#)
2. [Хранение файлов конфигурации Terraform в репозитории управления версиями](#)
3. [Включить сопоставление облачных зон](#)
4. [Интеграция репозитория с Cloud Assembly](#)

### Необходимые условия

Чтобы запускать операции Terraform в локальной версии vRealize Automation, необходима интеграция среды выполнения Terraform. См. раздел [Подготовка среды выполнения Cloud Assembly Terraform](#).

### Хранение файлов конфигурации Terraform в репозитории управления версиями

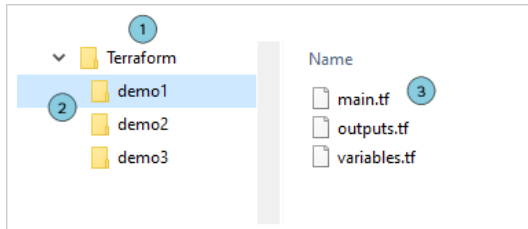
Cloud Assembly поддерживает следующие репозитории управления версиями для конфигураций Terraform.

- Облако GitHub, локальный GitHub Enterprise
- Облако GitLab, локальный GitLab Enterprise
- Локальный Bitbucket

В репозитории управления версиями создайте каталог по умолчанию с одним уровнем подкаталогов, каждый из которых содержит файлы конфигурации Terraform. Для каждой конфигурации Terraform создайте один подкаталог.

1. Каталог по умолчанию
2. Один уровень подкаталогов
3. Файлы конфигурации Terraform, готовые к развертыванию

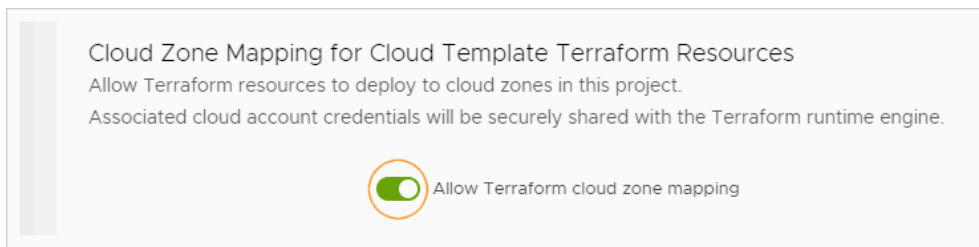
Не добавляйте файл состояния Terraform с файлами конфигурации. Если имеется `terraform.tfstate`, во время развертывания возникают ошибки.



## Включить сопоставление облачных зон

Если предполагается развертывание в облачной учетной записи, для модуля среды выполнения Terraform требуются следующие учетные данные облачной зоны.

На вкладке **Предоставление** проекта включите **Разрешить сопоставление облачной зоны Terraform**.



Несмотря на то что учетные данные передаются в защищенном режиме, для обеспечения дополнительного уровня безопасности оставьте данный параметр деактивированным, если пользователям проекта не нужно выполнять развертывание в облачной учетной записи.

## Интеграция репозитория с Cloud Assembly

В Cloud Assembly выберите **Инфраструктура > Подключения > Интеграции**.

Добавьте интеграцию в тот репозиторий, где были сохранены конфигурации Terraform: GitHub, GitLab или Bitbucket.

При добавлении проекта в интеграцию выберите тип **Конфигурации Terraform**, укажите репозиторий и ветвь.

**Папка** — это каталог предыдущей структуры по умолчанию.

Add Repository: testProject

Configure a repository to be used for this project.

Type *	Terraform Configurations	ⓘ
Repository *	parnassusdemo/repository1	ⓘ
Branch *	master	
Folder	/Terraform	

## Проектирование конфигураций Terraform в Cloud Assembly

Имея репозиторий и файлы конфигурации Terraform, можно разработать для них шаблон Cloud Assembly.

1. [Необходимые условия](#)
2. [Включение версий среды выполнения Terraform](#)
3. [Добавление ресурсов Terraform в проект](#)
4. [Развертывание облачного шаблона](#)

### Необходимые условия

Настройте и интегрируйте репозиторий управления версиями. См. раздел [Подготовка Cloud Assembly для конфигураций Terraform](#).

### Включение версий среды выполнения Terraform

При развертывании конфигураций Terraform можно определить версии среды выполнения Terraform, доступные для пользователей. Следует отметить, что конфигурации Terraform также могут включать в себя встроенные ограничения версий.

Чтобы создать список допустимых версий, выберите **Инфраструктура > Настроить > Версии Terraform**.

### Добавление ресурсов Terraform в проект

Создайте облачный шаблон, который включает в себя конфигурации Terraform.

1. В Cloud Assembly выберите **Проектирование > Облачные шаблоны** и щелкните **Создать из > Terraform**.

Появляется мастер настройки Terraform.

2. Следуйте подсказкам.

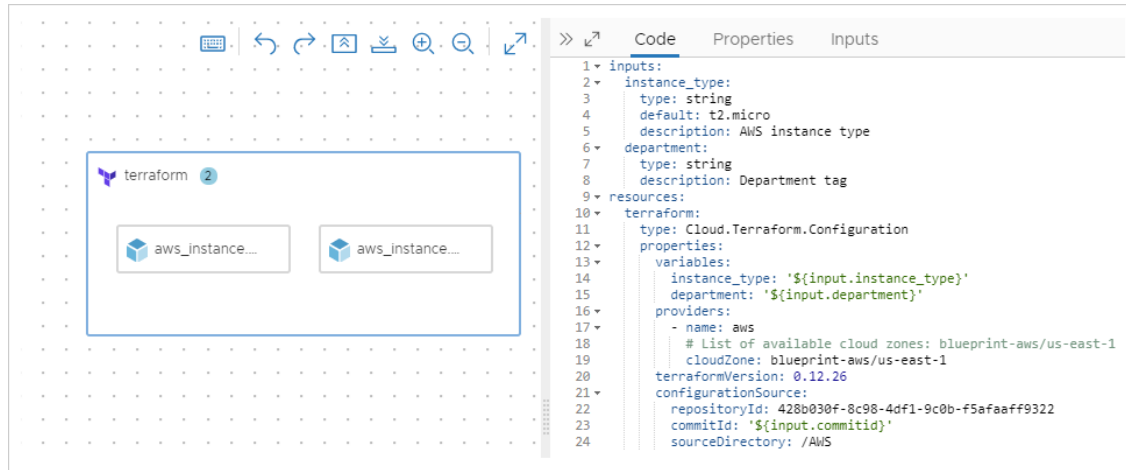
Страница мастера	Настройка	Значение
Создать облачный шаблон	Имя	Назначьте проекту уникальное имя.
	Описание	Объясните, для чего предназначен проект.
	Проект	Выберите проект, содержащий интеграцию репозитория, в котором хранится конфигурация Terraform.
Источник конфигурации	Репозиторий	Выберите интегрированный репозиторий, в котором сохранена конфигурация Terraform.

Страница мастера	Настройка	Значение
Завершение настройки	Фрагмент Commit	<p>Выберите фрагмент Commit из репозитория или оставьте поле пустым, чтобы использовать конфигурацию Terraform из заголовка head репозитория.</p> <p>Ограничение Bitbucket. Количество доступных для выбора фрагментов Commit может быть уменьшено из-за конфигурации сервера репозитория Bitbucket.</p>
	Исходный каталог	Выберите подкаталог в созданной структуре репозитория. Выше приведены примеры подкаталогов demo1, demo2 и demo3.
	Репозиторий	Проверьте правильность выбора репозитория.
	Исходный каталог	Проверьте правильность выбора каталога.
	Версия Terraform	Выберите версию среды выполнения Terraform для запуска при развертывании конфигурации Terraform.
	Поставщики	<p>Если в конфигурацию Terraform добавлен блок поставщика, проверьте его и облачную зону, в которой будет развернут этот облачный шаблон.</p> <p>Отсутствие поставщика не является проблемой. По окончании работы мастера нужно просто изменить поставщика и облачную зону в свойствах шаблона, чтобы добавить или изменить целевой объект развертывания.</p>
	Переменные	Выберите конфиденциальные значения шифрования, например пароли.
	Выходные данные	Проверьте выходные данные из конфигурации Terraform, преобразуемые в выражения, на которые код проекта может далее ссылаться.

### 3. Щелкните **Создать**.

Ресурс Terraform появляется на холсте облачного шаблона с кодом Cloud Assembly, отражающим развертываемую конфигурацию Terraform.





При необходимости в облачный шаблон можно добавить другие ресурсы Cloud Assembly, чтобы объединить код Terraform и код, отличный от Terraform, в гибридном проекте.

**Примечание** Обновление конфигураций Terraform в репозитории не синхронизирует изменения с облачным шаблоном. Автоматическая синхронизация может привести к угрозам безопасности, например из-за вновь добавленных конфиденциальных переменных.

Чтобы выявить изменения конфигурации Terraform, перезапустите мастер, выберите новый фрагмент Commit и определите все новые конфиденциальные переменные.

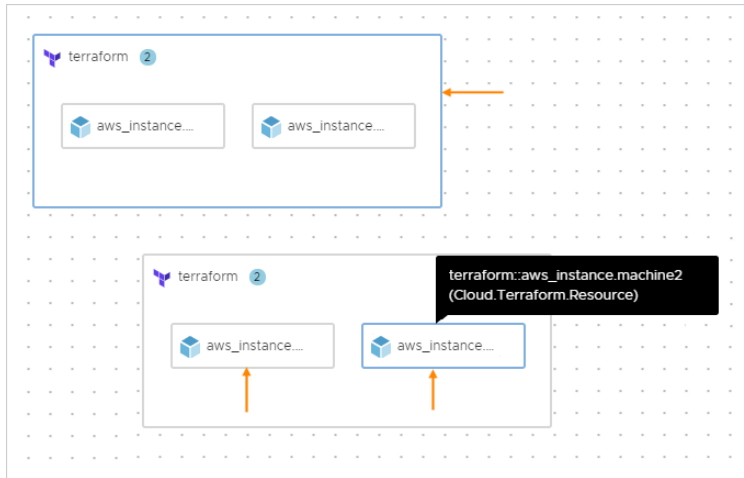
## Развертывание облачного шаблона

При развертывании облачного шаблона на вкладке **Журнал** можно развернуть событие, например этап выделения или создания, чтобы проверить журнал сообщений в Terraform CLI.

Утверждения. В дополнение к ожидаемым этапам Terraform, таким как ПЛАНИРОВАНИЕ, ВЫДЕЛЕНИЕ или СОЗДАНИЕ, Cloud Assembly реализует управление с помощью этапа утверждения. Дополнительные сведения об утверждении запросов см. в разделе [Настройка политик утверждения Service Broker](#).

Timestamp	Status	Resource type	Resource name	Details
Aug 3, 202...	PLAN_FINISHED	Cloud.Terraform.Configurati...	terraform	Creating 2 Terraform resources, updating 0 Terraform resources, deleting 0 Terraform resources
Aug 3, 202...	PLAN_IN_PROGRESS	Cloud.Terraform.Configurati...	terraform	<a href="#">Hide Logs</a>
<pre> 2:24:23 PM * provider.random: version = "~&gt; 2.3" 2:24:23 PM 2:24:23 PM Terraform has been successfully initialized! 2:24:28 PM Refreshing Terraform state in-memory prior to plan... 2:24:28 PM The refreshed state will be used to calculate this plan, but will not be 2:24:28 PM persisted to local or remote state storage. </pre>				
<a href="#">View as plain text</a>				
Aug 3, 202...	INITIALIZATION_FINISH...			
Aug 3, 202...	INITIALIZATION_IN_PRO...			

После развертывания появляется внешний ресурс, который представляет общий компонент Terraform с дочерними ресурсами внутри для отдельных компонентов, созданных Terraform. Родительский ресурс Terraform контролирует жизненный цикл дочерних ресурсов.



## Использование секретного свойства Cloud Assembly в конфигурации Terraform

Секретные зашифрованные значения можно применять в конфигурациях Terraform, которые добавляются в проекты облачных шаблонов Cloud Assembly.

1. В репозитории Git добавьте исходный файл конфигурации Terraform, который ссылается на секретные свойства как на переменные.

В этом примере источника конфигурации Terraform и ключи API-интерфейса и приложения являются секретными переменными.

```
variable "datadog_api_key" {
  description = "Datadog API Key"
}

variable "datadog_app_key" {
  description = "Datadog App Key"
}

provider "datadog" {
  api_key = "${var.datadog_api_key}"
  app_key = "${var.datadog_app_key}"
}

# Create a new monitor
resource "datadog_monitor" "default" {
  # ...
}

# Create a new timeboard
resource "datadog_timeboard" "default" {
  # ...
}
```

- В Cloud Assembly выберите **Инфраструктура > Администрирование > Секретные элементы** и введите значения секретных свойств.

Добавьте для секретных элементов имена и соответствующие значения. Для имен проще всего ввести то же имя, что и имя переменной из источника Terraform.

Дополнительные сведения можно найти в [Секретные свойства Cloud Assembly](#).

Name	Project	Value
datadog_api_key	Terraform	*****
datadog_app_key	Terraform	*****

- В Cloud Assembly импортируйте конфигурацию Terraform для использования в облачном шаблоне. Выберите **Проектирование > Облачные шаблоны** и щелкните **Создать из > Terraform**.

**Примечание** Несмотря на то что переменные отображаются для выбора на последней странице мастера, секретные переменные не нужно указывать как конфиденциальные. Секретные переменные Cloud Assembly уже будут зашифрованы, поэтому для них не требуется шифрование, выполняемое мастером.

Дополнительные сведения можно найти в [Проектирование конфигураций Terraform в Cloud Assembly](#).

Пример облачного шаблона должен иметь приблизительно следующий вид:

```
inputs:
  datadog_api_key:
    type: string
    description: Datadog API Key
  datadog_app_key:
    type: string
    description: Datadog App Key
resources:
  terraform:
    type: Cloud.Terraform.Configuration
    properties:
      variables:
        datadog_api_key: '${input.datadog_api_key}'
        datadog_app_key: '${input.datadog_app_key}'
      providers: []
      terraformVersion: 0.12.29
      configurationSource:
        repositoryId: 0fbf8f5e-54e1-4da3-9508-2b701gf25f51
        commitId: ed12424b249aa50439kr1c268942a4616bd751b6
        sourceDirectory: datadog
```

4. В редакторе кода для секретных значений вручную измените `input` на `secret`, как показано ниже.

```
terraform:
  type: Cloud.Terraform.Configuration
  properties:
    variables:
      datadog_api_key: '${secret.datadog_api_key}'
      datadog_app_key: '${secret.datadog_app_key}'
```

5. В разделе `inputs:` кода удалите входные записи, которые были заменены привязками на секретные свойства.

## Дополнительные сведения о конфигурациях Terraform в vRealize Automation

При внедрении конфигураций Terraform в качестве ресурса в vRealize Automation необходимо учитывать определенные ограничения и возможные проблемы.

### Ограничения конфигураций Terraform

- При проверке проекта с конфигурациями Terraform с помощью кнопки ТЕСТ можно проверить синтаксис Cloud Assembly, но нельзя проверить синтаксис кода Terraform.  
  
Кроме того, при нажатии кнопки ТЕСТ не проверяются идентификаторы фрагментов `commit`, связанные с конфигурациями Terraform.
- Чтобы клонировать облачный шаблон, включающий в себя конфигурации Terraform, в другом проекте, используйте следующее решение.
  - а) В новом проекте на вкладке **Интеграции** скопируйте `repositoryId` для интеграции.
  - б) Откройте шаблон-клон. В редакторе кода замените `repositoryId` скопированным значением.
- В репозитории управления версиями не включайте файл состояния Terraform с файлами конфигурации. Если имеется `terraform.tfstate`, во время развертывания возникают ошибки.

### Поддерживаемые действия по регулярному обслуживанию для родительского ресурса Terraform

Для родительского ресурса Terraform можно просмотреть или обновить файл состояния Terraform. Дополнительные сведения о действиях с файлами состояния см. в полном списке действий в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

### Поддерживаемые действия по регулярному обслуживанию для дочерних ресурсов

После развертывания конфигураций Terraform может потребоваться до 20 минут, чтобы действие по регулярному обслуживанию стало доступным для дочерних ресурсов.

Для дочерних ресурсов в конфигурации Terraform поддерживается только следующий набор действий по регулярному обслуживанию. Дополнительные сведения о действиях см. в полном списке действий в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

Поставщик	Тип ресурса Terraform	Поддерживаемые действия по регулярному обслуживанию
AWS	aws_instance	Включение
		Выключение
		Перезагрузить
		Сброс
Azure	azurerm_virtual_machine	Включение
		Выключение
		Перезапуск
		Приостановка
vSphere	vsphere_virtual_machine	Включение
		Выключение
		Перезагрузить
		Сброс
		Завершение работы
		Приостановка
		Создание моментальных снимков
		Удаление моментального снимка
		Откат моментальных снимков
GCP	google_compute_instance	Включение
		Выключение
		Создание моментальных снимков
		Удаление моментального снимка

## Устранение проблем с доступностью действий по регулярному обслуживанию

Если «готовые к использованию» (OOTB) действия по регулярному обслуживанию отсутствуют или деактивированы, может потребоваться устранение неполадок.

Проблема	Причина	Разрешение
Для ресурса Terraform в меню «Действия» отсутствует ожидаемое действие по регулярному обслуживанию, готовое к использованию.	Возможно, это действие не поддерживается для данного типа поставщика и ресурса, как указано в предыдущем списке. Кроме того, для отображения действия может потребоваться до 20 минут, в течение которых выполняется обнаружение и кеширование ресурсов.	Проверьте тип поставщика и ресурса в проекте. Подождите около 20 минут до завершения сбора данных.
Для ресурса Terraform отсутствует ожидаемое действие по регулярному обслуживанию даже по истечении 20 минут, которые могли потребоваться для сбора данных.	Действие не появляется из-за проблемы с обнаружением ресурсов. Одной из проблем является ситуация, когда ресурс случайно создан в облачной зоне вне рамок проекта. Например, проект включает в себя только облачную учетную запись и облачную зону в регионе us-east-1, но конфигурация Terraform содержит блок поставщика для us-west-1, что не было изменено при проектировании. Другая возможная проблема заключается в том, что процесс сбора данных не работает.	Проверьте соответствие облачных зон рабочего проекта и облачных зон проекта шаблона. Выберите <b>Инфраструктура &gt; Подключения &gt; Облачные учетные записи</b> и проверьте состояние сбора данных, а также время последней успешной операции сбора данных для этой облачной учетной записи.
Несмотря на отсутствие очевидных проблем с состоянием ресурса и сбором данных, действие по регулярному обслуживанию деактивировано (выделено серым).	Иногда возникают ошибки с временными параметрами и сбои при сборе данных.	Проблему необходимо устранить в течение 20 минут.
Ошибочно деактивировано действие по регулярному обслуживанию, которое должно быть активировано в соответствии с состоянием ресурса. Например, состояние «Выключение» активировано, а «Включение» деактивировано, несмотря на то, что ресурс был выключен с помощью интерфейса поставщика.	Период сбора данных может вызывать временное несоответствие. Если изменить состояние питания вне vRealize Automation, потребуется определенное время на ввод изменений в действие.	Подождите 20 минут.

## Использование настраиваемых поставщиков Terraform в vRealize Automation

Если вы хотите использовать настраиваемый поставщик Terraform, выполните следующие действия.

В репозитории системы управления версиями Git в каталоге Terraform, содержащем файл main.tf, добавьте следующую структуру подкаталогов и настраиваемый ZIP-файл поставщика Terraform.

```
terraform.d/plugins/<HOSTNAME>/<NAMESPACE>/<TYPE>/terraform-provider-
<TYPE_VERSION_TARGET>.zip
```

Например, при загрузке файла [azurerm version 3.12.0](#) необходимо создать следующую структуру.

```
terraform.d/plugins/registry.terraform.io/hashicorp/azurerm/terraform-provider-
azurerm_3.12.0_linux_amd64.zip
```

## Настраиваемые типы ресурсов для облачных шаблонов Cloud Assembly

При создании облачного шаблона в Cloud Assembly палитра типов ресурсов включает в себя типы ресурсов для поддерживаемой облачной учетной записи и конечных точек интеграции. В некоторых случаях необходимо создать облачные шаблоны на основе расширенного списка типов ресурсов. Можно создавать настраиваемые типы ресурсов, добавлять их на холст проекта, а также создавать облачные шаблоны, которые поддерживают потребности проектов и развертываний.

### Имя настраиваемого ресурса и тип ресурса

Имя настраиваемого ресурса определяет настраиваемый ресурс на палитре типов ресурсов в облачном шаблоне.

Тип ресурса настраиваемого ресурса должен начинаться с **Custom.** и каждый тип ресурса должны быть уникальными. Например, можно задать **Custom.ADUser** как тип ресурса для настраиваемого ресурса, который добавляет пользователей Active Directory. Несмотря на то, что включение **Custom.** не проверяется в этом текстовое поле; строка будет добавлена автоматически, если она была случайно удалена.

### Настраиваемые ресурсы действий расширяемости

Настраиваемые типы ресурсов позволяют использовать действия расширяемости в облачных шаблонах для создания сложных приложений. Например, можно использовать встроенную интеграцию действий расширяемости с Amazon Web Services и Microsoft Azure для простой интеграции с соответствующими службами. Чтобы создать настраиваемые ресурсы действий расширяемости, щелкните параметр **На основе** в редакторе настраиваемых ресурсов и выберите **Определенная пользователем схема ABX**.

### Действия жизненного цикла для настраиваемых ресурсов действий расширяемости

При использовании действия расширяемости для настраиваемого ресурса можно задать следующие действия жизненного цикла.

- **Создание.** Это действие расширяемости вызывается при запуске развертывания.
- **Чтение.** Это действие расширяемости используется для получения последнего состояния развернутого ресурса.
- **Обновление.** Это действие расширяемости вызывается при обновлении свойства облачного шаблона. Это действие запускается только в том случае, когда свойство не имеет отметки `recreateOnUpdate`.
- **Удаление.** Это действие расширяемости вызывается при удалении развертывания.

Эти действия жизненного цикла можно выбрать вручную из существующих действий расширяемости или создать автоматически, выбрав **Сгенерировать действия**. При выборе параметра **Сгенерировать действия** необходимо указать проект, в котором будет сгенерировано новое действие расширяемости.

---

**Примечание** Действия расширяемости, связанные с действиями жизненного цикла, можно изменить, щелкнув **Открыть** рядом с конкретным действием.

---

## Настраиваемые ресурсы vRealize Orchestrator

Каждый настраиваемый ресурс vRealize Orchestrator основан на типе иерархии SDK и создается с помощью рабочего процесса vRealize Orchestrator, который является экземпляром требуемого типа SDK. Для создания типов настраиваемых ресурсов простые типы, такие как `Properties`, `Date`, `string` и `number`, не поддерживаются.

---

**Примечание** Типы объектов SDK можно отличить от других типов свойств двоеточием (:), используемым для отделения имени подключаемого модуля от имени типа. Например, `AD:UserGroup` является типом объекта SDK, который используется для управления группами пользователей Active Directory.

---

Можно использовать встроенные рабочие процессы в vRealize Orchestrator или создать свои собственные. Использование vRealize Orchestrator для создания рабочих процессов «все как услуга» или «любой ресурс как услуга» означает, что можно создать облачный шаблон, который добавляет пользователя Active Directory к компьютерам во время развертывания, или добавить настраиваемую подсистему балансировки нагрузки F5 в развертывание. Чтобы создать настраиваемые ресурсы vRealize Orchestrator, щелкните параметр **На основе** в редакторе настраиваемых ресурсов и выберите **Иерархия vRO**.

## Внешний тип настраиваемого ресурса vRealize Orchestrator

Свойство внешнего типа определяет тип настраиваемого ресурса vRealize Orchestrator. При выборе рабочего процесса «Создание» в типе настраиваемого ресурса в Cloud Assembly под ним появляется раскрывающееся меню внешних типов. Раскрывающееся меню включает в себя свойства внешнего типа, которые выбираются из выходных параметров рабочего процесса vRealize Orchestrator. Выбранные выходные свойства рабочего процесса, включенные в раскрывающееся меню, должны быть типами объектов SDK, которые не являются массивами, например `VC:VirtualMachine` или `AD:UserGroup`.

---

**Примечание** При создании настраиваемых рабочих процессов, использующих подключаемый модуль динамического типа, убедитесь, что их переменные созданы с помощью метода `DynamicTypesManager.GetObject()`.

---

При определении типов настраиваемых ресурсов также определяется область доступности для выбранного внешнего типа. Выбранный внешний тип может быть следующим.

- Общедоступен для проектов.
- Доступен только для выбранного проекта.



Для каждой определенной области может быть задан только один тип настраиваемого ресурса с определенным значением внешнего типа. Например, при создании настраиваемого ресурса в проекте, в котором используется `VC:VirtualMachine` как внешний тип, нельзя создать другой настраиваемый ресурс для того же проекта, в котором применяется тот же внешний тип. Кроме того, нельзя создать два общедоступных настраиваемых ресурса, использующих один и тот же внешний тип.

## Проверка действия жизненного цикла vRealize Orchestrator

При добавлении рабочих процессов «Создание», «Удаление» и «Обновление» в виде действий жизненного цикла для настраиваемого ресурса Cloud Assembly проверяет, что выбранные рабочие процессы имеют корректные определения входных и выходных свойств.

- Рабочий процесс «Создание» должен иметь выходной параметр, который является типом объекта SDK, например `SSH:Host` или `SQL:Database`. Если выбранный рабочий процесс не прошел проверку, нельзя добавить рабочие процессы «Обновление» и «Удаление» или сохранить изменения в настраиваемом ресурсе.
- Рабочий процесс «Удаление» должен иметь входной параметр, являющийся типом объекта SDK, который соответствует внешнему типу настраиваемого ресурса.
- Рабочий процесс «Обновление» должен иметь как входной, так и выходной параметр, являющийся типом объекта SDK, который соответствует внешнему типу настраиваемого ресурса.

## Схема свойств настраиваемого ресурса

Чтобы изменить и просмотреть схему свойств настраиваемого ресурса, перейдите на вкладку **Свойства**. Схема содержит имя, тип данных, тип свойства и (при наличии) описание данного свойства. Схема также определяет, является ли свойство в облачном шаблоне обязательным или необязательным.

---

**Примечание** Для схемы свойств настраиваемых ресурсов действий расширяемости в облачном шаблоне обязательными являются все свойства.

---

При добавлении рабочих процессов vRealize Orchestrator в настраиваемый ресурс их входные и выходные параметры добавляются как свойства. Схему свойств настраиваемых ресурсов действий расширяемости необходимо создать вручную на вкладке **Свойства**. На этой вкладке также можно изменять и форматировать свойства настраиваемых ресурсов vRealize Orchestrator или действий с поддержкой расширяемости. Например, можно изменить отображаемое имя заданного свойства или добавить ограничения.

---

**Примечание** При добавлении ограничений в раздел элементов полей массива или раздел свойств полей объектов в схеме свойств убедитесь, что эти ограничения проверены, так как некорректно примененные ограничения могут вызвать проблемы с настраиваемым ресурсом. Например, при добавлении максимального ограничения к массиву чисел необходимо убедиться, что это ограничение не изменяет значение свойства по умолчанию.

---

Чтобы изменить схему свойств для настраиваемых ресурсов, перейдите на вкладку **Свойства** и используйте вкладку **Код** или **Форма**.

- **Код.** Изменение схемы свойств с помощью содержимого YAML.
- **Форма.** Чтобы создать новое свойство, щелкните **Новое свойство** и задайте его имя, отображаемое имя, описание, тип свойства и значение по умолчанию. Кроме того, свойства, которые не нужны и не являются вычисляемыми, можно скрыть в схеме, нажав кнопку **Удалить свойство**.

## Настраиваемые формы запросов операций по регулярному обслуживанию

Форму запроса операций по регулярному обслуживанию, включенных в настраиваемый ресурс, можно упростить путем добавления и изменения различных типов свойств ресурсов.

Значение входного параметра в форме запроса можно привязать к внешнему источнику, например к действию vRealize Orchestrator, которое извлекает имя развертывания или имя проекта. Кроме того, можно привязать значение определенного входного параметра к вычисленному значению двух других текстовых полей, включенных в ту же форму запроса.

---

**Примечание** Эта функция доступна как для настраиваемых ресурсов, так и для действий с ресурсами. Значение входных свойств формы запроса можно задать на вкладке **Значения** страницы **Параметры запроса** редактора настраиваемых ресурсов или действий с ресурсами.

---

## Проверка формы запроса операций по регулярному обслуживанию

Форму запроса операций по регулярному обслуживанию можно проверить путем добавления внешней проверки. Во время внешней проверки пользователь не может подать форму запроса до тех пор, пока не будут удовлетворены параметры проверки. Внешнюю проверку можно добавить на вкладке **Проверки** страницы **Запрос параметров** настраиваемого ресурса или в редакторе действий ресурса. После выбора вкладки можно перетащить элемент **Проверка с помощью оркестратора** на холст и добавить действие vRealize Orchestrator, которое необходимо использовать для проверки.

Например, можно создать настраиваемый ресурс, который включает в себя операцию по регулярному обслуживанию для изменения пароля пользователя. В таком случае можно добавить действие vRealize Orchestrator с входными параметрами newPassword и confirmPassword, в которых используется тип SecureString.

---

**Примечание** Это пример сценария для проверки пароля пользователя. Вы можете выбрать другой сценарий.

---

```
if (newPassword != confirmPassword) {
    return 'passwords are different';
}
if (newPassword.length < 7) {
    return 'password must be at least 10 symbols';
}
return null;
```

## Создание шаблона Cloud Assembly, обеспечивающего добавление пользователей в Active Directory

В дополнение к ресурсам облачного шаблона Cloud Assembly, которые используются при создании облачных шаблонов, также можно создать собственные настраиваемые ресурсы.

Настраиваемые ресурсы являются объектами vRealize Orchestrator или действий расширяемости, которыми можно управлять в vRealize Automation с помощью действий жизненного цикла, определенных в настраиваемом ресурсе. Служба облачных шаблонов автоматически вызывает соответствующие рабочие процессы vRealize Orchestrator или действия расширяемости при инициировании операции, связанной с определенным действием жизненного цикла. Для расширения функциональных возможностей типа ресурса также можно выбрать рабочие процессы vRealize Orchestrator или действия расширяемости, которые можно использовать как операции по регулярному обслуживанию.

В данном примере используются встроенные рабочие процессы из библиотеки vRealize Orchestrator. В нем содержатся рекомендованные значения или строки для демонстрации выполнения процесса. Их можно изменить в соответствии с требованиями среды.

Для справочных целей в данном примере используется проект с именем **DevOpsTesting**. Вместо этого примера проекта можно использовать любой проект в среде.

### Необходимые условия

- Убедитесь, что настроена интеграция vRealize Orchestrator. См. раздел [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).
- Убедитесь, что рабочие процессы, используемые для действий «создать», «обновить», «уничтожить» и «регулярное обслуживание», существуют в vRealize Orchestrator и успешно выполняются.
- В vRealize Orchestrator найдите тип ресурса, используемый рабочими процессами. Все рабочие процессы, включенные в этот настраиваемый ресурс, должны использовать один и тот же тип ресурса. Тип ресурса в данном примере — AD:User. Дополнительные сведения о проверке типа ресурса см. в разделе [Настраиваемые типы ресурсов для облачных шаблонов Cloud Assembly](#).
- С помощью встроенных рабочих процессов Active Directory в интеграции vRealize Orchestrator настройте сервер Active Directory.
- Вам необходимо знать, как настроить и развернуть облачный шаблон компьютера.

## Процедура

1. Создание настраиваемого ресурса Active Directory для добавления пользователя в группу.

На этом этапе настраиваемый ресурс добавляется на холст проекта облачного шаблона как тип ресурса.

- а) В Cloud Assembly выберите **Проектирование > Настраиваемые ресурсы** и щелкните **Создать настраиваемый ресурс**.
- б) Введите следующие значения.

Обратите внимание, что, за исключением имен рабочих процессов, приведенные значения являются только примерами.

Настройка	Пример значения
Имя	<b>Пользователь Active Directory</b> Это имя, которое отображается на палитре типов ресурсов для облачного шаблона.
Тип ресурса	<b>Custom.ADUser</b> Тип ресурса должен начинаться с <b>Custom.</b> и каждый тип ресурса должны быть уникальными. Несмотря на то, что включение <b>Custom.</b> не проверяется в этом текстовое поле; строка будет добавлена автоматически, если она была случайно удалена. Этот тип ресурса добавляется на палитру типов ресурсов, чтобы его можно было использовать в облачном шаблоне.

- в) Чтобы включить этот тип ресурса в списке типов ресурсов облачного шаблона, убедитесь, что параметр **Активировать** включен.
- г) Нажмите **область**, чтобы сделать тип ресурса доступным для любого проекта.
- д) В разделе **На основе** в качестве поставщика действий жизненного цикла должна быть выбрана **Иерархия vRO**.

- е) Выберите рабочие процессы, которые определяют ресурс и действия по регулярному обслуживанию.

**Примечание** Для выбранных рабочих процессов по регулярному обслуживанию необходим входной параметр того же типа, что и внешний тип. Входные данные внешнего типа не отображаются в настраиваемой форме по регулярному обслуживанию, которую запрашивает пользователь, так как она автоматически привязывается к настраиваемому ресурсу.

Настройка	Пример значения
Действия жизненного цикла — создание	<p>Выберите рабочий процесс <b>Создание пользователя с паролем в организационном подразделении</b>.</p> <p>Если у вас есть несколько интеграций vRealize Orchestrator, выберите рабочий процесс в экземпляре интеграции, который используется для запуска этих настраиваемых ресурсов.</p> <p>После выбора рабочего процесса появится раскрывающееся меню внешних типов, в котором будет автоматически выбран вариант <code>AD:User</code>.</p> <hr/> <p><b>Примечание</b> Тип внешнего источника можно использовать только один раз, если применяется общий доступ, и только один раз в каждом проекте. В этом примере использования настраиваемый ресурс будет предоставляться для всех проектов. Это означает, что <code>AD:User</code> нельзя использовать для любых других типов ресурсов во всех проектах. Если у вас есть другие рабочие процессы, для которых требуется тип <code>AD:User</code>, необходимо создать отдельные настраиваемые ресурсы для каждого проекта.</p>
Действия жизненного цикла — уничтожение	<p>Выберите рабочий процесс <b>Уничтожение пользователя</b>.</p>
Дополнительные действия	<p>Выберите рабочий процесс <b>Изменения пароля пользователя</b>.</p> <p>В окне <b>Добавить действие</b> добавьте имя действия, например <code>password_change</code>, и нажмите <b>Добавить</b>.</p> <p>Чтобы изменить форму запроса действия, которую пользователь заполняет при запросе, щелкните значок в столбце <b>Параметры запроса</b>.</p> <hr/> <p><b>Примечание</b> Для дополнительных рабочих процессов действий убедитесь, что в рабочем процессе есть входной параметр того же типа, что и внешний тип.</p>

В этом примере нет соответствующего приложения рабочего процесса обновления. Типичным примером рабочего процесса обновления, который вносит изменения в подготовленный настраиваемый ресурс, является горизонтальное или вертикальное масштабирование развертывания.

- ж) Просмотрите значения ключа и типа схемы на вкладке **Свойства**, чтобы уточнить входные данные рабочего процесса, которые можно настроить в облачном шаблоне.

В схеме перечислены обязательные и необязательные входные значения, определенные в рабочем процессе. Обязательные входные значения включены в YAML облачного шаблона.

В рабочем процессе «Создание пользователя» обязательными входными значениями являются `accountName`, `displayName` и `ouContainer`. Другие свойства схемы не являются обязательными. Схему также можно использовать, чтобы определить, где необходимо создать привязки со значениями других полей, рабочими процессами или действиями. В данном примере использования привязки отсутствуют.

- з) Чтобы завершить процедуру создания настраиваемого ресурса, щелкните **Создать**.
2. Создайте облачный шаблон, который добавляет пользователя на компьютер при его развертывании.
    - а) Выберите **Проектирование > Облачные шаблоны** и нажмите **Создать из > Пустой холст**.
    - б) Назовите облачный шаблон **Компьютер с пользователем AD**.
    - в) Выберите проект **DevOpsTesting** и нажмите **Создать**.
    - г) Добавление и настройка компьютера vSphere.
    - д) Из списка настраиваемых ресурсов, расположенного слева на странице проектирования шаблона облачных служб, перетащите на холст тип ресурса **Пользователь AD**.

---

**Примечание** Чтобы выбрать настраиваемый ресурс, можно прокрутить список вниз и выбрать ресурс на левой панели или найти ресурс в текстовом поле **Поиск типов ресурсов**. Если настраиваемый ресурс отсутствует, нажмите кнопку «Обновить» рядом с текстовым полем **Поиск типов ресурсов**.

---

- е) В правой части окна измените код YAML, чтобы добавить обязательные входные значения и пароль.

Добавьте раздел `inputs` в код, чтобы пользователи могли указывать имена добавляемых пользователей. В следующем примере некоторые из этих значений являются образцами данных. Реальные значения могут быть другими.

```
inputs:
  accountName:
    type: string
    title: Account name
    encrypted: true
  displayName:
    type: string
    title: Display name
  password:
    type: string
    title: Password
    encrypted: true
  confirmPassword:
    type: string
    title: Password
    encrypted: true
  ouContainer:
    type: object
    title: AD OU container
    $data: 'vro/data/inventory/AD:OrganizationalUnit'
    properties:
      id:
        type: string
      type:
        type: string
```

- ж) В разделе `resources` добавьте код `${input.input-name}` для запроса на выбор значений пользователем.

```
resources:
  Custom_ADUser_1:
    type: Custom.ADUser
    properties:
      accountName: '${input.accountName}'
      displayName: '${input.displayName}'
      ouContainer: '${input.ouContainer}'
      password: '${input.password}'
      confirmPassword: '${input.confirmPassword}'
```

### 3. Разверните облачный шаблон.

- На странице конструктора облачных шаблонов щелкните **Развернуть**.
- Введите в поле **Имя развертывания** значение **AD User Scott**.

- в) Выберите **Версия облачного шаблона** и нажмите кнопку **Далее**.
  - г) Заполните входные данные развертывания.
  - д) Щелкните **Развернуть**.
4. Отслеживайте запрос на подготовку на странице **Развертывания**, чтобы убедиться, что пользователь добавлен в Active Directory и что развертывание выполнено успешно.

#### Следующие шаги

После того как тестируемый облачный шаблон начал действовать, настраиваемый ресурс **Пользователь AD** можно использовать с другими облачными шаблонами.

## Создание шаблона Cloud Assembly, который включает в себя SSH

Можно создать настраиваемые ресурсы, которые будут использоваться для создания облачных шаблонов с помощью рабочих процессов vRealize Orchestrator. В этом примере использования необходимо добавить настраиваемый ресурс, который добавляет узел SSH. Затем ресурс можно добавить в облачные шаблоны. Эта процедура также добавляет рабочий процесс обновления, чтобы пользователи могли вносить изменения в конфигурацию SSH после развертывания, а не выполнять по отдельности действия по регулярному обслуживанию.

Настраиваемые ресурсы являются объектами vRealize Orchestrator или действий расширяемости, которыми можно управлять в vRealize Automation с помощью действий жизненного цикла, определенных в настраиваемом ресурсе. Служба облачных шаблонов автоматически вызывает соответствующие рабочие процессы vRealize Orchestrator или действия расширяемости при инициировании операции, связанной с определенным действием жизненного цикла. Для расширения функциональных возможностей типа ресурса также можно выбрать рабочие процессы vRealize Orchestrator или действия расширяемости, которые можно использовать как операции по регулярному обслуживанию.

В данном примере используются встроенные рабочие процессы из библиотеки vRealize Orchestrator. В нем содержатся рекомендованные значения или строки для демонстрации выполнения процесса. Их можно изменить в соответствии с требованиями среды.

Для справочных целей в данном примере используется проект с именем **DevOpsTesting**. Вместо проекта можно использовать уже существующий проект.

#### Необходимые условия

- Убедитесь, что настроена интеграция vRealize Orchestrator. См. раздел [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).
- Убедитесь, что рабочие процессы, используемые для действий «создать», «обновить», «уничтожить» и «регулярное обслуживание», существуют в vRealize Orchestrator и успешно выполняются.
- В vRealize Orchestrator найдите тип ресурса, используемый рабочими процессами. Все рабочие процессы, включенные в этот настраиваемый ресурс, должны использовать один и тот же тип ресурса. Тип ресурса в данном примере — `SSH:Host`. Дополнительные сведения о проверке типа ресурса см. в разделе [Настраиваемые типы ресурсов для облачных шаблонов Cloud Assembly](#).
- Вам необходимо знать, как настроить и развернуть облачный шаблон компьютера.



## Процедура

1. Создание настраиваемого ресурса узла SSH для добавления SSH в облачный шаблон.

На этом этапе настраиваемый ресурс добавляется на холст проекта облачного шаблона в качестве типа ресурса.

- а) В Cloud Assembly выберите **Проектирование > Настраиваемые ресурсы** и щелкните **Создать настраиваемый ресурс**.
- б) Введите следующие значения.

Обратите внимание, что, за исключением имен рабочих процессов, приведенные значения являются только примерами.

Таблица 6-3.

Настройка	Пример значения
Имя	<b>Узел SSH — проект DevOpsTesting</b> Это имя, которое отображается на палитре типов ресурсов для облачного шаблона.
Тип ресурса	<b>Custom.SSHHost</b> Тип ресурса должен начинаться с <b>Custom.</b> и каждый тип ресурса должны быть уникальными. Несмотря на то, что включение <b>Custom.</b> не проверяется в этом текстовое поле; строка будет добавлена автоматически, если она была случайно удалена. Этот тип ресурса добавляется на холст проекта, чтобы его можно было использовать в облачном шаблоне.

- в) Чтобы включить этот тип ресурса в списке типов ресурсов облачного шаблона, убедитесь, что параметр **Активировать** включен.
- г) Выберите параметр **Область**, чтобы тип ресурса был доступен для проекта **DevOpsTesting**.
- д) В разделе **На основе** в качестве поставщика действий жизненного цикла должна быть выбрана **Иерархия vRO**.

- е) Выберите рабочие процессы, которые определяют ресурс.

Настройка	Настройка
Действия жизненного цикла — создание	<p>Выберите рабочий процесс <b>Добавление узла SSH</b>.</p> <p>Если у вас есть несколько интеграций vRealize Orchestrator, выберите рабочий процесс в экземпляре интеграции, который используется для запуска этих настраиваемых ресурсов.</p> <p>После выбора рабочего процесса появится раскрывающееся меню внешних типов, в котором будет автоматически выбран вариант <code>SSH:Host</code>. Тип внешнего источника можно использовать только один раз, если применяется общий доступ, и только один раз в проекте. В этом примере использования настраиваемый ресурс будет предоставляться только для проекта <b>DevOpsTesting</b>.</p> <p>Если у вас есть другие рабочие процессы, для которых требуется тип <code>SSH:Host</code>, необходимо создать отдельные настраиваемые ресурсы для каждого проекта.</p>
Действия жизненного цикла — обновление	Выберите рабочий процесс <b>Обновление узла SSH</b> .
Действия жизненного цикла — уничтожение	Выберите рабочий процесс <b>Удаление узла SSH</b> .

- ж) Просмотрите значения ключа и типа схемы на вкладке **Свойства**, чтобы уточнить входные данные рабочего процесса, которые можно настроить в облачном шаблоне.

В схеме перечислены обязательные и необязательные входные значения, определенные в рабочем процессе. Обязательные входные значения включены в YAML облачного шаблона.

В рабочем процессе **Добавление узла SSH** обязательными входными значениями являются `hostname`, `port` и `username`. Другие свойства схемы не являются обязательными. Схему также можно использовать, чтобы определить, где необходимо создать привязки со значениями других полей, рабочими процессами или действиями. В данном примере использования привязки отсутствуют.

- з) Чтобы завершить процедуру создания настраиваемого ресурса, щелкните **Создать**.

## 2. Создание облачного шаблона, который добавляет узел SSH при его развертывании

- Выберите **Проектирование > Облачные шаблоны** и нажмите **Создать из > Пустой холст**.
- Назначьте облачному шаблону имя **Компьютер с узлом SSH**.
- Выберите проект **DevOpsTesting** и нажмите **Создать**.
- Добавление и настройка компьютера vSphere.

- д) Перетащите тип ресурса **Узел SSH — проект DevOpsTesting** из списка настраиваемых ресурсов слева на странице проекта облачного шаблона на холст.

---

**Примечание** Чтобы выбрать настраиваемый ресурс, можно прокрутить список вниз и выбрать ресурс на левой панели или найти ресурс в текстовом поле **Поиск типов ресурсов**. Если настраиваемый ресурс отсутствует, нажмите кнопку «Обновить» рядом с текстовым полем **Поиск типов ресурсов**.

---

Напоминание о наличии типа ресурса, так как он был настроен для проекта. Если вы создали облачный шаблон для другого проекта, то тип ресурса не будет виден.

- е) В правой части окна измените код YAML, чтобы добавить обязательные входные значения.

Добавьте раздел `inputs` в код, чтобы пользователи могли ввести имена пользователя и узла в ходе развертывания. В этом примере порт по умолчанию — 22. В следующем примере некоторые из этих значений являются образцами данных. Реальные значения могут быть другими.

```
inputs:
  hostname:
    type: string
    title: The hostname of the SSH Host
  username:
    type: string
    title: Username
```

- ж) В разделе `resources` добавьте код `${input.input-name}` для запроса на выбор значений пользователем.

```
resources:
  Custom_SSHTHost_1:
    type: Custom.SSHTHost
    properties:
      port: 22
      hostname: '${input.hostname}'
      username: '${input.username}'
```

### 3. Разверните облачный шаблон.

- На странице конструктора облачных шаблонов щелкните **Развернуть**.
- Введите **Имя развертывания Тест узла SSH**.
- Выберите **Версия облачного шаблона** и нажмите кнопку **Далее**.
- Заполните входные данные развертывания.
- Щелкните **Развернуть**.

4. Отслеживайте запрос на подготовку на странице **Развертывания**, чтобы убедиться, что узел SSH включен в развертывание и что развертывание выполнено успешно.

## Следующие шаги

После того как тестируемый облачный шаблон начал работать, настраиваемый ресурс `SSH Host` можно начать использовать с другими облачными шаблонами.

## Проекты Cloud Assembly, обеспечивающие подготовку к изменениям в рамках регулярного обслуживания

В дополнение к действиям по регулярному обслуживанию, уже связанным с типами ресурсов Cloud Assembly, существуют возможности проектирования, позволяющие заранее подготовиться к настраиваемым обновлениям, которые могут потребоваться пользователям.

---

**Осторожно!** Чтобы изменить развертывание, можно отредактировать его облачный шаблон и повторно его применить или использовать действия по регулярному обслуживанию. Однако в большинстве случаев одновременно использовать эти два этих подхода не нужно.

Изменения в рамках регулярного обслуживания в течение жизненного цикла, например включение или выключение питания, обычно безопасны, но при других изменениях, например при добавлении дисков, требуется соблюдать осторожность.

Например, если добавить диски с помощью действия по регулярному обслуживанию, а затем применить смешанный подход путем повторного применения облачного шаблона, этот шаблон может перезаписать изменение в рамках регулярного обслуживания, что может привести к удалению дисков и потере данных.

---

Подготовка процесса регулярного обслуживания может предусматривать непосредственное использование кода облачного шаблона или применение интерфейса проектирования Cloud Assembly.

- В коде облачного шаблона можно использовать входные данные. В этом случае при обновлении развертывания или развернутого ресурса интерфейс будет запрашивать новые значения.
- Cloud Assembly можно использовать для создания настраиваемого действия на основе рабочего процесса или действия расширяемости vRealize Orchestrator. Выполнение настраиваемого действия приводит к тому, что рабочий процесс или действие расширяемости вносит изменения в развертывание или развернутый ресурс.

## Использование входных данных облачного шаблона для обновлений при регулярном обслуживании vRealize Automation

При проектировании облачных шаблонов входные параметры vRealize Automation позволяют пользователям, выполняющим действия по регулярному обслуживанию, повторно вводить выбранные элементы из первоначального запроса на развертывание.

**Осторожно!** Некоторые изменения свойств приводят к необходимости повторного создания ресурса. Например, если `connection_string.name` изменить в `Cloud.Service.Azure.App.Service`, существующий ресурс будет удален и создан новый.

При проектировании входных данных для поддержки изменений в ходе регулярного обслуживания схема [Модели, размещенные на сайте code.vmware.com](#) может помочь найти свойства, обеспечивающие удаление и повторное создание ресурсов.

Сведения о создании входных данных см. в разделе [Входные данные пользователей в запросах vRealize Automation](#).

Пример действия по регулярному обслуживанию см. в следующем разделе.

### Перемещение развернутого компьютера в другую сеть

При обслуживании развертываний и сетей может потребоваться изменить местоположение компьютеров, развернутых с помощью Cloud Assembly.

Например, сначала можно выполнить развертывание в тестовой сети, а затем перейти в производственную сеть. Описанный здесь метод позволяет разработать облачный шаблон заранее, чтобы подготовиться к таким действиям по регулярному обслуживанию. Обратите внимание, что компьютер перемещен. Он не удален и не развернут повторно.

Данная процедура применима только к ресурсам **Cloud.vSphere.Machine**. Она не будет работать для компьютеров, независимых от облачной среды и развернутых в vSphere.

#### Необходимые условия

- Профиль сети Cloud Assembly должен содержать все подсети, к которым будет подключаться компьютер. В Cloud Assembly можно указать сети в разделе **Инфраструктура > Настроить > Профили сети**.  
Профиль сети должен быть в учетной записи и регионе, которые являются частью соответствующего проекта Cloud Assembly для пользователей.
- Разметьте две подсети разными тегами. В следующем примере предполагается, что теги имеют имена **test** и **prod**.
- У развернутого компьютера должен быть тот же тип назначения IP-адреса. При переходе в другую сеть статический адрес нельзя изменить на DHCP или наоборот.

#### Процедура

1. В Cloud Assembly перейдите в раздел **Проектирование** и создайте облачный шаблон для развертывания.

- В разделе «Входные данные» кода добавьте запись, которая позволяет пользователю выбрать сеть.

```
inputs:
  net-tagging:
    type: string
    enum:
      - test
      - prod
    title: Select a network
```

- В разделе «Ресурсы» кода добавьте **Cloud.Network** и подключите к нему компьютер vSphere.
- В разделе **Cloud.Network** создайте ограничение, которое ссылается на выбранные входные данные.

```
resources:
  ABCServer:
    type: Cloud.vSphere.Machine
    properties:
      name: abc-server
      . . .
    networks:
      - network: '${resource["ABCNet"].id}'
  ABCNet:
    type: Cloud.Network
    properties:
      name: abc-network
      . . .
    constraints:
      - tag: '${input.net-tagging}'
```

- Продолжите процесс проектирования и разверните проект, как обычно. При развертывании интерфейс предлагает выбрать сеть **test** или **prod**.
- Если необходимо внести изменение в рамках регулярного обслуживания, перейдите на страницу **Ресурсы > Развертывания** и найдите развертывание, связанное с облачным шаблоном.
- Справа от развертывания нажмите **Действия > Обновить**.
- На панели «Обновление» интерфейс также предлагает выбрать сеть **test** или **prod**.
- Чтобы изменить сети, выберите нужное значение, нажмите **Далее**, а затем **Отправить**.

## Создание действия с ресурсом Cloud Assembly для виртуальной машины vMotion

После развертывания облачного шаблона можно запускать действия по регулярному обслуживанию, которые вносят изменения в развертывание. Cloud Assembly позволяет выполнять много действий по регулярному обслуживанию, но могут потребоваться другие действия. Можно создать настраиваемые действия для работы с ресурсами и сделать их доступными пользователям в виде действий по регулярному обслуживанию.

Настраиваемые действия с ресурсами основаны на рабочих процессах vRealize Orchestrator.

Данный пример настраиваемого действия с ресурсом по регулярному обслуживанию позволяет познакомиться с процессом создания. Для эффективного использования действий с ресурсом необходимо иметь возможность создания рабочих процессов и действий vRealize Orchestrator, которые выполняют нужные задачи.

#### Необходимые условия

- Убедитесь, что настроена интеграция vRealize Orchestrator. См. раздел [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).
- Убедитесь, что рабочий процесс, используемый для действия по регулярному обслуживанию, существует в vRealize Orchestrator и успешно выполняется.

#### Процедура

1. Создайте настраиваемое действие с ресурсом, которое использует vMotion для перемещения виртуальной машины vSphere с одного узла на другой.
  - а) В Cloud Assembly выберите **Проектирование > Действия ресурсов** и щелкните **Создать действие ресурсов**.
  - б) Введите следующие значения.

Обратите внимание, что, за исключением имен рабочих процессов, приведенные значения являются только примерами.

Настройка	Пример значения
Имя	<b>vSphere_VM_vMotion</b> Это имя отображается в списке действий ресурса.
Отображаемое имя	<b>Переместить ВМ</b> Это имя, которое пользователи видят в меню действий по развертыванию.

- в) Щелкните **Активировать**, чтобы включить это действие в меню действий по регулярному обслуживанию для ресурсов, которые соответствуют заданному типу ресурса.
- г) Выберите тип ресурса и рабочий процесс, которые определяют действие по регулярному обслуживанию.

Настройка	Пример значения
Тип ресурса	<p>Выберите тип ресурса <b>Cloud.vSphere.Machine</b>.</p> <p>Это тип ресурса, который развертывается как компонент облачного шаблона, независимо от содержимого шаблона. Например, в шаблоне может быть независимый от облачной среды компьютер, но в случае его развертывания в vCenter Server будет создан компонент <b>Cloud.vSphere.Machine</b>. Так как действие применяется к развернутому типу, при определении действий с ресурсом не используйте типы, независимые от облачной среды.</p> <p>В этом примере vMotion работает только для компьютеров vSphere, но у вас могут быть другие действия, которые необходимо выполнять с ресурсами разного типа. Необходимо создать действие для каждого типа ресурсов.</p>
Рабочий процесс	<p>Выберите рабочий процесс <b>Миграция виртуальной машины с помощью vMotion</b>.</p> <p>Если у вас есть несколько интеграций vRealize Orchestrator, выберите рабочий процесс в экземпляре интеграции, который используется для выполнения настраиваемых действий ресурсов.</p>



2. Для свойств vRealize Orchestrator создайте привязку к свойствам схемы Cloud Assembly. Действия по регулярному обслуживанию в Cloud Assembly поддерживают три типа привязок.

Тип привязки	Описание
в запросе	Тип привязки для значения по умолчанию. Если выбран данный вариант, входное свойство отображается в форме запроса, а его значение должно быть введено пользователем в момент запроса.
с действием привязки	<p>Этот параметр доступен только для входных данных ссылочного типа, например:</p> <ul style="list-style-type: none"> <li>■ VC:VirtualMachine</li> <li>■ VC:Folder</li> </ul> <p>Пользователь выбирает действие, которое выполняет привязку. Выбранное действие должно возвращать тот же тип, что и входной параметр. Верное определение свойства — <code>\${properties.someProperty}</code>.</p>
прямая	Этот параметр доступен для входных свойств, которые используют простые типы данных. Если выбран данный вариант, свойство с подходящим типом сопоставляется непосредственно из схемы входного свойства. Пользователь выбирает свойство в дереве схемы. Свойства с разными типами отключены.

В этом случае привязка представляет собой действие vRealize Orchestrator, которое устанавливает соединение между типом входных данных VC:VirtualMachine в vRealize Orchestrator, используемым в рабочем процессе, и типом ресурса Cloud.vSphere.Machine в Cloud Assembly. Благодаря привязке действие по регулярному обслуживанию становится незаметным для пользователя, запрашивающего действие с vMotion на BM vSphere. Система предоставляет имя в рабочем процессе, чтобы пользователю не требовалось самому его создавать.

- а) После выбора рабочего процесса **Миграция виртуальной машины с помощью vMotion** перейдите в область **Привязка свойств**.
- б) Выберите привязку для входного свойства `vm`.
- в) В разделе **Привязка** выберите **С действием привязки**.

Автоматически выбирается действие **findVcVmByVcAndVmUuid**. Это действие предварительно настроено в интеграции vRealize Orchestrator в Cloud Assembly.

- г) Нажмите **Сохранить**.

3. Чтобы сохранить изменения в действии по регулярному обслуживанию, нажмите **Создать**.

4. Чтобы учитывать другие входные параметры в рабочем процессе, можно настроить форму запроса, которую пользователи видят при запросе действия.

- а) В разделе **Действия ресурсов** выберите последнее созданное действие по регулярному обслуживанию.
- б) Щелкните **Изменение параметров запроса**.

Вид страницы запроса для пользователей можно настроить.

Имя поля по умолчанию	Внешний вид	Значения	Ограничения
Целевой пул ресурсов для виртуальной машины. По умолчанию используется текущий пул ресурсов.	<ul style="list-style-type: none"> <li>■ Метка = целевой пул ресурсов</li> <li>■ Тип отображения = список выбора значений</li> </ul>		
Целевой узел, на который нужно выполнить миграцию виртуальной машины	<ul style="list-style-type: none"> <li>■ Метка = целевой узел</li> <li>■ Тип отображения = список выбора значений</li> </ul>		Обязательно = да
Приоритет задачи миграции	Метка = приоритет задачи	Параметры значений <ul style="list-style-type: none"> <li>■ Источник значения = константа</li> </ul> В текстовом поле введите список, разделенный запятыми. <div>             lowPriority Low,defaultPriority Default,highPriority High           </div>	Обязательно = да
(Необязательно) Миграцию виртуальной машины необходимо выполнять только в том случае, если ее состояние питания соответствует указанному состоянию.	Удалите это текстовое поле. vMotion может перемещать компьютеры в любом состоянии питания.		

- в) Нажмите **Сохранить**.

5. Чтобы ограничить период доступности действия, можно настроить условия.

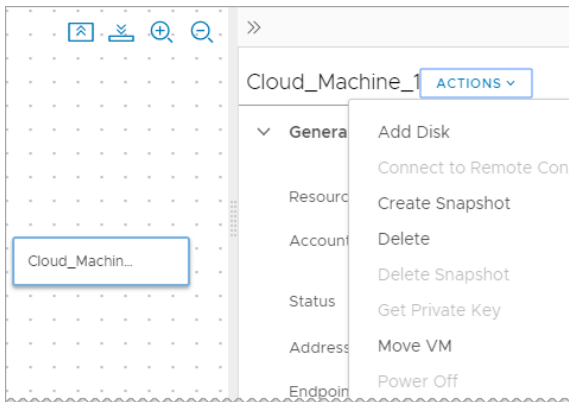
Например, необходимо, чтобы действие vMotion было доступно только в том случае, если компьютер содержит четыре ЦП или менее.

- а) Включите **Требуется условие**.
- б) Введите значение для условия.

Key	Оператор	Значение
\${properties.cpuCount}	lessThan	4

Если требуются сложные условия, см. [Создание расширенных условий для настраиваемых действий Cloud Assembly](#).

- в) Щелкните **Обновить**.
6. Убедитесь, что действие «Переместить ВМ» доступно для развернутых компьютеров, которые соответствуют критериям.
- а) Выберите **Развертывания**.
  - б) Выберите развертывание, которое содержит развернутый компьютер, соответствующий заданным критериям.
  - в) Откройте развертывание и выберите компьютер.
  - г) Щелкните действия на правой панели и убедитесь, что действие Move VM существует.

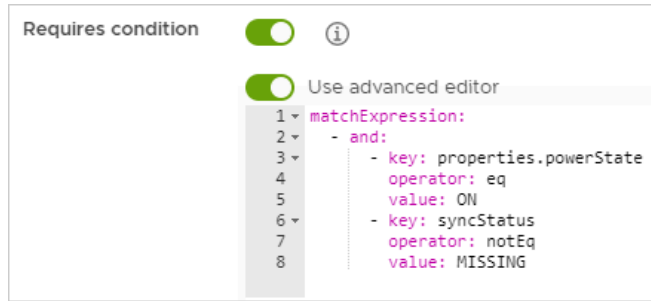


- д) Запустите действие.

## Создание расширенных условий для настраиваемых действий Cloud Assembly

В качестве альтернативы обычному списку условий в Cloud Assembly расширенный редактор позволяет создавать более сложные выражения критериев для определения условий доступности действия.

При создании нового действия с ресурсом выберите **Требуется условие** и **Использовать расширенный редактор**. Затем введите необходимое выражение критериев.



Выражение — это условие или список условий, для каждого из которых используется формат «ключ — оператор — значение». На предыдущем рисунке показаны критерии, при которых целевой объект должен быть в наличии и в работающем состоянии.

### Условия

Условие	Описание	Пример
и	Чтобы результатом выражения было true (истина), все подусловия должны иметь значение true (истина).	<p>Оценить как true только в том случае, если properties.powerState имеет значение «ВКЛ.», а syncStatus не имеет значения «ОТСУТСТВУЕТ».</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: eq       value: ON     - key: syncStatus       operator: notEq       value: MISSING</pre>
или	Чтобы результатом выражения было true (истина), одно или несколько подусловий должны иметь значение true (истина).	<p>Принимает значение true, если properties.powerState имеет значение «ВКЛ.» или «ВЫКЛ.».</p> <pre>matchExpression:   - or:     - key: properties.powerState       operator: eq       value: ON     - key: properties.powerState       operator: eq       value: OFF</pre>

## Операторы

Оператор	Описание	Пример
eq	Равно. Поиск точного совпадения.	<p>Принимает значение true, если properties.powerState имеет значение «ВКЛ.».</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: eq       value: ON</pre>
notEq	Не равно. Недопущение точного совпадения.	<p>Принимает значение true, если properties.powerState не имеет значение «ВЫКЛ.».</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: notEq       value: OFF</pre>
hasAny	Поиск совпадения в группе объектов.	<p>Принимает значение true, если массив storage.disks содержит объект EBS со значением IOPS, равным 100.</p> <pre>matchExpression:   - key: storage.disks     operator: hasAny     value:       matchExpression:         - and:           - key: iops             operator: eq             value: 100           - key: service             operator: eq             value: ebs</pre>
in	Поиск совпадения в наборе значений.	<p>Принимает значение true, если properties.powerState имеет значение «ВЫКЛ.» либо «ПРИОСТАНОВИТЬ».</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: in       value: OFF, SUSPEND</pre>

Оператор	Описание	Пример
notIn	Недопущение совпадения набора значений.	<p>Принимает значение true, если properties.powerState не имеет значения «ВЫКЛ.» и не имеет значения «ПРИОСТАНОВИТЬ».</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: notIn       value: OFF, SUSPEND</pre>
greaterThan	Поиск совпадения выше заданного порогового значения. Применяется только к числовым значениям.	<p>Принимает значение true, если первый объект в массиве storage.disks имеет значение IOPS больше 50.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: greaterThan       value: 50</pre>
lessThan	Поиск совпадения ниже заданного порогового значения. Применяется только к числовым значениям.	<p>Принимает значение true, если первый объект в массиве storage.disks имеет значение IOPS меньше 200.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: lessThan       value: 200</pre>
greaterThanEquals	Поиск совпадения, которое больше заданного порогового значения или равно ему. Применяется только к числовым значениям.	<p>Принимает значение true, если первый объект в массиве storage.disks имеет значение IOPS, равное 100 или больше.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: greaterThanEquals       value: 100</pre>

Оператор	Описание	Пример
lessThanEquals	Поиск совпадения, которое меньше или равно заданному пороговому значению. Применяется только к числовым значениям.	<p>Принимает значение true, если первый объект в массиве storage.disks имеет значение IOPS, равное 100 или меньше.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: lessThanEquals       value: 100</pre>
matchesRegex	Использование регулярного выражения для поиска совпадения.	<p>Принимает значение true, если properties.zone имеет значение us-east-1a или us-east-1c.</p> <pre>matchExpression:   - and:     - key: properties.zone       operator: matchesRegex       value: (us-east-1)+(a c)     {1,2}</pre>

## Примеры

Следующее выражение критериев принимает значение true, если properties.tags содержит тег ключа key1 и значение value1.

Во внешнем выражении используется hasAny, так как properties.tags — это массив, и выражение должно принимать значение true во всех случаях, когда в любой из пар «ключ — значение» в массиве появляется тег key1=value1.

Во внутреннем выражении есть два условия: одно для поля ключа и другое для поля значения. Массив properties.tags содержит пары тегов «ключ — значение», и необходимо совпадение в обоих полях (ключ и значение).

```
matchExpression:
  - key: properties.tags
    operator: hasAny
    value:
      matchExpression:
        - and:
          - key: key
            operator: eq
            value: key1
          - key: value
            operator: eq
            value: value1
```

Следующее выражение критериев похоже на предыдущий пример, но принимает значение true во всех случаях, когда `properties.tags` содержит тег `key1=value1` или `key2=value2`.

```
matchExpression:
  - or:
    - key: properties.tags
      operator: hasAny
      value:
        matchExpression:
          - and:
            - key: key
              operator: eq
              value: key1
            - key: value
              operator: eq
              value: value1
    - key: properties.tags
      operator: hasAny
      value:
        matchExpression:
          - and:
            - key: key
              operator: eq
              value: key2
            - key: value
              operator: eq
              value: value2
```

## Другие примеры кода Cloud Assembly

Код облачного шаблона в Cloud Assembly может быть практически неограниченным с точки зрения как своей структуры, так и применения.

Во многих случаях пример успешного кода представляет собой наилучшую отправную точку для дальнейшей разработки. Изучая пример, вносите изменения, чтобы применять параметры своего сайта, что касается имен ресурсов, значений и т. д.

## Документированный пример шаблона Cloud Assembly

Благодаря включению подробного набора комментариев этот пример позволяет просмотреть структуру и назначение разделов в шаблоне Cloud Assembly, который раньше назывался схемой элементов.

```
# *****
#
# This WordPress cloud template is enhanced with comments to explain its
# parameters.
#
# Try cloning it and experimenting with its YAML code. If you're new to
# YAML, visit yml.org for general information.
#
# The cloud template deploys a minimum of 3 virtual machines and runs scripts
# to install packages.
```



```

#
# *****
#
# -----
# Templates need a descriptive name and version if
# source controlled in git.
# -----
name: WordPress Template with Comments
formatVersion: 1
version: 1
#
# -----
# Inputs create user selections that appear at deployment time. Inputs
# can set placement decisions and configurations, and are referenced
# later, by the resources section.
# -----
inputs:
#
# -----
# Choose a cloud endpoint. 'Title' is the visible
# option text (oneOf allows for the friendly title). 'Const' is the
# tag that identifies the endpoint, which was set up earlier, under the
# Cloud Assembly Infrastructure tab.
# -----
platform:
  type: string
  title: Deploy to
  oneOf:
    - title: AWS
      const: aws
    - title: Azure
      const: azure
    - title: vSphere
      const: vsphere
  default: vsphere
#
# -----
# Choose the operating system. Note that the Cloud Assembly
# Infrastructure must also have an AWS, Azure, and vSphere Ubuntu image
# mapped. In this case, enum sets the option that you see, meaning there's
# no friendly title feature this time. Also, only Ubuntu is available
# here, but having this input stubbed in lets you add more operating
# systems later.
# -----
osimage:
  type: string
  title: Operating System
  description: Which OS to use
  enum:
    - Ubuntu
#
# -----
# Set the number of machines in the database cluster. Small and large
# correspond to 1 or 2 machines, respectively, which you see later,
# down in the resources section.

```

```

# -----
dbenvsize:
  type: string
  title: Database cluster size
  enum:
    - Small
    - Large
#
# -----
# Dynamically tag the machines that will be created. The
# 'array' of objects means you can create as many key-value pairs as
# needed. To see how array input looks when it's collected,
# open the cloud template and click TEST.
# -----
Mtags:
  type: array
  title: Tags
  description: Tags to apply to machines
  items:
    type: object
    properties:
      key:
        type: string
        title: Key
      value:
        type: string
        title: Value
#
# -----
# Create machine credentials. These credentials are needed in
# remote access configuration later, in the resources section.
# -----
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username
userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#\$]+'
  encrypted: true
  title: Database Password
  description: Database Password
#
# -----
# Set the database storage disk size.
# -----
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Size of database disk

```

```

#
# -----
# Set the number of machines in the web cluster. Small, medium, and large
# correspond to 2, 3, and 4 machines, respectively, which you see later,
# in the WebTier part of the resources section.
# -----
clusterSize:
  type: string
  enum:
    - small
    - medium
    - large
  title: Wordpress Cluster Size
  description: Wordpress Cluster Size
#
# -----
# Set the archive storage disk size.
# -----
archiveDiskSize:
  type: number
  default: 4
  maximum: 10
  title: Wordpress Archive Disk Size
  description: Size of Wordpress archive disk
#
# -----
# The resources section configures the deployment of machines, disks,
# networks, and other objects. In several places, the code pulls from
# the preceding interactive user inputs.
# -----
resources:
#
# -----
# Create the database server. Choose a cloud agnostic machine 'type' so
# that it can deploy to AWS, Azure, or vSphere. Then enter its property
# settings.
# -----
DBTier:
  type: Cloud.Machine
  properties:
#
# -----
# Descriptive name for the virtual machine. Does not become the hostname
# upon deployment.
# -----
  name: mysql
#
# -----
# Hard-coded operating system image to use. To pull from user input above,
# enter the following instead.
# image: '${input.osimage}'
# -----
  image: Ubuntu
#
# -----

```

```

# Hard-coded capacity to use. Note that the Cloud Assembly
# Infrastructure must also have AWS, Azure, and vSphere flavors
# such as small, medium, and large mapped.
# -----
#     flavor: small
#
# -----
# Tag the database machine to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with a site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
#     constraints:
#       - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Also tag the database machine with any free-form tags that were created
# during user input.
# -----
#     tags: '${input.Mtags}'
#
# -----
# Set the database cluster size by referencing the dbenvsize user
# input. Small is one machine, and large defaults to two.
# -----
#     count: '${input.dbenvsize == "Small" ? 1 : 2}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
#     networks:
#       - network: '${resource.WP_Network.id}'
#
# -----
# Enable remote access to the database server. Reference the credentials
# from the user input.
# -----
#     remoteAccess:
#       authentication: usernamePassword
#       username: '${input.username}'
#       password: '${input.userpassword}'
#
# -----
# You are free to add custom properties, which might be used to initiate
# an extensiblity subscription, for example.
# -----
#     ABC-Company-ID: 9393
#
# -----
# Run OS commands or scripts to further configure the database machine,
# via operations such as setting a hostname, generating SSH private keys,
# or installing packages.
# -----

```

```

cloudConfig: |
  #cloud-config
  repo_update: true
  repo_upgrade: all
  packages:
    - mysql-server
  runcmd:
    - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
    - service mysql restart
    - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
    - mysql -e "FLUSH PRIVILEGES;"
  attachedDisks: []

#
# -----
# Create the web server. Choose a cloud agnostic machine 'type' so that it
# can deploy to AWS, Azure, or vSphere. Then enter its property settings.
# -----
WebTier:
  type: Cloud.Machine
  properties:
#
# -----
# Descriptive name for the virtual machine. Does not become the hostname
# upon deployment.
# -----
    name: wordpress
#
# -----
# Hard-coded operating system image to use. To pull from user input above,
# enter the following instead:
# image: '${input.osimage}'
# -----
    image: Ubuntu
#
# -----
# Hard-coded capacity to use. Note that the Cloud Assembly
# Infrastructure must also have AWS, Azure, and vSphere flavors
# such as small, medium, and large mapped.
# -----
    flavor: small
#
# -----
# Set the web server cluster size by referencing the clusterSize user
# input. Small is 2 machines, medium is 3, and large defaults to 4.
# -----
    count: '${input.clusterSize== "small" ? 2 : (input.clusterSize == "medium" ? 3 : 4)}'
#
# -----
# Set an environment variable to display object information under the
# Properties tab, post-deployment. Another example might be
# {env.blueprintID}
# -----
  tags:
    - key: cas.requestedBy
      value: '${env.requestedBy}'

```

```

#
# -----
# You are free to add custom properties, which might be used to initiate
# an extensibility subscription, for example.
# -----
    ABC-Company-ID: 9393
#
# -----
# Tag the web server to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with your site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
    constraints:
      - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
    networks:
      - network: '${resource.WP_Network.id}'
#
# -----
# Run OS commands or scripts to further configure the web server,
# with operations such as setting a hostname, generating SSH private keys,
# or installing packages.
# -----
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all
      packages:
        - apache2
        - php
        - php-mysql
        - libapache2-mod-php
        - php-mcrypt
        - mysql-client
      runcmd:
        - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget
https://wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
        - i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
        - mysql -u root -pmysqlpassword -h ${resource.DBTier.networks[0].address} -e
"create database wordpress_blog;"
        - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
        - sed -i -e s/"define('DB_NAME', 'database_name_here');"/"define('DB_NAME',
'wordpress_blog');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i
-e s/"define('DB_USER', 'username_here');"/"define('DB_USER', 'root');"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define('DB_PASSWORD',

```

```
'password_here');"define('DB_PASSWORD', 'mysqlpassword');" /var/www/html/mywordpresssite/
wp-config.php && sed -i -e s/"define('DB_HOST', 'localhost');"define('DB_HOST', '$
{resource.DBTier.networks[0].address}');" /var/www/html/mywordpresssite/wp-config.php
    - service apache2 reload

#
# -----
# Create the network that the database and web servers connect to.
# Choose a cloud agnostic network 'type' so that it can deploy to AWS,
# Azure, or vSphere. Then enter its property settings.
# -----
WP_Network:
  type: Cloud.Network
  properties:
#
# -----
# Descriptive name for the network. Does not become the network name
# upon deployment.
# -----
    name: WP_Network
#
# -----
# Set the networkType to an existing network. You could also use a
# constraint tag to target a specific, like-tagged network.
# The other network types are private or public.
# -----
    networkType: existing
#
# *****
#
# VMware hopes that you found this commented template useful. Note that
# you can also access an API to create templates, or query for input
# schema that you intend to request. See the following Swagger
# documentation.
#
# www.mgmt.cloud.vmware.com/blueprint/api/swagger/swagger-ui.html
#
# *****
```

## Примеры ресурсов vSphere в Cloud Assembly

Эти примеры кода демонстрируют ресурсы компьютера vSphere в облачных шаблонах Cloud Assembly.

Ресурс	Пример облачного шаблона
Виртуальная машина vSphere с ЦП, памятью и операционной системой	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 1       totalMemoryMB: 1024       image: ubuntu</pre>
Компьютер vSphere с ресурсом хранилища данных	<pre>resources:   demo-vsphere-disk-001:     type: Cloud.vSphere.Disk     properties:       name: DISK_001       type: 'HDD'       capacityGb: 10       dataStore: 'datastore-01'       provisioningType: thick</pre>
Компьютер vSphere с подключенным диском	<pre>resources:   demo-vsphere-disk-001:     type: Cloud.vSphere.Disk     properties:       name: DISK_001       type: HDD       capacityGb: 10       dataStore: 'datastore-01'       provisioningType: thin   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 2       totalMemoryMB: 2048       imageRef: &gt;-         https://packages.vmware.com/photon/4.0/         Rev1/ova/photon-ova-4.0-ca7c9e9330.ova       attachedDisks:         - source: '\${demo-vsphere-disk-001.id}'</pre>



Ресурс	Пример облачного шаблона
Компьютер vSphere с динамически настраиваемым количеством дисков	<pre> inputs:   disks:     type: array     title: disks     items:       title: disks       type: integer     maxItems: 15 resources:   Cloud_Machine_1:     type: Cloud.vSphere.Machine     properties:       image: Centos       flavor: small       attachedDisks: '\$ {map_to_object(resource.Cloud_Volume_1[*].id, "source")}'   Cloud_Volume_1:     type: Cloud.Volume     allocatePerInstance: true     properties:       capacityGb: '\${input.disks[count.index]}'       count: '\${length(input.disks)}' </pre>
Компьютер vSphere из образа моментального снимка. Добавьте прямую косую черту и имя моментального снимка. Образ моментального снимка может быть связанным клоном.	<pre> resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       imageRef: 'demo-machine/snapshot-01'       cpuCount: 1       totalMemoryMB: 1024 </pre>
Компьютер vSphere в указанной папке в vCenter	<pre> resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 2       totalMemoryMB: 1024       imageRef: ubuntu       resourceGroupName: 'myFolder' </pre>

Ресурс	Пример облачного шаблона
Компьютер vSphere с несколькими сетевыми адаптерами	<pre> resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       image: ubuntu       flavor: small       networks:         - network: '\${network-01.name}'           deviceIndex: 0         - network: '\${network-02.name}'           deviceIndex: 1     network-01:       type: Cloud.vSphere.Network       properties:         name: network-01     network-02:       type: Cloud.vSphere.Network       properties:         name: network-02 </pre>
Компьютер vSphere с подключенным тегом в vCenter	<pre> resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       flavor: small       image: ubuntu       tags:         - key: env           value: demo </pre>

Ресурс	Пример облачного шаблона
Компьютер vSphere со спецификацией настройки	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       image: ubuntu       flavor: small       customizationSpec: Linux</pre>
Компьютер vSphere с удаленным доступом	<pre>inputs:   username:     type: string     title: Username     description: Username     default: testUser   password:     type: string     title: Password     default: VMware@123     encrypted: true     description: Password for the given username resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;-         https://cloud-images.ubuntu.com/releases/         16.04/release-20170307/ubuntu-16.04-server-cloudimg-         amd64.ova       cloudConfig:           ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false       users:         - default         - name: \${input.username}           lock_passwd: false           sudo: ['ALL=(ALL) NOPASSWD:ALL']           groups: [wheel, sudo, admin]           shell: '/bin/bash'       runcmd:         - echo "Defaults:\${input.username} !         requiretty" &gt;&gt; /etc/sudoers.d/\${input.username}</pre>

## Количество ядер на сокет и количество ЦП в Cloud Assembly

Код шаблона Cloud Assembly позволяет указать количество ядер на сокет для ресурса компьютера vSphere.

Можно указать количество ядер на виртуальный сокет или общее количество сокетов. Например, в условиях лицензирования может ограничиваться количество лицензий программного обеспечения на один сокет, доступные операционные системы могут распознавать только определенное число сокетов, что требует предоставления дополнительных ядер процессоров.

Добавьте свойство `coreCount` в облачный шаблон в ресурсе компьютера vSphere.

Значение `coreCount` должно быть меньше количества ЦП (`cpuCount`), указанного в сопоставлении конфигураций или в коде ресурса компьютера vSphere в облачном шаблоне, или равно ему.

Дополнительные сведения см. в разделе [Настройка количества ядер на ЦП в виртуальной машине \(1010184\)](#).

Свойство `coreCount` является необязательным и доступно только для ресурсов компьютеров vSphere.

Пример фрагмента ресурса компьютера vSphere показан ниже.

```
Cloud_vSphere_Machine_1:
  type: Cloud.vSphere.Machine
  properties:
    cpuCount: 8
    coreCount: 4
```

Дополнительные сведения о настройке сокетов и количества ядер для каждого сокета доступны в статье блога [Рекомендации по определению объема ресурсов виртуальных ЦП и vNUMA для виртуальной машины](#).

## Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation

В проектах облачных шаблонов и развертываниях можно использовать ресурсы сети, безопасности и подсистемы балансировки нагрузки.

Общие данные о параметрах кода проекта облачного шаблона см. в разделе [Схема типов ресурсов vRealize Automation](#).

Дополнительные сведения см. здесь:

- [Дополнительные сведения о сетевых ресурсах в облачных шаблонах vRealize Automation](#)
- [Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation](#)
- [Дополнительные сведения о ресурсах подсистем балансировки нагрузки в облачных шаблонах vRealize Automation](#)

Здесь приведены примеры ресурсов сетей, безопасности и подсистем балансировки нагрузки в стандартных проектах облачных шаблонов.

## Сети

Сценарий ресурсов	Пример кода проекта облачного шаблона
<p>Компьютер vSphere с несколькими сетевыми адаптерами, подключенными к сетям vSphere и NSX, с назначением IP-адреса DHCP</p>	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       image: ubuntu       flavor: small       networks:         - network: \${resource["demo-vSphere- Network"].id}           deviceIndex: 0         - network: \${resource["demo-NSX- Network"].id}           deviceIndex: 1     demo-vSphere-Network:       type: Cloud.vSphere.Network       properties:         networkType: existing     demo-NSX-Network:       type: Cloud.NSX.Network       properties:         networkType: outbound</pre>
<p>Добавление частной сети со статическим IP-адресом для развертывания VM Azure</p>	<pre>formatVersion: 1 inputs: {} resources:   Cloud_Azure_Machine_1:     type: Cloud.Azure.Machine     properties:       image: photon       flavor: Standard_B1ls       networks:         - network: '\$ {resource.Cloud_Network_1.id}'           assignment: static           address: 10.0.0.45           assignPublicIpAddress: false     Cloud_Network_1:       type: Cloud.Network       properties:         networkType: existing</pre>

Сценарий ресурсов	Пример кода проекта облачного шаблона
<p>Назначение статического IP-адреса можно использовать с vRealize IPAM (внутренним продуктом, поставляемым с vRealize Automation, или внешним продуктом на основе комплекта SDK IPAM в vRA, например для одного из подключаемых модулей Infoblox, доступных в VMware Marketplace). Другие варианты использования <code>assignment: static</code> не поддерживаются, как описано в разделе <i>Ограничения</i> в <a href="#">Дополнительные сведения о сетевых ресурсах в облачных шаблонах vRealize Automation</a>.</p>	<pre>resources:   demo_vm:     type: Cloud.vSphere.Machine     properties:       image: 'photon'       cpuCount: 1       totalMemoryMB: 1024       networks:         - network: \${resource.demo_nw.id}           assignment: static   demo_nw:     type: Cloud.vSphere.Network     properties:       networkType: existing</pre>

Сценарий ресурсов	Пример кода проекта облачного шаблона
<p>Добавьте или измените правила переадресации портов NAT и DNAT в ресурсе Cloud.NSX.NAT для существующего развертывания.</p>	<pre>resources:   gw:     type: Cloud.NSX.Gateway     properties:       networks:         - \${resource.akout.id}   nat:     type: Cloud.NSX.Nat     properties:       networks:         - \${resource.akout.id}       natRules:         - translatedInstance: \$           {resource.centos.networks[0].id}           index: 0           protocol: TCP           kind: NAT44           type: DNAT           sourceIPs: any           sourcePorts: 80           translatedPorts: 8080           destinationPorts: 8080           description: edit         - translatedInstance: \$           {resource.centos.networks[0].id}           index: 1           protocol: TCP           kind: NAT44           type: DNAT           sourceIPs: any           sourcePorts: 90           translatedPorts: 9090           destinationPorts: 9090           description: add           gateway: \${resource.gw.id}   centos:     type: Cloud.vSphere.Machine     properties:       image: WebTinyCentOS65x86       flavor: small       customizationSpec: Linux       networks:         - network: \${resource.akout.id}           assignment: static   akout:     type: Cloud.NSX.Network     properties:       networkType: outbound       constraints:         - tag: nsxt-nat-1-M2</pre>

Сценарий ресурсов	Пример кода проекта облачного шаблона
<p>Компьютер в общедоступном облаке для использования внутреннего IP-адреса вместо общедоступного IP-адреса. В этом примере используется конкретный идентификатор сети.</p> <p>Примечание. Вариант <code>network:</code> используется в параметре <code>networks:</code>, чтобы указать идентификатор целевой сети.</p> <p>Вариант <code>name:</code> в параметре <code>networks:</code> устарел и не должен использоваться.</p>	<pre>resources:   wf_proxy:     type: Cloud.Machine     properties:       image: ubuntu 16.04       flavor: small       constraints:         - tag: 'platform:vsphere'     networks:       - network: '\${resource.wf_net.id}'         assignPublicIpAddress: false</pre>
<p>Маршрутизируемая сеть для NSX-V или NSX-T с использованием типа сетевого ресурса NSX.</p>	<pre>Cloud_NSX_Network_1:   type: Cloud.NSX.Network   properties:     networkType: routed</pre>
<p>Добавьте тег к ресурсу сетевого адаптера компьютера в облачном шаблоне.</p>	<pre>formatVersion: 1 inputs: {} resources:   Cloud_Machine_1:     type: Cloud.vSphere.Machine     properties:       flavor: small       image: ubuntu     networks:       - name: '\${resource.Cloud_Network_1.name}'         deviceIndex: 0         tags:           - key: 'nic0'             value: null           - key: internal             value: true       - name: '\${resource.Cloud_Network_2.name}'         deviceIndex: 1         tags:           - key: 'nic1'             value: null           - key: internal             value: false</pre>
<p>Отметьте тегом логические коммутаторы NSX-T для исходящей сети.</p> <p>Теги поддерживаются для NSX-T и VMware Cloud on AWS.</p> <p>Дополнительные сведения об этом сценарии см. в статье блога сообщества <a href="#">Создание тегов в NSX с помощью Cloud Assembly</a>.</p>	<pre>Cloud_NSX_Network_1:   type: Cloud.NSX.Network   properties:     networkType: outbound   tags:     - key: app       value: opencart</pre>



## Группы безопасности

Сценарий ресурсов	Пример кода проекта облачного шаблона
<p>Существующая группа безопасности с тегом ограничения, применяемым к сетевому адаптеру компьютера.</p> <p>Чтобы использовать существующую группу безопасности, введите значение <i>existing</i> для свойства <code>securityGroupType</code>.</p> <p>Для выделения существующих групп безопасности с помощью тегов ограничений ресурсу <code>Cloud.SecurityGroup</code> можно назначить теги. В проекте облачного шаблона нельзя использовать группы безопасности, которые не содержат теги.</p> <p>Теги ограничений необходимо установить для ресурсов группы безопасности <code>securityGroupType: existing</code>. Эти ограничения должны совпадать с тегами, установленными для существующих групп безопасности. Теги ограничений нельзя устанавливать для ресурсов группы безопасности <code>securityGroupType: new</code>.</p>	<pre>formatVersion: 1 inputs: {} resources:   allowSsh_sg:     type: Cloud.SecurityGroup     properties:       securityGroupType: existing     constraints:       - tag: allowSsh   compute:     type: Cloud.Machine     properties:       image: centos       flavor: small       networks:         - network: '\${resource.prod-net.id}'           securityGroups:             - '\${resource.allowSsh_sg.id}'   prod-net:     type: Cloud.Network     properties:       networkType: existing</pre>
<p>Группа безопасности по требованию с двумя правилами брандмауэра, показывающими варианты доступа <code>Allow</code> и <code>Deny</code>.</p>	<pre>resources:   Cloud_SecurityGroup_1:     type: Cloud.SecurityGroup     properties:       securityGroupType: new     rules:       - ports: 5000         source:           'fc00:10:000:000:000:56ff:fe89:48b4'         access: Allow         direction: inbound         name: allow_5000         protocol: TCP       - ports: 7000         source:           'fc00:10:000:000:000:56ff:fe89:48b4'         access: Deny         direction: inbound         name: deny_7000         protocol: TCP   Cloud_vSphere_Machine_1:     type: Cloud.vSphere.Machine     properties:</pre>

Сценарий ресурсов	Пример кода проекта облачного шаблона
	<pre> image: photon cpuCount: 1 totalMemoryMB: 256 networks:   - network: '\$ {resource.Cloud_Network_1.id}'   assignIPv6Address: true   assignment: static   securityGroups:     - '\$ {resource.Cloud_SecurityGroup_1.id}'   Cloud_Network_1:     type: Cloud.Network     properties:       networkType: existing </pre>
<p>Сложный облачный шаблон с двумя группами безопасности, содержащий следующие элементы:</p> <ul style="list-style-type: none"> <li>■ 1 существующая группа безопасности</li> <li>■ 1 группа безопасности по требованию с примерами нескольких правил брандмауэра</li> <li>■ 1 компьютер vSphere</li> <li>■ 1 существующая сеть</li> </ul> <p>Этот пример показывает различные комбинации протоколов и портов, служб, значений IP CIDR для источника и назначения, диапазон IP-адресов в качестве источника или назначения, а также параметры для any, IPv6 и (::/O).</p> <p>Для сетевых адаптеров компьютеров можно указать подключенную сеть и группы безопасности. Кроме того, можно указать индекс или IP-адрес сетевого адаптера.</p>	<pre> formatVersion: 1 inputs: {} resources:   DEMO_ESG : <i>existing security group - security group 1</i>)     type: Cloud.SecurityGroup     properties:       constraints:         - tag: BlockAll         securityGroupType: <b>existing</b> (<i>designation of existing for security group 1</i>)   DEMO_ODSG: (<i>on-demand security group - security group 2</i>)     type: Cloud.SecurityGroup     properties:       rules: (<i>multiple firewall rules in this section</i>)         - <b>name: IN-ANY</b> (<i>rule 1</i>)           source: any           service: any           direction: inbound           <b>access: Deny</b>         - <b>name: IN-SSH</b> (<i>rule 2</i>)           source: any           service: SSH           direction: inbound           <b>access: Allow</b>         - <b>name: IN-SSH-IP</b> (<i>rule 3</i>)           source: 33.33.33.1-33.33.33.250           protocol: TCP           ports: 223           direction: inbound           <b>access: Allow</b>         - <b>name: IPv-6-ANY-SOURCE</b> (<i>rule 4</i>)           source: ':::/0'           protocol: TCP           ports: 223           direction: inbound           <b>access: Allow</b>         - <b>name: IN-SSH-IP</b> (<i>rule 5</i>)           source: 44.44.44.1/24           protocol: UDP           ports: 22-25           direction: inbound           <b>access: Allow</b>         - <b>name: IN-EXISTING-SG</b> (<i>rule 6</i>) </pre>

Сценарий ресурсов	Пример кода проекта облачного шаблона
	<pre> source: '\${resource["DEMO_ESG"].id}' protocol: ICMPv6 direction: inbound access: Allow - name: OUT-ANY (rule 7)   destination: any   service: any   direction: outbound   access: Deny - name: OUT-TCP-IPv6 (rule 8)   destination: '2001:0db8:85a3::8a2e:0370:7334/64'   protocol: TCP   ports: 22   direction: outbound   access: Allow - name: IPv6-ANY-DESTINATION (rule 9)   destination: '::/0'   protocol: UDP   ports: 23   direction: outbound   access: Allow - name: OUT-UDP-SERVICE (rule 10)   destination: any   service: NTP   direction: outbound   access: Allow securityGroupType: new (designation of on- demand for security group 2) DEMO_VC_MACHINE: (machine resource) type: Cloud.vSphere.Machine properties:   image: PHOTON   cpuCount: 1   totalMemoryMB: 1024 networks: (Machine network NICs) - network: '\${resource.DEMO_NW.id}'   securityGroups:     - '\${resource.DEMO_ODSG.id}'     - '\${resource.DEMO_ESG.id}' DEMO_NETWORK: (network resource) type: Cloud.vSphere.Network properties:   networkType: existing constraints: - tag: nsx62 </pre>

## Подсистемы балансировки нагрузки

Сценарий ресурсов	Пример кода проекта облачного шаблона
<p>Укажите уровень и алгоритм ведения журнала балансировки нагрузки, а также его размер.</p>	<p>Пример подсистемы балансировки нагрузки NSX, где показано использование уровня ведения журнала, алгоритма и размера:</p> <pre>resources:   Cloud_LoadBalancer_1:     type: Cloud.NSX.LoadBalancer     properties:       name: myapp-lb       network: '\${appnet-public.name}'       instances: '\${wordpress.id}'       routes:         - protocol: HTTP port: '80'           loggingLevel: CRITICAL           algorithm: LEAST_CONNECTION           type: MEDIUM</pre>
<p>Свяжите подсистему балансировки нагрузки с именованным компьютером или его сетевым адаптером. Чтобы добавить компьютер в пул подсистемы балансировки нагрузки, можно указать либо machine ID, либо machine network ID. Свойство instances поддерживает как компьютеры (machine by ID), так и сетевые адаптеры (machine by network ID).</p> <p>В первом примере в развертывании используется параметр machine by ID для балансировки нагрузки компьютера, когда он развернут в любой сети.</p> <p>Во втором примере в развертывании используется параметр machine by network ID для балансировки нагрузки компьютера только в том случае, когда компьютер развернут в сетевом адаптере именованного компьютера.</p> <p>В третьем примере показаны оба параметра, используемые в одном и том же варианте instances.</p>	<p>Для определения идентификатора компьютера или идентификатора сети компьютеров можно использовать свойство instances.</p> <ul style="list-style-type: none"> <li>■ Идентификатор компьютера <pre>Cloud_LoadBalancer_1:   type: Cloud.LoadBalancer   properties:     network: '\${resource.Cloud_Network_1.id}'     instances: '\$ {resource.Cloud_Machine_1.id}'</pre> </li> <li>■ Идентификатор сети компьютеров <pre>Cloud_LoadBalancer_1:   type: Cloud.LoadBalancer   properties:     network: '\${resource.Cloud_Network_1.id}'     instances: '\$ {resource.Cloud_Machine_1.networks[0].id}'</pre> </li> <li>■ Один компьютер, указанный для включения подсистемы балансировки нагрузки, и сетевой адаптер другого компьютера, указанный для включения подсистемы балансировки нагрузки: <pre>instances:   - resource.Cloud_Machine_1.id   - resource.Cloud_Machine_2.networks[2].id</pre> </li> </ul>

Сценарий ресурсов	Пример кода проекта облачного шаблона
<p>Добавьте параметры проверки работоспособности в подсистему балансировки нагрузки NSX. К дополнительным вариантам относятся <code>httpMethod</code>, <code>requestBody</code> и <code>responseBody</code>.</p>	<pre>myapp-lb:   type: Cloud.NSX.LoadBalancer   properties:     name: myapp-lb     network: '\${appnet-public.name}'     instances: '\${wordpress.id}'     routes:       - protocol: HTTP         port: '80'         algorithm: ROUND_ROBIN         instanceProtocol: HTTP         instancePort: '80'         healthCheckConfiguration:           protocol: HTTP           port: '80'           urlPath: /mywordpresssite/wp-admin/   install.php     intervalSeconds: 60     timeoutSeconds: 10     unhealthyThreshold: 10     healthyThreshold: 2     connectionLimit: '50'     connectionRateLimit: '50'     maxConnections: '500'     minConnections: ''     internetFacing: true{code}</pre>

Сценарий ресурсов	Пример кода проекта облачного шаблона
Сеть по требованию с подсистемой балансировки нагрузки с одним плечом.	<pre> inputs: {} resources:   mp-existing:     type: Cloud.Network     properties:       name: mp-existing       networkType: existing   mp-wordpress:     type: Cloud.vSphere.Machine     properties:       name: wordpress       count: 2       flavor: small       image: tiny       customizationSpec: Linux       networks:         - network: '\${resource["mp-private"].id}'   mp-private:     type: Cloud.NSX.Network     properties:       name: mp-private       networkType: private       constraints:         - tag: nsxt   mp-wordpress-lb:     type: Cloud.LoadBalancer     properties:       name: wordpress-lb       internetFacing: false       network: '\${resource.mp-existing.id}'       instances: '\${resource["mp-wordpress"].id}'       routes:         - protocol: HTTP           port: '80'           instanceProtocol: HTTP           instancePort: '80'           healthCheckConfiguration:             protocol: HTTP             port: '80'             urlPath: /index.pl             intervalSeconds: 60             timeoutSeconds: 30             unhealthyThreshold: 5             healthyThreshold: 2 </pre>
Существующая сеть с подсистемой балансировки нагрузки.	<pre> formatVersion: 1 inputs:   count:     type: integer     default: 1 resources:   ubuntu-vm:     type: Cloud.Machine     properties:       name: ubuntu       flavor: small       image: tiny       count: '\${input.count}'       networks: </pre>

Сценарий ресурсов	Пример кода проекта облачного шаблона
	<pre> - network: '\$ {resource.Cloud_NSX_Network_1.id}' Provider_LoadBalancer_1:   type: Cloud.LoadBalancer   properties:     name: OC-LB     routes:       - protocol: HTTP         port: '80'         instanceProtocol: HTTP         instancePort: '80'         healthCheckConfiguration:           protocol: HTTP           port: '80'           urlPath: /index.html           intervalSeconds: 60           timeoutSeconds: 5           unhealthyThreshold: 5           healthyThreshold: 2         network: '\$ {resource.Cloud_NSX_Network_1.id}'         internetFacing: false         instances: '\${resource["ubuntu-vm"].id}' Cloud_NSX_Network_1:   type: Cloud.NSX.Network   properties:     networkType: existing     constraints:       - tag: nsxt24prod </pre>

## Подробнее

Сценарии реализации сетей и групп безопасности см. в следующих публикациях в блогах VMware.

- [vRealize Automation Cloud Assembly Load Balancer with NSX-T Deep Dive](#) (Подсистема балансировки нагрузки vRealize Automation Cloud Assembly с NSX-T Deep Dive)
- [Автоматизация сети с помощью Cloud Assembly и NSX — часть 1](#) (включая использование облачных учетных записей NSX-T и vCenter и CIDR сети)
- [Автоматизация сети с помощью Cloud Assembly и NSX — часть 2](#) (включая использование существующей сети и исходящей сети)
- [Автоматизация сети с помощью Cloud Assembly и NSX — часть 3](#) (включая использование существующих групп безопасности и групп безопасности по требованию)
- [Автоматизация сети с помощью Cloud Assembly и NSX — часть 4](#) (включая использование существующей подсистемы балансировки нагрузки и подсистемы балансировки нагрузки по требованию)

## Дополнительные сведения о сетевых ресурсах в облачных шаблонах vRealize Automation

При создании или изменении облачных шаблонов vRealize Automation используйте сетевые ресурсы, которые лучше всего подходят для ваших целей. Ознакомьтесь с параметрами сети NSX и сети, независимой от облачной среды, доступными в облачном шаблоне.

Выберите один из трех доступных типов сетевых ресурсов с учетом компьютера и связанных условий в облачном шаблоне vRealize Automation.

### Независимый от облачной среды сетевой ресурс

Для добавления сети, независимой от облачной среды, можно использовать ресурс **Независимый от облачной среды > Сеть** на странице **Проектирование** облачного шаблона. Ресурс отображается в коде облачного шаблона как тип ресурса `Cloud.Network`. Ресурс по умолчанию отображается следующим образом:

```
Cloud_Network_1:
  type: Cloud.Network
  properties:
    networkType: existing
```

Если необходимо указать характеристики сети для типа целевого компьютера, который не подключен или не может быть подключен к сети NSX, используйте сеть, независимую от облачной среды.

Сетевой ресурс, независимый от облачной среды, доступен для следующих типов ресурсов.

- Компьютер, независимый от облачной среды
- vSphere
- Google Cloud Platform (GCP)
- Amazon Web Services (AWS)
- Microsoft Azure
- VMware Cloud on AWS (VMC)

Сетевой ресурс, независимый от облачной среды, доступен для следующих параметров типа сети (`networkType`).

- Общедоступный
- Частный
- Исходящий
- Существующий



## Сетевой ресурс vSphere

Чтобы добавить сеть vSphere, используйте ресурс **vSphere > Сеть** на странице **Проектирование** облачного шаблона. Ресурс отображается в коде облачного шаблона как тип ресурса `Cloud.vSphere.Network`. Ресурс по умолчанию отображается следующим образом:

```
Cloud_vSphere_Network_1:
  type: Cloud.vSphere.Network
  properties:
    networkType: existing
```

Если необходимо указать характеристики сети для типа компьютера vSphere (`Cloud.vSphere.Machine`), используйте сеть vSphere.

Сетевой ресурс vSphere доступен только для типа компьютера `Cloud.vSphere.Machine`.

Ресурс vSphere доступен для следующих параметров типа сети (`networkType`).

- Общедоступный
- Частный
- Существующий

См. пример в разделе [Использование параметров сети в профилях сетей и облачных шаблонах в vRealize Automation](#).

## Сетевой ресурс NSX

Чтобы добавить сеть NSX, используйте ресурс **NSX > Сеть** на странице **Проектирование** облачного шаблона. Ресурс отображается в коде облачного шаблона как тип ресурса `Cloud.NSX.Network`. Ресурс по умолчанию отображается следующим образом:

```
Cloud_NSX_Network_1:
  type: Cloud.NSX.Network
  properties:
    networkType: existing
```

Если необходимо подключить сетевой ресурс к одному или нескольким компьютерам, связанным с облачной учетной записью NSX-V или NSX-T, используйте сеть NSX. Сетевой ресурс NSX позволяет указать характеристики сети NSX для ресурса компьютера vSphere, который связан с облачной учетной записью NSX-V или NSX-T.

Ресурс `Cloud.NSX.Network` доступен для следующих параметров типа сети (`networkType`).

- Общедоступный
- Частный
- Исходящий
- Существующий
- routed — маршрутизируемые сети доступны только для NSX-V и NSX-T.

Если необходимо, чтобы несколько исходящих или маршрутизируемых сетей совместно использовали один и тот же маршрутизатор первого уровня NSX-T или шлюз служб Edge (ESG) NSX-V, перед первоначальным развертыванием подключите один ресурс шлюза NSX (`Cloud.NSX.Gateway`) к подключенным сетям в шаблоне. При добавлении шлюза после развертывания в рамках операции по регулярному обслуживанию или итеративной разработки каждая сеть создает свой собственный маршрутизатор.

Ресурс NAT NSX можно использовать в шаблоне для поддержки правил переадресации портов NAT и DNAT.

### Независимый от облачной среды сетевой ресурс для развертывания Azure, AWS или GCP

Для виртуальных машин общедоступных облачных сред могут потребоваться определенные комбинации свойств облачного шаблона, которые не всегда являются обязательными в развертываниях компьютеров на основе NSX или vSphere. Примеры кода облачного шаблона, который поддерживает некоторые подобные варианты использования, см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

### Ресурс шлюза NSX

В одном развертывании можно повторно или совместно использовать один маршрутизатор первого уровня NSX-T или шлюз служб Edge (ESG) NSX-V с помощью ресурса шлюза (`Cloud.NSX.Gateway`) в облачном шаблоне. Ресурс шлюза представляет собой службу первого уровня/ESG и может быть подключен к нескольким сетям в развертывании. Ресурс шлюза можно использовать только с сетями исходящего трафика или маршрутизируемыми сетями.

Ресурс `Cloud.NSX.Gateway` позволяет совместно использовать маршрутизатор первого уровня NSX-T или шлюз служб Edge (ESG) NSX-V в подключенных исходящих и маршрутизируемых сетях в развертывании.

Шлюз часто подключается к одной исходящей или маршрутизируемой сети. Тем не менее, если шлюз подключен к нескольким сетям, то все такие сети должны быть однотипными, например только исходящими или только маршрутизируемыми. Шлюз можно подключить к нескольким компьютерам или подсистемам балансировки нагрузки, которые подключены к одним и тем же исходящим или маршрутизируемым сетям. Шлюз должен быть подключен к подсистеме балансировки нагрузки в общедоступной сети по требованию, чтобы он мог повторно использовать маршрутизатор первого уровня NSX-T или шлюз служб Edge (ESG) NSX-V, созданный шлюзом.

Чтобы несколько исходящих или маршрутизируемых сетей могли использовать один и тот же маршрутизатор T1 или Edge, сначала необходимо подключить единый ресурс шлюза `Cloud.NSX.Gateway` ко всем сетям. Перед развертыванием облачного шаблона необходимо подключить друг к другу все целевые сети и единый шлюз. В противном случае каждая сеть создаст свой собственный маршрутизатор.

Для сети NSX, в которой содержится связанный шлюз вычислительных ресурсов, параметры шлюза применяются ко всем связанным сетям в развертывании. Для каждого развертывания создается один логический маршрутизатор уровня 1 NSX-T, который используется всеми сетями по требованию и подсистемами балансировки нагрузки в развертывании. Для каждого развертывания создается один маршрутизатор Edge NSX-V, который используется всеми сетями по требованию и подсистемами балансировки нагрузки в развертывании.

Ресурс шлюза можно присоединить к сети в ходе итеративного обновления развертывания. Однако при этом маршрутизатор T1 или Edge не будет создан: маршрутизатор создается при первоначальном развертывании сети.

Если для сетей NSX-T не используется связанный ресурс шлюза, число логических маршрутизаторов уровня 1, создаваемых в развертывании, будет соответствовать числу сетей по требованию в облачном шаблоне.

Если шлюз содержит правила NAT, для маршрутизатора первого уровня или маршрутизатора Edge можно перенастроить или удалить правила NAT или DNAT. Если шлюз изначально развернут без правил NAT, у него нет доступных действий по регулярному обслуживанию.

## NSX ресурса NAT

Ресурс `Cloud.NSX.NAT` позволяет присоединять правила DNAT и средства переадресации портов ко всем подключенным исходящим сетям с помощью ресурса шлюза. Ресурс NAT можно присоединить к ресурсу шлюза, для которого необходимо настроить правила DNAT.

---

**Примечание** Ресурс `Cloud.NSX.Gateway` был изначально доступен для правил DNAT. Однако использование `Cloud.NSX.Gateway` для определения правил DNAT и переадресации портов больше не поддерживается. Этот ресурс по-прежнему доступен в рамках обратной совместимости. Для правил DNAT и переадресации портов используйте ресурс облачного шаблона `Cloud.NSX.NAT`. При попытке использования типа ресурса `Cloud.NSX.Gateway` со спецификациями правил NAT в облачном шаблоне появится предупреждение.

---

Если ресурс `Cloud.NSX.NAT` подключен к сети исходящего трафика NSX-V или NSX-T, он поддерживает правила DNAT и переадресацию портов.

Параметр правил NAT для ресурса — `natRules:`. Чтобы настроить записи `natRules:` в шлюзе, ресурс NAT можно присоединить к ресурсу шлюза. В правилах DNAT, установленных в ресурсе, в качестве целевых объектов используются связанные компьютеры или подсистемы балансировки нагрузки.

Можно перенастроить сетевой адаптер компьютера или шлюз вычислительных ресурсов в существующем развертывании для изменения параметров `natRules:` путем добавления, упорядочения, редактирования или удаления правил переадресации портов DNAT. Правила DNAT нельзя использовать с кластерными компьютерами. Правила DNAT можно задать для отдельных компьютеров в кластере путем выполнения операции по регулярному обслуживанию.

## Параметры интеграции внешнего поставщика услуг управления IP-адресами

Сведения о свойствах, которые можно использовать в интеграциях IPAM Infoblox в проектах и развертываниях облачных шаблонов, см. в разделе [Использование свойств и расширяемых атрибутов, связанных с Infoblox, для интеграций IPAM в облачных шаблонах vRealize Automation](#).

## Ограничения по использованию назначения статического IP-адреса в облачном шаблоне

Назначение статического IP-адреса в облачном шаблоне vRealize Automation можно применять только при использовании vRealize Automation IPAM, то есть внутреннего IPAM, поставляемого вместе с vRealize Automation, или IPAM, образованного на основе подключаемого модуля внешнего поставщика, который был создан с помощью комплекта SDK для vRealize Automation IPAM, например одного

из подключаемых модулей Infoblox, которые можно загрузить с vRealize Automation Marketplace. Применение назначения статического IP-адреса (`assignment:static`) в облачном шаблоне не поддерживается в случае использования темы события «Настройка сети» (которая используется либо действием расширяемости Cloud Assembly (ABX), либо рабочим процессом vRealize Orchestrator). Неподдерживаемые назначения статического IP-адреса могут привести к сбою развертывания.

### Значение адреса в разделе «Общее» развернутого облачного шаблона

При проверке развернутого облачного шаблона значение **адреса** в разделе **Общее** — основной IP-адрес компьютера. Часто основной адрес — это общедоступный или иной доступный адрес компьютера. В развертываниях vSphere основной IP-адрес вычисляется vRealize Automation. При определении основного IP-адреса учитываются и ранжируются все IP-адреса всех сетевых адаптеров, включая их общедоступные, частные, статические, динамические и IPv6-свойства. Для развертываний, которые не относятся к vSphere, основной IP-адрес компьютера вычисляется системой ранжирования каждого поставщика облачной среды.

### Доступные операции регулярного обслуживания

Список типовых операций по регулярному обслуживанию, доступных для облачного шаблона и ресурсов развертывания, см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

Пример перемещения из одной сети в другую см. в разделе [Перемещение развернутого компьютера в другую сеть](#).

### Подробнее

Связанные с этой темой сведения и примеры сетевых ресурсов и настроек см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

Дополнительные сведения об определении сетевых ресурсов см. в разделе [Сетевые ресурсы в vRealize Automation](#).

Сведения об определении профилей сетей см. в разделе [Дополнительные сведения о профилях сетей в vRealize Automation](#).

### Дополнительные сведения о ресурсах групп безопасности и тегов в облачных шаблонах vRealize Automation

При создании или изменении облачных шаблонов vRealize Automation используйте ресурсы безопасности, которые лучше всего подходят для ваших целей.

## Независимый от облачной среды ресурс группы безопасности

Чтобы добавить ресурс группы безопасности, используйте ресурс **Независимый от облачной среды > Группа безопасности** на странице проектирования облачного шаблона. Ресурс отображается в коде облачного шаблона как тип ресурса `Cloud.SecurityGroup`. Ресурс по умолчанию отображается следующим образом:

```
Cloud_SecurityGroup_1:
  type: Cloud.SecurityGroup
  properties:
    constraints: []
    securityGroupType: existing
```

Ресурс группы безопасности указывается в проекте облачного шаблона как «существующий» (`securityGroupType: existing`) или «по требованию» (`securityGroupType: new`).

Существующую группу безопасности можно добавить в облачный шаблон, или можно использовать существующую группу безопасности, добавленную в профиль сети.

В NSX-V и NSX-T, а также в NSX-T с включенным диспетчером политик в сочетании с VMware Cloud on AWS при проектировании или изменении облачного шаблона можно добавить существующую группу безопасности или определить новую. Группы безопасности по требованию поддерживаются для NSX-T и NSX-V, а также для VMware Cloud on AWS при использовании с диспетчером политик NSX-T.

Для всех типов учетных записей облачной службы, кроме Microsoft Azure, можно связать одну или несколько групп безопасности с сетевым адаптером компьютера. Сетевой адаптер виртуальной машины Microsoft Azure (*machineName*) можно связать только с одной группой безопасности.

По умолчанию для свойства группы безопасности `securityGroupType` задано значение `existing`. Чтобы создать группу безопасности по требованию, введите значение `new` для свойства `securityGroupType`. Чтобы указать правила брандмауэра для группы безопасности по требованию, используйте свойство `rules` в разделе `Cloud.SecurityGroup` ресурса группы безопасности.

## Существующие группы безопасности

Существующие группы безопасности создаются в исходном ресурсе облачной учетной записи, например NSX-T или Amazon Web Services. Они представляют собой данные, которые vRealize Automation собирает из источника. Существующую группу безопасности можно выбрать в списке доступных ресурсов в рамках профиля сети vRealize Automation. В проекте облачного шаблона можно указать существующую группу безопасности либо по фактическому наличию в указанном профиле сети, либо по имени, используя параметр `securityGroupType: existing` ресурса группы безопасности. При добавлении группы безопасности в профиль сети добавьте в профиль сети хотя бы один тег возможности. При использовании ресурсов группы безопасности по требованию в проекте облачного шаблона требуется тег ограничения.

В проекте облачного шаблона ресурс группы безопасности можно связать с одним ресурсом компьютера или несколькими.

---

**Примечание** Если ресурс компьютера планируется использовать в проекте облачного шаблона для подготовки сетевого адаптера виртуальной машины (*machineName*) Microsoft Azure, этот ресурс необходимо связать только с одной группой безопасности.

---

### Группы безопасности по требованию

Группы безопасности по требованию можно настроить при определении или изменении проекта облачного шаблона с помощью параметра `securityGroupType: new` в коде ресурса группы безопасности.

С помощью группы безопасности по требованию для NSX-V и NSX-T, а также для Amazon Web Services (при условии использования с типом NSX-T Policy) можно применить определенный набор правил брандмауэра к ресурсу компьютера, подключенного к сети, или к набору объединенных ресурсов. Каждая группа безопасности может содержать несколько именованных правил брандмауэра. Группу безопасности по требованию можно использовать, чтобы указать службы или протоколы и порты. Следует отметить, что можно указать либо службу, либо протокол, но не то и другое одновременно. Кроме протокола, можно указать порт. Если указана служба, то порт указать нельзя. Если правило не содержит ни службу, ни протокол, значение службы по умолчанию Any.

Кроме того, в правилах брандмауэра можно указать IP-адреса и диапазоны IP-адресов. Некоторые примеры правил брандмауэра приведены в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

При создании правил брандмауэра в группе безопасности по требованию NSX-V или NSX-T значение по умолчанию разрешает указанный сетевой трафик, а также разрешает другой сетевой трафик. Для управления сетевым трафиком необходимо указать тип доступа для каждого правила. Типы правил для доступа

- Разрешить (по умолчанию). Разрешает сетевой трафик, указанный в данном правиле брандмауэра.
- Запретить (по умолчанию). Блокирует сетевой трафик, указанный в данном правиле брандмауэра. Активно сообщает клиенту, что подключение отклонено.
- Отклонить. Отклоняет сетевой трафик, указанный в правиле брандмауэра. Автоматически отбрасывает пакет, как если бы прослушиватель не был подключен к Интернету.

Пример проекта, в котором используется `access: Allow` и правило брандмауэра `access: Deny`, см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

---

**Примечание** Администратор облачных систем может создать проект облачного шаблона, содержащий только группу безопасности NSX по требованию, и может развернуть этот проект, чтобы создать повторно используемый ресурс существующей группы безопасности, который участники организации могут добавить в профили сети и проекты облачных шаблонов в качестве существующей группы безопасности.

---

Правила брандмауэра поддерживают значения CIDR в формате IPv4 или IPv6 для исходного и конечного IP-адресов. Пример проекта, в котором используются значения IPv6 CIDR в правиле брандмауэра, см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Существующие группы безопасности и группы безопасности по требованию для VMware Cloud on AWS

Группы безопасности по требованию можно определить для компьютера VMware Cloud on AWS в облачном шаблоне с помощью параметра `securityGroupType: new` в коде ресурса группы безопасности.

Ниже приведен пример фрагмента кода для группы безопасности по требованию.

```
resources:
  Cloud_SecurityGroup_1:
    type: Cloud.SecurityGroup
    properties:
      name: vmc-odsg
      securityGroupType: new
      rules:
        - name: datapath
          direction: inbound
          protocol: TCP
          ports: 5011
          access: Allow
          source: any
```

Также можно определить существующую группу безопасности для компьютера VMware Cloud on AWS, подключенного к сети, и при необходимости добавить теги ограничений, как показано в следующих примерах:

```
Cloud_SecurityGroup_2:
  type: Cloud.SecurityGroup
  properties:
    constraints: [xyz]
    securityGroupType: existing
```

```
Cloud_SecurityGroup_3:
  type: Cloud.SecurityGroup
  properties:
    securityGroupType: existing
    constraints:
      - tag: xyz
```

Поддерживается итеративная разработка облачных шаблонов.

- Если группа безопасности связана с одним или несколькими компьютерами в развертывании, при выполнении действия удаления отображается сообщение о том, что группу безопасности нельзя удалить.
- Если группа безопасности не связана с компьютерами в развертывании, то при выполнении действия удаления отображается сообщение о том, что группа безопасности будет удалена из этого развертывания и это действие будет нельзя отменить. Существующая группа безопасности удаляется из облачного шаблона, а группа безопасности по требованию уничтожается.

## Использование тегов безопасности NSX-V и тегов виртуальных машин NSX-T

Теги безопасности NSX-V, а также теги ВМ NSX-T и NSX-T с API-интерфейсом Policy можно просматривать и использовать в управляемых ресурсах облачных шаблонов vRealize Automation.

Теги безопасности NSX-V и NSX-T могут использоваться с vSphere. Также поддерживается использование тегов безопасности NSX-T с VMware Cloud on AWS.

---

**Примечание** Как и для виртуальных машин, развертываемых в vSphere, теги компьютера можно настроить для виртуальной машины, развертываемой в VMware Cloud on AWS. Тег компьютера также можно обновить после первоначального развертывания. Эти теги компьютера позволяют vRealize Automation динамически назначать виртуальную машину соответствующей группе безопасности NSX-T во время развертывания.

---

Для указания тегов безопасности NSX-V можно использовать `key: nsxSecurityTag` и значение тега в вычислительном ресурсе облачного шаблона, как показано в следующем примере, при условии, что компьютер подключен к сети NSX-V.

```
tags:
  - key: nsxSecurityTag
    value: security_tag_1
  - key: nsxSecurityTag
    value: security_tag_2
```

Указанное значение должно соответствовать тегу безопасности NSX-V. Если в NSX-V нет тегов безопасности, соответствующих указанному значению ключа `nsxSecurityTag`, произойдет сбой развертывания.

---

**Примечание** Установка тегов безопасности NSX-V требует подключения компьютера к сети NSX-V. Если компьютер подключен к сети vSphere, установка тегов безопасности NSX-V игнорируется. В любом случае для компьютера vSphere также устанавливаются теги.

---

В NSX-T нет отдельного тега безопасности. При указании любого тега в вычислительном ресурсе облачного шаблона развернутая ВМ будет связана со всеми тегами, указанными в NSX-T. Для NSX-T, включая NSX-T с API-интерфейсом Policy, теги ВМ также указываются в облачном шаблоне в виде пары «ключ — значение». Параметр `key` идентичен параметру `scope` в NSX-T, а параметр `value` идентичен параметру `Tag Name`, заданному в NSX-T.

Следует отметить, что если для переноса облачных учетных записей из NSX-V в NSX-T, включая NSX-T с API-интерфейсом Policy, используется помощник по переносу vRealize Automation NSX-V в NSX-T, то данный помощник создает пару «ключ — значение» `nsxSecurityTag`. В этом сценарии, или если облачный шаблон по какой-либо причине явно предлагает использовать `nsxSecurityTag` с NSX-T, включая NSX-T с API-интерфейсом Policy, развертывание создает тег ВМ с пустым параметром «Область» с именем тега, который совпадает с указанным значением `value`. При просмотре таких тегов в NSX-T столбец «Область» будет пустым.



Чтобы избежать путаницы, не используйте пары ключей `nsxSecurityTag` для NSX-T. Если вы указываете пару «ключ — значение» `nsxSecurityTag` для использования с NSX-T, включая NSX-T с API-интерфейсом Policy, развертывание создает тег VM с пустым параметром «Область» с именем тега, совпадающим с указанным значением `value`. При просмотре таких тегов в NSX-T столбец «Область» будет пустым.

### Использование политик изоляции приложений в правилах брандмауэра для группы безопасности по требованию

Чтобы разрешить только внутренний трафик между ресурсами, подготовленными в облачном шаблоне, можно использовать политику изоляции приложений. Благодаря изоляции приложений компьютеры, подготовленные с помощью облачного шаблона, могут обмениваться данными между собой, но не могут устанавливать соединения за пределами брандмауэра. Политику изоляции приложений можно создать в профиле сети. Кроме того, изоляцию приложений можно указать в проекте облачного шаблона, используя группу безопасности по требованию с правилом брандмауэра «Запретить», либо частную или исходящую сеть.

Политика изоляции приложений создается с низким приоритетом. При применении нескольких политик приоритет будет иметь политика с более высоким значением.

При создании политики изоляции приложений ее имя создается автоматически. Кроме того, политика доступна для повторного использования в других проектах облачных шаблонов и итерациях проектов, относящихся к конечной точке и проекту связанного ресурса. Имя политики изоляции приложений не отображается в облачном шаблоне, но отображается в виде настраиваемого свойства на странице проекта (**Инфраструктура > Администрирование > Проекты**) после развертывания проекта облачного шаблона.

Для той же связанной конечной точки в проекте любое развертывание, в котором для изоляции приложений требуется группа безопасности по требованию, может использовать ту же политику изоляции приложений. После создания политика не удаляется. При указании политики изоляции приложений система vRealize Automation выполняет поиск политики в проекте применительно к связанной конечной точке. Если политика будет найдена, она используется повторно. В противном случае создается новая политика. Имя политики изоляции приложений отображается только после первоначального развертывания в списке настраиваемых свойств проекта.

### Использование групп безопасности в итеративной разработке облачных шаблонов

Если группа безопасности не связана с компьютером в облачном шаблоне, то при изменении ограничений в процессе итеративной разработки группа безопасности обновляется в соответствии с параметрами итерации. Однако, если группа безопасности уже связана с компьютером, при повторном развертывании возникнет сбой. Во время итеративной разработки облачного шаблона следует отменять привязку существующих групп безопасности и (или) свойств ресурса `securityGroupType` к связанным компьютерам, а затем повторно привязывать их перед каждым следующим развертыванием. Если облачный шаблон уже развернут, необходимо выполнить следующие действия.

1. В конструкторе шаблонов Cloud Assembly отключите группу безопасности от всех компьютеров, связанных с ней в облачном шаблоне.
2. Повторно разверните шаблон, выбрав вариант **Обновить существующее развертывание**.

3. Удалите существующие теги ограничений группы безопасности и (или) свойства `securityGroupType` в шаблоне.
4. Добавьте новые теги ограничений группы безопасности и (или) свойства `securityGroupType` в шаблон.
5. Привяжите новые теги ограничений группы безопасности и (или) экземпляры свойств `securityGroupType` к компьютерам в шаблоне.
6. Повторно разверните шаблон, выбрав вариант **Обновить существующее развертывание**.

### Доступные операции регулярного обслуживания

Список типовых операций по регулярному обслуживанию, доступных для облачного шаблона и ресурсов развертывания, см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

### Подробнее

Сведения об использовании группы безопасности для изоляции сети см. в разделе [Ресурсы безопасности в vRealize Automation](#).

Сведения об использовании групп безопасности в профилях сети см. в разделах [Дополнительные сведения о профилях сетей в vRealize Automation](#) и [Использование параметров группы безопасности в профилях сетей и проектах облачных шаблонов в vRealize Automation](#).

Примеры использования групп безопасности в облачных шаблонах см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Дополнительные сведения о ресурсах подсистем балансировки нагрузки в облачных шаблонах vRealize Automation

При создании или изменении облачных шаблонов vRealize Automation используйте ресурсы подсистемы балансировки нагрузки, которые лучше всего подходят для ваших целей.

Для управления балансировкой нагрузки в развертывании можно использовать в облачном шаблоне ресурсы подсистемы балансировки нагрузки NSX, а также ресурсы подсистемы балансировки нагрузки, независимые от облачной среды.

Независимая от облака подсистема балансировки нагрузки может быть развернута в нескольких облаках. Для подсистемы балансировки нагрузки, предназначенной для облачной среды, можно указать дополнительные параметры и функции, доступные только для определенной облачной среды или топологии. Специальные свойства для облачной среды доступны в типе ресурса подсистемы балансировки нагрузки (Cloud.NSX.LoadBalancer) NSX. При добавлении этих свойств в подсистему балансировки нагрузки, независимую от облачной среды (Cloud.LoadBalancer), они будут игнорироваться, если, например, предоставлена подсистема балансировки нагрузки Amazon Web Services или Microsoft Azure, но будут учитываться, если предоставлена подсистема балансировки нагрузки NSX-V или NSX-T. Выберите один из доступных типов ресурсов подсистем балансировки нагрузки в зависимости от условий в облачном шаблоне vRealize Automation.

Ресурс подсистемы балансировки нагрузки нельзя подключить напрямую к ресурсу группы безопасности на холсте проекта.

## Независимый от облачной среды ресурс подсистемы балансировки нагрузки

Если необходимо указать характеристики сети для целевого компьютера любого типа, используйте независимую от облачной среды подсистему балансировки нагрузки.

Для добавления независимой от облачной среды подсистемы балансировки нагрузки можно использовать ресурс **Независимый от облачной среды > Подсистема балансировки нагрузки** на странице проектирования облачного шаблона. Ресурс отображается в коде облачного шаблона как тип ресурса `Cloud.LoadBalancer`. Ресурс по умолчанию отображается следующим образом:

```
Cloud_LoadBalancer_1:
  type: Cloud.LoadBalancer
  properties:
    routes: []
    network: ''
    instances: []
    internetFacing: false
```

## ресурс подсистемы балансировки нагрузки NSX

Если облачный шаблон содержит характеристики, относящиеся к NSX-V или NSX-T (методы Policy API или Manager API), используйте подсистему балансировки нагрузки NSX. К сети NSX-V или NSX-T и к компьютерам, связанным с сетью NSX-V или NSX-T, можно присоединить одну или несколько подсистем балансировки нагрузки.

Чтобы добавить подсистему балансировки нагрузки NSX, используйте ресурс **NSX > Подсистема балансировки нагрузки**. Ресурс отображается в коде облачного шаблона как тип ресурса `Cloud.NSX.LoadBalancer`. Ресурс по умолчанию отображается следующим образом:

```
Cloud_NSX_LoadBalancer_1:
  type: Cloud.NSX.LoadBalancer
  properties:
    routes: []
    network: ''
    instances: []
```

## Параметры подсистемы балансировки нагрузки в коде облачного шаблона

Добавление одного ресурса подсистемы балансировки нагрузки или нескольких в облачный шаблон позволяет указать следующие параметры. Некоторые примеры доступны в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

Протокол HTTP поддерживается для всех подсистем балансировки нагрузки по требованию.

Протокол HTTPS поддерживается только для подсистем балансировки нагрузки по требованию, связанных с облачной учетной записью NSX-T, для режима NSX которой задано значение **Policy**. Облачные учетные записи NSX-T, для режима NSX которых задано значение **Manager**, не могут использовать протокол HTTPS.

### ■ Спецификация компьютера

Можно указать именованные ресурсы компьютеров для включения в пул балансировки нагрузки. Кроме того, можно указать сетевой адаптер компьютера, который будет включен в пул подсистемы балансировки нагрузки.

Этот параметр доступен только для ресурса подсистемы балансировки нагрузки **NSX** (Cloud.NSX.LoadBalancer).

- ресурс. *Cloud\_Machine\_1.id*

Указывает, что подсистема балансировки нагрузки включает в себя компьютер, помеченный в коде облачного шаблона как *Cloud\_Machine\_1*.

- ресурс. *Cloud\_Machine\_2.networks[2].id*

Указывает, что подсистема балансировки нагрузки должна включать в себя компьютер, обозначенный в коде облачного шаблона как *Cloud\_Machine\_2*, только если он развернут на сетевом адаптере компьютера *Cloud\_Machine\_2.networks[2]*.

- Уровень ведения журналов

Значение уровня ведения журнала указывает уровень серьезности для журнала ошибок. Существуют следующие значения: NONE, EMERGENCY, ALERT, CRITICAL, ERROR, WARNING, INFO, DEBUG и NOTICE. Значение уровня ведения журнала применяется ко всем подсистемам балансировки нагрузки в облачном шаблоне. Этот параметр относится только к NSX. Для подсистем балансировки нагрузки, у которых есть родительский компонент, значение уровня ведения журнала переопределяет любое значение уровня ведения журнала для его дочерних элементов.

Дополнительные сведения см. в разделе [Добавление подсистем балансировки нагрузки](#) в документации по NSX.

- Тип

Используйте тип подсистемы балансировки нагрузки, чтобы указать объем масштабирования. Значение по умолчанию — small. Этот параметр относится только к NSX. Для подсистем балансировки нагрузки, у которых есть родительский компонент, значение родительского типа переопределяет любое значение типа для его дочерних элементов.

- Маленький

Соответствует compact в NSX-V и small в NSX-T.

- Средний

Соответствует large в NSX-V и medium в NSX-T.

- Большой

Соответствует quad-large в NSX-V и large в NSX-T.

- Очень большой

Соответствует xlarge в NSX-V и large в NSX-T.

Дополнительные сведения см. в соответствующих разделах, например [Масштабирование ресурсов подсистемы балансировки нагрузки](#) в документации по NSX.

Этот параметр доступен для ресурса подсистемы балансировки нагрузки **NSX** (`Cloud.NSX.LoadBalancer`).

#### ■ Алгоритм (пул серверов)

Используйте метод балансировки `algorithm` (алгоритм), чтобы контролировать способ распределения входящих подключений между участниками пула серверов. Алгоритм можно использовать в пуле серверов или непосредственно на сервере. Все алгоритмы балансировки нагрузки пропускают серверы, которые удовлетворяют любому из следующих условий.

- Для состояния администратора установлено значение `DISABLED`.
- Для состояния администратора установлено значение `GRACEFUL_DISABLED`, а соответствующая запись об устойчивости отсутствует.
- Активное или пассивное состояние проверки работоспособности — `DOWN`.
- Достигнуто максимальное количество одновременных подключений пула серверов.

Этот параметр относится только к **NSX**.

#### ■ `IP_HASH`

Выбирает сервер на основании хеша IP-адреса источника и общего веса всех работающих серверов.

Соответствует `IP-HASH` в **NSX-V** и **NSX-T**.

#### ■ `LEAST_CONNECTION`

Клиентские запросы распределяются между серверами в зависимости от количества подключений. Новые подключения отправляются на тот сервер, где их меньше. Вес участников пула серверов игнорируется, даже если он задан.

Соответствует `LEASTCONN` в **NSX-V** и `LEAST_CONNECTION` в **NSX-T**.

#### ■ `ROUND_ROBIN`

Входящие запросы клиентов проходят циклически по списку доступных серверов, которые способны обработать запрос. Вес членов пула серверов игнорируется, даже если он задан. По умолчанию.

Соответствует `ROUND_ROBIN` в **NSX-V** и **NSX-T**.

#### ■ `WEIGHTED_LEAST_CONNECTION`

Каждому серверу присваивается весовое значение, которое учитывает количество отправляемых на сервер запросов и указывает его производительность относительно других серверов в пуле. Задача этого алгоритма — равномерно распределить нагрузку между доступными ресурсами сервера. По умолчанию, если весовое значение не задано, оно равно 1 и включен параметр медленного запуска.

Соответствует `WEIGHTED_LEAST_CONNECTION` в **NSX-T**. Соответствующий параметр в **NSX-V** отсутствует.

#### ■ `WEIGHTED_ROUND_ROBIN`

Каждому серверу присваивается весовое значение, которое учитывает количество отправляемых на сервер запросов и указывает его производительность относительно других серверов в пуле. Задача этого алгоритма — равномерно распределить нагрузку между доступными ресурсами сервера.

Соответствует WEIGHTED\_ROUND\_ROBIN в NSX-T. Соответствующий параметр в NSX-V отсутствует.

#### ■ URI

Левая часть кода URI хешируется и делится на общий вес работающих серверов. Результат указывает сервер, который получает запрос. Благодаря этому URI всегда направляется на один и тот же сервер, пока количество работающих серверов остается неизменным. Этот параметр алгоритма URI имеет два компонента: `uriLength=<len>` и `uriDepth=<dep>`. Диапазон параметра длины должен быть  $1 \leq len < 256$ . Диапазон параметра глубины должен быть  $1 \leq dep < 10$ .

Параметры длины и глубины должны быть положительными целыми числами. Эти компоненты могут выполнять балансировку серверов, основываясь только на первой части URI. Параметр длины указывает, что алгоритм должен принимать во внимание только определенные символы в начале URI для вычисления хеша. Параметр глубины определяет максимальную глубину каталога, которую следует использовать для вычисления хеша. С каждой косой чертой в запросе добавляется один уровень. Если определены оба параметра, проверка прекращается при достижении любого из них.

Соответствует URI в NSX-V. Соответствующий параметр в NSX-T отсутствует.

#### ■ HTTPHEADER

Имя заголовка HTTP проверяется в каждом запросе HTTP. Имя заголовка в круглых скобках не зависит от регистра. Если заголовок отсутствует или не содержит никакого значения, применяется алгоритм циклического перебора (round robin). Параметр алгоритма HTTPHEADER имеет один вариант: `headerName=<name>`.

Соответствует HTTPHEADER в NSX-V. Соответствующий параметр в NSX-T отсутствует.

#### ■ URL-адрес

Поиск параметра URL-адреса, указанного в аргументе, выполняется в строке запроса для каждого запроса HTTP GET. Если за параметром следуют знак равенства (=) и значение, это значение хешируется и делится на общий вес работающих серверов. Результат указывает сервер, который получает запрос. Этот процесс используется для отслеживания идентификаторов пользователей в запросах, благодаря чему один и тот же идентификатор пользователя всегда отправляется на один и тот же сервер, пока количество работающих серверов остается неизменным. Если значение или параметр не найдены, применяется алгоритм циклического перебора. Этот параметр алгоритма URL-адреса имеет один вариант: `urlParam=<url>`.

Соответствует URL в NSX-V. Соответствующий параметр в NSX-T отсутствует.

Дополнительные сведения см. в соответствующих разделах, например [Добавление пула серверов для балансировки нагрузки](#) в документации по продукту NSX.

#### ■ Монитор работоспособности

С помощью параметров монитора работоспособности проверяйте, доступен ли сервер. Поддерживается активный мониторинг работоспособности для протоколов HTTP, ICMP, TCP и UDP. Пассивный мониторинг работоспособности доступен только для NSX-T.

Этот параметр относится только к NSX.

- **httpMethod**

Метод HTTP, используемый для определения состояния сервера при запросе проверки работоспособности. Методы: GET, HOST, OPTIONS, HEAD или PUT.

- **requestBody**

Содержимое текста запроса проверки работоспособности. Используется протоколами HTTP, TCP и UDP и является обязательным для них.

- **responseBody**

Содержимое текста предполагаемого ответа на запрос проверки работоспособности. Если полученная строка соответствует тексту ответа, сервер считается работоспособным. Используется протоколами HTTP, TCP и UDP и является обязательным для них.

---

**Примечание** При использовании протокола монитора UDP обязательными параметрами являются UDP Data Sent и UDP Data Expected. Свойства requestBody и responseBody сопоставляются с этими параметрами.

---

Этот параметр доступен для ресурса подсистемы балансировки нагрузки NSX (Cloud.NSX.LoadBalancer).

Дополнительные сведения см. в соответствующих разделах, например [Настройка активного монитора работоспособности](#) в документации по NSX.

- **Проверка работоспособности**

С помощью параметров проверки работоспособности укажите, как подсистема балансировки нагрузки должна выполнять проверки работоспособности.

Этот параметр доступен только для ресурса подсистемы балансировки нагрузки NSX (Cloud.NSX.LoadBalancer).

Пример доступных настроек проверки работоспособности см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Типы сетей NSX-V и NSX-T, а также параметры подсистем балансировки нагрузки

Параметры подсистемы балансировки нагрузки зависят от сети, с которой компонент подсистемы балансировки нагрузки связан в облачном шаблоне. Подсистему балансировки нагрузки можно настроить с учетом типа и условий сети.

- **Сеть по требованию**

Если вычислительные ресурсы подсистемы балансировки нагрузки привязаны к сети по требованию, создается новый маршрутизатор первого уровня, который подключается к маршрутизатору нулевого уровня, указанному в профиле сети. После этого подсистема балансировки нагрузки привязывается к маршрутизатору первого уровня. Объявление виртуального IP-адреса маршрутизатора первого уровня включается, если виртуальный IP-адрес находится в существующей сети. Если для DHCP настроена сеть по требованию, то эта сеть по требованию и подсистема балансировки нагрузки совместно используют маршрутизатор первого уровня.

#### ■ Существующая сеть

Если подсистема балансировки нагрузки подключена к существующей сети, создается подсистема балансировки нагрузки с маршрутизатором первого уровня существующей сети. Создается новая подсистема балансировки нагрузки, если к маршрутизатору первого уровня не подключены подсистемы балансировки нагрузки. Если подсистема балансировки нагрузки уже существует, новые виртуальные серверы подключаются к ней. Если существующая сеть не подключена к маршрутизатору первого уровня, создается новый маршрутизатор первого уровня, который подключается к маршрутизатору нулевого уровня, определенному в профиле сети, и объявление виртуального IP-адреса маршрутизатора первого уровня не включается.

vRealize Automation не поддерживает двунаправленную (встроенную) подсистему балансировки нагрузки NSX-T в двух разных существующих сетях. Обратите внимание, что при использовании двунаправленной подсистемы балансировки нагрузки канал исходящей связи виртуального IP-адреса находится в существующей сети, а компьютеры, входящие в пул, подключены к сети по требованию. Чтобы при использовании существующей сети можно было выбрать подсистему балансировки нагрузки, необходимо настроить однонаправленную подсистему балансировки нагрузки. В ней используется одна и та же существующая сеть для виртуального IP-адреса подсистемы и компьютеров в пуле. Однако, начиная с vRealize Automation 8.4.2, при использовании подсистемы балансировки нагрузки, выбранной в профиле сети, можно балансировать нагрузку между компьютерами в двух разных существующих сетях, если между этими сетями существует возможность подключения.

#### ■ Изоляция сети, определенная в профиле сети

Для типов сетей `outbound` или `private` в профиле сети можно указать параметры изоляции сети, чтобы смоделировать новую группу безопасности. Так как компьютеры подключены к существующей сети, а в профиле определены параметры изоляции, этот параметр аналогичен подсистеме балансировки нагрузки, созданной в существующей сети. Разница заключается в том, что IP-адрес порта исходящей связи первого уровня добавляется в группу безопасности изоляции, чтобы активировать путь к данным.

Параметры подсистемы балансировки нагрузки для сетей, связанных с NSX, можно задать с помощью ресурса подсистемы балансировки нагрузки NSX в проекте облачного шаблона.

Дополнительные сведения см. в записи блога [VMware vRA Cloud Assembly Load Balancer with NSX-T Deep Dive \(Подсистема балансировки нагрузки vRA Cloud Assembly с NSX-T Deep Dive\)](#).



**Перенастройка параметров уровня или типа ведения журнала в том случае, если несколько подсистем балансировки нагрузки используют маршрутизатор уровня 1 NSX-T или пограничный маршрутизатор NSX-V**

Если используется облачный шаблон, содержащий несколько подсистем балансировки нагрузки, которые совместно используют маршрутизатор уровня 1 в конечной точке NSX-T или пограничный маршрутизатор в конечной точке NSX-V, перенастройка параметров типа и уровня ведения журнала в одном из ресурсов подсистемы балансировки нагрузки не влечет за собой обновление параметров других подсистем балансировки нагрузки. Несоответствие параметров приводит к несогласованности в NSX. Чтобы избежать несогласованности при перенастройке этих параметров типа и (или) уровня ведения журнала, используйте одинаковые значения перенастройки для всех ресурсов подсистемы балансировки нагрузки в облачном шаблоне, которые совместно используют маршрутизатор уровня 1 или пограничный маршрутизатор в связанной конечной точке NSX.

### Доступные операции регулярного обслуживания

При горизонтальном или вертикальном масштабировании развертывания, в котором содержится подсистема балансировки нагрузки, эта подсистема настраивается так, чтобы в нее попадали добавленные компьютеры или чтобы останавливать работу компьютеров балансировки нагрузки, которые планируется отключить.

Список типовых операций по регулярному обслуживанию, доступных для облачных шаблонов и развертываний, см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

### Подробнее

Сведения об определении параметров подсистемы балансировки нагрузки в профиле сети см. в разделе [Дополнительные сведения о профилях сетей в vRealize Automation](#).

Примеры проектов облачных шаблонов, содержащих подсистемы балансировки нагрузки, см. в разделе [Сети, ресурсы безопасности и подсистемы балансировки нагрузки в vRealize Automation](#).

## Облачный шаблон с возможностями Puppet и доступом по имени пользователя и паролю

В этом примере управление конфигурацией Puppet добавляется в облачный шаблон, развернутый на базе вычислительного ресурса vCenter, с доступом по имени пользователя и паролю.

В этой процедуре приведен пример создания развертываемого ресурса с поддержкой Puppet, для которого требуется проверка подлинности с помощью имени пользователя и пароля. Доступ по имени пользователя и паролю означает, что пользователь должен вручную войти на основной компьютер Puppet из вычислительного ресурса, чтобы запустить управление конфигурацией Puppet.

Кроме того, можно настроить проверку подлинности в режиме удаленного доступа, которая настраивает управление конфигурацией в облачном шаблоне, чтобы вычислительный ресурс выполнял проверку подлинности с помощью основного компьютера Puppet. Если удаленный доступ включен, вычислительный ресурс автоматически создает ключ для проверки подлинности с помощью пароля. Допустимое имя пользователя по-прежнему является обязательным.

Дополнительные примеры настройки различных сценарии Puppet в схемах элементов Cloud Assembly см. в разделах [Примеры облачных шаблонов для управления конфигурацией AWS Puppet](#) и [Примеры облачных шаблонов vCenter с конфигурацией Puppet](#).

### Необходимые условия

- Настройте экземпляр Puppet Enterprise в соответствующей сети.
- Добавьте экземпляр Puppet Enterprise в Cloud Assembly с помощью компонента «Интеграции». См. раздел [Настройка интеграции Puppet Enterprise в службе Cloud Assembly](#).
- Настройте учетную запись vSphere и вычислительный ресурс vCenter.

## Процедура

1. Добавьте компонент управления конфигурацией Puppet к вычислительному ресурсу vSphere на холсте требуемого облачного шаблона.
  - а) Выберите **Инфраструктура > Управление > Интеграции**.
  - б) Щелкните **Добавить интеграцию** и выберите Puppet.
  - в) Введите соответствующую информацию на странице «Конфигурация Puppet».

Конфигурация	Описание	Пример значения
Hostname	Имя узла или IP-адрес основного компьютера Puppet.	Puppet-Ubuntu
SSH Port	Порт SSH для обмена данными между Cloud Assembly и основным компьютером Puppet. (Необязательно)	Н/д
Autosign secret	Общий секретный ключ, настроенный на основном компьютере Puppet, который должны предоставлять узлы для поддержки запросов самозаверяющего сертификата.	Отдельно для каждого пользователя
Размещение	Указывает, находится ли основной компьютер Puppet в частном или общедоступном облаке.  <b>Примечание</b> Развертывание в нескольких облачных средах поддерживается только при наличии связи между вычислительным ресурсом развертывания и основным компьютером Puppet.	
Cloud proxy	Не требуется для общедоступных учетных записей облачной службы, например, Microsoft Azure или Amazon Web Services. При использовании учетной записи облачной службы на основе vCenter выберите соответствующий cloud proxy для своей учетной записи.	Н/д
Имя пользователя	Имя пользователя SSH и RBAC для основного компьютера Puppet.	Отдельно для каждого пользователя Значение YAML: "\${input.username}"
Пароль	Пароль SSH и RBAC для основного компьютера Puppet.	Отдельное значение YAML для пользователя: "\${input.password}"
Использовать команды sudo для этого пользователя	Выберите режим использования команд sudo для procidd.	истина
Имя	Имя основного компьютера Puppet.	PEMasterOnPrem
Описание		

2. Добавьте свойства имени пользователя и пароля в YAML для Puppet, как показано в следующем примере.

3. Убедитесь, что для свойства `remoteAccess` в коде YAML облачного шаблона Puppet задано значение `authentication: username and password`, как показано в примере ниже.

### Пример. Код YAML для имени пользователя и пароля в vCenter

В следующем примере приведен пример кода YAML для добавления проверки подлинности по имени пользователя и паролю в вычислительном ресурсе vCenter.

```
inputs:
  username:
    type: string
    title: Username
    description: Username to use to install Puppet agent
    default: puppet
  password:
    type: string
    title: Password
    default: VMware@123
    encrypted: true
    description: Password for the given username to install Puppet agent
resources:
  Puppet-Ubuntu:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      imageRef: >-
        https://cloud-images.ubuntu.com/releases/16.04/release-20170307/ubuntu-16.04-server-
cloudimg-amd64.ova
      remoteAccess:
        authentication: usernamePassword
        username: '${input.username}'
        password: '${input.password}'
  Puppet_Agent:
    type: Cloud.Puppet
    properties:
      provider: PEMasterOnPrem
      environment: production
      role: 'role::linux_webserver'
      username: '${input.username}'
      password: '${input.password}'
      host: '${Puppet-Ubuntu.*}'
      useSudo: true
      agentConfiguration:
        certName: '${Puppet-Ubuntu.address}'
```

### Примеры облачных шаблонов для управления конфигурацией AWS Puppet

Существует несколько вариантов настройки облачных шаблонов для поддержки управления конфигурацией на основе Puppet в вычислительных ресурсах AWS.

## Управление Puppet в AWS с использованием имени пользователя и пароля

Пример...	Пример YAML схемы элементов
<p>проверка подлинности конфигурации облачной среды на основе любого поддерживаемого образа компьютера Amazon.</p>	<pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 resources:   Webserver:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       image: centos       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false       users:         - default         - name: \${input.username}           lock_passwd: false           sudo: ['ALL=(ALL) NOPASSWD:ALL']           groups: [wheel, sudo, admin]           shell: '/bin/bash'           ssh-authorized-keys:             - ssh-rsa               AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6/+vGbmKXoRpX               dmettem@dmettem-m01.vmware.com       runcmd:         - echo "Defaults:\${input.username} !requiretty"         &gt;&gt; /etc/sudoers.d/\${input.username}   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEOAWS       environment: production       role: 'role::linux_webserver'       host: '\${Webserver.*}'       osType: linux       username: '\${input.username}'       password: '\${input.password}'       useSudo: true </pre>
<p>Проверка подлинности конфигурации облачной среды на основе настраиваемого образа компьютера Amazon с существующим пользователем.</p>	<pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 </pre>

Пример...	Пример YAML схемы элементов
	<pre> resources:   Webserver:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       image: centos       cloudConfig:           #cloud-config       runcmd:         - sudo sed -e 's/.*PasswordAuthentication no.*/PasswordAuthentication yes/' -i /etc/ssh/sshd_config         - sudo service sshd restart   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEOAWS       environment: production       role: 'role::linux_webserver'       host: '\${Webserver.*}'       osType: linux       username: '\${input.username}'       password: '\${input.password}'       useSudo: true </pre>

## Управление Puppet в AWS с созданием PublicPrivateKey

Пример...	Пример YAML схемы элементов
проверка подлинности remoteAccess.authentication в AWS с созданием ключей доступа PublicPrivateKey.	<pre> inputs: {} resources:   Machine:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       imageRef: ami-a4dc46db       remoteAccess:         authentication: generatedPublicPrivateKey   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: puppet-BlueprintProvisioningITSuite       environment: production       role: 'role::linux_webserver'       host: '\${Machine.*}'       osType: linux       username: ubuntu       useSudo: true       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'       useSudo: true </pre>

## Примеры облачных шаблонов vCenter с конфигурацией Puppet

Существует несколько вариантов настройки облачных шаблонов для поддержки управления конфигурацией на основе Puppet в вычислительных ресурсах vCenter.

## **Puppet vSphere с проверкой подлинности при использовании имени пользователя и пароля**

Ниже приведен пример кода YAML для Puppet в vSphere OVA с проверкой подлинности на основе имени пользователя и пароля.

Таблица 6-4.

Пример...	Пример YAML схемы элементов
<p>кода YAML для Puppet в vSphere OVA с проверкой подлинности на основе имени пользователя и пароля.</p>	<pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'   Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024       imageRef: &gt;- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa                 AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX                 dmettem@dmettem-m01.vmware.com             runcmd:               - echo "Defaults:\${input.username} </pre>
<p>кода YAML для Puppet в vSphere OVA с проверкой подлинности на основе имени пользователя и пароля в вычислительном ресурсе.</p>	<pre> inputs:   username:     type: string     title: Username     default: puppet </pre>



Таблица 6-4. (продолжение)

Пример...	Пример YAML схемы элементов
	<pre> password:   type: string   title: Password   encrypted: true   default: VMware@123 resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'   Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024       imageRef: &gt;- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa                 AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX                 dmettem@dmettem-m01.vmware.com       runcmd:         - echo "Defaults:\${input.username} </pre>
<p>кода YAML для Puppet в vCenter OVA с проверкой подлинности на основе пароля в вычислительном ресурсе с поддержкой удаленного доступа.</p>	<pre> inputs:   username:     type: string     title: Username     description: Username to use to install Puppet agent     default: puppet   password:     type: string     title: Password     default: VMware@123     encrypted: true </pre>

Таблица 6-4. (продолжение)

Пример...	Пример YAML схемы элементов
	<pre> description: Password for the given username to install Puppet agent resources:   Puppet-Ubuntu:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;-         https://cloud-images.ubuntu.com/releases/16.04/         release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       remoteAccess:         authentication: usernamePassword         username: '\${input.username}'         password: '\${input.password}'   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEMasterOnPrem       environment: production       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       host: '\${Puppet-Ubuntu.*}'       useSudo: true       agentConfiguration:         certName: '\${Puppet-Ubuntu.address}' </pre>

## Puppet в vSphere с созданной проверкой подлинности на основе PublicPrivateKey

Таблица 6-5.

Пример...	Пример YAML схемы элементов
<p>кода YAML для Puppet в vSphere OVA с проверкой подлинности на основе созданного PublicPrivateKey в вычислительном ресурсе.</p>	<pre> inputs: {} resources:   Machine:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;-         https://cloud-images.ubuntu.com/releases/16.04/         release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       remoteAccess:         authentication: generatedPublicPrivateKey   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: puppet-BlueprintProvisioningITSuite       environment: production       role: 'role::linux_webserver'       host: '\${Machine.*}'       osType: linux       username: ubuntu       useSudo: true       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'         - echo "Defaults:\${input.username}" </pre>

## Схема свойств ресурса vRealize Automation

Редактор «инфраструктура как код» vRealize Automation позволяет получить контекстную справку по синтаксису и созданию кода. Чтобы просмотреть полный набор свойств ресурсов облачного шаблона, которые иногда называют настраиваемыми свойствами, обратитесь к схеме консолидированных ресурсов.

Данная схема доступна на сайте VMware {code}. Перейдите по ссылке и нажмите кнопку **Модели**, чтобы просмотреть список объектов ресурсов, доступных для облачных шаблонов, которые ранее назывались схемами элементов.

- [Схема типов ресурсов vRealize Automation на VMware {code}](#)

## Специальные свойства Cloud Assembly

В Cloud Assembly поддерживается несколько свойств, которые можно использовать вне производственных сред или в других особых ситуациях. Эти свойства не отображаются в схеме.

**Осторожно!** Следующие свойства следует применять только в том случае, если не планируется настраивать гостевую ОС или тестировать ее настройку.

awaitIp	По умолчанию подготовка vRealize Automation не отображается как завершенная до тех пор, пока не будет полностью подготовлена ОС и выполнена настройка. Использование параметра <code>awaitIp: false</code> позволяет завершить подготовку, даже если не выполнена полная настройка. <b>ВНИМАНИЕ!</b> При использовании этого параметра процесс подготовки выполняется быстрее, однако в результате компьютер может быть не настроен и не иметь IP-адреса.
awaitHostName	Как и в случае с <code>awaitIp</code> , использование параметра <code>awaitHostName: false</code> позволяет завершить подготовку, даже если для компьютера не настроено имя узла.

## Другие способы создания шаблонов Cloud Assembly

В дополнение к созданию шаблона Cloud Assembly на основе пустого холста можно воспользоваться существующим кодом.

### Клонирование облачного шаблона

Для клонирования шаблона выберите **Проект**, затем источник и щелкните **Клонировать**. Клонирование облачного шаблона позволяет создать копию на основе источника. Затем клон назначается в новый проект или используется в качестве начального кода для нового приложения.

## Отправка и загрузка

Отправить и загрузить код YAML облачного шаблона, а также предоставить к нему общий доступ можно любым способом, подходящим для вашего сайта. Код облачного шаблона можно также редактировать с использованием внешних редакторов и сред разработки.

---

**Примечание** Хорошим способом проверки кода общего шаблона является его тестирование в редакторе кода Cloud Assembly на странице проектирования.

---

## Интеграция Cloud Assembly с репозиторием

Интегрированный репозиторий системы управления версиями Git может предоставить пользователям с соответствующими правами доступ к облачным шаблонам в качестве основы для нового развертывания. См. раздел [Как использовать интеграцию Git в Cloud Assembly](#).

## Увеличение и автоматизация жизненных циклов приложений с помощью действий расширяемости

Для увеличения жизненного цикла приложений можно использовать действия расширяемости или рабочие процессы vRealize Orchestrator с подписками с поддержкой расширяемости.

Средства расширяемости Cloud Assembly позволяют назначить для события действие расширяемости или рабочий процесс vRealize Orchestrator на основе подписок. Если происходит указанное событие, подписка запускает необходимое действие или рабочий процесс, о чем уведомляются все подписчики.

### Действия расширяемости

Действия расширяемости — это небольшие, упрощенные сценарии в виде фрагментов кода, в которых указывается действие и способ его выполнения. Действия расширяемости можно импортировать из предварительно созданных шаблонов действий Cloud Assembly или из ZIP-файла. Для создания настраиваемых сценариев для действий расширяемости также можно использовать редактор действий. Если несколько сценариев действий связаны между собой в рамках одного сценария, можно создать поток действий. С помощью потоков действий можно создать последовательность действий. Дополнительные сведения об использовании потоков действий см. в разделе [Что такое поток действий](#).

### Рабочие процессы в vRealize Orchestrator

Интеграция Cloud Assembly с существующей средой vRealize Orchestrator позволяет использовать рабочие процессы в подписках с поддержкой расширяемости.

### Подписки на действия с поддержкой расширяемости

Чтобы расширить жизненный цикл приложения, подписке Cloud Assembly можно назначить действие расширяемости.

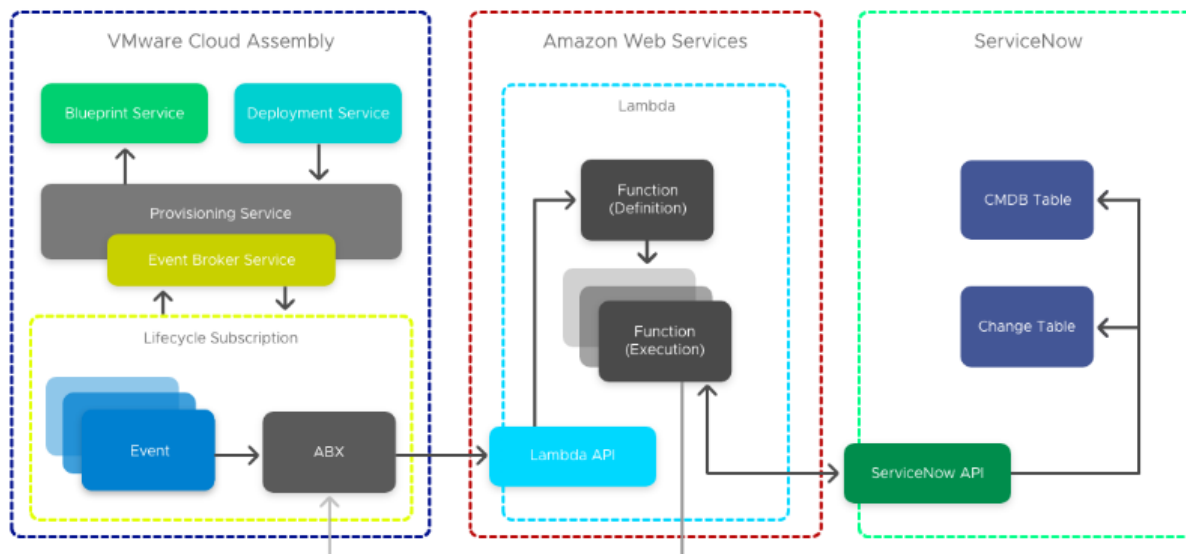
---

**Примечание** Следующие подписки приведены в качестве примеров и не представляют исчерпывающего обзора всех возможностей действий с поддержкой расширяемости.

---

## Интеграция службы Cloud Assembly и ServiceNow с помощью действий с поддержкой расширяемости

Используя действия с поддержкой расширяемости, службу Cloud Assembly можно интегрировать с корпоративными решениями для управления ИТ-службами (ITSM), такими как ServiceNow.

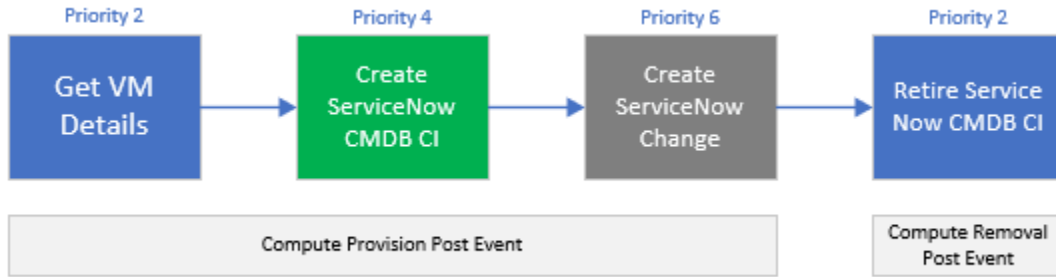


В корпоративной среде платформа управления облаком часто интегрируется с платформой управления ИТ-службами (ITSM) и базой данных управления конфигурациями (CMDB) для соответствия нормативным требованиям. Следуя данному примеру, можно интегрировать службу Cloud Assembly с решением ServiceNow для базы данных управления конфигурациями (CMDB) и управления ИТ-службами (ITSM) с помощью сценариев действий расширяемости.

**Примечание** Кроме того, интегрировать ServiceNow со службой Cloud Assembly можно, используя рабочие процессы vRealize Orchestrator. Дополнительные сведения об интеграции с решением ServiceNow с помощью рабочих процессов см. в разделе [Интеграция службы Cloud Assembly с решением ServiceNow для управления ИТ-службами \(ITSM\) с помощью рабочих процессов vRealize Orchestrator](#).

Чтобы создать эту интеграцию, потребуется четыре сценария действий с поддержкой расширяемости. Первые три сценария инициируются последовательно во время предоставления ресурсов событием «После предоставления вычислительных ресурсов». Четвертый сценарий запускается событием «После удаления вычислительных ресурсов».

Дополнительные сведения о темах событий см. в разделе [Темы событий, предоставленные с помощью Cloud Assembly](#).



### Получение сведений о ВМ

Сценарий «Получение сведений о ВМ» получает дополнительные сведения о полезных данных, необходимые для создания непрерывной интеграции, и маркер идентификации из Amazon Web Services Systems Manager Parameter Store (SSM). Кроме того, этот сценарий обновляет раздел `customProperties`, добавляя в него дополнительные свойства для дальнейшего использования.

### Создание непрерывной интеграции базы данных управления конфигурациями в ServiceNow

Сценарий «Создание непрерывной интеграции базы данных управления конфигурациями в ServiceNow» передает URL-адрес экземпляра ServiceNow в качестве входных данных и сохраняет экземпляр в SSM для соответствия требованиям безопасности. Этот сценарий также считывает ответ уникального идентификатора записи базы данных управления конфигурациями ServiceNow (`sys_id`). Он передает этот ответ в качестве выходного значения и записывает настраиваемое свойство `serviceNowSysId` в процессе создания. Это значение используется для того, чтобы пометить непрерывную интеграцию как списанную при удалении экземпляра.

---

**Примечание** Может потребоваться выделение дополнительных разрешений для роли vRealize Automation services Amazon Web Services, чтобы разрешить Lambda доступ к SSM Parameter Store.

---

### Создание изменения в ServiceNow

Этот сценарий завершает интеграцию ITSM путем передачи URL-адреса экземпляра ServiceNow в качестве входных данных и сохранения учетных данных ServiceNow в качестве SSM в соответствии с требованиями безопасности.

### Создание изменения в ServiceNow

Сценарий списания непрерывной интеграции базы данных управления конфигурациями в ServiceNow приводит к остановке ServiceNow и помечает непрерывную интеграцию как списанную в соответствии с настраиваемым свойством `serviceNowSysId`, созданным сценарием создания интеграции.

## Необходимые условия

- Перед настройкой интеграции отфильтруйте все подписки на события с помощью условного свойства облачного шаблона: `event.data["customProperties"]["enable_servicenow"] == "true"`

---

**Примечание** Облачные шаблоны, которым требуется интеграция с ServiceNow, включают в себя это свойство.

---

- Загрузите и установите Python.

Дополнительные сведения о фильтрации подписок см. в разделе [Создание подписки с поддержкой расширяемости](#).

## Процедура

1. Откройте командную строку на виртуальной машине.
2. Запустите сценарий «Получение сведений о ВМ».

```
from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    baseUrl = inputs['url']
    casToken = client.get_parameter(Name="casToken",WithDecryption=True)

    url = baseUrl + "/iaas/login"
    headers = {"Accept":"application/json","Content-Type":"application/json"}
    payload = {"refreshToken":casToken['Parameter']['Value']}

    results = requests.post(url,json=payload,headers=headers)

    bearer = "Bearer "
    bearer = bearer + results.json()["token"]

    deploymentId = inputs['deploymentId']
    resourceId = inputs['resourceIds'][0]

    print("deploymentId: " + deploymentId)
    print("resourceId:" + resourceId)

    machineUri = baseUrl + "/iaas/machines/" + resourceId
    headers = {"Accept":"application/json","Content-Type":"application/json",
    "Authorization":bearer }
    resultMachine = requests.get(machineUri,headers=headers)
    print("machine: " + resultMachine.text)

    print( "serviceNowCPUCount: " + json.loads(resultMachine.text)["customProperties"]
    ["cpuCount"] )
    print( "serviceNowMemoryInMB: " + json.loads(resultMachine.text)["customProperties"]
    ["memoryInMB"] )
```

```

#update customProperties
outputs = {}
outputs['customProperties'] = inputs['customProperties']
outputs['customProperties']['serviceNowCPUCount'] = int(json.loads(resultMachine.text)
["customProperties"]["cpuCount"])
outputs['customProperties']['serviceNowMemoryInMB'] = json.loads(resultMachine.text)
["customProperties"]["memoryInMB"]
return outputs

```

### 3. Запустите действие по созданию элемента конфигурации CMDB.

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):

    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    table_name = "cmdb_ci_vmware_instance"
    url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'name': inputs['customProperties']['serviceNowHostname'],
        'cpus': int(inputs['customProperties']['serviceNowCPUCount']),
        'memory': inputs['customProperties']['serviceNowMemoryInMB'],
        'correlation_id': inputs['deploymentId'],
        'disks_size': int(inputs['customProperties']['provisionGB']),
        'location': "Sydney",
        'vcenter_uuid': inputs['customProperties']['vcUuid'],
        'state': 'On',
        'sys_created_by': inputs['__metadata']['userName'],
        'owned_by': inputs['__metadata']['userName']
    }
    results = requests.post(
        url,
        json=payload,
        headers=headers,
        auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
    )
    print(results.text)

    #parse response for the sys_id of CMDB CI reference
    if json.loads(results.text)['result']:
        serviceNowResponse = json.loads(results.text)['result']
        serviceNowSysId = serviceNowResponse['sys_id']
        print(serviceNowSysId)

    #update the serviceNowSysId customProperty

```



```

outputs = {}
outputs['customProperties'] = inputs['customProperties']
outputs['customProperties']['serviceNowSysId'] = serviceNowSysId;
return outputs

```

#### 4. Запустите сценарий действия «Создание».

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    table_name = "change_request"
    url = "https://" + inputs['instanceUrl'] + "/api/now/table/{0}".format(table_name)
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'short_description': 'Provision CAS VM Instance'
    }
    results = requests.post(
        url,
        json=payload,
        headers=headers,
        auth=(snowUser['Parameter']['Value'], snowPass['Parameter']['Value'])
    )
    print(results.text)

```

#### Результаты

Служба Cloud Assembly успешно интегрирована с системой управления ИТ-службами ServiceNow.

#### Следующие шаги

При необходимости эту непрерывную интеграцию можно пометить как списанную, используя действие списания элемента конфигурации CMDB.

```

from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    snowUser = client.get_parameter(Name="serviceNowUserName",WithDecryption=False)
    snowPass = client.get_parameter(Name="serviceNowPassword",WithDecryption=True)
    tableName = "cmdb_ci_vmware_instance"
    sys_id = inputs['customProperties']['serviceNowSysId']
    url = "https://" + inputs['instanceUrl'] + "/api/now/" + tableName + "/" + sys_id
    headers = {'Content-type': 'application/json', 'Accept': 'application/json'}
    payload = {
        'state': 'Retired'
    }

```

```

results = requests.put(
    url,
    json=payload,
    headers=headers,
    auth=(inputs['username'], inputs['password'])
)
print(results.text)

```

Дополнительные сведения об использовании действий с поддержкой расширяемости для интеграции системы ServiceNow со службой Cloud Assembly см. в разделе [Расширение Cloud Assembly с помощью действий ABX для интеграции с ServiceNow](#).

## Добавление тегов к виртуальным машинам в ходе предоставления ресурсов с помощью действий с поддержкой расширяемости

Используя действия с поддержкой расширяемости в сочетании с подписками, можно автоматизировать и упростить добавление тегов к виртуальным машинам.

Администратор облачных систем может создавать развертывания, в которых теги расставляются автоматически с указанными входными и выходными данными с помощью действий и подписок с поддержкой расширяемости. При создании нового развертывания для проекта, содержащего подписку на расстановку тегов в виртуальной машине, событие развертывания запускает сценарий «Добавление тегов к ВМ» (Tag VM), в результате чего выполняется автоматическая расстановка тегов. Это позволяет сэкономить время и способствует повышению эффективности, благодаря чему упрощается управление развертыванием.

### Необходимые условия

- Доступ к учетным данным администратора облачных систем.
- Роль Amazon Web Services для функций Lambda.

### Процедура

1. Выберите **Расширяемость > Библиотека > Действия > Создать действие** и создайте действие со следующими параметрами.

Параметр	Описание
Название действия	Название действия с поддержкой расширяемости (желательно, чтобы оно содержало <b>TagVM</b> в качестве префикса или суффикса).
Проект	Проект, в котором выполняется тестирование действия с поддержкой расширяемости.
Шаблон действия	<b>Добавление тегов к ВМ (Tag VM)</b>
Среда выполнения	Python
Источник сценария	Написать сценарий

2. Укажите **Обработчик** в качестве значения параметра **Основная функция**.

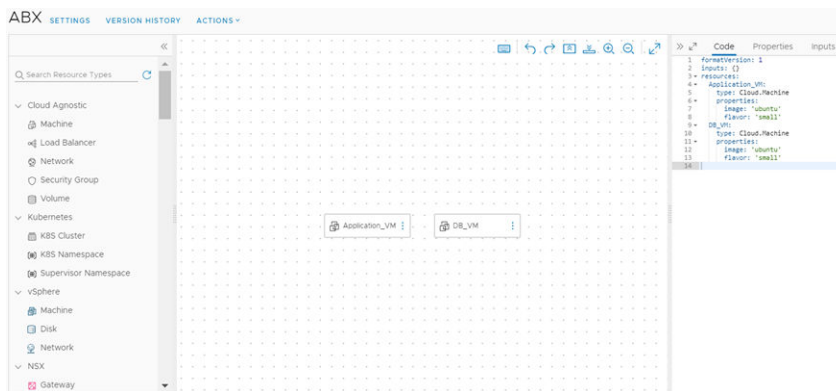
- Добавьте входные данные для расстановки тегов, предназначенные для тестирования действия расширяемости.

Например, `resourceNames = ["DB_VM"]` и `target = world`.

- Чтобы сохранить действие, нажмите **Сохранить**.
- Чтобы протестировать действие, нажмите **Тестировать**.
- Чтобы выйти из редактора действий, нажмите **Заккрыть**.
- Выберите **Расширяемость > Подписки**.
- Щелкните **Создать подписку**.
- Введите следующие сведения о подписке.

Сведения	Настройка
Тема события	<p>Выберите тему события, связанную со стадией расстановки тегов виртуальной машины. Например, «Выделение вычислительных ресурсов».</p> <p><b>Примечание</b> Теги должны быть частью параметров события для выбранной темы событий.</p>
Блокирующая	Установите время ожидания для подписки, равное 1 минуте.
Действие или рабочий процесс	Выберите тип готовности к запуску действия с поддержкой расширяемости и настраиваемое действие с поддержкой расширяемости.

- Чтобы сохранить подписку на настраиваемое действие с поддержкой расширяемости, нажмите кнопку **Сохранить**.
- Перейдите в раздел **Проектирование > Облачные шаблоны** и создайте облачный шаблон на основе пустого холста.
- Добавьте две виртуальные машины в облачный шаблон: `Application_VM` и `DB_VM`.



- Чтобы развернуть виртуальные машины, нажмите **Развернуть**.
- Во время развертывания убедитесь, что событие инициировано и действие с поддержкой расширяемости запущено.

15. Чтобы убедиться в правильности расстановки тегов, выберите **Ресурсы > Ресурсы > Виртуальные машины**.

## Настройка имени контроллера сетевого интерфейса с помощью действий расширяемости

Имя интерфейса контроллера сетевого интерфейса (сетевого адаптера) можно настроить с помощью вызовов API-интерфейса инфраструктуры как услуги, применяемых с помощью действий расширяемости.

Чтобы настроить имя интерфейса сетевого адаптера, необходимо выполнить вызовы GET и PATCH к API-интерфейсу инфраструктуры как услуги vRealize Automation. При выполнении вызова GET к `https://your_vRA_fqdn/iaas/api/machines/{id}` можно извлечь ссылку сетевого адаптера для вычислительного ресурса, который требуется изменить. Затем можно выполнить вызов PATCH к `https://your_vRA_fqdn/iaas/api/machines/{id}/network-interfaces/{nicId}`, который содержит имя сетевого адаптера в качестве полезных данных, что позволяет добавить новое имя сетевого адаптера.

В следующем примере используется сценарий Python, который можно применить для настройки имени интерфейса сетевого адаптера. Вы можете использовать другой сценарий и язык сценария, например Node.js.

### Необходимые условия

- Имя интерфейса сетевого адаптера можно настроить только до подготовки вычислительного ресурса. Поэтому для соответствующих подписок с поддержкой расширяемости можно выбрать только тему события **Compute Provision**.
- Имена интерфейса можно настроить только для сетевых адаптеров, которые используют Microsoft Azure в качестве поставщика.

### Процедура

1. Создайте действие расширяемости.
  - а) Выберите **Расширяемость > Действия**.
  - б) Щелкните **Создать действие**.
  - в) Введите имя и проект для действия расширяемости и щелкните **Далее**.

- г) Добавьте сценарий настройки сетевого адаптера.

Ниже приведен пример сценария Python.

```
import json

def handler(context, inputs):

    # Get the machine info, which contains machine nic link
    response = context.request('/iaas/api/machines/'+inputs["resourceIds"][0], "GET",
    {})

    # Build PATCH machine nic payload here
    name = "customized-nic-02";
    data = {'name':name};

    # Convert machine data string to json object
    response_json = json.loads(response["content"])

    # Patch machine nic
    response_patch = context.request(response_json["_links"]["network-interfaces"]
    ["hrefs"][0] + "?apiVersion=2021-07-15", 'PATCH', data)

    # return value is empty since we are not changing any compute provisioning
    parameters
    outputs = {}
    return outputs
```

В предыдущем примере сценарий выполняет две основные операции через API-интерфейс инфраструктуры как услуги. Сначала сценарий использует вызов GET для извлечения ссылки сетевого адаптера, а затем использует вызов PATCH, чтобы применить имя интерфейса. В этом примере имя интерфейса сетевого адаптера жестко запрограммировано в сценарии как "customized-nic-02".

- д) Чтобы завершить редактирование нового действия расширяемости, щелкните **Сохранить**.

## 2. Создание подписки с поддержкой расширяемости.

- а) Выберите **Расширяемость > Подписки**.
- б) Щелкните **Создать подписку**.
- в) Введите имя подписки с поддержкой расширяемости.
- г) В поле **Тема события** в качестве темы события для подписки с поддержкой расширяемости выберите **Compute Provision**.
- д) В поле **Действие/рабочий процесс** выберите действие расширяемости, созданное для настройки сетевого адаптера.

- е) Включите блокировку событий.

Включение блокировки позволяет убедиться в том, что процесс подготовки заблокирован до тех пор, пока не будет завершено выполнение действия расширяемости.

- ж) Чтобы завершить редактирование подписки с поддержкой расширяемости, щелкните **Сохранить**.

## Результаты

Новая подписка с поддержкой расширяемости выполняется при инициировании события подготовки вычислительного ресурса и обеспечивает настройку имени интерфейса сетевого адаптера для подготавливаемых вычислительных ресурсов.

## Дополнительные сведения о действиях расширяемости

В функции «расширяемость на основе действий» для автоматизации действий расширяемости в Cloud Assembly используются оптимизированные сценарии в виде кода.

Расширяемость на основе действий предоставляет упрощенный и гибкий интерфейс модуля среды выполнения, в котором можно определять небольшие действия на основе сценариев и настраивать их запуск при возникновении определенных событий, указанных в подписках с поддержкой расширяемости.

В Cloud Assembly или локальной среде можно создать такие сценарии действий расширяемости в виде кода и назначить их подпискам. Сценарии действий расширяемости используются для упрощения процесса автоматизации задач и этапов. Дополнительные сведения об интеграции Cloud Assembly с сервером vRealize Orchestrator см. в разделе [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).

Расширяемость на основе действий предоставляет следующие возможности.

- Альтернатива рабочим процессам vRealize Orchestrator, использующим небольшие и многократно используемые действия на основе сценариев, для упрощения интеграции и настройки.
- Способ повторного использования шаблонов действий, которые содержат многократно используемые параметризованные действия.

Действия расширяемости можно создать, написав код сценария пользовательского действия или импортировав предварительно созданный код сценария в виде пакета ZIP. Расширяемость на основе действий поддерживает среды выполнения Node.js, Python и PowerShell. Среда выполнения Node.js и Python основаны на Amazon Web Services Lambda. Следовательно, необходимо иметь активную подписку с функцией «Управление идентификацией и доступом» (IAM) в Amazon Web Services и настроить Amazon Web Services как конечную точку в Cloud Assembly. Сведения о начале работы со службой Amazon Web Services Lambda см. в разделе [ABX: расширяемость служб Cloud Assembly без использования сервера](#).

---

**Примечание** Действия расширяемости зависят от конкретного проекта.

---

## Создание действия с поддержкой расширяемости

С помощью Cloud Assembly можно создавать действия расширяемости для использования в подписках с поддержкой расширяемости.

Действия расширяемости — это настраиваемые, компактные и гибкие средства для расширения жизненного цикла приложений с помощью пользовательских сценариев на основе кода и шаблонов действий.

Шаблоны действий содержат предварительно определенные параметры, которые помогают задать основу для действия с поддержкой расширяемости.

Создать действие расширяемости можно двумя способами.

- Написание пользовательского кода для сценария действий с поддержкой расширяемости.

---

**Примечание** Для написания пользовательского кода в редакторе действий расширяемости может потребоваться активное подключение к Интернету.

---

- Импорт пакета развертывания в виде ZIP-файла для действия расширяемости. Сведения о создании пакета ZIP для действий расширяемости см. в разделах [Создание пакета ZIP для действий расширяемости в среде выполнения Python](#), [Создание пакета ZIP для действий расширяемости в среде выполнения Node.js](#) или [Создание пакета ZIP для действий расширяемости в среде выполнения PowerShell](#).

Ниже описана процедура создания действия расширяемости, в которой в качестве поставщика FaaS используется Amazon Web Services.

#### Необходимые условия

- Участие в активном действующем проекте.
- Настроенная роль Amazon Web Services для функций Lambda. Например, `AWSLambdaBasicExecutionRole`.
- Роль с правами администратора облачных систем или активными разрешениями `iam:PassRole`.

#### Процедура

1. Выберите **Расширяемость > Библиотека > Действия**.
2. Щелкните **Создать действие**.
3. Введите имя действия и выберите проект.
4. (дополнительно) Добавьте описание для действия.
5. Нажмите кнопку **Далее**.
6. Найдите и выберите шаблон действия.

---

**Примечание** Чтобы создать настраиваемое действие без использования шаблона действия, нажмите кнопку **Настраиваемый сценарий**.

---

Отобразятся новые настраиваемые параметры.

7. Нажмите кнопку **Написать сценарий** или **Импорт пакета**.
8. Выберите время выполнения действий.

9. Введите имя **основной функции** для точки входа действия.

---

**Примечание** Для действий, импортированных из пакета ZIP, основная функция также должна включать в себя имя файла сценария, который содержит точку входа. Например, если основной файл сценария называется `main.py`, а точка входа `handler (context, inputs)`, то имя основной функции должно быть `main.handler`.

---

10. Определите входные и выходные параметры действия.
11. (дополнительно) Добавьте секретные элементы или константы действия с поддержкой расширяемости во входные данные по умолчанию.

---

**Примечание** Дополнительные сведения о секретных элементах и константах действий с поддержкой расширяемости см. в разделах [Создание секретных элементов для использования в действиях с поддержкой расширяемости](#) и [Создание констант действий с поддержкой расширяемости](#).

---

12. (дополнительно) Добавьте зависимости приложения в действие.

---

**Примечание** Для сценариев PowerShell можно определить зависимости приложений, чтобы они могли обрабатываться с использованием репозитория PowerShell Gallery. Для определения зависимости приложений таким образом, чтобы они обрабатывались из общедоступного репозитория, используйте следующий формат.

```
@{
    Name = 'Version'
}

e.g.

@{
    Pester = '4.3.1'
}
```

---

**Примечание** Для действий, импортируемых из пакета ZIP, зависимости приложений добавляются автоматически.

---

13. Чтобы определить ограничения времени ожидания и памяти, включите параметр **Установить настраиваемое время ожидания и ограничения**.
14. Чтобы протестировать действие, нажмите кнопку **Сохранить**, а затем кнопку **Тест**.



## Следующие шаги

После создания и проверки действия с поддержкой расширяемости его можно назначить соответствующей подписке.

---

**Примечание** В подписках с поддержкой расширяемости используется последняя опубликованная версия действия расширяемости. После создания новой версии действия щелкните **Версии** в верхнем правом углу окна редактора. Чтобы опубликовать версию действия, которую нужно использовать в подписке, нажмите **Опубликовать**.

---

## Создание пакета ZIP для действий расширяемости в среде выполнения Python

Можно создать пакет ZIP, содержащий сценарий Python и зависимости, используемые действиями расширяемости Cloud Assembly.

Существует два способа построения сценария для действий расширяемости.

- Написание сценария непосредственно в редакторе действий расширяемости в Cloud Assembly.
- Создание сценария в локальной среде и добавление его с соответствующими зависимостями в пакет ZIP.

С помощью пакета ZIP можно создать настраиваемый шаблон сценариев действий и зависимостей, которые можно импортировать в Cloud Assembly для использования в действиях расширяемости.

Кроме того, пакет ZIP можно использовать в ситуациях, когда модули, связанные с зависимостями в сценарии действий, не могут быть обработаны в службе Cloud Assembly, например, если в среде отсутствует доступ к Интернету.

Кроме того, пакет ZIP можно использовать для создания действий расширяемости, содержащих несколько файлов сценариев Python. Использование нескольких файлов сценариев может быть полезно для организации структуры кода действий расширяемости.

## Необходимые условия

Если используется Python 3.3 или более ранняя версия, загрузите и настройте установщик пакета PIP. См. раздел [Индекс пакета Python](#).

## Процедура

1. На локальном компьютере создайте папку для сценария действий и зависимостей.

Например, /home/user1/zip-action.

2. Добавьте в папку основной сценарий или сценарии Python.

Например, /home/user1/zip-action/main.py.

### 3. (дополнительно) Добавьте в папку все зависимости для сценария Python.

- а) Создайте файл `requirements.txt`, содержащий зависимости. См. раздел [Файлы требований](#).
- б) Откройте оболочку Linux.

---

**Примечание** Среда выполнения расширяемости на основе действий в Cloud Assembly основана на Linux. Поэтому любые зависимости Python, скомпилированные в среде Windows, могут сделать созданный пакет ZIP непригодным для создания действий расширяемости. Следовательно, необходимо использовать оболочку Linux.

---

- в) Установите файл `requirements.txt` в папке сценария, выполнив следующую команду:

```
pip install -r requirements.txt --target=home/user1/zip-action
```

### 4. В назначенной папке выберите элементы сценария и, если применимо, файл `requirements.txt` и сожмите их в виде пакета ZIP.

---

**Примечание** Сценарий и элементы зависимостей должны храниться на корневом уровне пакета ZIP. При создании пакета ZIP в среде Linux может возникнуть проблема, если содержимое пакета не хранится на корневом уровне. При возникновении этой проблемы создайте пакет, запустив команду `zip -r` в оболочке командной строки.

---

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

#### Следующие шаги

Используйте пакет ZIP для создания сценария действий расширяемости. См. раздел [Создание действия с поддержкой расширяемости](#).

#### Создание пакета ZIP для действий расширяемости в среде выполнения Node.js

Можно создать пакет ZIP, содержащий сценарий Node.js и зависимости, используемые действиями расширяемости Cloud Assembly.

Существует два способа построения сценария для действий расширяемости.

- Написание сценария непосредственно в редакторе действий расширяемости в Cloud Assembly.
- Создание сценария в локальной среде и добавление его с соответствующими зависимостями в пакет ZIP.

С помощью пакета ZIP можно создать настраиваемый шаблон сценариев действий и зависимостей, которые можно импортировать в Cloud Assembly для использования в действиях расширяемости.

Кроме того, пакет ZIP можно использовать в ситуациях, когда модули, связанные с зависимостями в сценарии действий, не могут быть обработаны в службе Cloud Assembly, например, если в среде отсутствует доступ к Интернету.

Кроме того, пакеты можно использовать для создания действий расширяемости, содержащих несколько файлов сценариев Node.js. Использование нескольких файлов сценариев может быть полезно для организации структуры кода действий расширяемости.

### Процедура

1. На локальном компьютере создайте папку для сценария действий и зависимостей.

Например, `/home/user1/zip-action`.

2. Добавьте в папку основной сценарий или сценарии Node.js.

Например, `/home/user1/zip-action/main.js`.

3. (дополнительно) Добавьте в папку все зависимости для сценария Node.js.

а) Создайте файл `package.json` с зависимостями в папке сценария. См. разделы [Создание файла package.json](#) и [Указание зависимостей и devDependencies в файле package.json](#).

б) Откройте оболочку командной строки.

в) Перейдите в папку, созданную для сценария действий и зависимостей.

```
cd /home/user1/zip-action
```

- г) Установите файл `package.json` в папке сценария, выполнив следующую команду:

```
npm install --production
```

---

**Примечание** Эта команда создает каталог `node_modules` в папке.

---

4. В назначенной папке выберите элементы сценария и, если применимо, каталог `node_modules` и сожмите их в виде пакета ZIP.

---

**Примечание** Сценарий и элементы зависимостей должны храниться на корневом уровне пакета ZIP. При создании пакета ZIP в среде Linux может возникнуть проблема, если содержимое пакета не хранится на корневом уровне. При возникновении этой проблемы создайте пакет, запустив команду `zip -r` в оболочке командной строки.

---

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

### Следующие шаги

Используйте пакет ZIP для создания сценария действий расширяемости. См. раздел [Создание действия с поддержкой расширяемости](#).

### Создание пакета ZIP для действий расширяемости в среде выполнения PowerShell

Можно создать пакет ZIP, содержащий сценарий PowerShell и модули зависимостей для использования в действиях расширяемости.

Существует два способа построения сценария для действий расширяемости.

- Написание сценария непосредственно в редакторе действий расширяемости в Cloud Assembly.
- Создание сценария в локальной среде и добавление его с соответствующими зависимостями в пакет ZIP.

С помощью пакета ZIP можно создать настраиваемый шаблон сценариев действий и зависимостей, которые можно импортировать в Cloud Assembly для использования в действиях расширяемости.

---

**Примечание** Командлеты PowerCLI не нужно определять в виде зависимостей или объединять их в пакет ZIP. Командлеты PowerCLI предварительно настроены в среде выполнения PowerShell службы Cloud Assembly.

---

Кроме того, пакет ZIP можно использовать в ситуациях, когда модули, связанные с зависимостями в сценарии действий, не могут быть обработаны в службе Cloud Assembly, например, если в среде отсутствует доступ к Интернету.

Кроме того, пакет ZIP можно использовать для создания действий расширяемости, содержащих несколько файлов сценариев PowerShell. Использование нескольких файлов сценариев может быть полезно для организации структуры кода действий расширяемости.

#### Необходимые условия

Для этого требуется знакомство с PowerShell и PowerCLI. В [Docker Hub](#) можно найти образ Docker, содержащий PowerShell Core, PowerCLI 10, PowerNSX, а также несколько модулей и образцов сценариев, разработанных участниками сообщества.

#### Процедура

1. На локальном компьютере создайте папку для сценария действий и зависимостей.

Например, /home/user1/zip-action.

2. Добавьте в папку основной сценарий PowerShell с расширением .psm1.

Следующий сценарий представляет собой простую функцию PowerShell, которая называется main.psm1.

```
function handler($context, $payload) {

    Write-Host "Hello " $payload.target

    return $payload
}
```

---

**Примечание** Выходные данные действия расширяемости PowerShell основаны на последней переменной, отображаемой в теле функции. Все другие переменные во включенной функции отбрасываются.

---

3. (дополнительно) Добавьте конфигурацию прокси-сервера в основной сценарий PowerShell с использованием параметров `context`. См. раздел [Использование контекстных параметров для добавления конфигурации прокси-сервера в сценарии PowerShell](#).
4. (дополнительно) Добавьте зависимости для сценария PowerShell.

---

**Примечание** Сценарий зависимости PowerShell должен использовать расширение `.psm1`. Используйте одно и то же имя для сценария и вложенной папки, в которой сохранен сценарий.

---

- а) Войдите в оболочку Linux PowerShell.

---

**Примечание** Среда выполнения расширяемости на основе действий в Cloud Assembly основана на Linux. Любые зависимости PowerShell, скомпилированные в среде Windows, могут сделать созданный пакет ZIP непригодным для использования. Любые установленные сторонние зависимости должны быть совместимы с VMware Photon OS, когда сценарии PowerShell выполняются в Photon OS.

---

- б) Перейдите в папку `/home/user1/zip-action`.
- в) Загрузите и сохраните модуль PowerShell, содержащий зависимости, запустив командлет `Save-Module`.

```
Save-Module -Name <module name> -Path ./
```

- г) Повторите предыдущий шаг для всех дополнительных модулей зависимостей.

---

**Важно!** Убедитесь, что каждый модуль зависимости находится в отдельной вложенной папке. Дополнительные сведения о создании модулей PowerShell и управлении ими см. в разделе [Написание модуля сценариев PowerShell](#).

---

5. В назначенной папке выберите элементы сценария и, если применимо, вложенные папки модуля зависимостей и сожмите их в виде пакета ZIP.

---

**Примечание** Вложенные папки сценариев и модулей зависимостей должны храниться на корневом уровне пакета ZIP. При создании пакета ZIP в среде Linux может возникнуть проблема, если содержимое пакета не хранится на корневом уровне. При возникновении этой проблемы создайте пакет, запустив команду `zip -r` в оболочке командной строки.

---

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

#### Следующие шаги

Используйте пакет ZIP для создания сценария действий расширяемости. См. раздел [Создание действия с поддержкой расширяемости](#).

Использование контекстных параметров для добавления конфигурации прокси-сервера в сценарии PowerShell

Связь с сетевым прокси-сервером в сценарии PowerShell можно включить с помощью параметров `context`.

Для некоторых командлетов PowerShell может быть необходимо настроить сетевой прокси-сервер в качестве переменной среды в функции PowerShell. Настройки прокси-сервера передаются в функцию PowerShell с параметрами `$context.proxy.host` и `$context.proxy.port`.

Эти параметры `context` можно добавить в начале сценария PowerShell.

```
$proxyString = "http://" + $context.proxy.host + ":" + $context.proxy.port
$Env:HTTP_PROXY = $proxyString
$Env:HTTPS_PROXY = $proxyString
```

Если командлеты поддерживают параметр `-Proxy`, можно также передать значение прокси-сервера непосредственно в отдельные командлеты PowerShell.

### Настройка действий расширяемости, связанных с облачной средой

Для облачных учетных записей можно настроить действия расширяемости.

В ходе создания действия расширяемости его можно настроить и привязать к различным облачным учетным записям.

- Microsoft Azure
- Amazon Web Services

#### Необходимые условия

Для этого требуется действующая облачная учетная запись.

#### Процедура

1. Выберите **Расширяемость > Библиотека > Действие**.
2. Щелкните **Создать действие**.
3. Задайте необходимые параметры действия.
4. В раскрывающемся меню **Поставщик FaaS** выберите поставщика облачной учетной записи или нажмите **Автоматический выбор**.

---

**Примечание** При выборе значения **Автоматически** действие автоматически определяет поставщика услуг FaaS (функция как услуга).

---

5. Нажмите **Сохранить**.

#### Результаты

Использование действия расширяемости привязано к настроенной облачной учетной записи.

### Настройка локальных действий расширяемости

Для действий расширяемости можно настроить использование локального поставщика FaaS вместо облачной учетной записи Amazon Web Services или Microsoft Azure.

С помощью локального поставщика FaaS для действий расширяемости можно использовать локальные службы, например LDAP, CMDB или центры обработки данных vCenter в подписках Cloud Assembly с поддержкой расширяемости.

### Процедура

1. Выберите **Расширяемость > Библиотека > Действия**.
2. Щелкните **Создать действие**.
3. Введите имя и проект для действия расширяемости.
4. (дополнительно) Введите описание действия расширяемости.
5. Нажмите кнопку **Далее**.
6. Создайте или импортируйте сценарий действий расширяемости.
7. Щелкните раскрывающееся меню **Поставщик FaaS** и выберите **Локально**.
8. Чтобы сохранить новое действие расширяемости, нажмите кнопку **Сохранить**.

### Следующие шаги

Используйте созданное действие расширяемости в подписках Cloud Assembly с поддержкой расширяемости.

### Создание секретных элементов для использования в действиях с поддержкой расширяемости

Зашифрованные входные данные можно добавлять в действие с поддержкой расширяемости с помощью секретных элементов на уровне проекта.

С помощью секретных элементов можно добавлять зашифрованные входные значения в действия с поддержкой расширяемости. Шифрование полезно использовать в тех случаях, когда входные данные применяются для управления конфиденциальными данными, например паролями и сертификатами. Секретные элементы доступны для всех поставщиков FaaS и сред выполнения.

---

**Примечание** Кроме того, зашифрованные входные значения можно добавлять с помощью констант действий. См. раздел [Создание констант действий с поддержкой расширяемости](#).

---

Доступ к секретным элементам зависит от проекта, в котором они создаются. Например, секретные элементы, созданные в проекте А, доступны только тем пользователям, которые являются участниками проекта А.

Секретные секреты используют функцию `context.getSecret()` для дешифрования значения секретного элемента при его добавлении в сценарий. Эта функция использует имя секретного элемента в качестве параметра. Например, в качестве зашифрованного входного параметра в действии можно использовать секретный элемент с именем `abxsecret`. Чтобы добавить этот входной параметр в сценарий действия, необходимо использовать `context.getSecret(inputs["abxsecret"])`.

## Процедура

### 1. Создание нового секретного элемента.

- а) Выберите пункт **Инфраструктура > Администрирование > Секретные элементы**.
- б) Выберите **Создать секретный элемент**.
- в) Введите имя проекта, которому назначен секретный элемент.

---

**Примечание** Действие с поддержкой расширяемости, которому нужно назначить секретный элемент, должно быть частью того же проекта, что и этот секретный элемент.

---

- г) Введите имя секретного элемента.
- д) Введите значение, которое нужно назначить секретному элементу.
- е) (дополнительно) Введите описание.
- ж) Щелкните **Создать**.

### 2. Добавление секретного элемента в действие с поддержкой расширяемости.

- а) Выберите существующее или создайте новое действие с поддержкой расширяемости.
- б) В разделе **Входные данные по умолчанию** установите флажок **Секретный элемент**.
- в) Найдите секретный элемент и добавьте его во входные данные действия с поддержкой расширяемости.
- г) Добавьте секретный элемент в сценарий действия с поддержкой расширяемости с помощью функции `context.getSecret()`.
- д) Чтобы протестировать секретный элемент, нажмите **Тестировать**.

## Создание констант действий с поддержкой расширяемости

Можно создавать и хранить константы для использования в действиях с поддержкой расширяемости.

Благодаря константам действий с поддержкой расширяемости можно добавлять зашифрованные входные значения в действия с поддержкой расширяемости. Шифрование полезно использовать в тех случаях, когда входные данные применяются для управления конфиденциальными данными, например паролями и сертификатами. Константы доступны для всех поставщиков FaaS и сред выполнения.

---

**Примечание** В отличие от секретных элементов, константы действий с поддержкой расширяемости можно использовать только для секретных элементов с поддержкой расширяемости. Дополнительные сведения о секретных элементах см. в разделе [Создание секретных элементов для использования в действиях с поддержкой расширяемости](#).

---

Константы действий с поддержкой расширяемости доступны всем пользователям в организации.



Константы используют функцию `context.getSecret()` для запуска в рамках сценария. Эта функция использует имя константы в качестве параметра. Например, в качестве зашифрованного входного параметра в действии можно использовать константу действия с поддержкой расширяемости с именем `abxconstant`. Чтобы добавить этот входной параметр в сценарий действия, необходимо использовать `context.getSecret(inputs["abxconstant"])`.

## Процедура

1. Создание константы действия с поддержкой расширяемости.
  - а) Выберите **Расширяемость > Библиотека > Действия**.
  - б) Выберите **Константы действий**.
  - в) Чтобы создать константу, нажмите **Создать константу действия**.
  - г) Введите имя и значение константы, а затем нажмите **Сохранить**.
2. Добавьте константу в действие с поддержкой расширяемости.
  - а) Выберите существующее или создайте новое действие с поддержкой расширяемости.
  - б) В разделе **Входные данные по умолчанию** установите флажок **Секретный элемент**.
  - в) Найдите константу и добавьте ее во входные данные действий с поддержкой расширяемости.
  - г) Добавьте константу в сценарий действия с поддержкой расширяемости с помощью функции `context.getSecret()`.
  - д) Чтобы протестировать константу действия с поддержкой расширяемости, нажмите **Тестировать**.

## Создание общих действий расширяемости

Администратор Cloud Assembly создает действия расширяемости, которые могут совместно использоваться в разных проектах без экспорта и импорта действия.

Сведения об экспорте и импорте действий расширяемости см. в разделе [Экспорт и импорт действий с поддержкой расширяемости](#).

## Необходимые условия

Создайте два или более проектов в организации Cloud Assembly.

## Процедура

1. Выберите **Расширяемость > Библиотека > Действия**.
2. Щелкните **Создать действие**.
3. Введите имя для действия расширяемости.
4. (дополнительно) Введите описание для действия расширяемости.
5. Выберите проект, в котором создано действие расширяемости.
6. Установите флажок **Предоставить общий доступ участникам всех проектов в этой организации**.
7. Нажмите кнопку **Далее**.

8. Создайте или импортируйте сценарий действий и сохраните действие расширяемости.

---

**Примечание** В разделе **Параметры** можно включить или отключить общий доступ. Если действие расширяемости используется в подписках, общий доступ отключить нельзя. Чтобы отключить общий доступ, необходимо удалить действие расширяемости из подписок.

---

9. Создайте подписку с поддержкой расширяемости, добавьте общее действие расширяемости и установите для области подписки значение **Любой проект**.

---

**Примечание** Дополнительные сведения о создании подписок с поддержкой расширяемости см. в разделе [Создание подписки с поддержкой расширяемости](#).

---

Подписка с поддержкой расширяемости запускается соответствующими событиями в любом из ваших проектов.

#### Следующие шаги

Кроме того, можно импортировать общие действия расширяемости в качестве источника содержимого в каталог Service Broker. При выборе исходного проекта укажите проект, в котором было создано действие расширяемости. Дополнительные сведения о добавлении действий расширяемости в Service Broker см. в разделе [Добавление действий расширяемости в каталог Service Broker](#).

#### Ведение журнала Azure для действий с поддержкой расширяемости на основе Python

Теперь функции ведения журнала Microsoft Azure 3.x можно использовать в сценарии действий с поддержкой расширяемости.

В настоящее время в действиях с поддержкой расширяемости в Cloud Assembly используется API-интерфейс для создания сценариев Microsoft Azure 3.x, который заменил предыдущую версию 1.x. API-интерфейс для создания сценариев Microsoft Azure 3.x основан на Linux и работает в среде контейнера.

Вследствие изменения версии функции ведения журнала, внедренные в сценарий действий с поддержкой расширяемости, которые используют Microsoft Azure в качестве поставщика возможностей FaaS («функция как услуга»), работают иначе. В следующих двух примерах сценариев показаны различные функции ведения журнала, используемые в двух версиях API-интерфейса.

Пример сценария Microsoft Azure 1.x.

```
def handler(context, inputs):
    greeting = "Hello, {0}!".format(inputs["target"])
    print(greeting)

    outputs = {
        "greeting": greeting
    }

    return outputs
```

Пример сценария Microsoft Azure 3.x.

```
import logging
```

```
def handler(context, inputs):
    greeting = "Hello, {0}!".format(inputs["target"])
    logging.info(greeting)

    outputs = {
        "greeting": greeting
    }

    return outputs
```

В предыдущем примере показано, что версия 3.x добавляет функцию `import logging` в начале сценария и заменяет функцию `print()` на `logging.info()`. Чтобы продолжить ведение журнала с помощью действий с поддержкой расширяемости, созданных в API-интерфейсе Microsoft Azure 1.x, необходимо изменить функции ведения журнала в сценарии так, чтобы он совпадал с примером на основе Microsoft Azure 3.x.

Дополнительные сведения о ведении журнала см. в [руководстве разработчика Python по возможностям Azure](#).

### Экспорт и импорт действий с поддержкой расширяемости

С помощью Cloud Assembly можно экспортировать и импортировать действия расширяемости для использования в разных проектах.

#### Необходимые условия

Существующее действие расширяемости.

#### Процедура

##### 1. Экспорт действия расширяемости.

- а) Выберите **Расширяемость > Библиотека > Действия**.
- б) Выберите действие расширяемости и нажмите **Экспорт**.

Сценарий действий и его зависимости сохраняются в локальной среде в виде ZIP-файла.

##### 2. Импорт действия расширяемости.

- а) Выберите **Расширяемость > Библиотека > Действия**.
- б) Щелкните **Импортировать**.
- в) Выберите экспортированное действие расширяемости и назначьте его проекту.
- г) Щелкните **Импортировать**.

---

**Примечание** Если импортированное действие расширяемости уже назначено указанному проекту, система предложит выбрать политику разрешения конфликтов.

---

### Что такое поток действий

Потоки действий — это набор сценариев действий расширяемости, которые используются для продления жизненного цикла и расширения автоматизации.

Все потоки действий начинаются с параметра `flow_start` и заканчиваются параметром `flow_end`. Для связи между собой нескольких сценариев действий по расширению можно использовать следующие элементы потока действий:

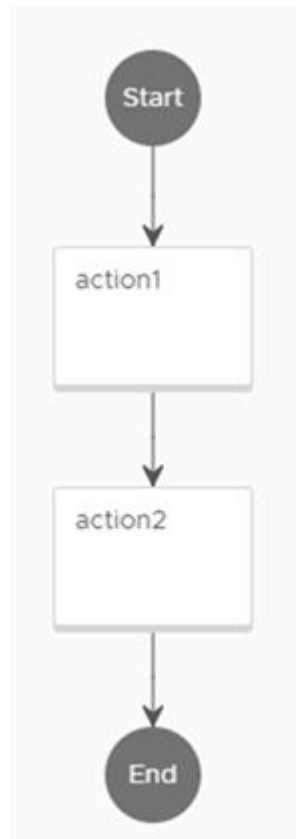
- **Последовательные потоки действий** — несколько сценариев действий по расширению, выполняемых последовательно.
- **Последовательности действий с ветвлением** — несколько сценариев действий по расширению или потоков, которые выполняются по разным путям, но приходят в одну выходную точку.
- **Потоки действий с объединением** — несколько сценариев действий по расширению или потоков, которые объединяются и приходят в одну выходную точку.
- **Последовательности действий с условиями** — несколько сценариев действий по расширению или потоков, которые выполняются при соблюдении некоторого условия.

### Последовательные потоки действий

Несколько сценариев с действиями, обеспечивающих расширяемость и выполняемых последовательно.

```
version: "1"
flow:
  flow_start:
    next: action1
  action1:
    action: <action_name>
    next: action2
  action2:
    action: <action_name>
    next: flow_end
```

**Примечание** Можно заиклить поток на предыдущем действии, указав его в качестве значения для параметра `next`. Например, в этом примере можно вместо `next: flow_end` ввести `next: action1`, чтобы повторно запустить действие `action1` и тем самым перезапустить последовательность действий.



### Последовательности действий с ветвлением

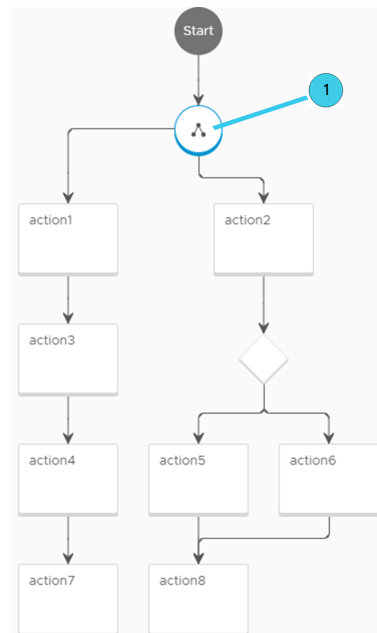
Несколько сценариев или последовательностей действий, обеспечивающих расширяемость; они создают ветви, выходные данные которых формируют единый результат.

```

version: "1"
flow:
  flow_start:
    next: forkAction
  forkAction:
    fork:
      next: [action1, action2]
  action1:
    action: <action_name>
    next: action3
  action3:
    action: <action_name>
    next: action4
  action4:
    action: <action_name>
    next: action7
  action7:
    action: <action_name>
  action2:
    action: <action_name>

```

**Примечание** Можно заиклить поток на предыдущем действии, указав его в качестве значения для параметра `next`. Например, вместо того чтобы завершить последовательность действий, введя `next: flow_end`, можно задать повторный запуск действия `action1`, введя `next: action1`, и тем самым перезапустить последовательность действий.



1 Элемент «Ветка»

### Потоки действий с объединением

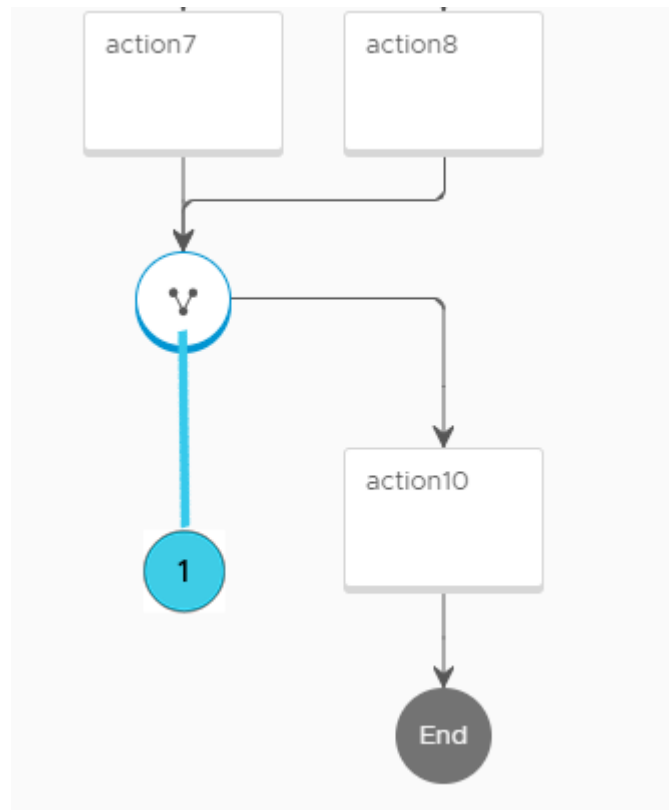
Несколько сценариев или последовательностей действий, обеспечивающих расширяемость; они объединяют ветви, выходные данные которых формируют единый результат.

```

version: "1"
action7:
  action: <action_name>
  next: joinElement
action8:
  action: <action_name>
  next: joinElement
joinElement:
  join:
    type: all
    next: action10
action10:
  action: <action_name>
  next: flow_end

```

**Примечание** Можно зациклить поток на предыдущем действии, указав его в качестве значения для параметра `next`. Например, в этом примере можно вместо `next: flow_end` ввести `next: action1`, чтобы повторно запустить действие `action1` и тем самым перезапустить последовательность действий.



1 Элемент «Объединение»

### Последовательности действий с условиями

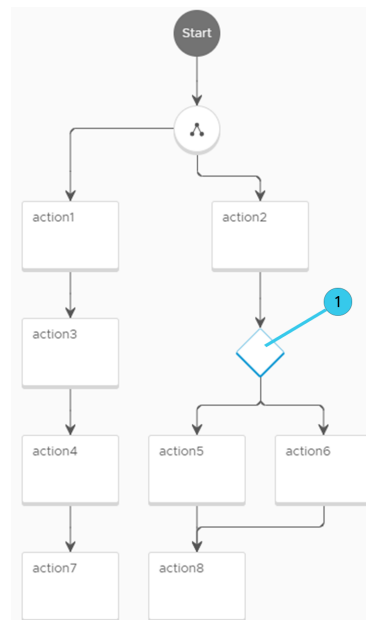
Несколько сценариев или последовательностей действий, обеспечивающих расширяемость; они запускаются элементом-переключателем при выполнении определенного условия.

В некоторых случаях для запуска действия требуется, чтобы заданное условие было истинным (`true`). В других случаях, как показано в этом примере, для запуска действия требуется, чтобы параметры имели определенные значения. Если ни одно из условий не выполняется, то рабочий процесс, содержащий действия, не запускается.

```

version: 1
id: 1234
name: Test
inputs: ...
outputs: ...
flow:
  flow_start:
    next: forkAction
  forkAction:
    fork:
      next: [action1, action2]
  action1:
    action: <action_name>
    next: action3
  action3:
    action: <action_name>
    next: action4
  action4:
    action: <action_name>
    next: action7
  action7:
    action: <action_name>
    next: joinElement
  action2:
    action: <action_name>
    next: switchAction
  switchAction:
    switch:
      "${1 == 1}": action5
      "${1 != 1}": action6
  action5:
    action: <action_name>
    next: action8
  action6:
    action: <action_name>
    next: action8
  action8:
    action: <action_name>

```



1 Элемент-переключатель

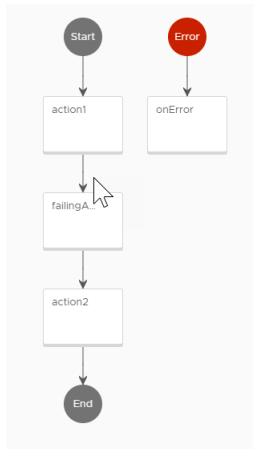
**Примечание** Можно зациклить поток на предыдущем действии, указав его в качестве значения для параметра `next`. Например, вместо того чтобы завершить последовательность действий, введя `next: flow_end`, можно задать повторный запуск действия `action1`, введя `next: action1`, и тем самым перезапустить последовательность действий.

## Использование обработчика ошибок при работе с последовательностями действий

С помощью обработчика ошибок можно настроить последовательность действий таким образом, чтобы на определенных этапах последовательности создавалось сообщение об ошибке.

Обработчику ошибок требуется два вида входных данных:

- определенное сообщение о сбое действия;
- входные данные для последовательности действий.



Если последовательность действий включает в себя элемент «Обработчик ошибок», то при неудачном завершении какого-либо действия появится соответствующее сообщение об ошибке. Обработчик ошибок сам по себе также является действием. Следующий сценарий представляет собой пример обработчика ошибок, который может использоваться в последовательности действий.

```
def handler(context, inputs):

    errorMsg = inputs["errorMsg"]
    flowInputs = inputs["flowInputs"]

    print("Flow execution failed with error {0}".format(errorMsg))
    print("Flow inputs were: {0}".format(flowInputs))

    outputs = {
        "errorMsg": errorMsg,
        "flowInputs": flowInputs
    }

    return outputs
```

Сведения об успешно и неудачно выполненных действиях можно просмотреть в окне «Выполненные действия».

	Status	Run ID	Action
<input type="checkbox"/>	Completed	8a76996b6839fe3c01684...	error-handler
<input type="checkbox"/>	Failed	8a76996b6839fe3c01684...	failing-action
<input type="checkbox"/>	Completed	8a76996b6839fe3c01684...	simple-hello
<input type="checkbox"/>	Completed	8a76996b6839fe3c01684...	flow-with-handler

В этом примере последовательность действий flow-with-handler (поток, включающий обработчик ошибок) был успешно выполнен. Однако одно из действий завершилось сбоем, в результате чего был запущен обработчик ошибок и создано сообщение об ошибке.



## Отслеживание выполнения действий

На вкладке выполнения действий приводится журнал, в котором отражены действия с поддержкой расширяемости, запущенные подписками, и их статусы.

Чтобы просмотреть журнал выполнения действий, выберите **Расширяемость > Действие > Выполненные действия**. Кроме того, список циклов выполнения действий отфильтровать по одному или нескольким свойствам одновременно.

## Устранение проблем при выполнении действий по расширению

В случае неудачного выполнения действия по расширению можно осуществить соответствующие действия, чтобы устранить проблему.

Если действие завершается сбоем, может поступить сообщение об ошибке, о состоянии сбоя, а также журнальная запись об ошибке. Если при выполнении действия возникла ошибка, это может быть вызвано ошибкой развертывания или кода.

Проблема	Решение
Ошибка развертывания	Такие ошибки являются результатом проблем, связанных с настройкой учетных записей облачной службы, развертыванием действий или другими зависимостями, которые могут препятствовать развертыванию. Убедитесь, что используемый проект определен в настроенной учетной записи облачной службы, и обладает разрешениями на выполнение функций. Перед повторным выполнением действия его можно протестировать на определенном проекте на странице сведений о действии.
Ошибка кода	Такие ошибки являются результатом недопустимых сценариев или кода. Для устранения неполадок и исправления некорректных сценариев используйте журнал выполненных действий.

## Подписки на рабочие процессы с поддержкой расширяемости

Чтобы расширить жизненный цикл приложений, можно использовать размещенные рабочие процессы vRealize Orchestrator вместе со службой Cloud Assembly.

## Изменение свойств виртуальной машины с помощью подписки на рабочие процессы vRealize Orchestrator

Используя существующий рабочий процесс vRealize Orchestrator, можно изменить свойства виртуальной машины и добавить виртуальные машины в Active Directory.

Параметры темы события определяют формат полезных данных для сообщений службы брокера событий (Event Broker Service, EBS). Чтобы получать и использовать полезные данные из сообщений службы EBS в рамках рабочего процесса, необходимо определить входные параметры рабочего процесса `inputProperties`.

### Необходимые условия

- Роль пользователя с правами администратора облака
- Существующие локальные рабочие процессы vRealize Orchestrator.
- Успешная интеграция и подключение к серверу клиента vRealize Orchestrator.

### Процедура

1. Выберите **Расширяемость > Подписки**.
2. Щелкните **Создать подписку**.
3. Создайте подписку, используя следующие параметры:

Параметр	Значение
Имя	<b>RenameVM</b>
Тема события	Выберите тему события, которая подходит для требуемой интеграции с vRealize Orchestrator. Например, «Выделение вычислительных ресурсов».
Блокирующая/неблокирующая	Неблокирующая
Действие или рабочий процесс	Выберите тип готовности к запуску vRealize Orchestrator. Выберите нужный рабочий процесс. Например, «Задание имени ВМ».

4. Чтобы сохранить подписку, нажмите **Сохранить**.
5. Назначьте и активируйте подписку, создав новый облачный шаблон или развернув существующий.

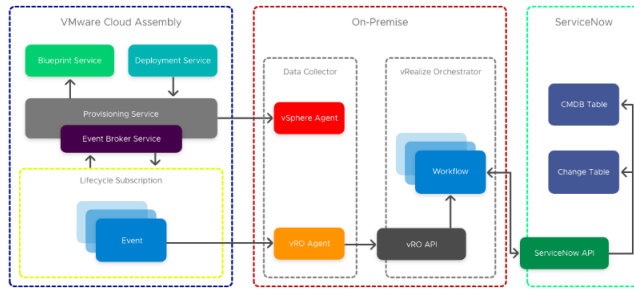
### Следующие шаги

Убедитесь в том, что рабочий процесс успешно запущен, используя один из следующих методов.

- Проверьте журнал выполнения рабочих процессов: **Расширяемость > Действия > Выполненные рабочие процессы**.
- Откройте клиент vRealize Orchestrator и проверьте статус рабочего процесса, перейдя в рабочий процесс и отобразив его статус либо открыв вкладки конкретных журналов.

## Интеграция службы Cloud Assembly с решением ServiceNow для управления ИТ-службами (ITSM) с помощью рабочих процессов vRealize Orchestrator

С помощью размещенных рабочих процессов vRealize Orchestrator службу Cloud Assembly можно интегрировать с решением ServiceNow для обеспечения соответствия нормативным требованиям в отношении управления ИТ-службами (ITSM).



В корпоративной среде платформа управления облаком часто интегрируется с платформой управления ИТ-службами (ITSM) и базой данных управления конфигурациями (CMDB) для соответствия нормативным требованиям. Следуя данному примеру, можно интегрировать службу Cloud Assembly с решением ServiceNow для CMDB и ITSM с помощью размещенных рабочих процессов vRealize Orchestrator. При использовании интеграций и рабочих процессов vRealize Orchestrator теги возможностей могут быть особенно полезны в случаях, когда вы имеете дело с несколькими экземплярами для разных сред. Дополнительные сведения о тегах возможностей см. в разделе [Использование тегов возможностей в Cloud Assembly](#).

**Примечание** Кроме того, интегрировать ServiceNow со службой Cloud Assembly можно, используя сценарии действий с поддержкой расширяемости. Дополнительные сведения об интеграции решения ServiceNow с помощью сценариев действий с поддержкой расширяемости см. в разделе [Интеграция службы Cloud Assembly и ServiceNow с помощью действий с поддержкой расширяемости](#).

В этом примере интеграция с решением ServiceNow состоит из трех рабочих процессов высшего уровня. У каждого рабочего процесса есть свои подписки, что позволяет выполнять обновление и итерацию для каждого компонента по отдельности.

- Точка входа подписки на событие — «Базовое ведение журнала» — идентифицирует пользователя, отправившего запрос, и виртуальную машину vCenter (при необходимости).
- Рабочий процесс интеграции разделяет объекты и подает входные данные в технический рабочий процесс, обрабатывает журналы, свойства и обновления выходных данных.
- Технический рабочий процесс осуществляет интеграцию нисходящих систем для API-интерфейса решения ServiceNow и создает API-интерфейс инфраструктуры как услуги Cloud Assembly, базы данных управления конфигурациями, непрерывной интеграции и непрерывного реагирования с дополнительными свойствами виртуальной машины за пределами полезных данных.

#### Необходимые условия

- Автономная или кластерная среда vRealize Orchestrator.
- Интеграция vRealize Orchestrator в Cloud Assembly. Дополнительные сведения об интеграции автономного экземпляра vRealize Orchestrator со службой Cloud Assembly см. в разделе [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).

## Процедура

1. Создайте и сохраните в службе vRealize Orchestrator файл конфигурации, содержащий общую конфигурацию, которая используется в нескольких рабочих процессах.
2. Сохраните маркер API-интерфейса Cloud Assembly в том же расположении, что и файл конфигурации на шаге 1.

---

**Примечание** Срок действия маркера API-интерфейса Cloud Assembly ограничен.

---

3. Создайте рабочий процесс в службе vRealize Orchestrator с использованием предоставленного элемента сценария. Этот сценарий ссылается на узел REST и определяет его расположение. Он также стандартизирует действия REST, в которых используется дополнительный параметр маркера, который добавляется в качестве заголовка дополнительной проверки подлинности.

```
var configPath = "CS"
var configName = "environmentConfig"
var attributeName = "CASRestHost"

//get REST Host from configuration element
var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath,configName,attribute
Name)

var ConfigurationElement =
System.getModule("au.com.cs.example").getConfigurationElementByName(configName,configPath);
System.debug("ConfigurationElement:" + ConfigurationElement);
var casToken = ConfigurationElement.getAttributeWithKey("CASToken")["value"]
if(!casToken){
    throw "no CAS Token";
}
//REST Template
var opName = "casLogin";
var opTemplate = "/iaas/login";
var opMethod = "POST";

// create the REST operation:
var opLogin =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

//cas API Token
var contentObject = {"refreshToken":casToken}
postContent = JSON.stringify(contentObject);

var loginResponse =
System.getModule("au.com.cs.example").executeOp(opLogin,null,postContent,null) ;

try{
    var tokenResponse = JSON.parse(loginResponse)['token']
    System.debug("token: " + tokenResponse);
} catch (ex) {
    throw ex + " No valid token";
}
```

```

//REST Template Machine Details
var opName = "machineDetails";
var opTemplate = "/iaas/machines/" + resourceId;
var opMethod = "GET";

var bearer = "Bearer " + tokenResponse;

var opMachine =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

// (Rest Operation, Params, Content, Auth Token)
var vmResponse =
System.getModule("au.com.cs.example").executeOp(opMachine,null,"",bearer) ;

try{
    var vm = JSON.parse(vmResponse);
} catch (ex) {
    throw ex + " failed to parse vm details"
}

System.log("cpuCount: " + vm["customProperties"]["cpuCount"]);
System.log("memoryInMB: " + vm["customProperties"]["memoryInMB"]);

cpuCount = vm["customProperties"]["cpuCount"];
memoryMB = vm["customProperties"]["memoryInMB"];

```

Этот сценарий отправляет выходные значения `cpuCount` и `memoryMB` в родительский рабочий процесс и обновляет существующие свойства `customProperties`. Эти значения можно использовать в последующих рабочих процессах при создании базы данных управления конфигурациями (CMDB).

4. Добавьте элемент сценария создания непрерывной интеграции базы данных управления конфигурациями ServiceNow в рабочий процесс. Этот элемент находит узел REST для ServiceNow, используя элемент конфигурации, создает операцию REST для таблицы `cmdb_ci_vmware_instance`, создает строку объекта содержимого на основе входных данных рабочего процесса для пост-данных и выводит возвращенное значение `sys_id`.

```

var configPath = "CS"
var configName = "environmentConfig"
var attributeName = "serviceNowRestHost"
var tableName = "cmdb_ci_vmware_instance"

//get REST Host from configuration element
var restHost =
System.getModule("au.com.cs.example").getRestHostFromConfig(configPath,configName,attribute
Name)

//REST Template
var opName = "serviceNowCreatCI";
var opTemplate = "/api/now/table/" + tableName;
var opMethod = "POST";

// create the REST operation:

```

```

var opCI =
System.getModule("au.com.cs.example").createOp(restHost,opName,opMethod,opTemplate);

//cmdb_ci_vm_vmware table content to post;
var contentObject = {};
contentObject["name"] = hostname;
contentObject["cpus"] = cpuTotalCount;
contentObject["memory"] = MemoryInMB;
contentObject["correlation_id"]= deploymentId
contentObject["disks_size"]= diskProvisionGB
contentObject["location"] = "Sydney";
contentObject["vcenter_uuid"] = vcUuid;
contentObject["state"] = "On";
contentObject["owned_by"] = owner;

postContent = JSON.stringify(contentObject);
System.log("JSON: " + postContent);

// (Rest Operation, Params, Content, Auth Token)
var ciResponse =
System.getModule("au.com.cs.example").executeOp(opCI,null,postContent,null) ;

try{
    var cmdbCI = JSON.parse(ciResponse);
} catch (ex) {
    throw ex + " failed to parse ServiceNow CMDB response";
}

serviceNowSysId = cmdbCI['result']['sys_id'];

```

- Используя выходные данные дочернего рабочего процесса, создайте объект свойства на основе существующих значений `customProperties` и перезапишите свойство `serviceNowSysId`, используя значение, полученное из ServiceNow. Этот уникальный идентификатор используется в базе данных управления конфигурациями для маркировки удаляемого экземпляра как списанного.

## Результаты

Служба Cloud Assembly успешно интегрирована с системой управления ИТ-службами ServiceNow. Дополнительные сведения об использовании рабочих процессов для интеграции системы ServiceNow со службой Cloud Assembly см. в разделе [Расширение Cloud Assembly с помощью vRealize Orchestrator для интеграции с ServiceNow](#).

## Дополнительные сведения о подписках на рабочие процессы

Благодаря интеграции vRealize Orchestrator с Cloud Assembly можно расширять жизненный цикл приложений с помощью рабочих процессов.

vRealize Automation включает в себя встроенное развертывание vRealize Orchestrator. В подписках можно использовать библиотеку рабочих процессов встроенного развертывания vRealize Orchestrator. Рабочие процессы можно создавать, изменять и удалять с помощью клиента vRealize Orchestrator.

Кроме того, можно интегрировать внешнее развертывание vRealize Orchestrator в Cloud Assembly. См. раздел [Настройка интеграции с решением vRealize Orchestrator в службе Cloud Assembly](#).

## Практические рекомендации по созданию рабочих процессов vRealize Orchestrator

Подписка на рабочий процесс основана на определенной теме события и параметрах события этой темы. Чтобы в рамках подписки могли инициализироваться рабочие процессы vRealize Orchestrator, необходимо выбрать для них правильные входные параметры для обработки сведений о событиях.

### Входные параметры рабочих процессов

Настраиваемый рабочий процесс может содержать все параметры или только один, который будет получать все полезные данные.

Чтобы использовать один параметр, настройте параметр типа `Properties` с именем `inputProperties`.

### Выходные параметры рабочих процессов

Настраиваемый рабочий процесс может содержать выходные параметры, относящиеся к последующим событиям, необходимым для определенного типа темы события с ответом.

Если тема события предусматривает ответ, выходные параметры рабочего процесса должны соответствовать параметрам схемы ответа.

## Отслеживание выполнения рабочих процессов

В окне **Выполненные рабочие процессы** отображаются журналы рабочих процессов, инициированных подписками, и их статус.

Для просмотра журналов выполняемых рабочих процессов выберите **Расширяемость > Действие > Выполненные рабочие процессы**.

## Устранение проблем с подписками на рабочий процесс

В случае сбоя подписки на рабочий процесс можно осуществить соответствующие действия, чтобы устранить проблему.

Ошибка при выполнении рабочего процесса может привести к невозможности запуска подписки на рабочий процесс или ее неудачному завершению. Сбои при выполнении рабочего процесса могут возникать из-за нескольких распространенных проблем.

Проблема	Причина	Решение
Подписка на рабочий процесс vRealize Orchestrator не запущена или выполнена с ошибками.	Подписка на рабочий процесс настроена на запуск настраиваемого рабочего процесса при получении сообщения о событии, но рабочий процесс не запускается или выполняется с ошибками.	<ol style="list-style-type: none"> <li>1 Убедитесь, что подписка на рабочий процесс сохранена корректно.</li> <li>2 Убедитесь, что условия подписки на рабочий процесс настроены правильно.</li> <li>3 Убедитесь, что vRealize Orchestrator содержит указанный рабочий процесс.</li> <li>4 Убедитесь, что рабочий процесс правильно настроен в vRealize Orchestrator.</li> </ol>
Подписка на рабочий процесс vRealize Orchestrator для запроса на подтверждение не запущена.	Настроена подписка на рабочий процесс предварительного или последующего подтверждения для запуска рабочего процесса vRealize Orchestrator. Рабочий процесс не запускается, если компьютер, соответствующий определенным критериям, запрашивается в каталоге служб.	<p>Чтобы успешно запустить подписку на рабочий процесс подтверждения, необходимо убедиться, что все компоненты настроены правильно.</p> <ol style="list-style-type: none"> <li>1 Убедитесь, что политика подтверждения активна и правильно применена.</li> <li>2 Убедитесь, что подписка на рабочий процесс правильно настроена и сохранена.</li> <li>3 Проверьте в журналах событий сообщения, относящиеся к подтверждениям.</li> </ol>
Подписка на рабочий процесс vRealize Orchestrator для запроса на подтверждение отклонена.	<p>Вы настроили подписку на рабочий процесс предварительного или последующего подтверждения, который запускает указанный рабочий процесс vRealize Orchestrator, но запрос отклонен на уровне внешнего подтверждения.</p> <p>Одной из возможных причин является внутренняя ошибка при запуске рабочего процесса в vRealize Orchestrator. Например, отсутствует рабочий процесс или не запущен сервер vRealize Orchestrator.</p>	<ol style="list-style-type: none"> <li>1 Проверьте в журналах сообщения, относящиеся к подтверждениям.</li> <li>2 Убедитесь, что сервер vRealize Orchestrator работает.</li> <li>3 Убедитесь, что vRealize Orchestrator содержит указанный рабочий процесс.</li> </ol>

## Дополнительные сведения о подписках с поддержкой расширяемости

Для увеличения жизненного цикла приложений можно использовать действия расширяемости или размещенные рабочие процессы vRealize Orchestrator с подписками с поддержкой расширяемости.

При возникновении в среде инициирующего события активируется подписка и запускается указанный рабочий процесс или действие расширяемости. В журнале событий можно просматривать системные события, выполняемые рабочие процессы в окне «Циклы выполнения рабочих процессов», а также выполняемые действия в окне «Циклы выполнения действий». Подписки зависят от проекта, т. е. связаны с облачными шаблонами и развертываниями в указанном проекте.



## Расширяемость: терминология

При работе с подписками с поддержкой расширяемости в службе Cloud Assembly может встретиться специфическая терминология, касающаяся подписок и службы брокера событий.

Таблица 6-6. Расширяемость: терминология

Термин	Описание
Тема события	<p>Описывает набор событий с одинаковой логической целью и структурой. Каждое событие является экземпляром темы события.</p> <p>Некоторым темам событий можно назначить блокирующие параметры. Дополнительные сведения см. в разделе <a href="#">Темы блокирующих событий</a>.</p>
Событие	Обозначает изменение состояния в авторе или объектах под его управлением. Событие является объектом, записывающим информацию о наступлении события.
Служба брокера события	Служба, подготавливающая к отправке сообщения, которые автор опубликовал для подписчиков.
Полезные данные	Данные по событию, содержащие все соответствующие свойства, связанные с темой этого события.
Подписка	Показывает, что подписчик проявил интерес в получении уведомлений о событии, подписавшись на тему события и указав критерии отправки уведомления. Посредством подписок рабочие процессы и действия с поддержкой расширяемости привязываются к событиям-триггерам, которые используются для автоматизации определенных частей жизненного цикла приложений.
Подписчик	Пользователи, получающие уведомления о событиях, опубликованных в службе брокера событий, в рамках определения подписки. Подписчик также можно называться потребителем или клиентом.
Системный администратор	Пользователь с правом создания, чтения, обновления и удаления подписки на рабочие процессы арендатора и системы с использованием Cloud Assembly.
Подписка на рабочий процесс	Определяет условия и тему события для запуска рабочего процесса vRealize Orchestrator.
Подписка на действия	Определяет условия и темы события для запуска действия с поддержкой расширяемости.
Рабочий процесс	Рабочий процесс vRealize Orchestrator, интегрированный в службу Cloud Assembly. Эти рабочие процессы можно связать с событиями посредством подписок.

Таблица 6-6. Расширяемость: терминология (продолжение)

Термин	Описание
Действие с поддержкой расширяемости	Оптимизированный сценарий в виде кода, выполняемый после наступления события-триггера в подписке. Действия с поддержкой расширяемости аналогичны рабочим процессам, но являются более компактными. Действия с поддержкой расширяемости можно настроить в службе Cloud Assembly.
Выполнение действий	Для доступа выберите вкладку <b>Выполненные действия</b> . Здесь приводится подробный журнал действий с поддержкой расширяемости, выполненных в ответ на наступление события-триггера.

### Темы блокирующих событий

Некоторые темы событий поддерживают блокирующие события. Характеристики подписки с поддержкой расширяемости зависят от того, поддерживает ли тема эти типы событий, а также от способа настройки подписки.

В подписках с поддержкой расширяемости Cloud Assembly могут использоваться два основных типа тем событий: темы неблокирующих событий и блокирующих событий. Тип темы события определяет особенности подписки с поддержкой расширяемости.

### Темы неблокирующих событий

Темы неблокирующих событий позволяют создавать только неблокирующие подписки. Неблокирующие подписки запускаются асинхронно, поэтому не следует полагаться на порядок их запуска.

### Темы блокирующих событий

Некоторые темы событий поддерживают возможность блокировки. Если подписка помечена как блокирующая, никакие другие подписки с соответствующими условиями не получают сообщений, которые соответствуют заданным условиям до тех пор, пока не будет запущен готовый к запуску элемент блокирующей подписки.

Блокирующие подписки запускаются в порядке приоритета. Высший приоритет имеет значение 0. При наличии нескольких блокирующих подписок для одной и той же темы события с одинаковым уровнем приоритета подписки запускаются в обратном алфавитном порядке по имени подписки. После обработки всех блокирующих подписок отправляются одновременные сообщения всем неблокирующим подпискам. Поскольку блокирующие подписки запускаются синхронно, при отправке уведомлений последующим подпискам в измененные полезные данные событий включается обновленное событие.

Темы блокирующего события можно использовать для управления несколькими подписками, которые зависят друг от друга.

Например, могут существовать две подписки на рабочий процесс предоставления, одна из которых зависит от результатов другой. Первая подписка изменяет свойство во время предоставления, а вторая записывает новое свойство, например имя компьютера, в файловую систему. Приоритет подписки `ChangeProperty` имеет значение 0, а `RecordProperty` — 1, поскольку вторая подписка использует результаты первой подписки. После предоставления ресурсов для компьютера запускается подписка на процесс `ChangeProperty`. Так как условия подписки `RecordProperty` основаны на условии выполнения

после предоставления, событие инициирует подписку `RecordProperty`. Однако поскольку рабочий процесс `ChangeProperty` является блокирующим, событие не будет получено, пока процесс не будет завершен. После изменения имени компьютера и завершения первой подписки на рабочий процесс запускается вторая подписка на рабочий процесс, и имя компьютера записывается в файловой системе.

### Готовый к запуску элемент для восстановления

Для тем блокирующих событий в подписку можно добавить готовый к запуску элемент для восстановления. Готовый к запуску элемент для восстановления в подписке запускается в случае сбоя основного готового к запуску элемента. Например, можно создать подписку на рабочий процесс, где основным готовым к запуску элементом является рабочий процесс, который создает записи в системе CMDB, такой как ServiceNow. Даже если произойдет сбой подписки на рабочий процесс, некоторые записи смогут быть созданы в системе CMDB. В данном сценарии можно использовать готовый к запуску элемент для восстановления, чтобы очистить записи, оставшиеся в системе CMDB из-за сбоя готового к запуску элемента.

Для сценариев использования, содержащих несколько зависимых друг от друга подписок, в готовый к запуску элемент для восстановления можно добавить свойство `ebs.recover.continuation`. С помощью этого свойства можно указать, должна ли служба расширяемости продолжить работу со следующей подпиской в цепочке в случае сбоя текущей подписки.

## Темы событий, предоставленные с помощью Cloud Assembly

Cloud Assembly включает в себя предварительно определенные темы событий.

### Темы событий

Темы событий — это категории, которые группируют сходные события. Когда тема события назначается подписке, это определяет, какое событие будет запускать эту подписку. Следующие темы событий предоставляются службой Cloud Assembly по умолчанию. Все темы могут использоваться для добавления или обновления настраиваемых свойств или тегов для ресурса. Если рабочий процесс vRealize Orchestrator или действие расширяемости завершается ошибкой, соответствующая задача также завершается ошибкой.

Таблица 6-7. Темы событий Cloud Assembly

Тема события	Возможность блокировки	Описание
Cloud template configuration	Нет	Выдается, если происходит событие настройки облачного шаблона, например создание или удаление облачного шаблона. Эта тема события может быть полезной для уведомления внешних систем о таких событиях.
Cloud template version configuration	Нет	Выдается, если происходит новое событие с версией облачного шаблона, например создание, выпуск, отмена выпуска или восстановление версии. Эта тема события полезна при интеграции с системами управления версиями сторонних производителей.

Таблица 6-7. Темы событий Cloud Assembly (продолжение)

Тема события	Возможность блокировки	Описание
Compute allocation	Да	Выдается до выделения resourcenames и hostselections. На данном этапе можно изменить оба этих свойства. Выдается один раз для кластера компьютеров.
Compute gateway post provisioning	Да	Выдается после подготовки шлюза вычислительных ресурсов.
Compute gateway post removal	Да	Выдается после удаления шлюза вычислительных ресурсов.
Compute gateway provisioning	Да	Выдается до подготовки шлюза вычислительных ресурсов.
Compute gateway removal	Да	Выдается до удаления шлюза вычислительных ресурсов.
Compute initial power on	Да	Выдается после подготовки ресурса на уровне гипервизора до первого включения питания ресурса. В настоящее время эта тема событий поддерживается только для vSphere. События отправляются для каждого компьютера в кластере.  <b>Примечание</b> Первоначальное включение ресурса можно пропустить.
Compute nat post provisioning	Да	Выдается после подготовки вычислительного ресурса NAT.
Compute nat post removal	Да	Выдается после удаления вычислительного ресурса NAT.
Compute nat provisioning	Да	Выдается до подготовки вычислительного ресурса NAT.
Compute nat removal	Да	Выдается до удаления вычислительного ресурса NAT.
Compute post provision	Да	Выдается после подготовки ресурса. События отправляются для каждого компьютера в кластере.
Compute post removal	Да	Выдается после удаления вычислительного ресурса. События отправляются для каждого компьютера в кластере.

Таблица 6-7. Темы событий Cloud Assembly (продолжение)

Тема события	Возможность блокировки	Описание
Compute provision	Да	Выдается до предоставления ресурса на уровне гипервизора. События отправляются для каждого компьютера в кластере.  <b>Примечание</b> Можно изменить полученный в ходе выделения ресурсов IP-адрес.
Compute removal	Да	Выдается до удаления ресурса. События отправляются для каждого компьютера в кластере.
Compute reservation	Да	Выдается в момент резервирования. Выдается один раз для кластера компьютеров.  <b>Примечание</b> Можно изменить порядок размещения.
Custom resource post provision	Да	Выдается для событий, происходящих после предоставления и запущенных настраиваемыми операциями с ресурсами.
Custom resource pre provision	Да	Выдается для событий, происходящих до предоставления и запущенных настраиваемыми операциями с ресурсами.
Deployment action completed	Да	Выдается по окончании действия с развертыванием.
Deployment action requested	Да	Выдается до окончания действия с развертыванием.
Deployment completed	Да	Выдается после развертывания облачного шаблона или запроса в каталог.
Deployment onboarded	Нет	Выдается после внедрения нового развертывания.
Deployment requested	Да	Выдается до развертывания облачного шаблона или запроса в каталог.
Deployment resource action completed	Да	Выдается после развертывания действия ресурса.
Deployment resource action requested	Да	Выдается до развертывания действия ресурса.
Deployment resource completed	Да	Выдается после предоставления ресурса развертывания.
Deployment resource requested	Да	Выдается до предоставления ресурса развертывания.

Таблица 6-7. Темы событий Cloud Assembly (продолжение)

Тема события	Возможность блокировки	Описание
Disk allocation	Да	Выдается для предварительного выделения дисковых ресурсов.
Disk attach	Да	<p>Выдается до подключения диска к компьютеру. Disk attach — это событие чтения и записи. Для обратной записи поддерживаются следующие свойства диска.</p> <ul style="list-style-type: none"> <li>■ diskFullPaths</li> <li>■ diskDatastoreNames</li> <li>■ diskParentDirs</li> </ul> <p>Для обновления необходимо задать все три свойства диска, относящиеся к vSphere. Все остальные свойства доступны только для чтения.</p> <p><b>Примечание</b> Обратная запись является необязательной для дисков первого класса vSphere.</p>
Disk detach	Да	Выдается после отключения диска от компьютера. Disk detach — это событие категории «только для чтения».
Disk post removal	Да	Выдается после удаления дискового ресурса.
Disk post resize	Да	Выдается после изменения размера дискового ресурса.
Kubernetes cluster allocation	Да	Выдается для предварительного выделения ресурсов для кластера Kubernetes.
Kubernetes cluster post provision	Да	Выдается после подготовки кластера Kubernetes.
Kubernetes cluster post removal	Да	Выдается после удаления кластера Kubernetes.
Kubernetes cluster provision	Да	Выдается до подготовки кластера Kubernetes.
Kubernetes cluster removal	Да	Выдается до инициирования процесса удаления кластера Kubernetes.
Kubernetes namespace allocation	Да	Выдается во время предварительного выделения ресурсов пространства имен Kubernetes.
Kubernetes namespace post provision	Да	Выдается после подготовки ресурса пространства имен Kubernetes.
Kubernetes namespace post removal	Да	Выдается после удаления ресурса пространства имен Kubernetes.

Таблица 6-7. Темы событий Cloud Assembly (продолжение)

Тема события	Возможность блокировки	Описание
Kubernetes namespace provision	Да	Выдается до подготовки пространства имен Kubernetes.
Kubernetes namespace removal	Да	Выдается до удаления ресурса кластера пространств имен.
Kubernetes supervisor namespace allocation	Да	Выдается во время предварительного выделения ресурсов пространства имен управляющей программы Kubernetes.
Kubernetes supervisor namespace post provision	Да	Выдается после подготовки пространства имен управляющей программы.
Kubernetes supervisor namespace post removal	Да	Выдается после удаления ресурса пространства имен управляющей программы.
Kubernetes supervisor namespace provision	Да	Выдается до подготовки пространства имен управляющей программы.
Kubernetes supervisor namespace removal	Да	Выдается до удаления ресурса пространств имен управляющей программы.
Load balancer post provision	Да	Выдается после предоставления подсистемы балансировки нагрузки.
Load balancer post removal	Да	Выдается после удаления подсистемы балансировки нагрузки.
Load balancer provision	Да	Выдается до подготовки подсистемы балансировки нагрузки.
Load balancer removal	Да	Выдается до удаления подсистемы балансировки нагрузки.
Network Configure	Да	Выдается при настройке сети в ходе выделения вычислительных ресурсов.  <b>Примечание</b> Тема «Настройка сети» поддерживает использование нескольких IP-адресов и сетевых адаптеров.
Network post provisioning	Да	Выдается после подготовки сетевого ресурса.
Network post removal	Да	Выдается после удаления сетевого ресурса.
Network provisioning	Да	Выдается до подготовки сетевого ресурса.
Network removal	Да	Выдается до удаления сетевого ресурса.
Project Lifecycle Event Topic	Нет	Выдается при создании, обновлении или удалении проекта.
Provisioning request	Да	Выдается перед удалением группы безопасности.

Таблица 6-7. Темы событий Cloud Assembly (продолжение)

Тема события	Возможность блокировки	Описание
Security group post provision	Да	Выдается после подготовки группы безопасности.
Security group post removal	Да	Выдается после удаления группы безопасности.
Security group provisioning	Да	Выдается до подготовки группы безопасности.
Security group removal	Да	Выдается перед удалением группы безопасности.

### Параметры события

После добавления темы события можно просмотреть параметры этой темы. Они определяют структуру полезных данных события (или `inputProperties`). Некоторые параметры событий нельзя изменить, и они помечены как доступные только для чтения. Чтобы определить, относится ли данный параметр к категории «только для чтения», щелкните значок сведений справа от параметра.

### Журнал событий расширяемости

На странице событий расширяемости отображается список всех событий, произошедших в пределах среды.

Для просмотра журналов событий расширяемости выберите **Расширяемость > События**. Кроме того, список событий можно отфильтровать по одному или нескольким свойствам. Чтобы просмотреть дополнительные сведения о конкретном событии, выберите его идентификатор.

ID	Timestamp	Event Topic	User Name	Target ID	Description
cbaf56ce-a324-f5ae-5dd1-66d1e5911a6	04/28/20, 1:10 PM	N/A	N/A	endpoints	CREATE
6f621151-2906-dce2-14ab-68c17132a756	03/25/20, 4:22 PM	N/A	N/A	endpoints	CREATE
468e8e55-cf27-e77e-0179-1b5b736717b3	03/25/20, 10:12 AM	N/A	N/A	endpoints	CREATE
d9482883-d1ae-5899-fb06-852c202cc178	03/20/20, 2:41 PM	N/A	N/A	endpoints	CREATE
38584d40-a663-631f-7098-3747aa528d12	01/30/20, 5:35 PM	N/A	N/A	endpoints	CREATE

### Создание подписки с поддержкой расширяемости

Используя возможности интеграции с vRealize Orchestrator и действия расширяемости в службе Cloud Assembly, можно создавать подписки, позволяющие расширять возможности приложений.

Подписки с поддержкой расширяемости позволяют расширять возможности приложений путем запуска рабочих процессов или действий при наступлении определенных событий в рамках жизненного цикла. Кроме того, к подпискам можно применять фильтры, чтобы назначать логические условия для указанного события. Например, можно сделать так, чтобы событие и рабочий процесс или действие запускались только в том случае, если логическое выражение является `'true'`. Это удобно в тех случаях, когда необходимо управлять запуском событий, действий или рабочих процессов.



## Необходимые условия

- Убедитесь в том, что вам назначена роль пользователя с правами администратора облака.
- Если используются рабочие процессы vRealize Orchestrator.
  - Библиотека встроенного клиента vRealize Orchestrator или библиотека любого интегрированного внешнего экземпляра vRealize Orchestrator.
- Если используются действия расширяемости.
  - Существующие сценарии действий расширяемости. Дополнительные сведения см. в разделе [Создание действия с поддержкой расширяемости](#).

## Процедура

1. Выберите **Расширяемость > Подписки**.
2. Щелкните **Создать подписку**.
3. Введите сведения о подписке.
4. Задайте **область организации** для подписки.

---

**Примечание** Дополнительные сведения о создании подписок с поддержкой расширяемости для поставщиков и арендаторов организации см. в разделе [Создание подписок с поддержкой расширяемости для поставщиков или арендаторов](#).

---

5. Выберите **Тема события**.
6. (дополнительно) Задайте условия для темы события.

---

**Примечание** Условия можно создать с помощью выражения синтаксиса JavaScript. Это выражение может включать в себя логические операторы, такие как "&&" (AND), "||" (OR), "^" (XOR) и "!" (NOT). Кроме того, можно использовать арифметические операторы, такие как "==" (equal to), "!=" (not equal to), ">=" (greater than or equal), "<=" (less than or equal), ">" (greater than) и "<" (lesser than). Более сложные логические выражения можно создавать на основе более простых выражений. Чтобы получить доступ к полезным данным события в соответствии с указанными параметрами темы, используйте 'event.data' или любое свойство заголовка события: sourceType, sourceIdentity, timeStamp, eventType, eventTopicId, correlationType, correlationId, description, targetType, targetId, userName или orgId.

---

7. В разделе **Действие/рабочий процесс** выберите готовый к запуску элемент для своей подписки с поддержкой расширяемости.
8. (дополнительно) При необходимости настройте поведение блокировки для темы события.
9. (дополнительно) Чтобы задать область проекта для подписки с поддержкой расширяемости, отмените выбор параметра **Любой проект** и нажмите кнопку **Добавить проекты**.

---

**Примечание** Если область организации для подписки задана как **Любая организация арендатора**, область проекта всегда будет задана как **Любой проект** и ее невозможно будет изменить. Область проекта можно изменить, только если в качестве области организации указана организация поставщика.

---

## 10. Чтобы сохранить подписку, нажмите **Сохранить**.

### Результаты

Подписка создана. При наступлении события, подпадающего под выбранную тему событий, запускается связанный рабочий процесс vRealize Orchestrator или действие расширяемости и все подписчики получают соответствующее уведомление.

### Следующие шаги

После создания подписки можно создать или развернуть облачный шаблон для привязки и использования подписки. Кроме того, можно проверить состояние выполнения рабочего процесса или действия расширяемости на вкладке **Расширяемость** в Cloud Assembly. Если подписки содержат рабочие процессы vRealize Orchestrator, можно также осуществлять мониторинг выполнения рабочих процессов и их состояний в клиенте vRealize Orchestrator.

## Использование подписок с поддержкой расширяемости для управления истечением срока действия развертывания

Для управления развертываниями с истекшим сроком действия и их ресурсами можно использовать действие `Expire` и существующие темы события.

После истечения срока аренды развертывания в среде можно использовать темы события расширяемости для выполнения таких задач, как остановка резервного копирования или мониторинга любых ресурсов развертывания. Для выполнения этих операций по регулярному обслуживанию API-интерфейс vRealize Automation использует действие уровня системы `Expire`. Это действие автоматически запускается системой каждый раз, когда истекает срок аренды развертывания в организации. Запуск действия `Expire` предшествует событию выключения всех ресурсов, связанных с этим развертыванием.

---

**Примечание** В предыдущих выпусках продукта событие выключения запускалось на уровне развертывания после истечения срока аренды. Теперь событие выключения запускается на уровне ресурса для каждого ресурса развертывания, который находится во включенном состоянии.

---

Действие `Expire` включено в полезные данные существующих тем события, таких как **Действие развертывания запрошено** и **Действие развертывания выполнено**. Оно использует параметр `deploymentid` для выполнения задач, связанных с ресурсами развертывания, до и после истечения срока аренды.

---

**Примечание** Действие `Expire` запускается приблизительно через 10–15 минут после истечения срока аренды развертывания. Система не запускает события истечения срока аренды до фактического истечения срока. Действие `Expire` выполняется на уровне системы, и пользователи не могут запускать связанные с ним события вручную.

---

В текущем примере использования тема события **Действие развертывания запрошено** и действие `Expire` применяются для резервного копирования виртуальной машины в развертывании в качестве шаблона. В этом случае резервное копирование выполняется с помощью рабочего процесса vRealize Orchestrator, но ту же самую задачу можно выполнить, используя действие расширяемости в качестве готового к запуску элемента подписки.

## Процедура

1. Перейдите в раздел **Расширяемость > Подписки** и щелкните **Создать подписку**.
2. Введите имя подписки.
3. Просмотрите **Состояние**, чтобы убедиться, что подписка включена.
4. В разделе **Тема события** выберите **Действие развертывания запрошено**.
5. Включите параметр **Условие** и добавьте фильтр для действия истечения срока.

```
event.data.actionName == 'Expire'
```

**Примечание** Тема события **Действие развертывания запрошено** может быть запущена различными операциями по регулярному обслуживанию развертывания, такими как изменением срока действия аренды развертывания. Добавление фильтра действий истечения срока аренды гарантирует, что подписка запускается только для событий истечения срока действия.

6. В разделе **Действие/рабочий процесс** добавьте рабочий процесс vRealize Orchestrator.

Схема этого примера рабочего процесса включает в себя задачу с поддержкой сценариев и элемент рабочего процесса, который содержит рабочий процесс **Клонирование виртуальной машины без настройки**, предварительно настроенный в vRealize Orchestrator. Элемент задачи с поддержкой сценариев включает в себя следующий пример сценария.

```
System.log("Lease expiry action triggered to clone a VM...")

System.log("Deployment Id is: " + inputProperties.deploymentId);
inputHeaders = new Properties();
deploymentId = inputProperties.deploymentId;
pathUriVariable = "/deployment/api/deployments/" + deploymentId + "/resources";
var restClient = vRAHost.createRestClient();
var request = restClient.createRequest("GET", pathUriVariable, null);
var keys = inputHeaders.keys;
for(var key in keys){
    request.setHeader(keys[key], inputHeaders.get(keys[key]));
}
var response = restClient.execute(request);
System.log("Content as string: " + response.contentAsString);
var content = response.contentAsString;
var obj = JSON.parse(content);

var object = new Properties(obj);
var contentJson = object.content;
for (var i = 0; i < contentJson.length; i++) {
    var resources = contentJson[i];

    var resourceProperties = resources.properties;
    System.log("Resource name is: " + resourceProperties.resourceName);
    resourceName = resourceProperties.resourceName;
}

var query = "xpath:name='" + resourceName + "'";
```

```
var vms=Server.findAllForType("VC:VirtualMachine", query);
vcVM=vms[0];

System.log("VM input is: " + vcVM);
dataStoreOutput = datastore
template= true;
name="test-vm-name"
```

7. Решите, будет подписка задана как блокирующая или неблокирующая.

**Примечание** Блокирующая подписка означает, что событие выключения для ресурсов развертывания запускается только после того, как готовый к запуску элемент, в данном случае рабочий процесс истечения срока аренды, будет успешно выполнен. Неблокирующая подписка означает, что событие выключения для ресурсов развертывания запускается независимо от состояния выполнения рабочего процесса.

8. Чтобы завершить редактирование подписки, щелкните **Сохранить**.

#### Следующие шаги

После запуска подписки с поддержкой расширяемости с помощью события истечения срока аренды и успешного выполнения рабочего процесса перейдите в vSphere Web Client и убедитесь, что виртуальная машина преобразована в шаблон.

## Устранение проблем с подпиской с поддержкой расширяемости

Устраните ошибки подписки с поддержкой расширяемости.

Обычно сбой подписки происходит из-за ошибок в рабочем процессе или сценарии действий расширяемости.

### Просмотр параметров темы и полезных данных

Для просмотра определенных параметров и полезных данных виртуальной машины на любом заданном этапе событий службы можно использовать сценарий параметров темы подписки на вывод дампа.

В первую очередь этот сценарий полезен для отладки и проверки доступных входных данных для рабочего процесса vRealize Orchestrator. Чтобы просмотреть все параметры виртуальной машины, используйте в рабочем процессе следующий сценарий:

```
function dumpProperties(props, lvl) {
    var keys = props.keys;
    var prefix = ""
    for (var i=0; i<lvl; i++){
        prefix = prefix + " ";
    }
    for (k in keys){
        var key = keys[k];
        var value = props.get(keys[k])
        if ("Properties" == System.getObjectType(value)){
            System.log(prefix + key + "[")
            dumpProperties(value, (lvl+2));
        }
    }
}
```

```

        System.log(prefix+ "]")
    } else{
        System.log( prefix + key + ":" + value)
    }
}

dumpProperties(inputProperties, 0)

customProps = inputProperties.get("customProperties")

```

## Журнал версий подписок

При сбросе подписки можно просмотреть журнал версий.

### Просмотр журнала версий подписок

На вкладке **Журнал версий** в редакторе подписок можно увидеть историю изменений подписки, включая автора и дату изменения. Можно также сравнить разные версии подписки, нажав кнопку **Сравнить с**.

Если подписка работает с ошибками или некорректно, журнал версий может помочь определить причину ошибки.

# Управление развертываниями и ресурсами в Cloud Assembly

# 7

Администраторы облачных систем и разработчики облачных шаблонов могут управлять ресурсами на вкладке «Ресурсы». Эти ресурсы могут быть развернуты вами, обнаружены для ваших облачных учетных записей, обнаружены и внедрены вами либо по иной причине доступны для управления с помощью Cloud Assembly.

В эту главу входят следующие разделы:

- [Управление развертываниями Cloud Assembly](#)
- [Управление ресурсами в Cloud Assembly](#)

## Управление развертываниями Cloud Assembly

Администраторы облачных систем Cloud Assembly и разработчики облачных шаблонов могут управлять развертываниями и связанными ресурсами на странице «Развертывания». Вы можете устранить неполадки при сбое процессов подготовки, внести изменения в ресурсы или уничтожить неиспользуемые развертывания.

Развертывания включают в себя развернутые облачные шаблоны и внедренные ресурсы. Ресурсы, которые созданы с помощью API-интерфейса инфраструктуры как услуги, также могут отображаться в виде развертываний.

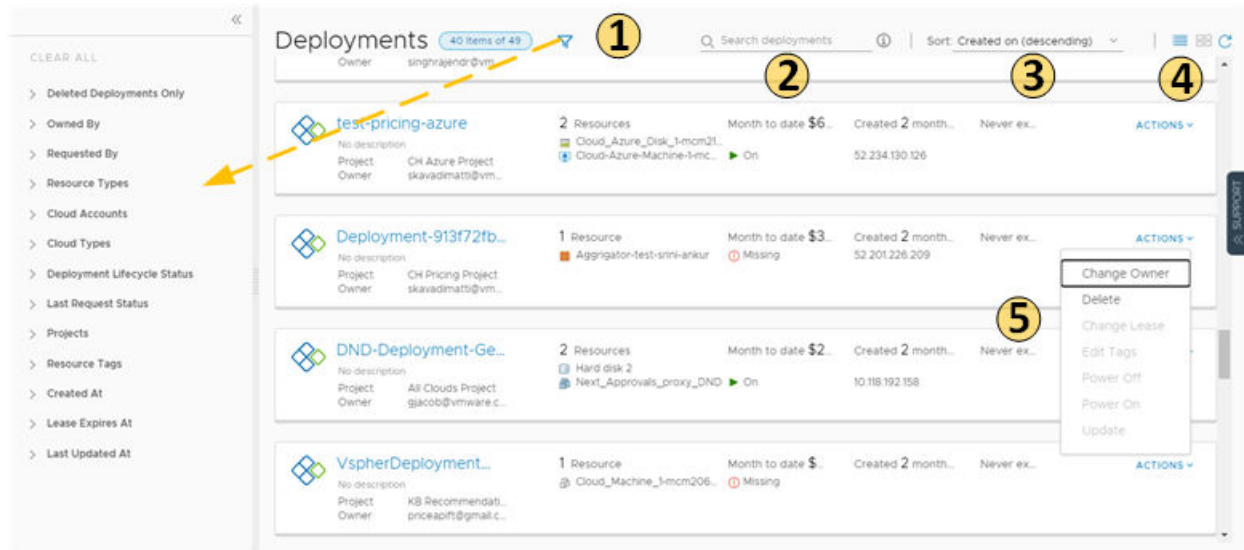
При управлении небольшим количеством развертываний карточки развертывания позволяют получить графическое представление для управления ими. При управлении большим количеством развертываний более надежное представление для управления можно получить с помощью списка развертываний и списка ресурсов.

Для просмотра развертываний выберите **Ресурсы > Развертывания**.

## Работа с карточками развертываний и списком развертываний

Развертывания можно искать и настраивать с помощью списка карточек. Можно выполнять фильтрацию или поиск конкретных развертываний, а затем выполнять действия с этими развертываниями.

Рис. 7-1. Страница развертываний, представление карточек



1. Отфильтровывать запросы на основе атрибутов.

Например, можно выполнять фильтрацию по владельцу, проектам, дате истечения срока аренды или другим параметрам. Также можно найти все развертывания для двух проектов с определенным тегом. При создании фильтра для определенных проектов и образца тегов результаты соответствуют следующим критериям: (Проект1 ИЛИ Проект2) И Тег1.

Значения, отображаемые на панели фильтра, зависят от текущих развертываний, для просмотра которых или управления которыми имеются разрешения.

Большинство фильтров, а также способ их применения легко понятны. Дополнительные сведения о некоторых таких фильтрах см. ниже.

2. Выполнять поиск развертываний по ключевым словам или инициатору запроса.
3. Сортировать список по времени или по имени.
4. Переключаться между представлениями карточек развертываний и списка развертываний.
5. Выполнять действия на уровне развертывания, в том числе удаление неиспользуемых развертываний, чтобы освободить ресурсы.

Также можно увидеть затраты на развертывание, даты истечения срока действия и состояние.

Между представлениями карточек и списка можно переключаться в верхнем правом углу страницы, справа от текстового поля «Сортировка». Представление списка можно использовать для управления большим количеством развертываний на меньшем числе страниц.

Рис. 7-2. Страница развертываний, представление списка

Actions	Address	Owner	Project	Status	Expires on	Price
<ul style="list-style-type: none"> <li>shared-ip-ranges-d...</li> <li>nikola-ipam-test-0...</li> <li>net.90</li> </ul>	192.168.0.6	bratanov@vmware.com	bratanov-ipa...	On	Never	
shared-ip-ranges-d...		bratanov@vmware.com	bratanov-ipa...	Never	Never	
test-depl		bratanov@vmware.com	bratanov-ipa...	Create — Failed	Never	
test2222		tdimitrova@vmware.com	vraikov	Never	Never	
afds4234		vraikov@vmware.com	vraikov	Never	Never	
4erasd		vraikov@vmware.com	vraikov	Never	Never	
grigor test 2412412		gganekov@vmware.com	vp-project	Never	Never	

## Работа с выбранными фильтрами для развертывания

Следующая таблица не представляет собой точный перечень возможных фильтров. Для большинства представленных вариантов объяснение не требуется. Однако для использования некоторых фильтров требуются дополнительные сведения.



Таблица 7-1. Сведения о выбранном фильтре

Имя фильтра	Описание
Только ресурсы, поддерживающие оптимизацию	Если вы интегрировали vRealize Operations Manager и используете эту интеграцию для выявления высвобождаемых ресурсов, можно включить фильтр, чтобы ограничить список соответствующих развертываний.
Состояние жизненного цикла развертывания	<p>Фильтры «Состояние жизненного цикла развертывания» и «Состояние последнего запроса» можно использовать по отдельности или совместно, в частности для управления большим количеством развертываний. Примеры приведены в конце раздела «Состояние последнего запроса» ниже.</p> <p>Фильтры «Состояние жизненного цикла развертывания» для текущего состояния развертывания на основании операций управления.</p> <p>Этот фильтр недоступен для удаленных развертываний.</p> <p>Значения, отображаемые на панели фильтра, зависят от текущего состояния развертываний, указанных в списке. Могут отображаться не все возможные значения. В списке ниже показаны все возможные значения. Действия по регулярному обновлению относятся к состоянию «Обновление».</p> <ul style="list-style-type: none"> <li>■ Создание — успешно</li> <li>■ Создание — выполняется</li> <li>■ Создание — сбой</li> <li>■ Обновление — успешно</li> <li>■ Обновление — выполняется</li> <li>■ Обновление — сбой</li> <li>■ Удаление — выполняется</li> <li>■ Удаление — сбой</li> </ul>
Фильтры «Состояние последнего запроса»	<p>Фильтры «Состояние последнего запроса» для последней операции или действия, которое выполнялось в развертывании.</p> <p>Этот фильтр недоступен для удаленных развертываний.</p> <p>Значения, отображаемые на панели фильтра, зависят от последних операций, которые выполнялись в развертываниях, указанных в списке. Могут отображаться не все возможные значения. В списке ниже показаны все возможные значения.</p> <ul style="list-style-type: none"> <li>■ Ожидание. Первый этап запроса, когда запрос на действие отправлен, но процесс развертывания еще не начался.</li> <li>■ Сбой. На одном из этапов процесса развертывания произошел сбой запроса.</li> <li>■ Отменено. Пользователь отменил запрос, когда процесс развертывания выполнялся, но еще не был завершен.</li> <li>■ Успешно. Запрос выполнен: развертывание создано (обновлено или удалено).</li> <li>■ Выполняется. Процесс развертывания выполняется. Дополнительные состояния развертывания, например «Инициализация» и «Завершение», отображаемые на вкладке «Журнал» развертывания, недоступны в качестве фильтров. Чтобы найти развертывания в этих состояниях, можно использовать фильтр «Выполняется».</li> </ul>

Таблица 7-1. Сведения о выбранном фильтре (продолжение)

Имя фильтра	Описание
	<ul style="list-style-type: none"> <li>■ Ожидается подтверждение. В результате запроса инициирована одна или несколько политик подтверждения. Процесс ожидает ответа на запрос на подтверждение.</li> <li>■ Запрос на подтверждение отклонен. Запрос отклонен утверждающими в инициированных политиках подтверждения. Выполнение запроса завершается.</li> </ul> <p>В следующих примерах показано, как использовать фильтры «Состояние жизненного цикла развертывания» и «Состояние последнего запроса» по отдельности или вместе.</p> <ul style="list-style-type: none"> <li>■ Чтобы найти все неудачные запросы на удаление, выберите <b>Удаление — сбой</b> в фильтре «Состояние жизненного цикла развертывания».</li> <li>■ Чтобы найти все запросы, ожидающие подтверждения, выберите <b>Ожидается подтверждение</b> в фильтре «Состояние последнего запроса».</li> <li>■ Чтобы найти запросы на удаление, в которых запрос на подтверждение все еще находится в состоянии ожидания, выберите <b>Удаление — выполняется</b> в фильтре «Состояние жизненного цикла развертывания» и <b>Ожидается подтверждение</b> в фильтре «Состояние последнего запроса».</li> </ul>

## Мониторинг развертываний в Cloud Assembly

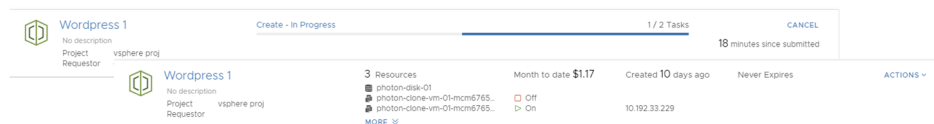
После развертывания облачного шаблона Cloud Assembly можно выполнить мониторинг запроса и убедиться, что необходимые ресурсы подготовлены и выполняются. Сначала можно проверить статус предоставления ресурсов, открыв карточку нужного развертывания. Затем можно ознакомиться с подробными сведениями о развертывании. Наконец, можно просмотреть и отфильтровать удаленные развертывания в течение 90 дней после их удаления.

### Процедура

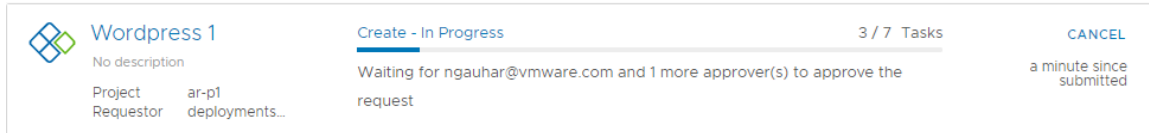
1. Выберите **Ресурсы > Развертывания** и найдите развертывание, используя фильтр и поиск, если необходимо.
2. Проверьте статус в карточке.

В процессе выполнения развертывания на панели хода выполнения отображается количество оставшихся задач. Если развертывание успешно завершено, в карточке отображаются основные

сведения о нем.

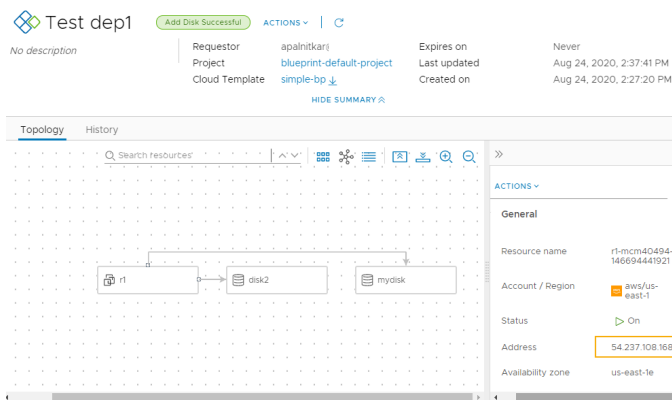


Если для запроса запущена политика утверждения, может отображаться запрос в состоянии «выполняется» с именем по крайней мере одного утверждающего. Политики утверждения определяет администратор в Service Broker. Утверждающие определяются в политике. Утверждающие утверждают запросы в Service Broker. Вы также можете столкнуться с утверждениями для действий по регулярному обслуживанию.



3. Чтобы определить, где развернуты ресурсы, щелкните имя развертывания и просмотрите сведения на странице «Топология».

Скорее всего, потребуется узнать IP-адрес основного компонента. Щелкая компоненты по очереди, можно просмотреть информацию, касающуюся компонента, выбранного в данный момент. В этом примере выделен IP-адрес.



Наличие внешней ссылки зависит от поставщика облачных служб. Если она имеется, для доступа к компоненту у вас должны быть учетные данные от этого поставщика.

#### Следующие шаги

- Можно внести изменения в развертывание. См. раздел [Управление жизненным циклом завершеного развертывания Cloud Assembly](#).
- Если развертывание выполнено неудачно, см. раздел [Что можно сделать в случае неудачного развертывания Cloud Assembly](#).

## Что можно сделать в случае неудачного развертывания Cloud Assembly

Запрос на развертывание может завершиться сбоем по ряду причин. Это может быть вызвано сетевым трафиком, отсутствием ресурсов у целевого поставщика облачных служб или ошибочной спецификацией развертывания. Кроме того, развертывание может быть выполнено успешно, но при этом не работать. При помощи Cloud Assembly можно проанализировать развертывание, просмотреть сообщения об ошибках и определить, с чем связана ошибка: со средой, запрошенной спецификацией рабочей нагрузки или чем-то еще.

Этот рабочий процесс используется для начального анализа проблемы. В ходе его выполнения может выясниться, что сбой произошел из-за временной ошибки среды. Повторное развертывание элемента по запросу, после того как вы убедились, что условия предоставления улучшились, приводит к устранению таких проблем. В других случаях для анализа может потребоваться более подробное ознакомление со сведениями в других областях.

Как участник проекта вы можете просмотреть сведения о запросе в Cloud Assembly.

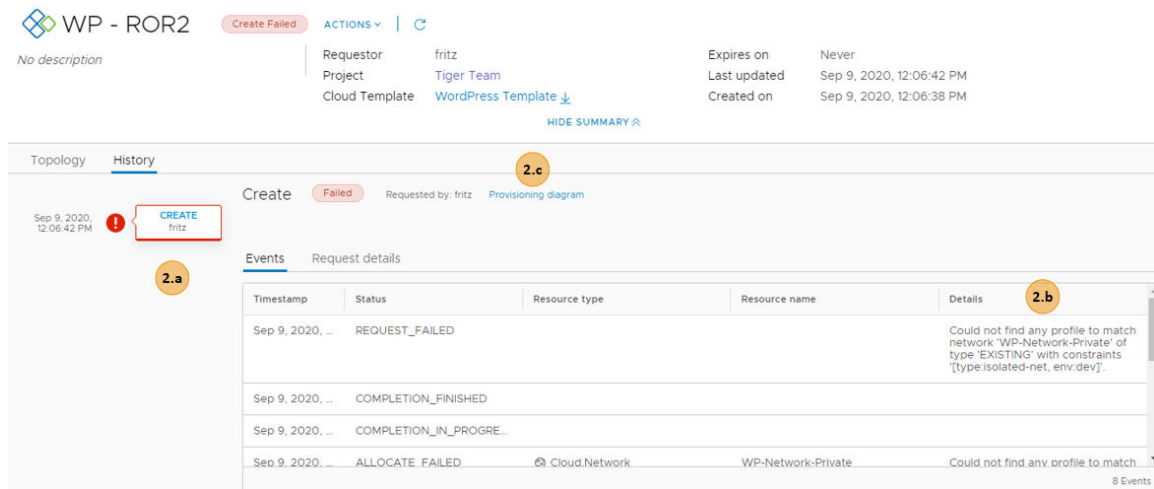
## Процедура

1. Чтобы определить, остался ли запрос невыполненным, выберите **Ресурсы > Развертывания** и найдите карточку развертывания.



В карточке указаны неудачные развертывания.

- а) Просмотрите сообщение об ошибке.
  - б) Для получения сведений о развертывании щелкните имя развертывания.
2. На странице сведений о развертывании перейдите на вкладку **Журнал**.



- а) Просмотрите дерево событий и определите, где произошла ошибка процесса предоставления. Это дерево удобно использовать, если вы изменяете развертывание и происходит ошибка.

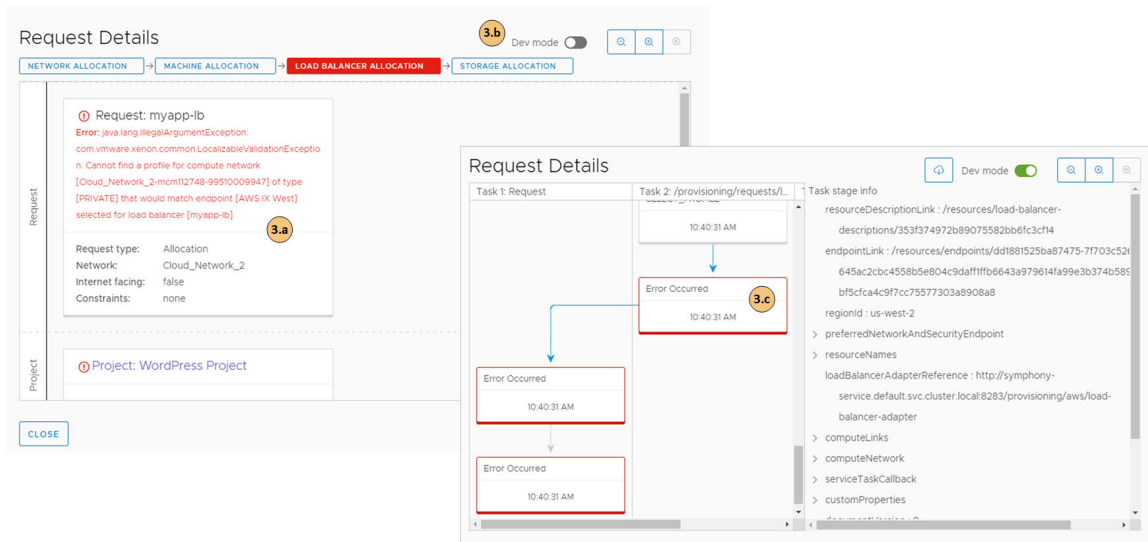
В дереве также отображаются выполняемые действия по развертыванию. Дерево можно использовать для устранения проблем, связанных с ошибками при внесении изменений.

- б) В разделе **Сведения** выводится более подробное сообщение об ошибке.
- в) Если запрошенный элемент — это облачный шаблон Cloud Assembly, нажмите ссылку справа от сообщения, чтобы открыть Cloud Assembly и посмотреть **Сведения о запросе**.

3. **Сведения о запросе** позволяют создать рабочий процесс предоставления компонентов с ошибками, чтобы понять проблему.

Журнал запросов хранится в течение 48 часов.

Просмотр и фильтрация удаленного журнала развертывания в течение 90 дней после удаления

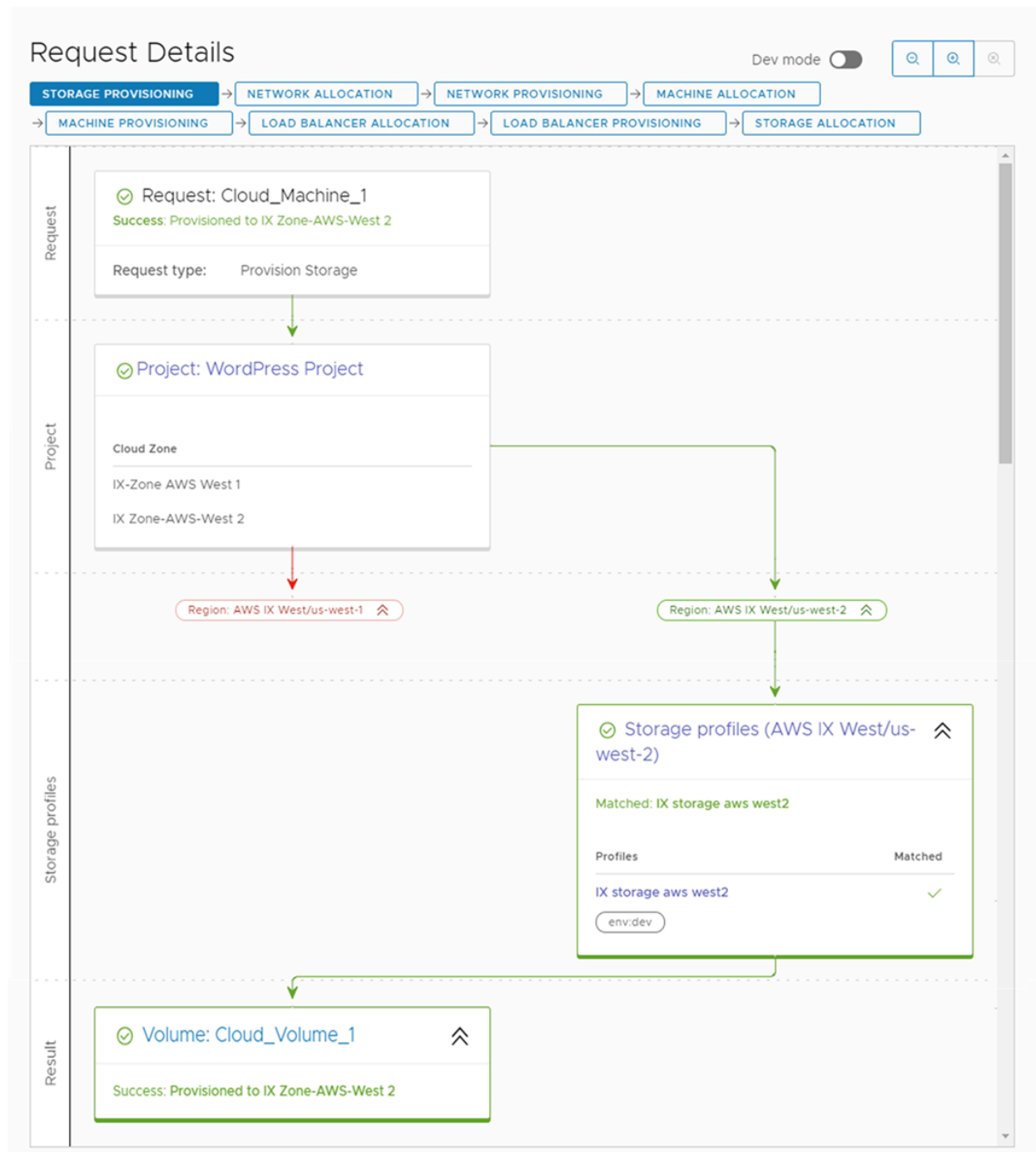


- Просмотрите сообщение об ошибке.
  - Для переключения между простым рабочим процессом предоставления и более подробным рабочим процессом можно включить **Режим разработки**.
  - Щелкните карточку, чтобы просмотреть сценарий развертывания.
4. Устраните ошибки и повторно разверните облачный шаблон.

Ошибки могут быть в структуре шаблона, или они могут быть связаны с настройкой инфраструктуры.

#### Следующие шаги

Если ошибки устранены и облачный шаблон развернут, то в сведениях о запросе можно посмотреть данные, аналогичные тем, что приведены в примере ниже. Для просмотра сведений о запросе выберите **Инфраструктура > Действие > Запросы**.



## Управление жизненным циклом завершено развертывания Cloud Assembly

После того, как развертывание подготовлено и запущено, им можно управлять, выполняя определенный набор доступных действий. К действиям по управлению жизненным циклом могут относиться операции включения и отключения, изменения размера и удаления развертывания. Используя действия, можно также управлять отдельными компонентами.

## Процедура

1. Выберите **Ресурсы > Развертывания** и найдите нужное развертывание.
2. Для доступа к сведениям о развертывании щелкните его имя.

Сведения о развертывании позволяют понять, как развернуты ресурсы и какие изменения внесены. Также можно получить сведения о ценах, текущей работоспособности развертывания и ресурсах, которые нужно изменить.

The screenshot displays the vRealize Automation Cloud Assembly interface for a deployment named 'sb-demo-03'. The interface is organized into several tabs: Topology, History, Price, Monitor, Alerts, and Optimize. The 'Create' tab shows a successful creation of a Cloud\_vSphere\_Machine\_1[0] and Cloud\_vSphere\_Machine\_1[1]. The 'Price' tab shows a price analysis for the month of March 2021, with a price of \$0.38. The 'Monitor' tab shows a graph of CPU usage for the deployment, with a peak of 0.123. The 'Alerts' tab shows a list of alerts, including 'Definition\_Deployment\_VM' and 'AlertDefinition\_Deployment\_has\_cost'. The 'Optimize' tab shows a list of underutilized VMs, including 'Cloud\_vSphere\_Machine\_1-mcm306191-163093649552'.

**Deployment Details:**

- Health: Good
- Owner: sbhandari@vmware.com
- Requestor: sbhandari@vmware.com
- Project: demo-vcenter-project
- Cloud Template: sb-demo
- Expires on: Never
- Last updated: Mar 17, 2021, 11:31:09 AM
- Created on: Mar 2, 2021, 8:47:06 AM

**Price Analysis:**

Price month to date: \$0.38  
Last month: \$0.38

**Monitor:**

CPU: 4, Memory: 6144 MB, Storage: 1 GB, Guest OS: Red Hat Enterprise Linux 6 (32-bit)

**Alerts:**

Definition\_Deployment\_VM (Dismiss)  
Created: Mar 8, 2021, 3:20:27 AM | Updated: Mar 17, 2021, 11:41:17 AM  
Deployment: sb-demo-03

**Optimize:**

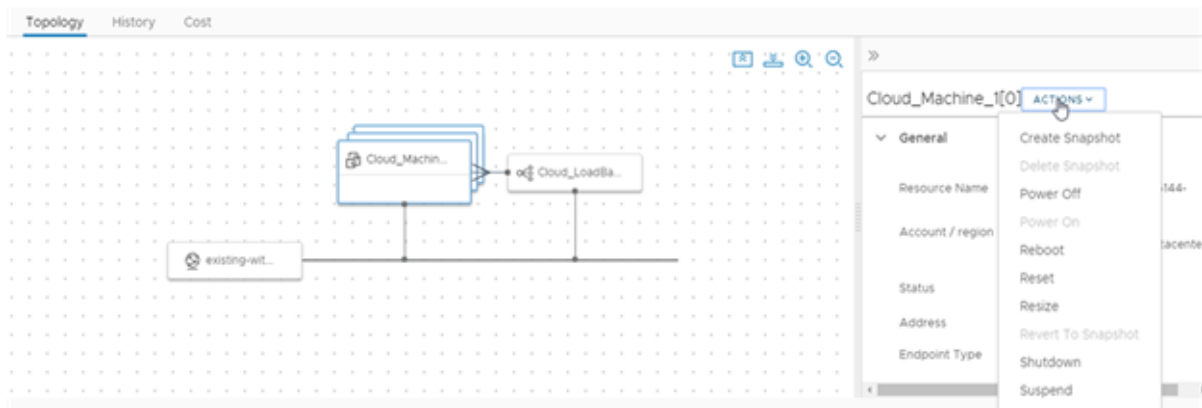
Underutilized VMs: 2 Idle VMs, 0 Powered Off VMs

Status	VM	Allocated CPU (vCPUs)	Allocated Memory (MB)	Allocated Storage (GB)
Idle since Mar 09, 2021	Cloud_vSphere_Machine_1-mcm306191-163093649552	4	6,144	1
Idle since Mar 09, 2021	Cloud_vSphere_Machine_1-mcm306192-163093649552	4	6,144	1

- Вкладка **Топология**. Структура и ресурсы развертывания представлены на вкладке "Топология".

- Вкладка **Журнал**. На вкладке "Журнал" содержатся сведения о всех событиях предоставления и любых событиях, связанных с действиями, которые выполняются после развертывания запрошенного элемента. Если в процессе предоставления возникли проблемы, сведения о событиях на вкладке "Журнал" помогут их устранить.
  - Вкладка **Ценообразование**. На карточке ценообразования содержатся сведения о стоимости развертывания для организации. Сведения о ценообразовании основаны на интеграциях vRealize Operations Manager или CloudHealth.
  - Вкладка **Мониторинг**. На вкладке «Мониторинг» содержатся сведения о работоспособности развертывания. Они основаны на данных, полученных из vRealize Operations Manager.
  - Вкладка **Оповещения**. На вкладке «Оповещения» отображаются активные оповещения о ресурсах развертывания. Оповещение можно отклонить или добавить в него заметки для справки. Оповещения основаны на данных, получаемых из службы vRealize Operations Manager.
  - Вкладка **Оптимизация**. На вкладке «Оптимизация» содержатся сведения об использовании развертывания, а также рекомендации по освобождению или иному изменению ресурсов для оптимизации их потребления. Сведения об оптимизации основаны на данных, полученных из vRealize Operations Manager.
3. Если текущая конфигурация развертывания связана со слишком большими издержками и требуется изменить размер какого-либо компонента, выберите нужный компонент на странице топологии, а затем нажмите **Действия > Изменить размер** на странице компонента.

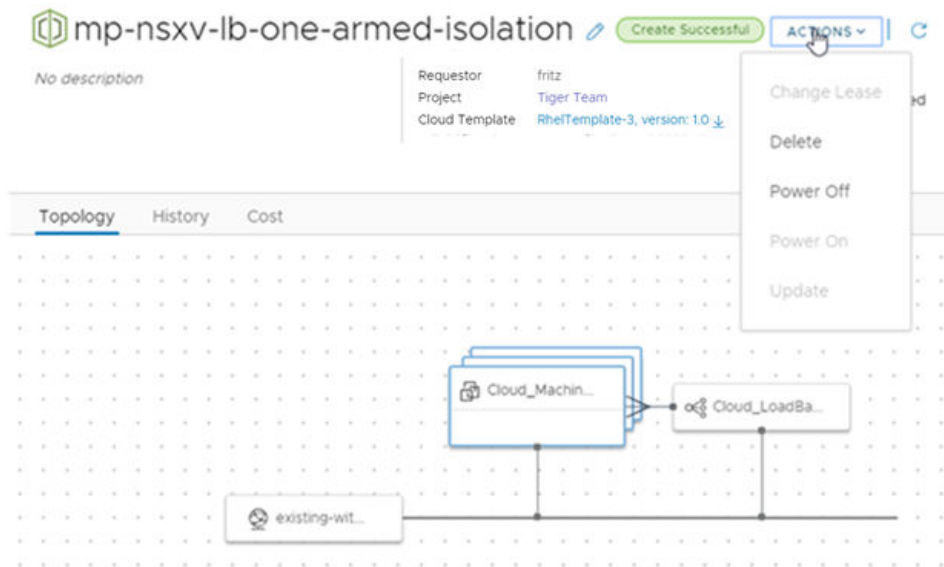
Доступные действия зависят от компонента, облачной учетной записи и разрешений.



4. В рамках жизненного цикла разработки наступает момент, когда одно из развертываний больше не нужно. Чтобы удалить это развертывание и освободить ресурсы, выберите пункт **Действия > Удалить**.

Доступные действия зависят от состояния развертывания.

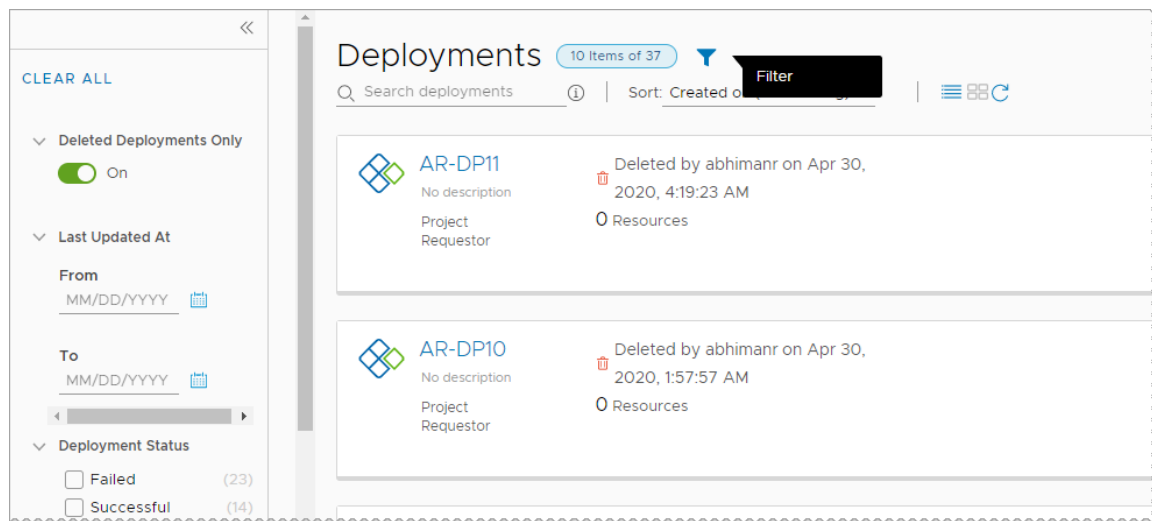




- Чтобы просмотреть удаленные развертывания, щелкните фильтр на странице **Развертывания**, а затем включите переключатель **Только удаленные развертывания**.

Теперь список содержит только развертывания, которые были удалены. Возможно, понадобится просмотреть журнал конкретного развертывания, например, чтобы получить имя удаленного компьютера.

Удаленные развертывания отображаются в течение 90 дней.



### Следующие шаги

Дополнительные сведения о вариантах действий см. в разделе [Какие действия можно выполнять в развертываниях Cloud Assembly](#).

## Какие действия можно выполнять в развертываниях Cloud Assembly

После развертывания облачных шаблонов можно выполнять действия в Cloud Assembly, позволяющие управлять ресурсами. Спектр доступных действий зависит от типа ресурсов и от того, какие действия поддерживаются определенной облачной учетной записью или платформой интеграции.

Доступные пользователю действия также зависят от полномочий, предоставленных ему администратором.

Администратор или администратор проекта может настраивать политики действий по регулярному обслуживанию в Service Broker. См. раздел [Предоставление пользователям права на политики действий по регулярному обслуживанию Service Broker](#).

Вы также можете увидеть действия, которые не включены в список. Скорее всего, это настраиваемые действия, добавленные администратором. Например, [Создание действия с ресурсом Cloud Assembly для виртуальной машины vMotion](#).

Таблица 7-2. Список возможных действий

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем	ресурса	
Добавление диска	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Добавление дополнительных дисков к существующим виртуальным машинам.</p> <p>При добавлении диска на компьютер Azure постоянный или непостоянный диск развертывается в группе ресурсов, в которой содержится компьютер.</p> <p>При добавлении диска на компьютер Azure можно также зашифровать новый диск, используя набор шифрования диска Azure, настроенный в профиле хранилища.</p> <p>На компьютер Azure с неуправляемым диском диск добавить нельзя.</p> <p>При добавлении диска на компьютеры vSphere можно выбрать развернутый контроллер SCSI, для которого указан порядок в облачном шаблоне. Кроме того, можно указать номер элемента для нового диска. Номер элемента без выбранного контроллера указать нельзя.</p> <p>Если вы не выбрали контроллер или не указали номер элемента, новый диск развертывается на первом доступном контроллере и ему назначается следующий доступный номер элемента в этом контроллере.</p> <p>При добавлении диска на компьютер vSphere в проекте с заданными ограничениями хранилища размер этого диска не должен превышать ограничения для хранилища.</p> <p>Если используется DRS хранилища VMware (SDRS) и в профиле хранилища настроен кластер, диски, содержащиеся в SDRS, можно добавлять на компьютеры vSphere.</p>
Применение конфигурации Salt	Компьютеры	<ul style="list-style-type: none"> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Установите служебный сервер Salt или обновите конфигурацию Salt на виртуальной машине.</p> <p>Параметр «Применить конфигурацию Salt» доступен, если настроена интеграция SaltStack Config.</p> <p><b>Примечание</b> Прежде чем использовать этот метод для установки служебного сервера Salt, следует рассмотреть более надежный вариант, при котором служебный сервер добавляется в облачный шаблон. Метод на основе шаблона включает в себя тип ресурса SaltStack Config в развертывании. Дополнительные сведения см. в разделе <a href="#">Как добавить ресурс SaltStack Config в шаблоны</a>.</p> <p>Чтобы применить конфигурацию, необходимо выбрать метод проверки подлинности. Параметр <b>Удаленный доступ с существующими учетными данными</b> использует учетные данные удаленного доступа, включенные в развертывание. Если изменить учетные данные на компьютере после развертывания, это может привести к сбою. Если новые учетные данные известны, используйте проверку подлинности с помощью пароля.</p>

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем		
Отмена	<ul style="list-style-type: none"> <li>■ Развертывания</li> <li>■ Различные типы ресурсов в развертываниях</li> </ul>	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p><b>Пароль и Закрытый ключ</b> используют имя пользователя и пароль или ключ для проверки учетных данных и подключения к виртуальной машине с помощью SSH.</p> <p>Если не задать значение для идентификатора главного и служебного сервера, Salt задаст их автоматически.</p> <p>Отмена развертывания либо действия по регулярному обслуживанию в развертывании или ресурсе во время обработки запроса.</p> <p>Запрос можно отменить на карточке развертывания или в сведениях о развертывании. После отмены запроса он отображается как невыполненный запрос на странице <b>Развертывания</b>. Чтобы освободить развернутые ресурсы и очистить список развертываний, используйте действие <b>Удалить</b>.</p> <p>Отмена запроса, который, по вашему мнению, выполняется слишком долго, — это один из способов управления временем развертывания. Однако более эффективный способ — установить в проектах параметр <b>Время ожидания запроса</b>. Время ожидания по умолчанию составляет два часа. Если развертывание рабочей нагрузки для проекта требует больше времени, это значение можно увеличить.</p>
Изменение аренды	Развертывания	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Изменение даты и времени истечения аренды.</p> <p>По истечении срока аренды развертывание уничтожается и ресурсы освобождаются.</p> <p>Политики аренды настраиваются в Service Broker.</p>
Изменить владельца	Развертывания	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Выбор пользователя в качестве владельца развертывания. Выбранный пользователь (как отдельный пользователь или участник группы) должен быть администратором или участником того же проекта, где развернут запрос.</p> <p>Если разработчик облачных шаблонов развертывает шаблон, он выступает одновременно и инициатором запроса, и владельцем. При этом инициатор запроса может сделать владельцем другого участника проекта.</p> <p>С помощью политик можно управлять возможностями владельца в отношении развертывания: ограничивать или расширять набор разрешений.</p>

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем	к ресурса	
Изменение проекта	Развертывания	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ NSX-T</li> <li>■ NSX-V</li> <li>■ VMware Cloud Director</li> <li>■ VMware Cloud Foundation</li> <li>■ VMware Cloud on AWS</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Действие изменения проекта используется для перемещения развертывания из одного проекта в другой.</p> <p>Действие изменения проекта доступно для развертываний с развернутыми ресурсами и развертываний с внедренными ресурсами. Это действие не поддерживается для развертываний, содержащих как внедренные, так и развернутые ресурсы. Это действие недоступно для перенесенных развертываний.</p> <p>Поддерживаемые ресурсы включают в себя следующие типы ресурсов и ограничения.</p> <ul style="list-style-type: none"> <li>■ Развертывания с развернутыми ресурсами могут содержать виртуальные машины, диски, подсистемы балансировки нагрузки, сети, группы безопасности, группы Azure, NAT и шлюзы.</li> <li>■ Развертывания с внедренными ресурсами могут содержать виртуальные машины, диски и сети.</li> <li>■ В случае добавления неподдерживаемого типа ресурса в любой тип развертывания (с развернутыми или внедренными ресурсами) действие изменения проекта выполнить нельзя. Например, при добавлении конфигурации Terraform в развертывание действие изменения проекта будет недоступно.</li> </ul> <p>Роли, рекомендации и ограничения для развертываний с развернутыми ресурсами</p> <ul style="list-style-type: none"> <li>■ Чтобы изменить проект развертывания с развернутыми ресурсами, пользователь-инициатор должен иметь следующую роль: <ul style="list-style-type: none"> <li>■ администратор облачных систем.</li> </ul> </li> <li>■ Проект можно изменить только в том случае, если целевой проект содержит все облачные зоны, в которых развернуты компьютеры и диски развертывания. Для перемещенного развертывания будут заданы настроенные ограничения целевого проекта, в том числе количество экземпляров, память, ЦП и хранилище. После перемещения ресурсы, используемые в исходном проекте, освобождаются.</li> <li>■ После перемещения развертывания в целевой проект к нему будут применены политики целевого проекта. Например, аренда, действия по регулярному обслуживанию, квота ресурсов и другие политики. Для перемещения развертывания срок аренды развертывания, заданный политикой аренды в целевом проекте, должен действовать в течение следующих 24 часов.</li> </ul>

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим типам ресурсов	Доступно для этих типов облачных систем	Источник ресурса	Описание
				<p>Роли, рекомендации и ограничения для развертываний с внедренными ресурсами</p> <ul style="list-style-type: none"> <li>■ Чтобы переместить развертывание с внедренными ресурсами, пользователь-инициатор должен иметь хотя бы одну из следующих ролей. <ul style="list-style-type: none"> <li>■ администратор облачных систем.</li> <li>■ Разрешение на управление развертываниями. Это разрешение можно определить как настраиваемую роль.</li> <li>■ Администратор целевого проекта.</li> <li>■ Участник целевого проекта и развертывания являются общедоступными для всех пользователей в этом проекте.</li> </ul> </li> <li>■ Несмотря на то что внедренные ресурсы можно переместить в проект, который не содержит те же самые облачные зоны, если в целевом проекте нет тех же облачных зон, то любые последующие действия по регулярному обслуживанию, связанные с применяемой облачной учетной записью или ресурсами региона, могут не работать.</li> </ul> <p>Общие рекомендации</p> <ul style="list-style-type: none"> <li>■ Если вы являетесь администратором, перемещающим развертывание, вы сможете переместить его в проект, участником которого владелец проекта не является, в результате чего он потеряет к нему доступ. Владельца можно добавить в целевой проект или переместить развертывание в проект, участником которого он является.</li> </ul>

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем	к ресурса	
Изменение групп безопасности	Компьютеры	■ VMware vSphere	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>В развертывании группы безопасности можно связать с сетями компьютеров и удалить такую связь. Действие «Изменение» применяется к существующим группам безопасности и группам безопасности по требованию для NSX-V и NSX-T. Это действие доступно только для отдельных компьютеров, а не кластеров компьютеров.</p> <p>Чтобы связать группу безопасности с сетью компьютеров, эта группа уже должна существовать в развертывании.</p> <p>Удаление связи для группы безопасности из всех сетей всех компьютеров в развертывании не приводит к удалению группы безопасности из развертывания.</p> <p>Эти изменения не влияют на группы безопасности, применяемые в рамках профилей сети.</p> <p>Это действие изменяет конфигурацию группы безопасности компьютера без повторного создания компьютера. Такое изменение не вызывает нарушение работы.</p> <ul style="list-style-type: none"> <li>■ Чтобы изменить конфигурацию группы безопасности компьютера, выберите компьютер на панели «Топология», затем меню <b>Действие</b> на правой панели и нажмите <b>Изменение групп безопасности</b>. Теперь можно добавить связь с группами безопасности с сетями компьютеров или удалить ее.</li> </ul>
Подключение к удаленной консоли	Компьютеры	■ VMware vSphere	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Обновленные</li> <li>■ Внедренные</li> </ul>	<p>Открытие удаленного сеанса на выбранном компьютере. Для успешного подключения ознакомьтесь со следующими требованиями.</p> <ul style="list-style-type: none"> <li>■ Потребитель развертывания должен проверить, что подготовленный компьютер включен.</li> </ul>

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем	ресурса	
Создание момента снимка диска	Компьютеры и диски	<ul style="list-style-type: none"> <li>■ Microsoft Azure</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> <li>■ Новые</li> </ul>	<p>Создание моментального снимка диска виртуальной машины или диска хранилища.</p> <ul style="list-style-type: none"> <li>■ Для компьютеров создаются моментальные снимки отдельных дисков компьютера, включая загрузочный диск, диски образов и диски хранилища.</li> <li>■ Для дисков хранилища создаются моментальные снимки независимых управляемых дисков; снимки неуправляемых дисков не создаются.</li> </ul> <p>Помимо имени, можно указать следующие данные моментального снимка.</p> <ul style="list-style-type: none"> <li>■ Инкрементный моментальный снимок. Поставьте флажок, чтобы создать снимок, содержащий не все данные, а только изменения с момента создания последнего снимка.</li> <li>■ Группа ресурсов. Введите имя целевой группы ресурсов, в которой нужно создать моментальный снимок. По умолчанию моментальный снимок создается в группе ресурсов, которая используется родительским диском.</li> <li>■ Идентификатор набора средств шифрования. Выберите ключ шифрования для моментального снимка. По умолчанию моментальный снимок шифруется с помощью ключа, который используется родительским диском.</li> <li>■ Теги. Введите любые теги, которые позволяют управлять моментальными снимками в Microsoft Azure.</li> </ul>
Создание моментальных снимков	Компьютеры	<ul style="list-style-type: none"> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> <li>■ Новые</li> </ul>	<p>Создание моментального снимка виртуальной машины. Если в vSphere разрешается создать только два моментальных снимка и они уже созданы, эта команда станет доступной только после удаления одного моментального снимка.</p>
Удалить	Развертывания	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> <li>■ Новые</li> </ul>	<p>Уничтожение развертывания.</p> <p>Все ресурсы будут удалены и освобождены.</p> <p>Если действие удаления завершается сбоем, это действие можно запустить в развертывании еще раз. Во время второй попытки можно выбрать вариант <b>Игнорировать ошибки удаления</b>. В случае выбора этого варианта развертывание будет удалено, но ресурсы могут быть не освобождены. Необходимо проверить системы, в которых развертывание было подготовлено, и убедиться, что все ресурсы удалены. Если это не так, удалите вручную остаточные ресурсы в этих системах.</p>

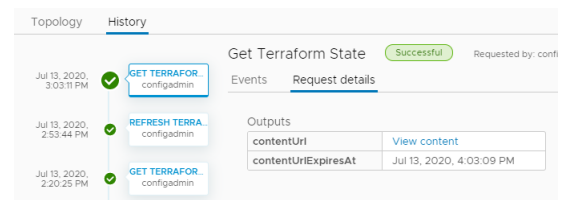


Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем		
	Шлюз NSX	■ NSX	■ Развернутые ■ Внедренные	Удалите правила переадресации портов NAT из шлюза NSX-T или NSX-V.
	Компьютеры и подсистемы балансировки и нагрузки	■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere ■ VMware NSX	■ Развернутые ■ Внедренные	Удалите компьютер или подсистему балансировки нагрузки из развертывания. В результате выполнения этого действия развертывание может стать непригодным для использования.
	Группы безопасности	■ NSX-T ■ NSX-V	■ Развернутые ■ Внедренные	Если группа безопасности не связана ни с одним компьютером в развертывании, она удаляется из развертывания. ■ Если группа безопасности является группой по требованию, она будет уничтожена в конечной точке. ■ Если к группе безопасности предоставлен общий доступ, произойдет сбой действия.
	Кластеры Tanzu Kubernetes	■ VMware vSphere	■ Развернутые ■ Внедренные	Удалите кластер Tanzu Kubernetes из развертывания.
Удаление момента снимка диска	Компьютеры и диски	■ Microsoft Azure	■ Развернутые ■ Внедренные	Удаление диска виртуальной машины Azure или моментального снимка управляемого диска. Это действие доступно при наличии хотя бы одного моментального снимка.
Удаление момента снимка	Компьютеры	■ VMware vSphere ■ Google Cloud Platform	■ Развернутые ■ Внедренные	Удаление моментального снимка виртуальной машины.
Отключить диагностику загрузки	Компьютеры	■ Microsoft Azure	■ Развернутые ■ Внедренные	Отключите функцию отладки виртуальной машины Azure. Отключение доступно только в том случае, если эта функция включена.

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем		
Редактирование тегов	Развертывания	<ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>Развернутые</li> <li>Внешние</li> </ul>	Добавление или изменение тегов ресурсов, которые применяются к отдельным ресурсам развертывания.
Включение диагностики загрузки	Компьютеры	<ul style="list-style-type: none"> <li>Microsoft Azure</li> </ul>	<ul style="list-style-type: none"> <li>Развернутые</li> <li>Внешние</li> </ul>	<p>Включите функцию отладки виртуальной машины Azure, чтобы диагностировать ошибки загрузки виртуальных машин. Сведения о диагностике загрузки доступны в консоли Azure.</p> <p>Включение доступно только в том случае, если функция выключена.</p>
Получение состояния Terraform	Конфигурация Terraform	<ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>Развернутые</li> <li>Внешние</li> </ul>	<p>Отображение файла состояния Terraform</p> <p>Чтобы просмотреть изменения, внесенные на компьютерах Terraform на облачных платформах, где они были развернуты, и обновить развертывание, сначала запустите действие «Обновление состояния Terraform», а затем выполните действие «Получение состояния Terraform».</p> <p>Когда файл отображается в диалоговом окне. Файл доступен в течение примерно 1 часа, прежде чем потребуется выполнить новое действие обновления. Если файл будет нужен позже, его можно скопировать.</p> <p>Файл также можно просмотреть на вкладке «Журнал» развертывания. Выберите событие «Получение состояния Terraform» на вкладке «События», а затем щелкните <b>Сведения о запросе</b>. Если срок действия файла не истек, щелкните <b>Просмотреть содержимое</b>. Если срок действия файла истек, выполните действия «Обновить» и «Получить» еще раз.</p>



Для ресурсов Terraform, встроенных в конфигурацию, можно запустить другое действие по регулярному обслуживанию. Доступные действия зависят от типа ресурса, облачной платформы, в которой они развернуты, и от того, имеете ли вы право запускать действия в соответствии с политикой регулярного обслуживания.

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типов ресурсов	Доступно для этих типов облачных систем	ресурса	
Выключение	Развертывания	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Обновленные</li> <li>■ Внедренные</li> </ul>	Выключите питание для развертывания после первой попытки завершения работы гостевых операционных систем. Если гибкое выключение питания завершается сбоем, жесткое выключение может быть по-прежнему выполнено.
	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Выключите питание компьютера после первой попытки завершения работы гостевых операционных систем. Если гибкое выключение питания завершается сбоем, жесткое выключение может быть по-прежнему выполнено.
Включение	Развертывания	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Включение развертывания. Если ресурсы были приостановлены, нормальное функционирование будет возобновлено с той точки, в которой они были приостановлены.
	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Обновленные</li> <li>■ Внедренные</li> </ul>	Включение компьютера. Если работа компьютера была приостановлена, нормальное функционирование будет возобновлено с той точки, в которой оно было приостановлено.
Перезагрузить	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Перезагрузка гостевой операционной системы на виртуальной машине. Чтобы использовать это действие, на компьютере (vSphere) необходимо установить VMware Tools.

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем		
Повторная настройка	Подсистемы балансировки и нагрузки	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware NSX</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Изменение размера подсистемы балансировки нагрузки и уровня ведения журнала.</p> <p>Кроме того, можно добавлять или удалять маршруты, а также изменять параметры протокола, порта, конфигурации работоспособности и пула участников.</p> <p>Для подсистем балансировки нагрузки NSX можно включить или отключить проверку работоспособности и изменить параметры работоспособности. Для NSX-T можно настроить активную или пассивную проверку. NSX-V не поддерживает пассивную проверку работоспособности.</p>
		<ul style="list-style-type: none"> <li>■ NSX-T</li> <li>■ NSX-V</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Добавьте, измените или удалите правила переадресации портов NAT из шлюза NSX-T или NSX-V.</p>
		<ul style="list-style-type: none"> <li>■ NSX-T</li> <li>■ NSX-V</li> <li>■ VMware Cloud</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Правила брандмауэра или ограничения можно добавлять, изменять или удалять в зависимости от типа группы безопасности (группа по требованию или существующая группа).</p> <ul style="list-style-type: none"> <li>■ Группа безопасности по требованию</li> </ul> <p>Добавление, изменение или удаление правил брандмауэра для групп безопасности по требованию в NSX-T и VMware Cloud.</p> <ul style="list-style-type: none"> <li>■ Чтобы добавить или удалить правило, выберите группу безопасности на панели топологии, перейдите в меню <b>Действие</b> на правой панели и выберите <b>Перенастройка</b>. Теперь можно добавлять, изменять или удалять правила.</li> <li>■ Существующая группа безопасности</li> </ul> <p>Добавление, изменение или удаление ограничений для существующих групп безопасности в NSX-V, NSX-T и VMware Cloud.</p> <ul style="list-style-type: none"> <li>■ Чтобы добавить или удалить ограничение, выберите группу безопасности на панели топологии, перейдите в меню <b>Действие</b> на правой панели и выберите <b>Перенастройка</b>. Теперь можно добавлять, изменять или удалять ограничения.</li> </ul>

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем		
Обновление состояния Terraform	Конфигурация Terraform	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Получение данных о последней итерации файла состояния Terraform</p> <p>Чтобы получить сведения об изменениях, внесенных на компьютерах Terraform на облачных платформах, где они были развернуты, и обновить развертывание, сначала запустите действие «Обновление состояния Terraform».</p> <p>Чтобы просмотреть файл, запустите действие <b>Получение состояния Terraform</b> в конфигурации.</p> <p>Для отслеживания процесса обновления используйте вкладку «Журнал» развертывания.</p>
Удаление диска	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Удаление дисков из существующих виртуальных машин.</p> <p>Если вы запускаете действие по регулярному обслуживанию в развертывании, которое состоит из компьютеров и дисков vSphere, то указанное количество дисков освобождается, так как оно относится к ограничениям хранилища проекта. Ограничения хранилища проекта не применяются к дополнительным дискам, добавленным после развертывания с помощью действия по регулярному обслуживанию.</p>
Сброс	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Компьютер принудительно перезагружается без завершения работы гостевых операционных систем.</p>
Изменить размер	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Увеличение или уменьшение ЦП и памяти виртуальной машины.</p>
Изменение размера загрузочного диска	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Увеличение или уменьшение размера загрузочного диска.</p> <p>Если вы запускаете действие по регулярному обслуживанию в развертывании, состоящем из компьютеров и дисков vSphere, и такое действие завершается сбоем с сообщением «Запрошенный объем хранилища превышает доступный объем сегмента хранилища», скорее всего, это связано с ограничениями для хранилища, установленными в шаблонах виртуальных машин vSphere и библиотеке содержимого, которые определены в проекте. Ограничения хранилища проекта не применяются к дополнительным дискам, добавленным после развертывания с помощью действия по регулярному обслуживанию.</p>

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем	ресурса	
Изменение размера диска	Дисковый накопитель	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Увеличение емкости дискового накопителя.</p> <p>Если вы запускаете действие по регулярному обслуживанию в развертывании, состоящем из компьютеров и дисков vSphere, и такое действие завершается сбоем с сообщением «Запрошенный объем хранилища превышает доступный объем сегмента хранилища», скорее всего, это связано с ограничениями для хранилища, установленными в шаблонах виртуальных машин vSphere и библиотеке содержимого, которые определены в проекте. Ограничения хранилища проекта не применяются к дополнительным дискам, добавленным после развертывания с помощью действия по регулярному обслуживанию.</p>
	Компьютеры	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Увеличение или уменьшение размера дисков, включенных в шаблон образа компьютера, и всех подключенных дисков.
Перезапуск	Компьютеры	<ul style="list-style-type: none"> <li>■ Microsoft Azure</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Завершение работы и перезапуск работающего компьютера.
Возврат к моментальному снимку	Компьютеры	<ul style="list-style-type: none"> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Восстановление состояния этого компьютера из предыдущего моментального снимка.</p> <p>Для использования этого действия требуется моментальный снимок.</p>
Запуск задачи Puppet	Управляемые ресурсы	<ul style="list-style-type: none"> <li>■ Puppet Enterprise</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	<p>Выполнение выбранной задачи на компьютерах в развертывании.</p> <p>Задачи определены в экземпляре Puppet. Необходимо иметь возможность определить задачу и указать входные параметры.</p>
Масштабирование рабочих узлов	Кластеры Tanzu Kubernetes	<ul style="list-style-type: none"> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Увеличьте или уменьшите количество виртуальных машин рабочих узлов Tanzu Kubernetes в развертывании.

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типов ресурсов	Доступно для этих типов облачных систем	ресурса	
Завершение работы	Компьютеры	<ul style="list-style-type: none"> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> </ul>	Завершение работы гостевой операционной системы и выключение компьютера. Чтобы использовать это действие, на компьютере необходимо установить VMware Tools.
Приостановка	Компьютеры	<ul style="list-style-type: none"> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Приостановка работы компьютера для того, чтобы его невозможно было использовать и он не потреблял никакие системные ресурсы, кроме используемых в настоящее время ресурсов хранилища.
Обновить	Развертывания	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Изменение развертывания на основе входных параметров. См. пример в разделе <a href="#">Перемещение развернутого компьютера в другую сеть</a> . Если развертывание основано на ресурсах vSphere, а компьютеры и диски содержат параметр «Количество», то при увеличении заданного количества могут применяться ограничения хранилища, которые заданы в проекте. Если действие завершается сбоем с сообщением «Запрошенный объем хранилища превышает доступный объем сегмента хранилища», скорее всего, это связано с ограничениями хранилища, установленными в шаблонах виртуальных машин vSphere, которые определены в проекте. Ограничения хранилища проекта не применяются к дополнительным дискам, добавленным после развертывания с помощью действия по регулярному обслуживанию.
Обновление тегов	Компьютеры и диски	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<ul style="list-style-type: none"> <li>■ Развернутые</li> <li>■ Внедренные</li> </ul>	Добавление, изменение или удаление тега, который применяется к отдельному ресурсу.

Таблица 7-2. Список возможных действий (продолжение)

Действие	Применяется к этим		Источники	Описание
	типам ресурсов	Доступно для этих типов облачных систем	к ресурса	
Обновление версии Tanzu	Кластеры Tanzu Kubernetes	■ VMware vSphere	■ Развернутые ■ Внедренные	Обновите текущую версию Kubernetes до более поздней версии.
Отмена регистрации	Компьютеры	■ Amazon Web Service ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere	■ Развернутые ■ Внедренные	Действие отмены регистрации доступно только для компьютеров внедренного развертывания. Компьютеры, регистрация которых отменяется, удаляются из развертывания вместе со всеми подключенными дисками. После удаления ресурсов рабочий процесс внедрения можно повторно запустить для компьютера с отмененной регистрацией. Возможно, будет необходимо снова внедрить ресурс, но уже в новый проект.  Если внести в компьютер какие-либо изменения, например добавить диск, перед отменой регистрации такого компьютера, действие отмены регистрации завершится сбоем.

## Управление ресурсами в Cloud Assembly

Администраторы облачных систем Cloud Assembly или разработчики облачных шаблонов могут управлять облачными ресурсами на вкладке «Ресурсы». Вкладка «Ресурсы» представляет собой центр ресурсов, где можно отслеживать ресурсы в различных облаках, вносить в них изменения и даже удалять и уничтожать их.

Поиск ресурсов и управление ими можно осуществлять с помощью различных представлений. Можно фильтровать списки, просматривать сведения о ресурсах, а затем выполнять действия с отдельными элементами. Доступные действия зависят от состояния ресурса и политик регулярного обслуживания.

Администраторы Cloud Assembly также могут просматривать обнаруженные компьютеры и управлять ими.

Для просмотра ресурсов выберите **Ресурсы > Ресурсы**.

### Работа со списками ресурсов

Списки ресурсов можно использовать для управления следующими типами ресурсов: компьютеры, тома хранилища, сети, подсистемы балансировки нагрузки и группы безопасности в развертываниях. В таком списке управление ресурсами можно выполнять в группах типов ресурсов, а не на уровне развертываний.

#### ■ Все ресурсы

Включают в себя все обнаруженные, развернутые, перенесенные и внедренные ресурсы, описанные в следующих разделах.



- Виртуальные машины

Отдельные виртуальные машины. Они могут быть частью крупных развертываний.

- Тома

Тома хранилища, которые были обнаружены или связаны с развертываниями.

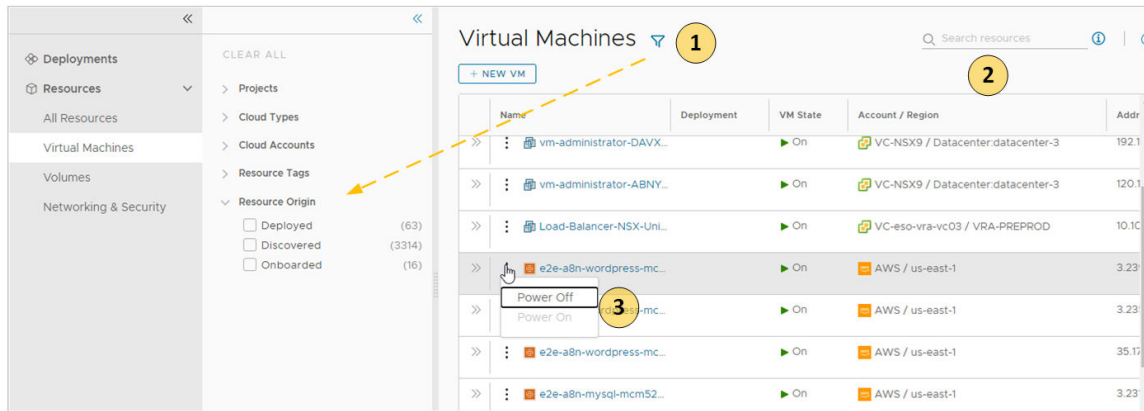
- Сети и безопасность

Включают в себя сети, подсистемы балансировки нагрузки и группы безопасности.

Как и в представлении списка развертываний, можно фильтровать список, выбрать тип ресурса, выполнить поиск, сортировку, а также действия.

Если щелкнуть имя ресурса, откроются сведения о нем.

Рис. 7-3. Список на странице ресурсов



1. Фильтруйте список на основе атрибутов ресурсов.

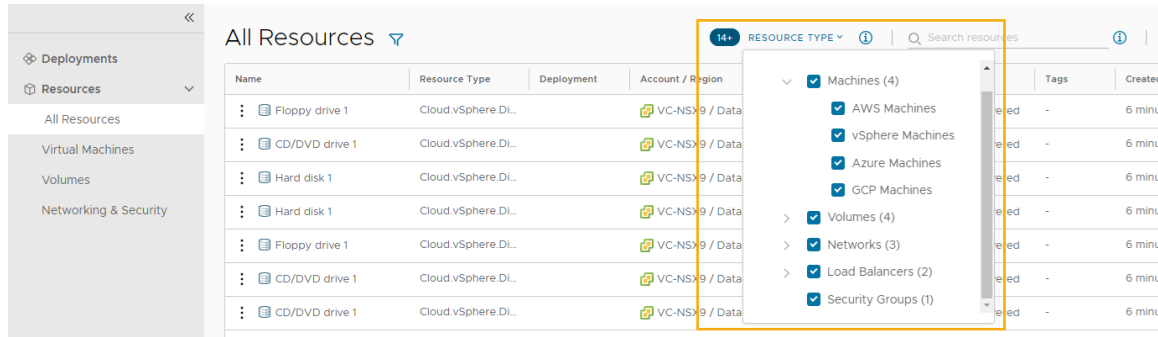
Например, можно выполнить фильтрацию по проектам, типам облачных систем, источнику или другим атрибутам.

2. Ищите ресурсы по имени, регионам учетной записи или другим значениям.

3. Выполняйте доступные действия по регулярному обслуживанию, предназначенные для типа и состояния ресурса.

Например, можно включить обнаруженный компьютер, если он выключен. Или можно изменить размер внедренного компьютера.

В дополнение к возможностям поиска и фильтрации на каждой странице на странице «Все ресурсы» есть средство выбора типа ресурса, в котором можно создать фильтр для всех ресурсов.



## Список управляемых ресурсов по источнику

На вкладке «Ресурсы» можно управлять следующими типами ресурсов.

Таблица 7-3. Источники ресурсов

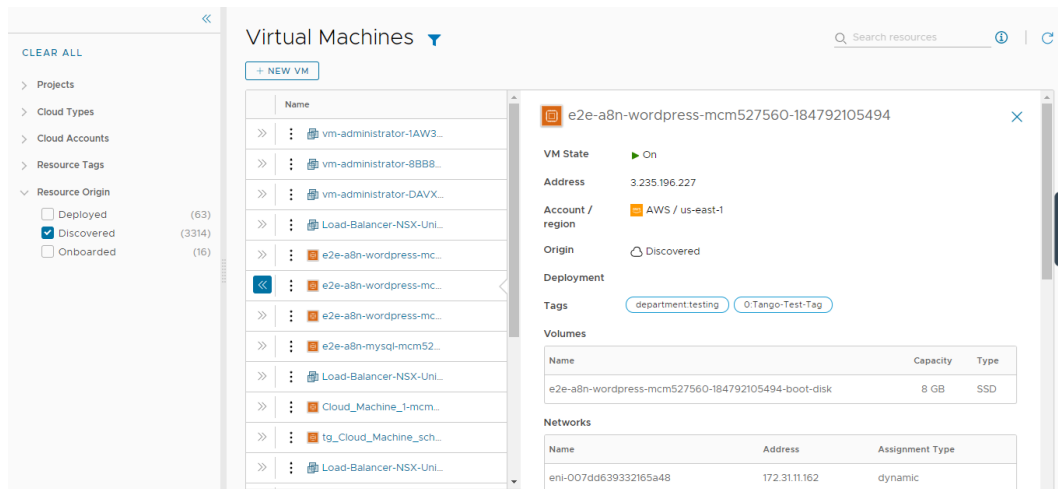
Управляемые ресурсы	Описание
Развернутые	<p>Развертывания — это полностью управляемые рабочие нагрузки, которые являются развернутыми облачными шаблонами или внедренными ресурсами. Ресурсы рабочей нагрузки могут включать в себя компьютеры, тома хранилища, сети, подсистемы балансировки нагрузки и группы безопасности.</p> <p>Развертываниями можно управлять в разделе «Развертывания» или разделе «Ресурсы».</p>
Обнаруженные	<p>Обнаруженные ресурсы — это компьютеры, тома хранилища, сети, подсистемы балансировки нагрузки и группы безопасности, обнаруженные для каждого добавленного региона облачной учетной записи.</p> <p>Только администраторы Cloud Assembly могут просматривать обнаруженные ресурсы в разделе «Ресурсы» и управлять ими.</p>
Перенесенные	<p>Перенесенные ресурсы — это развертывания версии 7.X, перенесенные в vRealize Automation. Перенесенные ресурсы могут включать в себя компьютеры, тома хранилища, сети, подсистемы балансировки нагрузки и группы безопасности. Управление перенесенными ресурсами аналогично управлению развертываниями.</p> <p>Перенесенными ресурсами можно управлять в разделе «Развертывания» или разделе «Ресурсы».</p>
Внедренные	<p>Внедренные ресурсы — это обнаруженные ресурсы, переведенные под более надежное управление vRealize Automation. Управление внедренными ресурсами аналогично управлению развертываниями.</p> <p>Внедренными ресурсами можно управлять в разделе «Развертывания» или разделе «Ресурсы».</p>

## Что такое представление сведений о ресурсе

Представление сведений о ресурсе можно использовать для более подробного изучения выбранного ресурса. В зависимости от ресурса сведения могут включать в себя сети, порты и другую информацию, собранную о компьютере. Глубина информации зависит от типа облачной учетной записи и источника.

Чтобы открыть панель сведений, щелкните имя ресурса или двойные стрелки.

Рис. 7-4. Панель сведений о ресурсах



## Действия по регулярному обслуживанию, которые можно выполнять на ресурсах

Доступные действия по регулярному обслуживанию зависят от источника, типа и состояния ресурса, а также облачной учетной записи.

Таблица 7-4. Список действий по источнику

Источник ресурса	Действия по регулярному обслуживанию
Развернутые	Действия, которые можно выполнять с ресурсами, зависят от типа и состояния ресурса, а также облачной учетной записи. См. подробный список в разделе <a href="#">Какие действия можно выполнять в развертываниях Cloud Assembly</a> .
Обнаруженные	<p>Доступные действия для обнаруженных ресурсов ограничены виртуальными машинами. В зависимости от состояния можно выполнять следующие действия.</p> <ul style="list-style-type: none"> <li>■ Выключение</li> <li>■ Включение</li> </ul> <p>Дополнительное действие с виртуальной машиной vSphere.</p> <ul style="list-style-type: none"> <li>■ Подключение к удаленной консоли</li> </ul>

Таблица 7-4. Список действий по источнику (продолжение)

Перенесенные	Для перенесенных ресурсов доступны те же возможности управления действиями по регулярному обслуживанию, что и для развертываний. Действия, которые можно выполнять с перенесенными ресурсами, зависят от типа и состояния ресурса, облачной учетной записи и политик регулярного обслуживания. См. подробный список в разделе <a href="#">Какие действия можно выполнять в развертываниях Cloud Assembly</a> .
Внедренные	Для внедренных ресурсов доступны те же возможности управления действиями по регулярному обслуживанию, что и для развертываний. Действия, которые можно выполнять с внедренными ресурсами, зависят от типа и состояния ресурса, а также облачной учетной записи. См. подробный список в разделе <a href="#">Какие действия можно выполнять в развертываниях Cloud Assembly</a> .

## Работа с отдельными ресурсами в Cloud Assembly

Администратор облачных систем или участник проекта с ресурсами для проекта может использовать раздел «Ресурсы» на вкладке «Ресурсы» для управления развернутыми, внедренными и перенесенными ресурсами в качестве отдельных ресурсов по типу ресурса.

Рабочий процесс, ориентированный на управление виртуальными машинами, содержит общее руководство по управлению жизненным циклом ресурсов, которое можно применить к другим типам ресурсов.

### Поиск ресурсов виртуальных машин

Развернутые, внедренные и перенесенные виртуальные машины доступны на страницах «Все ресурсы» и «Виртуальные машины» на вкладке «Ресурсы». В этом примере рассматриваются виртуальные машины, но к другим типам ресурсов можно применить тот же рабочий процесс.

1. Выберите **Ресурсы > Ресурсы > Виртуальные машины**.
2. Найдите виртуальную машину.

Для поиска конкретных ресурсов можно использовать фильтры.

Name	Deployment	VM State	Account / Region	Address	Project	Origin	Tags
vm-administrator-VLDX...		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
vm-administrator-N6CE...		On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...	192.167.211.142		Discovered	-
mcm-20211203215331-0...	Google Cloud Create VM_6f...	On	yingzhi-GCP / us-east1	34.74.168.22	Create VM Proj...	Deployed	-

### Просмотр сведений о виртуальной машине

Сведения о ресурсе позволяют быстро просматривать информацию о компьютере, включая сети, настраиваемые свойства и другую собранную информацию.

1. Найдите компьютер в списке виртуальных машин.
2. Щелкните имя ресурса или двойные стрелки в левом столбце таблицы.

В правой части списка откроется панель сведений.

The screenshot shows the 'Virtual Machines' section of the vRealize Automation Cloud Assembly interface. On the left, a list of VMs is displayed, including 'mcm-20211203215331-0...'. On the right, the details for the selected VM are shown.

**Virtual Machines**

**VM State** ▶ On

**Address** 34.74.168.22

**Account / region** yingzhi-GCP / us-east1

**Origin** Deployed

**Deployment** Google Cloud Create VM\_6f6d0315-ddc8-4f5d-9e1e-563c4f9a836d

**Tags**

**Volumes**

Name	Capacity	Type
create-vm-new-disk-1-524598563851	4 GB	HDD
mcm-20211203215331-000020	10 GB	HDD

**Networks**

Name	Address	Assignment Type
default	10.142.0.56	dynamic

**Custom Properties**

Name	Value
resourceId	3b43b1a6-105c-4d68-8562-f84d545d07a0
zone_overlapping_migrated	true
project	d952119a-7354-4dc2-afd5-718755917230
zone	us-east1-b
environmentName	Google Cloud Platform
providerId	1393403671676923083
id	/resources/compute/3b43b1a6-105c-4d68-8562-f84d545d07a0

3. Чтобы закрыть панель, щелкните двойные стрелки или имя ресурса.

## Выполнение действий по регулярному обслуживанию на виртуальной машине

Для управления ресурсами используются действия по регулярному обслуживанию. Доступные действия зависят от типа и состояния ресурса, а также применяемых политик действий по регулярному обслуживанию.

1. Найдите компьютер в списке виртуальных машин.
2. Щелкните вертикальное троеточие, чтобы просмотреть доступные действия.
3. Щелкните действие.

The screenshot shows the 'Virtual Machines' section of the vRealize Automation Cloud Assembly interface. A list of VMs is displayed, including 'mcm-20211203215331-0...'. A context menu is open for the selected VM, showing various actions.

**Virtual Machines**

Name	Deployment	VM State	Account / Region	Address	Project	Origin	Tags
vm-administrator-VLDX...		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
vm-administrator-N6CE...		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...	192.167.211.142		Discovered	-
mcm-20211203215331-0...	Google Cloud Create VM_6f...	▶ On	yingzhi-GCP / us-east1	34.74.168.22	Create VM Proj...	Deployed	-
vm-administrator-7COL...		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
vm-administrator-0628...		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
vm-administrator-BBJM...		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-
vm-administrator-7RQZ...		▶ On	https://cmbu-w01-vc08.eng.vmware.com / w01-vc08...			Discovered	-

**Context Menu Actions:**

- Add Disk
- Create Snapshot
- Delete
- Power Off
- Resize
- Resize Boot Disk
- Resize Disk
- Update Tags

## Работа с обнаруженными ресурсами в Cloud Assembly

Администратор Cloud Assembly может использовать раздел «Ресурсы» на вкладке «Ресурсы» для управления обнаруженными компьютерами. Только администраторы будут видеть обнаруженные ресурсы на различных страницах.

Этот рабочий процесс ориентирован на управление обнаруженными виртуальными машинами.

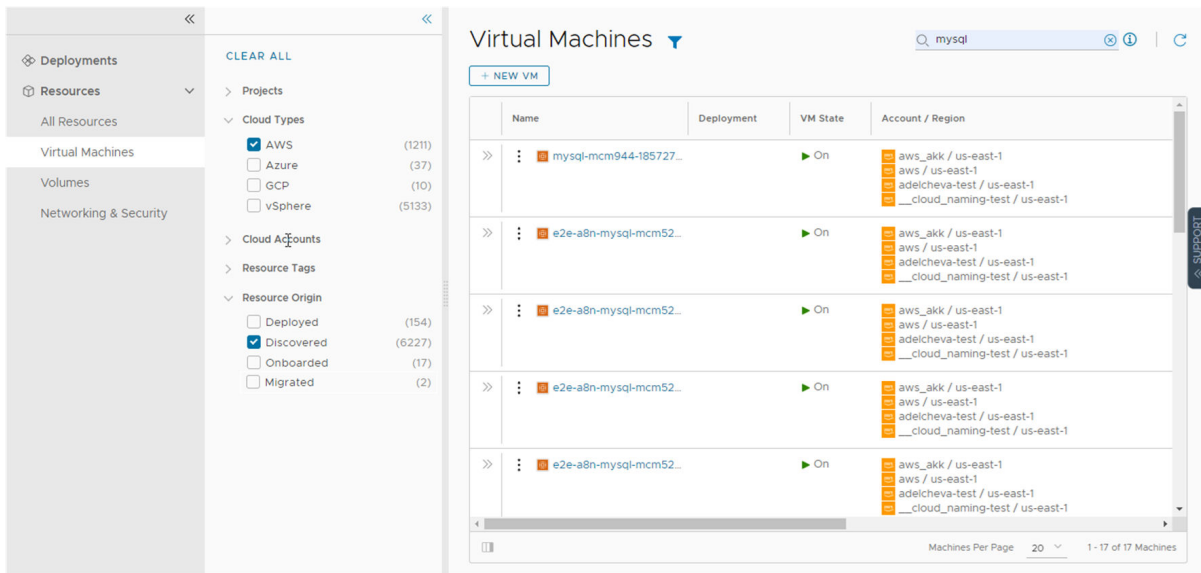
### Начало работы

- Добавьте облачную учетную запись для ресурсов, которые необходимо обнаружить. В этом рабочем процессе в качестве примера используется компьютер Amazon Web Services. Сведения о добавлении облачной учетной записи см. в разделе [Добавление учетных записей облачной службы в Cloud Assembly](#).

### Поиск обнаруженных виртуальных машин

Обнаруженные ресурсы собираются из региона облачной учетной записи и добавляются к ресурсам на вкладке «Ресурс». В этом примере основное внимание уделяется виртуальным машинам, но собираются и другие типы ресурсов, в том числе информация о хранилище и сети.

1. Выберите **Ресурсы > Ресурсы > Виртуальные машины**.



2. Чтобы найти виртуальные машины AWS, щелкните значок **Фильтр** рядом с названием страницы.
3. В списке фильтров разверните **Типы облачных систем** и выберите **AWS**.

Теперь список ограничен виртуальными машинами AWS. Вы могли видеть источники «Развернутые», «Обнаруженные» и другие.

4. В списке фильтров разверните **Источник ресурса** и выберите **Обнаруженные**.

Теперь этот список ограничен обнаруженными виртуальными машинами AWS.

5. Чтобы найти конкретный компьютер, можно использовать **Поиск ресурсов** для поиска по имени, IP-адресу, тегам или значениям.

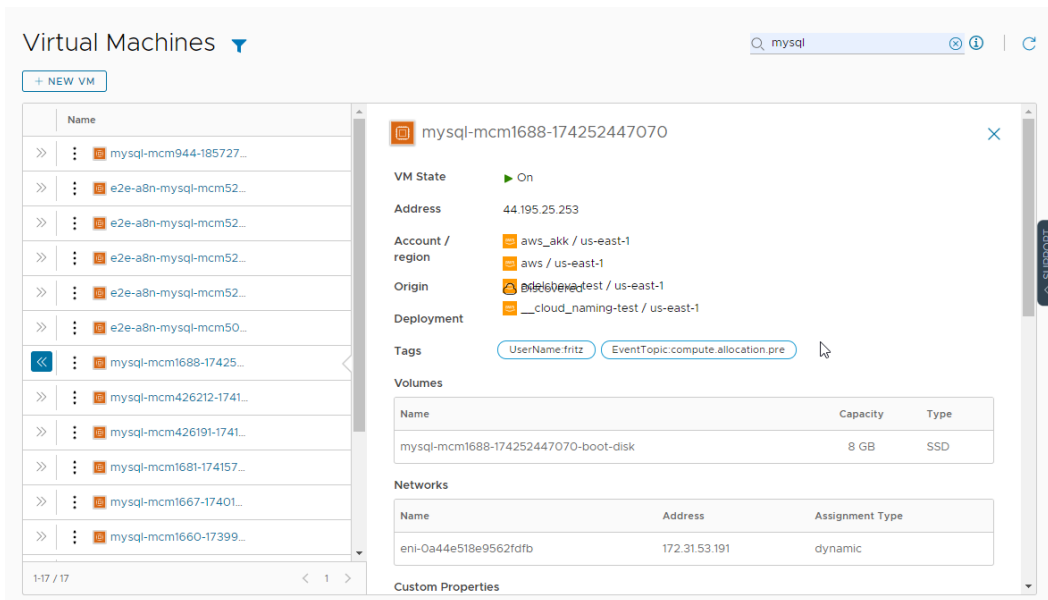
В этом примере ключевым словом для поиска является **mysql**.

## Просмотр сведений о виртуальной машине

Сведения о ресурсе включают в себя всю собранную о нем информацию. Ее можно использовать для понимания ресурса и любых связей с другими ресурсами.

1. Найдите виртуальную машину в списке виртуальных машин.
2. Чтобы просмотреть сведения о ресурсе, щелкните имя компьютера или двойные стрелки в левом столбце.

В правой части списка откроется панель сведений.



3. Просмотрите сведения, в том числе хранилище, сети, настраиваемые свойства и другую собранную информацию.
4. Чтобы закрыть панель, щелкните двойные стрелки или имя ресурса.

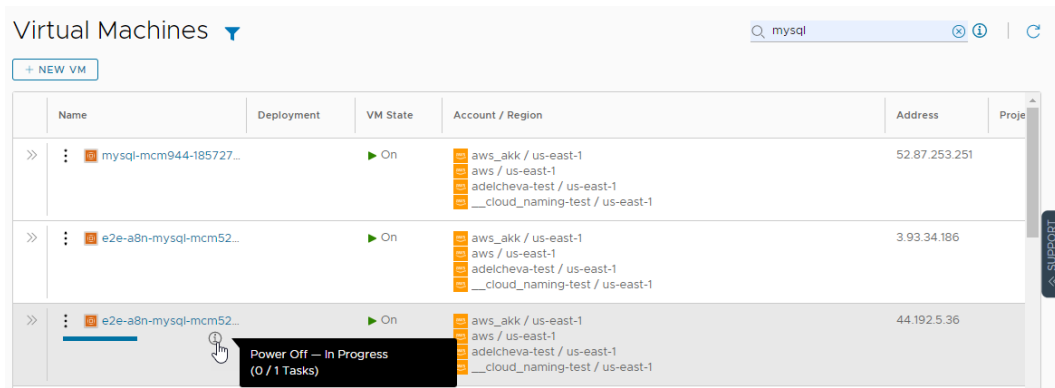
## Выполнение действий по регулярному обслуживанию на виртуальной машине

Действия по регулярному обслуживанию используются для управления ресурсами. К текущим действиям для обнаруженных виртуальных машин относятся включение и выключение. При управлении виртуальной машиной vSphere также можно использовать подключение к консоли удаленного доступа.

1. Найдите компьютер в списке виртуальных машин.
2. Щелкните вертикальное троеточие, чтобы просмотреть доступные действия.

Возможными действиями для виртуальной машины AWS являются «Выключить» и «Включить». Действие «Включить» неактивное, так как машина уже включена.

3. Щелкните **Выключить** и отправьте запрос.



После завершения процесса компьютер будет выключен. Теперь его можно снова включить.

## Что еще можно сделать с обнаруженной виртуальной машиной

Чтобы обеспечить полное управление обнаруженными ресурсами, их можно внедрить. См. раздел [Что такое планы внедрения в Cloud Assembly](#).