

Verwenden und Verwalten von vRealize Automation Cloud Assembly

Oktober 2022

vRealize Automation 8.0

Die aktuellste technische Dokumentation finden Sie auf der VMware-Website unter:

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

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

VMware Global, Inc.
Zweigniederlassung Deutschland
Willy-Brandt-Platz 2
81829 München
Germany
Tel.: +49 (0) 89 3706 17 000
Fax: +49 (0) 89 3706 17 333
www.vmware.com/de

Copyright © 2022 VMware, Inc. Alle Rechte vorbehalten. [Urheberrechts- und Markenhinweise](#).

Inhalt

1	Was ist vRealize Automation Cloud Assembly?	7
	Funktionsweise von vRealize Automation Cloud Assembly	8
2	Einrichten von vRealize Automation Cloud Assembly für Ihre Organisation	11
	Definition der vRealize Automation Cloud Assembly-Benutzerrollen	11
	Hinzufügen von Cloud-Konten	15
	Zum Arbeiten mit Cloud-Konten sind Anmeldedaten erforderlich	15
	Erstellen Sie ein Microsoft Azure-Cloud-Konto in vRealize Automation.	34
	Erstellen eines Amazon Web Services-Cloud-Kontos in vRealize Automation	35
	Erstellen eines Google Cloud Platform-Cloud-Kontos	36
	Erstellen eines vCenter-Cloud-Kontos	38
	Erstellen eines NSX-V-Cloud-Kontos	39
	Erstellen eines NSX-T-Cloud-Kontos	41
	Erstellen eines VMware Cloud on AWS-Cloud-Kontos	42
	Integrieren in andere Anwendungen	44
	Vorgehensweise zum Verwenden der GitLab- und GitHub-Integration	44
	Konfigurieren eines externen IPAM-Integrationspunkts	49
	Vorgehensweise zum Upgrade auf ein neueres IPAM-Integrationspaket	51
	Konfigurieren der My VMware-Integration in vRealize Automation Cloud Assembly	52
	Konfigurieren der vRealize Orchestrator-Integration in Cloud Assembly	53
	Vorgehensweise zum Arbeiten mit Kubernetes in vRealize Automation Cloud Assembly	56
	Definition der Konfigurationsverwaltung in vRealize Automation Cloud Assembly	65
	Vorgehensweise zum Erstellen einer Active Directory-Integration in vRealize Automation Cloud Assembly	72
	Definition von Onboarding-Plänen	73
	Integrieren ausgewählter Maschinen als Einzelbereitstellung	74
	Integrieren von durch Regeln gefilterten Maschinen als separate Bereitstellungen	78
	Erweiterte Konfiguration	83
	Vorgehensweise zum Integrieren eines Internet-Proxyservers	83
	Vorgehensweise zum Einrichten einer Windows-Vorlage mit cloud-init oder cloudbase-init	88
	Vorgehensweise zum Verwenden des IPAM-SDK zur Erstellung eines anbieterspezifischen, externen IPAM-Integrationspakets	88
3	vRealize Automation Cloud Assembly-Anwendungsfälle	90
	Anwendungsbeispiel: WordPress	90
	Erstellen der Infrastruktur	91
	Erstellen eines Projekts	99
	Erstellen und Erweitern eines Blueprints	100

VMware Cloud on AWS-Anwendungsbeispiel	118
Konfigurieren eines grundlegenden VMware Cloud on AWS-Workflows	119
Konfigurieren eines isolierten Netzwerks in VMware Cloud on AWS	133
Anwendungsfall für eine anbieterspezifische externe IPAM-Integration	138
Hinzufügen erforderlicher erweiterbarer Attribute in der Infoblox-Anwendung vor der Bereitstellung des Download-Pakets	140
Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets	141
Erstellen einer Ausführungsumgebung für einen IPAM-Integrationspunkt	143
Hinzufügen eines externen IPAM-Integrationspunkts	145
Konfigurieren eines Netzwerks und eines Netzwerkprofils zur Verwendung von IPAM-Anbieterwerten	148
Definieren und Bereitstellen eines Blueprints, der die Bereichszuweisung des IPAM-Anbieters verwendet	151
Verwenden Infoblox-spezifischer Eigenschaften für IPAM-Integrationen	154

4 Erstellen der Ressourceninfrastruktur 157

Vorgehensweise zum Hinzufügen von Cloud-Zonen	157
Weitere Informationen zu Cloud-Zonen	158
Vorgehensweise zum Hinzufügen von Konfigurationszuordnungen	160
Weitere Informationen zu Konfigurationszuordnungen	160
Vorgehensweise zum Hinzufügen von Image-Zuordnungen	161
Weitere Informationen zu Image-Zuordnungen	161
Vorgehensweise zum Hinzufügen von Netzwerkprofilen	165
Weitere Informationen zu Netzwerkprofilen	165
Arbeiten mit IP-Adressen in Netzwerken und Netzwerkprofilen	172
Verwenden von Netzwerken und Netzwerkprofilen	173
Verwenden der Einstellungen des Lastausgleichsdiensts	177
Vorgehensweise zum Hinzufügen von Speicherprofilen	179
Weitere Informationen zu Speicherprofilen	179
Vorgehensweise zum Verwenden von Tags	180
Erstellen einer Tagging-Strategie	182
Verwenden von Funktions-Tags in vRealize Automation Cloud Assembly	184
Verwenden von Einschränkungs-Tags in vRealize Automation Cloud Assembly	185
Standard-Tags	187
Wie vRealize Automation Cloud Assembly Tags verarbeitet	188
Vorgehensweise zum Einrichten einer einfachen Tagging-Struktur	189
Vorgehensweise zum Arbeiten mit Ressourcen	191
Computing-Ressourcen	191
Netzwerkressourcen	191
Sicherheitsressourcen	193
Speicherressourcen	195
Maschinenressourcen	195

Volume-Ressourcen	195
Weitere Informationen zu Ressourcen	196

5 Hinzufügen und Verwalten von Projekten 202

Vorgehensweise zum Hinzufügen eines Projekts für mein Entwicklungsteam	202
Weitere Informationen zu Projekten	204
Verwenden von Projekt-Tags und benutzerdefinierten Eigenschaften	204
Funktionsweise von Projekten zur Bereitstellungszeit	206

6 Entwerfen Ihrer Bereitstellungen 208

Vor dem Erstellen eines Blueprints	209
Möglichkeiten zum Erstellen von Blueprints	209
Vorgehensweise zum Erstellen eines einfachen Blueprints von Grund auf	211
Vorgehensweise zum Auswählen und Hinzufügen von Komponenten zu einem Blueprint	212
Vorgehensweise zum Verbinden von Blueprint-Ressourcen	212
Vorgehensweise zum Erstellen eines gültigen Blueprint-Codes	213
Vorgehensweise zum Verbessern eines einfachen Blueprints	215
Anpassen eines Blueprints mithilfe von Benutzereingaben	216
Vorgehensweise zum Festlegen der Reihenfolge der Komponentenbereitstellung	221
Vorgehensweise zum Verwenden von Ausdrücken zur flexibleren Gestaltung von Blueprint-Code	223
Vorgehensweise zum automatischen Initialisieren einer Maschine in einem Blueprint	232
Vorgehensweise zum Aktivieren des Remotezugriffs in Blueprints	241
Vorgehensweise zum Speichern verschiedener Versionen eines Blueprints	244
Anpassen der Namen von bereitgestellten Ressourcen	246
Eigenschaften der Ressource	249
Beispiele für Blueprint-Code	249
Beispiele für vSphere-Komponenten in -Blueprints	250
Überprüfbarer Blueprint	253
Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in Blueprints	260
Puppet-fähiger Blueprint mit Zugriff auf Benutzername und Kennwort	264
Vorgehensweise zum Verwenden des Marketplace	275
Vorgehensweise zum Erweitern und Automatisieren der Lebenszyklen von Anwendungen mit Erweiterbarkeit	276
Abonnements für Erweiterbarkeitsaktionen	276
Abonnements für Erweiterbarkeits-Workflows	297
Weitere Informationen zu Erweiterbarkeitsabonnements	306

7 Verwalten von Bereitstellungen 316

Vorgehensweise zum Überwachen aktiver Bereitstellungen	317
Vorgehensweise beim Fehlschlagen einer vRealize Automation Cloud Assembly-Bereitstellung	318

Vorgehensweise zum Verwalten des Lebenszyklus einer abgeschlossenen Bereitstellung	322
Welche Aktionen kann ich auf Bereitstellungen ausführen?	324

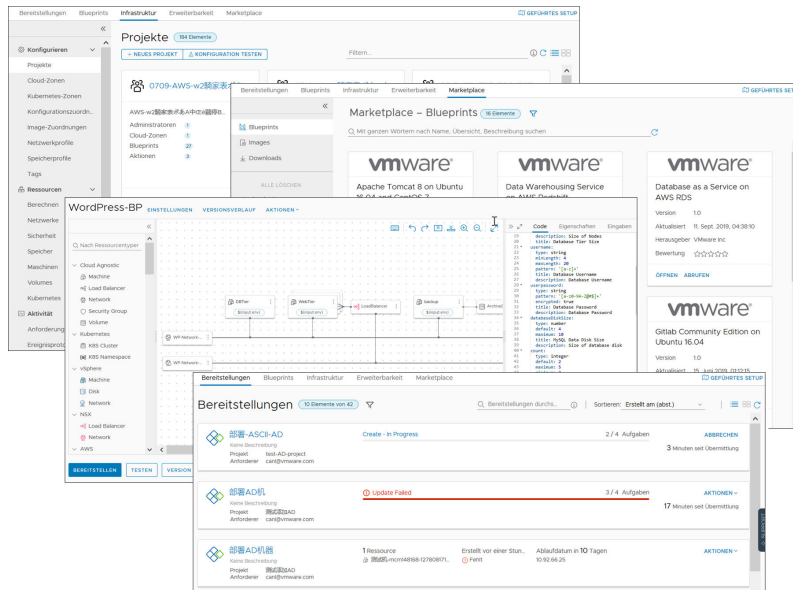
Was ist vRealize Automation Cloud Assembly?

1

Mit vRealize Automation Cloud Assembly stellen Sie eine Verbindung zu Ihren Public und Private Cloud-Anbietern her, damit Sie Maschinen, Anwendungen und Dienste bereitstellen können, die Sie für Ihre Ressourcen erstellen. Sie und ihre Teams entwickeln Blueprints-as-Code in einer Umgebung, die einen iterativen Workflow unterstützt, von der Entwicklung über die Tests bis zur Freigabe für die Produktionsumgebung. Bei der Bereitstellung können Sie eine Reihe von Cloud-Anbietern einsetzen. Der Dienst ist ein verwaltetes VMware-Framework auf SaaS- und NaaS-Basis.

Als Übersicht über vRealize Automation Cloud Assembly werden hier die folgenden allgemeinen Funktionen beschrieben.

- Auf der Registerkarte „Infrastruktur“ können Sie Ihre Cloud-Anbieter-Ressourcen und -Benutzer hinzufügen und organisieren. Diese Registerkarte enthält auch Informationen zu den bereitgestellten Blueprints.
- Die Registerkarte „Download-Center“ bietet VMware Solution Exchange-Blueprints und -Images, mit denen Sie Ihre Blueprint-Bibliothek erstellen und auf unterstützende OVAs oder OVFes zugreifen können.
- Die Registerkarte „Blueprints“ ist Ihre Entwicklungs-Startseite. Die Arbeitsfläche und den YAML-Editor verwenden Sie zur Entwicklung und Bereitstellung von Maschinen und Anwendungen.
- Auf der Registerkarte „Entwicklung“ wird der aktuelle Status Ihrer bereitgestellten Ressourcen angezeigt. Sie können auf Details und Verlauf zugreifen, die Sie zum Verwalten Ihrer Bereitstellungen verwenden.



Dieses Kapitel enthält die folgenden Themen:

- Funktionsweise von vRealize Automation Cloud Assembly

Funktionsweise von vRealize Automation Cloud Assembly

Bei vRealize Automation Cloud Assembly handelt es sich um einen Entwicklungs- und Bereitstellungsdienst für Blueprints. Sie und ihre Teams verwenden den Dienst, um Ihren Cloud-Anbieterressourcen Maschinen, Anwendungen und Dienste bereitzustellen.

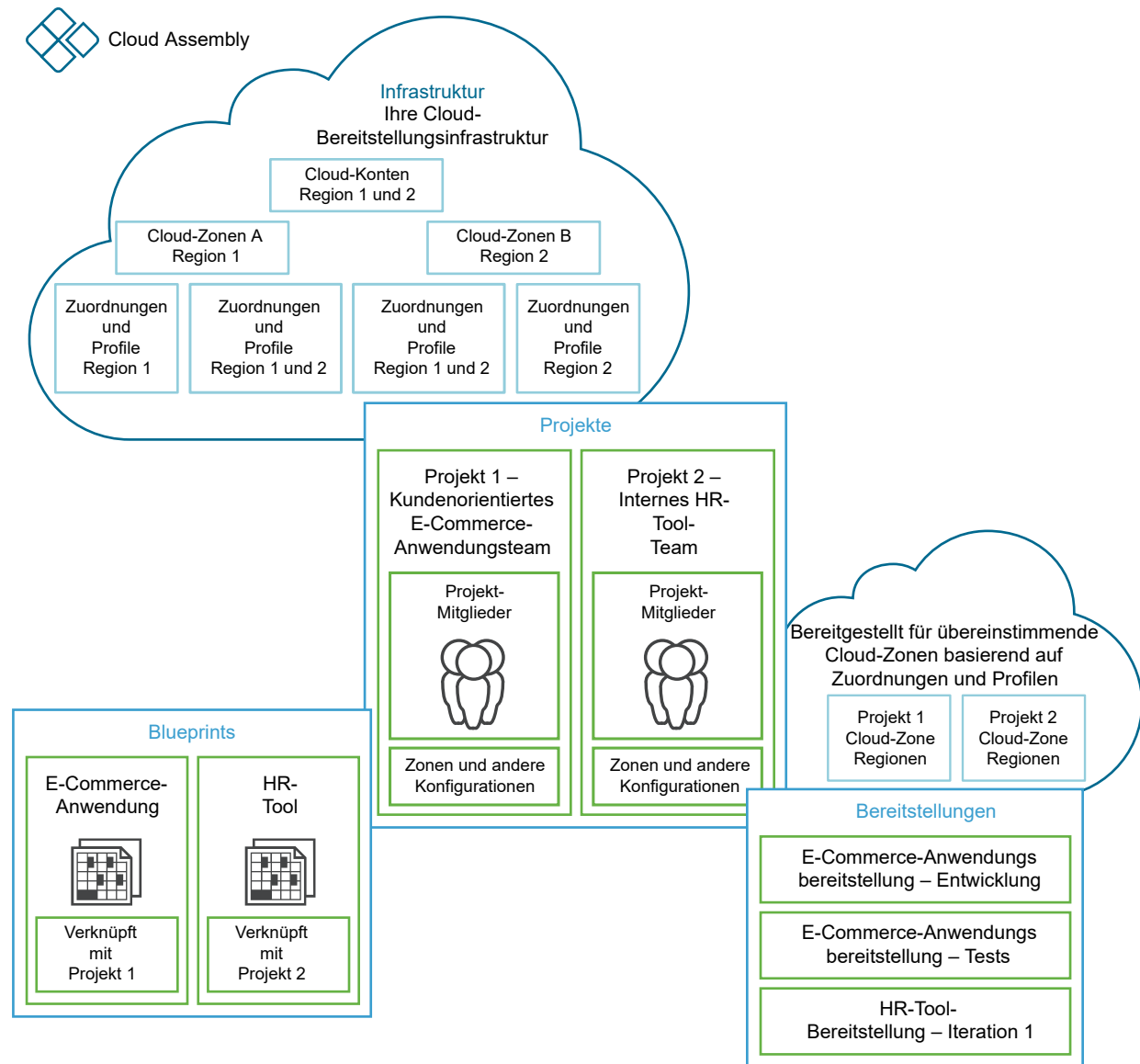
Als Cloud Assembly-Administrator, in der Regel als Cloud-Administrator bezeichnet, richten Sie die Bereitstellungsinfrastruktur ein und erstellen die Projekte, die Benutzer und Ressourcen gruppieren.

- Fügen Sie Ihre Cloud-Anbieterkonten hinzu. Weitere Informationen hierzu finden Sie unter [Hinzufügen von Cloud-Konten zu vRealize Automation Cloud Assembly](#).
- Legen Sie die Regionen und Datenspeicher fest, die als Cloud-Zonen für Bereitstellungen der Entwickler dienen sollen. Weitere Informationen hierzu finden Sie unter [Weitere Informationen zu vRealize Automation Cloud Assembly-Cloud-Zonen](#).
- Erstellen Sie Richtlinien, die die Cloud-Zonen definieren. Weitere Informationen hierzu finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#).
- Erstellen Sie Projekte, die die Entwickler mit den Cloud-Zonen gruppieren. Weitere Informationen hierzu finden Sie unter [Verwenden von vRealize Automation Cloud Assembly-Projekt-Tags und benutzerdefinierten Eigenschaften](#).

Als Blueprint-Entwickler sind Sie Mitglied eines oder mehrerer Projekte. Sie erstellen Blueprints und stellen sie in den Cloud-Zonen bereit, die mit einem Ihrer Projekte verknüpft sind.

- Entwickeln Sie Blueprints für Projekte mithilfe der Arbeitsfläche. Ihr Projektadministrator kann den Marketplace verwenden, um Blueprints und unterstützende Images aus VMware Solution Exchange herunterzuladen. Weitere Informationen finden Sie unter [Kapitel 6 Entwerfen Ihrer vRealize Automation Cloud Assembly-Bereitstellungen](#) und [Vorgehensweise zum Verwenden des vRealize Automation Cloud Assembly-Marketplace](#).
- Stellen Sie Ihre Blueprints basierend auf Richtlinien und Einschränkungen in den Cloud-Zonen eines Projekts bereit.
- Verwalten Sie Ihre Bereitstellungen und löschen Sie nicht verwendete Anwendungen. Weitere Informationen hierzu finden Sie unter [Kapitel 7 Verwalten von vRealize Automation Cloud Assembly-Bereitstellungen](#).

Willkommen bei vRealize Automation Cloud Assembly. Wenn Sie ein Beispiel für die Definition der Infrastruktur benötigen und anschließend einen Blueprint erstellen und bereitstellen möchten, finden Sie weitere Informationen unter [Anwendungsbeispiel: WordPress](#).



Einrichten von vRealize Automation Cloud Assembly für Ihre Organisation

2

Als Cloud Assembly-Administrator müssen Sie die Benutzerrollen verstehen und Verbindungen mit dem Cloud-Kontoanbieter und den Integrationsanwendungen einrichten.

Wenn Sie die Cloud-Konten und -Integrationen konfigurieren, richten Sie die Kommunikation zwischen Cloud Assembly und diesen Zielsystemen ein.

Dieses Kapitel enthält die folgenden Themen:

- [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#)
- [Hinzufügen von Cloud-Konten zu vRealize Automation Cloud Assembly](#)
- [Integrieren von vRealize Automation mit anderen Anwendungen](#)
- [Definition von Onboarding-Plänen in vRealize Automation Cloud Assembly](#)
- [Erweiterte Konfiguration für vRealize Automation Cloud Assembly-Umgebung](#)

Definition der vRealize Automation Cloud Assembly-Benutzerrollen

Über die Benutzerrollen wird festgelegt, was Benutzern in vRealize Automation Cloud Assembly angezeigt wird und welche Aufgaben sie ausführen können. Bestimmte Rollen werden auf der Organisationsebene definiert, andere wiederum sind vRealize Automation Cloud Assembly-spezifisch.

Benutzerrollen

Benutzerrollen werden von einem Organisationsbesitzer für die Organisation in der vRealize Automation-Konsole definiert. Es gibt zwei Arten von Rollen: Organisationsrollen und Dienstrollen.

Die Organisationsrollen sind global und gelten für alle Dienste in der Organisation. Die Rollen auf Organisationsebene sind die Rolle des Organisationsbesitzers oder des Organisationsmitglieds.

Weitere Informationen zu den Organisationsrollen finden Sie unter [Verwalten von vRealize Automation](#).

Die vRealize Automation Cloud Assembly-Dienstrollen, die dienstspezifische Berechtigungen sind, werden auch auf Organisationsebene in der Konsole zugewiesen.

Tabelle 2-1. Dienstrollen

Rolle	Beschreibung
Cloud Assembly-Administrator	Muss über Lese- und Schreibzugriff auf die gesamte Benutzeroberfläche und die API-Ressourcen verfügen. Dies ist die einzige Benutzerrolle, mit der angezeigt und durchgeführt werden kann, einschließlich Cloud-Konten hinzufügen, neue Projekte erstellen und einen Projektadministrator zuweisen.
Cloud Assembly-Benutzer	Jeder Benutzer, der nicht über die Cloud Assembly-Administratorrolle verfügt. In einem vRealize Automation Cloud Assembly-Projekt fügt der Administrator Benutzer zu Projekten als Projektmitglieder hinzu. Der Administrator kann auch einen Projektadministrator hinzufügen. Die Berechtigung für diese beiden Rollen wird nachfolgend definiert.

Projektrollen und -berechtigungen

Die Projektrollen des Projektadministrators und Projektmitglieds werden in vRealize Automation Cloud Assembly definiert und können zwischen Projekten variieren.

Beachten Sie in den folgenden Tabellen, in denen die Berechtigungen definiert sind, dass der Cloud-Administrator über sämtliche Berechtigungen für alle Bereiche der Benutzeroberfläche verfügt.

Projektadministratoren nutzen die vom Cloud-Administrator erstellte Infrastruktur, um sicherzustellen, dass die Projektmitglieder über die Ressourcen verfügen, die sie für ihre Entwicklungsarbeit benötigen.

Tabelle 2-2. Berechtigungen des Projektadministrators

Registerkarte	Knoten oder Bereich	Ansicht	Erstellen	Ändern/Löschen
Infrastruktur	Konfigurieren – Projekte	Ja (nur Ihre Projekte)	Nein	Ja (nur Ihre Projekte)
	Konfigurieren – Cloud-Zonen	Nein	Nein	Nein
	Konfigurieren – Konfigurationszuordnungen	Ja	Nein	Nein
	Konfigurieren – Image-Zuordnungen	Ja	Nein	Nein
	Konfigurieren – Netzwerkprofile	Ja	Nein	Nein
	Konfigurieren – Speicherprofile	Ja	Nein	Nein
	Konfigurieren – Tags	Ja	Nein	Nein

Tabelle 2-2. Berechtigungen des Projektadministrators (Fortsetzung)

Registerkarte	Knoten oder Bereich	Ansicht	Erstellen	Ändern/Löschen
	Ressourcen – Computing	Ja	Nein	Nein
	Ressourcen – Netzwerk	Ja	Nein	Nein
	Ressourcen – Speicher	Ja	Nein	Nein
	Ressourcen – Maschinen	Ja (nur Ihre Projekte)	Ja	Ja (nur Ihre Projekte)
	Ressourcen – Volumes			
	Aktivität – Anforderungen	Ja (nur Ihre Projekte)	Nicht verfügbar	Ja (nur Ihre Projekte)
	Aktivität – Ereignisse	Ja (nur Ihre Projekte)	Nicht verfügbar	Ja (nur Ihre Projekte)
	Verbindungen – Cloud-Konten	Nein	Nein	Nein
	Verbindungen – Integrationen		Nein	Nein
	Verbindungen – Cloud-Proxys		Nein	Nein
	Kosten – VMC-Bewertung	Ja	Nein	Nein
	Kosten – Private Clouds	Ja	Nein	Nein
	Onboarding		Nein	Nein
Blueprints	Blueprints	Ja (nur für Ihre Projekte)	Ja (nur für Ihre Projekte)	Ja (nur für Ihre Projekte)
Bereitstellungen	Bereitstellungen	Ja (nur für Ihre Projekte)	Nicht verfügbar	Ja (nur für Ihre Projekte)

Bei den Projektmitgliedern handelt es sich in der Regel um Entwickler, die Blueprints erstellen und bereitstellen.

Tabelle 2-3. Berechtigungen der Projektmitglieder

Registerkarte	Knoten oder Bereich	Ansicht	Erstellen	Ändern/Löschen
Infrastruktur	Konfigurieren – Projekte	Ja (nur die Projekte, in denen Sie Mitglied sind)	Nein	Nein
	Konfigurieren – Cloud-Zonen	Nein	Nein	Nein

Tabelle 2-3. Berechtigungen der Projektmitglieder (Fortsetzung)

Registerkarte	Knoten oder Bereich	Ansicht	Erstellen	Ändern/Löschen
	Konfigurieren – Konfigurationszuordnungen	Ja	Nein	Nein
	Konfigurieren – Image-Zuordnungen	Ja	Nein	Nein
	Konfigurieren – Netzwerkprofile	Ja	Nein	Nein
	Konfigurieren – Speicherprofile	Ja	Nein	Nein
	Konfigurieren – Tags	Ja	Nein	Nein
	Ressourcen – Computing	Ja	Nein	Nein
	Ressourcen – Netzwerk	Ja	Nein	Nein
	Ressourcen – Speicher	Ja	Nein	Nein
	Ressourcen – Maschinen	Ja (nur diejenigen, die Sie bereitgestellt haben)	Ja	Ja (nur diejenigen, die Sie bereitgestellt haben)
	Ressourcen – Volumes			
	Aktivität – Anforderungen	Ja (nur diejenigen, die Sie bereitgestellt haben)	Nicht verfügbar	Ja (nur diejenigen, die Sie bereitgestellt haben)
	Aktivität – Ereignisse	Ja (nur diejenigen, die Sie bereitgestellt haben)	Nicht verfügbar	Ja (nur diejenigen, die Sie bereitgestellt haben)
	Verbindungen – Cloud-Konten	Nein	Nein	Nein
	Verbindungen – Integrationen			
	Verbindungen – Cloud-Proxys			
	Kosten – VMC-Bewertung	Ja	Nein	Nein
	Kosten – Private Clouds	Ja	Nein	Nein
	Onboarding			

Tabelle 2-3. Berechtigungen der Projektmitglieder (Fortsetzung)

Registerkarte	Knoten oder Bereich	Ansicht	Erstellen	Ändern/Löschen
Blueprints	Blueprints	Ja (nur für Ihre Projekte)	Ja (nur für Ihre Projekte)	Ja (nur für Ihre Projekte)
Bereitstellungen	Bereitstellungen	Ja (nur für Ihre Bereitstellungen, es sei denn, die Projektbereitstellungen werden von allen Projektmitgliedern gemeinsam verwendet.)	Nicht verfügbar	Ja (nur für Ihre Bereitstellungen, es sei denn, Projektbereitstellungen werden von allen Projektmitgliedern gemeinsam genutzt und Sie sind berechtigt, die Tag-2-Aktionen auszuführen.)

Hinzufügen von Cloud-Konten zu vRealize Automation Cloud Assembly

Bei Cloud-Konten handelt es sich um die konfigurierten Berechtigungen, die von vRealize Automation Cloud Assembly zum Erfassen von Daten aus den Regionen oder Datencentern und zum Bereitstellen von Blueprints für diese Regionen verwendet werden.

Die erfassten Daten enthalten die Regionen, die Sie später mit Cloud-Zonen verknüpfen.

Wenn Sie Cloud-Zonen, Zuordnungen und Profile zu einem späteren Zeitpunkt konfigurieren, wählen Sie das Cloud-Konto aus, dem sie zugeordnet sind.

Als Cloud-Administrator erstellen Sie Cloud-Konten für die Projekte, in denen Teammitglieder arbeiten. Ressourceninformationen, wie z. B. Inhalte zu Netzwerk und Sicherheit, Berechnungen, Speicher und Tags, werden aus Ihren Cloud-Konten abgerufen.

Hinweis Wenn das Cloud-Konto über zugeordnete Maschinen verfügt, die bereits in der Region bereitgestellt wurden, können Sie diese Maschinen mithilfe eines Onboarding-Plans in die vRealize Automation Cloud Assembly-Verwaltung integrieren. Weitere Informationen hierzu finden Sie unter [Definition von Onboarding-Plänen in vRealize Automation Cloud Assembly](#).

Wenn Sie ein Cloud-Konto entfernen, das in einer Bereitstellung verwendet wird, werden Ressourcen, die Teil dieser Bereitstellung sind, nicht mehr verwaltet.

Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich

Zum Arbeiten mit und Konfigurieren von Cloud-Konten in vRealize Automation müssen Sie sicherstellen, dass Sie über die folgenden Anmeldedaten verfügen.

Erforderliche Anmeldedaten für Cloud-Konto

Aufgabe	Voraussetzungen
Registrieren und Anmelden bei vRealize Automation Cloud Assembly.	<p>Eine VMware-ID.</p> <ul style="list-style-type: none"> ■ Einrichten eines My VMware-Kontos unter Verwendung Ihrer geschäftlichen E-Mail-Adresse.
Herstellen einer Verbindung zu vRealize Automation-Diensten	<p>Offener HTTPS-Port 443 für ausgehenden Datenverkehr mit Zugriff über die Firewall auf:</p> <ul style="list-style-type: none"> ■ *.vmwareidentity.com ■ gaz.csp-vidm-prod.com ■ *.vmware.com <p>Weitere Informationen zu Ports und Protokollen finden Sie unter VMware-Ports und -Protokolle.</p> <p>Weitere Informationen zu den erforderlichen Ports und Protokollen finden Sie unter:</p> <ul style="list-style-type: none"> ■ Ports und Protokolle in der Hilfe zur <i>Installation</i> ■ Portanforderungen in der Hilfe zur <i>Referenzarchitektur</i>

Aufgabe	Voraussetzungen
Hinzufügen eines Amazon Web Services (AWS)-Cloud-Kontos	<p>Bereitstellen eines Hauptbenutzerkontos mit Lese- und Schreibberechtigungen. Das Benutzerkonto muss Mitglied der Zugriffsrichtlinie für Hauptbenutzer (PowerUserAccess) im AWS-IAM-System (Identity and Access Management) sein.</p> <ul style="list-style-type: none"> ■ 20-stellige Zugriffsschlüssel-ID und entsprechender geheimer Zugriffsschlüssel <p>Wenn Sie einen externen HTTP-Internet-Proxy verwenden, muss dieser für IPv4 konfiguriert sein.</p> <p>Möglicherweise sind für die aktionsbasierte Erweiterbarkeit (ABX) und die externe IPAM-Integration von vRealize Automation zusätzliche Berechtigungen erforderlich.</p> <p>Die folgenden AWS-Berechtigungen werden empfohlen, um automatische Skalierungsfunktionen zuzulassen:</p> <ul style="list-style-type: none"> ■ Aktionen für die automatische Skalierung: <ul style="list-style-type: none"> ■ autoscaling:DescribeAutoScalingInstances ■ autoscaling:AttachInstances ■ autoscaling>DeleteLaunchConfiguration ■ autoscaling:DescribeAutoScalingGroups ■ autoscaling>CreateAutoScalingGroup ■ autoscaling:UpdateAutoScalingGroup ■ autoscaling>DeleteAutoScalingGroup ■ autoscaling:DescribeLoadBalancers ■ Ressourcen für die automatische Skalierung: <ul style="list-style-type: none"> ■ * <p>Stellen Sie alle Berechtigungen für Ressourcen für die automatische Skalierung bereit.</p> <p>Die folgenden Berechtigungen sind erforderlich, damit die Funktionen des AWS Security Token Service (AWS STS) temporäre Anmeldedaten mit eingeschränkten Rechten für AWS-Identität und -Zugriff unterstützen können:</p> <ul style="list-style-type: none"> ■ AWS STS-Ressourcen: <ul style="list-style-type: none"> ■ * <p>Stellen Sie alle Berechtigungen für STS-Ressourcen bereit.</p> <p>Die folgenden AWS-Berechtigungen sind erforderlich, um EC2-Funktionen zuzulassen:</p> <ul style="list-style-type: none"> ■ EC2-Aktionen: <ul style="list-style-type: none"> ■ ec2:AttachVolume ■ ec2:AuthorizeSecurityGroupIngress ■ ec2>DeleteSubnet ■ ec2>DeleteSnapshot ■ ec2:DescribeInstances ■ ec2>DeleteTags ■ ec2:DescribeRegions ■ ec2:DescribeVolumesModifications ■ ec2>CreateVpc ■ ec2:DescribeSnapshots ■ ec2:DescribeInternetGateways ■ ec2>DeleteVolume ■ ec2:DescribeNetworkInterfaces ■ ec2:StartInstances ■ ec2:DescribeAvailabilityZones ■ ec2:CreateInternetGateway

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none"> ■ ec2:CreateSecurityGroup ■ ec2:DescribeVolumes ■ ec2:CreateSnapshot ■ ec2:ModifyInstanceAttribute ■ ec2:DescribeRouteTables ■ ec2:DescribeInstanceStatus ■ ec2:DetachVolume ■ ec2:RebootInstances ■ ec2:AuthorizeSecurityGroupEgress ■ ec2:ModifyVolume ■ ec2:TerminateInstances ■ ec2:DescribeSpotFleetRequestHistory ■ ec2:DescribeTags ■ ec2:CreateTags ■ ec2:RunInstances ■ ec2:DescribeNatGateways ■ ec2:StopInstances ■ ec2:DescribeSecurityGroups ■ ec2:CreateVolume ■ ec2:DescribeSpotFleetRequests ■ ec2:DescribeImages ■ ec2:DescribeVpcs ■ ec2>DeleteSecurityGroup ■ ec2>DeleteVpc ■ ec2:CreateSubnet ■ ec2:DescribeSubnets ■ ec2:RequestSpotFleet <hr/> <p>Hinweis Die SpotFleet-Anforderungsberechtigung ist für die aktionsbasierte Erweiterbarkeit (ABX) oder die externen IPAM-Integrationen von vRealize Automation nicht erforderlich.</p> <hr/> <ul style="list-style-type: none"> ■ EC2-Ressourcen: <ul style="list-style-type: none"> ■ * <p>Stellen Sie alle Berechtigungen für EC2-Ressourcen bereit.</p> <p>Die folgenden AWS-Berechtigungen sind erforderlich, um Funktionen für den elastischen Lastausgleich zuzulassen:</p> ■ Lastausgleichsdienstaktionen: <ul style="list-style-type: none"> ■ elasticloadbalancing>DeleteLoadBalancer ■ elasticloadbalancing:DescribeLoadBalancers ■ elasticloadbalancing:RemoveTags ■ elasticloadbalancing>CreateLoadBalancer ■ elasticloadbalancing:DescribeTags ■ elasticloadbalancing:ConfigureHealthCheck ■ elasticloadbalancing:AddTags ■ elasticloadbalancing>CreateTargetGroup ■ elasticloadbalancing>DeleteLoadBalancerListeners

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none"> ■ elasticloadbalancing:DeregisterInstancesFromLoadBalancer ■ elasticloadbalancing:RegisterInstancesWithLoadBalancer ■ elasticloadbalancing:CreateLoadBalancerListeners ■ Lastausgleichsdienstressourcen: <ul style="list-style-type: none"> ■ * <p>Stellen Sie alle Berechtigungen für Lastausgleichsdienstressourcen bereit.</p> <p>Die folgenden Berechtigungen für die Identitäts- und Zugriffsverwaltung (IAM) von AWS können aktiviert werden, sind aber nicht erforderlich:</p> <ul style="list-style-type: none"> ■ iam:SimulateCustomPolicy ■ iam:GetUser ■ iam:ListUserPolicies ■ iam:GetUserPolicy ■ iam:ListAttachedUserPolicies ■ iam:GetPolicyVersion ■ iam:ListGroupsForUser ■ iam:ListGroupPolicies ■ iam:GetGroupPolicy ■ iam:ListAttachedGroupPolicies ■ iam:ListPolicyVersions

Aufgabe	Voraussetzungen
Hinzufügen eines Microsoft Azure-Cloud-Kontos	<p>Konfigurieren Sie eine Instanz von Microsoft Azure und rufen Sie ein gültiges Microsoft Azure-Abonnement ab, dessen Abonnement-ID verwendet werden kann.</p> <p>Erstellen Sie eine Active Directory-Anwendung entsprechend der Beschreibung unter Vorgehensweise für die Verwendung des Portals zum Erstellen einer Azure AD-Anwendung und eines Dienstprinzips für den Zugriff auf Ressourcen in der Microsoft Azure-Produktdokumentation.</p> <p>Wenn Sie einen externen HTTP-Internet-Proxy verwenden, muss dieser für IPv4 konfiguriert sein.</p> <p>Notieren Sie sich die folgenden Informationen:</p> <ul style="list-style-type: none"> ■ Abonnement-ID <p>Ermöglicht den Zugriff auf Ihre Microsoft Azure-Abonnements.</p> ■ Mandanten-ID <p>Der Autorisierungs-Endpoint für die Active Directory-Anwendungen, die Sie in Ihrem Microsoft Azure-Konto erstellen.</p> ■ Client-Anwendungs-ID <p>Bietet Zugriff auf Microsoft Active Directory in Ihrem individuellen Microsoft Azure-Konto.</p> ■ Geheimer Schlüssel der Client-Anwendung <p>Der eindeutige geheime Schlüssel, der zur Kopplung mit Ihrer Client-Anwendungs-ID erzeugt wurde.</p> <p>Die folgenden Berechtigungen sind für das Erstellen und Validieren von Microsoft Azure-Cloud-Konten erforderlich:</p> <ul style="list-style-type: none"> ■ Microsoft Compute <ul style="list-style-type: none"> ■ Microsoft.Compute/virtualMachines/extensions/write ■ Microsoft.Compute/virtualMachines/extensions/read ■ Microsoft.Compute/virtualMachines/extensions/delete ■ Microsoft.Compute/virtualMachines/deallocate/action ■ Microsoft.Compute/virtualMachines/delete ■ Microsoft.Compute/virtualMachines/powerOff/action ■ Microsoft.Compute/virtualMachines/read ■ Microsoft.Compute/virtualMachines/restart/action ■ Microsoft.Compute/virtualMachines/start/action ■ Microsoft.Compute/virtualMachines/write ■ Microsoft.Compute/availabilitySets/write ■ Microsoft.Compute/availabilitySets/read ■ Microsoft.Compute/availabilitySets/delete ■ Microsoft.Compute/disks/delete ■ Microsoft.Compute/disks/read ■ Microsoft.Compute/disks/write ■ Microsoft Network <ul style="list-style-type: none"> ■ Microsoft.Network/loadBalancers/backendAddressPools/join/action ■ Microsoft.Network/loadBalancers/delete ■ Microsoft.Network/loadBalancers/read ■ Microsoft.Network/loadBalancers/write ■ Microsoft.Network/networkInterfaces/join/action ■ Microsoft.Network/networkInterfaces/read ■ Microsoft.Network/networkInterfaces/write ■ Microsoft.Network/networkInterfaces/delete

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none"> ■ Microsoft.Network/networkSecurityGroups/join/action ■ Microsoft.Network/networkSecurityGroups/read ■ Microsoft.Network/networkSecurityGroups/write ■ Microsoft.Network/networkSecurityGroups/delete ■ Microsoft.Network/publicIPAddresses/delete ■ Microsoft.Network/publicIPAddresses/join/action ■ Microsoft.Network/publicIPAddresses/read ■ Microsoft.Network/publicIPAddresses/write ■ Microsoft.Network/virtualNetworks/read ■ Microsoft.Network/virtualNetworks/subnets/delete ■ Microsoft.Network/virtualNetworks/subnets/join/action ■ Microsoft.Network/virtualNetworks/subnets/read ■ Microsoft.Network/virtualNetworks/subnets/write ■ Microsoft.Network/virtualNetworks/write ■ Microsoft Resources <ul style="list-style-type: none"> ■ Microsoft.Resources/subscriptions/resourcegroups/delete ■ Microsoft.Resources/subscriptions/resourcegroups/read ■ Microsoft.Resources/subscriptions/resourcegroups/write ■ Microsoft Storage <ul style="list-style-type: none"> ■ Microsoft.Storage/storageAccounts/delete ■ Microsoft.Storage/storageAccounts/listKeys/action ■ Microsoft.Storage/storageAccounts/read ■ Microsoft.Storage/storageAccounts/write ■ Microsoft Web <ul style="list-style-type: none"> ■ Microsoft.Web/sites/read ■ Microsoft.Web/sites/write ■ Microsoft.Web/sites/delete ■ Microsoft.Web/sites/config/read ■ Microsoft.Web/sites/config/write ■ Microsoft.Web/sites/config/list/action ■ Microsoft.Web/sites/publishxml/action ■ Microsoft.Web/serverfarms/write ■ Microsoft.Web/serverfarms/delete ■ Microsoft.Web/sites/hostruntime/functions/keys/read ■ Microsoft.Web/sites/hostruntime/host/read ■ Microsoft.web/sites/functions/masterkey/read <p>Wenn Sie Microsoft Azure mit aktionsbasierter Erweiterbarkeit verwenden, sind neben den minimalen Berechtigungen die folgenden Berechtigungen erforderlich:</p> <ul style="list-style-type: none"> ■ Microsoft.Web/sites/read ■ Microsoft.Web/sites/write ■ Microsoft.Web/sites/delete ■ Microsoft.Web/sites/config/read ■ Microsoft.Web/sites/config/write ■ Microsoft.Web/sites/config/list/action ■ Microsoft.Web/sites/publishxml/action

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none">■ Microsoft.Web/serverfarms/write■ Microsoft.Web/serverfarms/delete■ Microsoft.Web/sites/hostruntime/functions/keys/read■ Microsoft.Web/sites/hostruntime/host/read■ Microsoft.Web/sites/functions/masterkey/read <p>Wenn Sie Microsoft Azure mit aktionsbasierter Erweiterbarkeit mit Erweiterungen verwenden, sind die folgenden Berechtigungen ebenfalls erforderlich:</p> <ul style="list-style-type: none">■ Microsoft.Compute/virtualMachines/extensions/write■ Microsoft.Compute/virtualMachines/extensions/read■ Microsoft.Compute/virtualMachines/extensions/delete

Aufgabe	Voraussetzungen
Hinzufügen eines Google Cloud Platform (GCP)-Cloud-Kontos	<p>Das Cloud-Konto von Google Cloud Platform interagiert mit der Computing-Engine von Google Cloud Platform.</p> <p>Zum Erstellen und Validieren von Google Cloud Platform-Cloud-Konten sind die Anmeldedaten des Projektadministrators und des Projektbesitzers erforderlich.</p> <p>Wenn Sie einen externen HTTP-Internet-Proxy verwenden, muss dieser für IPv4 konfiguriert sein.</p> <p>Der Computing-Engine-Dienst muss aktiviert werden. Verwenden Sie beim Erzeugen des Cloud-Kontos in vRealize Automation das Dienstkonto, das beim Initialisieren der Computing-Engine erstellt wurde.</p> <p>Außerdem sind die folgenden Berechtigungen für die Computing-Engine erforderlich, je nachdem, welche Aktionen der Benutzer durchführen darf:</p> <ul style="list-style-type: none"> ■ <code>roles/compute.admin</code> <p>Bietet vollständige Kontrolle über alle Computing-Engine-Ressourcen.</p> ■ <code>roles/iam.serviceAccountUser</code> <p>Bietet Zugriff auf Benutzer, die VM-Instanzen verwalten, die für die Ausführung als Dienstkonto konfiguriert sind. Gewährt Zugriff auf die folgenden Ressourcen und Dienste:</p> <ul style="list-style-type: none"> ■ <code>compute.*</code> ■ <code>resourceManager.projects.get</code> ■ <code>resourceManager.projects.list</code> ■ <code>serviceUsage.quotas.get</code> ■ <code>serviceUsage.services.get</code> ■ <code>serviceUsage.services.list</code> ■ <code>roles/compute.imageUser</code> <p>Bietet ausschließlich die Berechtigung zum Auflisten und Lesen von Images, jedoch keine anderen Berechtigungen für das Image. Wenn die Rolle „compute.imageUser“ auf Projektebene zugewiesen wird, haben Benutzer die Möglichkeit, alle Images im Projekt aufzulisten. Außerdem können Benutzer Ressourcen (z. B. Instanzen und persistente Festplatten) auf der Basis von Images im Projekt erstellen.</p> <ul style="list-style-type: none"> ■ <code>compute.images.get</code> ■ <code>compute.images.getFromFamily</code> ■ <code>compute.images.list</code> ■ <code>compute.images.useReadOnly</code> ■ <code>resourceManager.projects.get</code> ■ <code>resourceManager.projects.list</code> ■ <code>serviceUsage.quotas.get</code> ■ <code>serviceUsage.services.get</code> ■ <code>serviceUsage.services.list</code> ■ <code>roles/compute.instanceAdmin</code> <p>Bietet Berechtigungen zum Erstellen, Ändern und Löschen von VM-Instanzen. Dazu gehören Berechtigungen zum Erstellen, Ändern und Löschen von Festplatten sowie zum Konfigurieren von abgeschirmten VMBETA-Einstellungen.</p> <p>Erteilen Sie diese Rolle Benutzern, die VM-Instanzen (aber keine Netzwerk- oder Sicherheitseinstellungen oder -instanzen, die als Dienstkonten ausgeführt werden) verwalten, für die Organisation, den Ordner oder das Projekt, welche die Instanzen enthalten, oder für die einzelnen Instanzen.</p> <p>Benutzer, die VM-Instanzen verwalten, welche für die Ausführung als Dienstkonto konfiguriert sind, benötigen zudem die Rolle <code>roles/iam.serviceAccountUser</code>.</p> <ul style="list-style-type: none"> ■ <code>compute.acceleratorTypes</code>

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none"> ■ compute.addresses.get ■ compute.addresses.list ■ compute.addresses.use ■ compute.autoscalers ■ compute.diskTypes ■ compute.disks.create ■ compute.disks.createSnapshot ■ compute.disks.delete ■ compute.disks.get ■ compute.disks.list ■ compute.disks.resize ■ compute.disks.setLabels ■ compute.disks.update ■ compute.disks.use ■ compute.disks.useReadOnly ■ compute.globalAddresses.get ■ compute.globalAddresses.list ■ compute.globalAddresses.use ■ compute.globalOperations.get ■ compute.globalOperations.list ■ compute.images.get ■ compute.images.getFromFamily ■ compute.images.list ■ compute.images.useReadOnly ■ compute.instanceGroupManagers ■ compute.instanceGroups ■ compute.instanceTemplates ■ compute.instances ■ compute.licenses.get ■ compute.licenses.list ■ compute.machineTypes ■ compute.networkEndpointGroups ■ compute.networks.get ■ compute.networks.list ■ compute.networks.use ■ compute.networks.useExternallp ■ compute.projects.get ■ compute.regionOperations.get ■ compute.regionOperations.list ■ compute.regions ■ compute.reservations.get ■ compute.reservations.list ■ compute.subnetworks.get ■ compute.subnetworks.list ■ compute.subnetworks.use

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none"> ■ compute.subnetworks.useExternalIp ■ compute.targetPools.get ■ compute.targetPools.list ■ compute.zoneOperations.get ■ compute.zoneOperations.list ■ compute.zones ■ resourceManager.projects.get ■ resourceManager.projects.list ■ serviceusage.quotas.get ■ serviceusage.services.get ■ serviceusage.services.list ■ roles/compute.instanceAdmin.v1 <p>Bietet vollständige Kontrolle über Instanzen, Instanzgruppen, Festplatten, Snapshots und Images der Computing-Engine. Bietet auch Lesezugriff auf alle Netzwerkressourcen der Computing-Engine.</p> <hr/> <p>Hinweis Wenn Sie einem Benutzer diese Rolle auf der Instanzebene zuweisen, kann dieser Benutzer keine neuen Instanzen erstellen.</p> <hr/> <ul style="list-style-type: none"> ■ compute.acceleratorTypes ■ compute.addresses.get ■ compute.addresses.list ■ compute.addresses.use ■ compute.autoscalers ■ compute.backendBuckets.get ■ compute.backendBuckets.list ■ compute.backendServices.get ■ compute.backendServices.list ■ compute.diskTypes ■ compute.disks ■ compute.firewalls.get ■ compute.firewalls.list ■ compute.forwardingRules.get ■ compute.forwardingRules.list ■ compute.globalAddresses.get ■ compute.globalAddresses.list ■ compute.globalAddresses.use ■ compute.globalForwardingRules.get ■ compute.globalForwardingRules.list ■ compute.globalOperations.get ■ compute.globalOperations.list ■ compute.healthChecks.get ■ compute.healthChecks.list ■ compute.httpHealthChecks.get ■ compute.httpHealthChecks.list ■ compute.httpsHealthChecks.get ■ compute.httpsHealthChecks.list

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none"> ■ compute.images ■ compute.instanceGroupManagers ■ compute.instanceGroups ■ compute.instanceTemplates ■ compute.instances ■ compute.interconnectAttachments.get ■ compute.interconnectAttachments.list ■ compute.interconnectLocations ■ compute.interconnects.get ■ compute.interconnects.list ■ compute.licenseCodes ■ compute.licenses ■ compute.machineTypes ■ compute.networkEndpointGroups ■ compute.networks.get ■ compute.networks.list ■ compute.networks.use ■ compute.networks.useExternallp ■ compute.projects.get ■ compute.projects.setCommonInstanceMetadata ■ compute.regionBackendServices.get ■ compute.regionBackendServices.list ■ compute.regionOperations.get ■ compute.regionOperations.list ■ compute.regions ■ compute.reservations.get ■ compute.reservations.list ■ compute.resourcePolicies ■ compute.routers.get ■ compute.routers.list ■ compute.routes.get ■ compute.routes.list ■ compute.snapshots ■ compute.sslCertificates.get ■ compute.sslCertificates.list ■ compute.sslPolicies.get ■ compute.sslPolicies.list ■ compute.sslPolicies.listAvailableFeatures ■ compute.subnetworks.get ■ compute.subnetworks.list ■ compute.subnetworks.use ■ compute.subnetworks.useExternallp ■ compute.targetHttpProxies.get ■ compute.targetHttpProxies.list ■ compute.targetHttpsProxies.get

Aufgabe	Voraussetzungen
	<ul style="list-style-type: none"> ■ compute.targetHttpsProxies.list ■ compute.targetInstances.get ■ compute.targetInstances.list ■ compute.targetPools.get ■ compute.targetPools.list ■ compute.targetSslProxies.get ■ compute.targetSslProxies.list ■ compute.targetTcpProxies.get ■ compute.targetTcpProxies.list ■ compute.targetVpnGateways.get ■ compute.targetVpnGateways.list ■ compute.urlMaps.get ■ compute.urlMaps.list ■ compute.vpnTunnels.get ■ compute.vpnTunnels.list ■ compute.zoneOperations.get ■ compute.zoneOperations.list ■ compute.zones ■ resourcemanager.projects.get ■ resourcemanager.projects.list ■ serviceusage.quotas.get ■ serviceusage.services.get ■ serviceusage.services.list
Hinzufügen eines NSX-T-Cloud-Kontos	<p>Bereitstellen eines Kontos mit den folgenden Lese- und Schreibberechtigungen:</p> <ul style="list-style-type: none"> ■ NSX-T-Enterprise-Administrator-Rolle und -Zugriffsanmeldedaten ■ IP-Adresse oder FQDN von NSX-T <p>Administratoren benötigen <i>außerdem</i> Zugriff auf den vCenter Server. Der folgende Abschnitt <i>Anforderungen des vSphere-Agenten für vCenter-basierte Cloud-Konten</i> auf dieser Seite enthält eine Beschreibung hierzu.</p>
Hinzufügen eines NSX-V-Cloud-Kontos	<p>Bereitstellen eines Kontos mit den folgenden Lese- und Schreibberechtigungen:</p> <ul style="list-style-type: none"> ■ NSX-V-Enterprise-Administrator-Rolle und -Zugriffsanmeldedaten ■ IP-Adresse oder FQDN von NSX-V <p>Administratoren benötigen <i>außerdem</i> Zugriff auf den vCenter Server. Der folgende Abschnitt <i>Anforderungen des vSphere-Agenten für vCenter-basierte Cloud-Konten</i> auf dieser Seite enthält eine Beschreibung hierzu.</p>

Aufgabe	Voraussetzungen
Hinzufügen eines vCenter-Cloud-Kontos	<p>Bereitstellen eines Kontos mit den folgenden Lese- und Schreibberechtigungen:</p> <ul style="list-style-type: none"> ■ IP-Adresse oder FQDN von vCenter <p>Administratoren benötigen <i>außerdem</i> Zugriff auf den vCenter Server. Der folgende Abschnitt <i>Anforderungen des vSphere-Agenten für vCenter-basierte Cloud-Konten</i> auf dieser Seite enthält eine Beschreibung hierzu.</p>
Hinzufügen eines VMC-Cloud-Kontos (VMware Cloud on AWS)	<p>Bereitstellen eines Kontos mit den folgenden Lese- und Schreibberechtigungen:</p> <ul style="list-style-type: none"> ■ Des Kontos „cloudadmin@vmc.local“ oder eines beliebigen Benutzerkontos in der Gruppe „CloudAdmin“ ■ NSX-Enterprise-Administrator-Rolle und -Zugriffsanmeldedaten ■ NSX-Cloud-Administratorzugriff auf die VMware Cloud on AWS-SDDC-Umgebung Ihres Unternehmens ■ Administratorzugriff auf die VMware Cloud on AWS-SDDC-Umgebung Ihres Unternehmens ■ Das VMware Cloud on AWS-API-Token für Ihre VMware Cloud on AWS-Umgebung im VMware Cloud on AWS-Dienst Ihres Unternehmens ■ IP-Adresse oder FQDN von vCenter <p>Administratoren benötigen <i>auch</i> Zugriff auf das vCenter, das von dem VMware Cloud on AWS-Ziel-SDDC verwaltet wird, dessen gesamte Berechtigungen im folgenden Abschnitt <i>Anforderungen des vSphere-Agenten für vCenter-basierte Cloud-Konten</i> auf dieser Seite aufgelistet sind.</p> <p>Weitere Informationen zu den Berechtigungen, die zum Erstellen und Verwenden von VMware Cloud on AWS-Cloud-Konten erforderlich sind, finden Sie unter <i>Verwalten des VMware Cloud on AWS-Datencenters</i> in der VMware Cloud on AWS -Produktdokumentation.</p>

Anforderungen des vSphere-Agenten für vCenter-basierte Cloud-Konten

In der folgenden Tabelle werden die Berechtigungen aufgeführt, die zum Verwalten von VMware Cloud on AWS- und vCenter-Cloud-Konten erforderlich sind. Die Berechtigungen müssen für alle Cluster im vCenter Server und nicht nur für Cluster aktiviert sein, die Endpoints hosten.

Für alle vCenter Server-basierten Cloud-Konten, einschließlich NSX-V, NSX-T, vCenter und VMware Cloud on AWS, muss der Administrator über Anmeldedaten des vSphere-Endpoints oder diejenigen Anmeldedaten verfügen, unter denen der Agent-Dienst in vCenter ausgeführt wird und die administrativen Zugriff auf den Host-vCenter Server bereitstellen.

Weitere Informationen zu den Anforderungen des vSphere-Agenten finden Sie in der [VMware vSphere-Produktdokumentation](#).

Tabelle 2-4. Erforderliche Berechtigungen für vSphere-Agenten zur Verwaltung von vCenter Server-Instanzen

Attributwert	Berechtigung
Datenspeicher	<ul style="list-style-type: none"> ■ Speicher zuteilen ■ Datenspeicher durchsuchen ■ Dateivorgänge auf niedriger Ebene
Datenspeicher-Cluster	Einen Datenspeicher-Cluster konfigurieren
Ordner	<ul style="list-style-type: none"> ■ Ordner erstellen ■ Ordner löschen

Tabelle 2-4. Erforderliche Berechtigungen für vSphere-Agenten zur Verwaltung von vCenter Server-Instanzen (Fortsetzung)

Attributwert	Berechtigung
Global	<ul style="list-style-type: none"> ■ Benutzerdefinierte Attribute verwalten ■ Benutzerdefiniertes Attribut festlegen
Netzwerk	Netzwerk zuweisen
Berechtigungen	Berechtigung ändern
Ressource	<ul style="list-style-type: none"> ■ VM zu Ressourcenpool zuweisen ■ Ausgeschaltete virtuelle Maschine migrieren ■ Einschaltete virtuelle Maschine migrieren
Inhaltsbibliothek	<p>Um eine Berechtigung für eine Inhaltsbibliothek zuzuweisen, muss ein Administrator dem Benutzer die Berechtigung als globale Berechtigung erteilen. Weitere Informationen finden Sie im Abschnitt Hierarchische Vererbung von Berechtigungen für Inhaltsbibliotheken unter <i>Verwaltung virtueller vSphere-Maschinen</i> in der VMware vSphere-Dokumentation.</p> <ul style="list-style-type: none"> ■ Bibliothekselement hinzufügen ■ Lokale Bibliothek erstellen ■ Abonnierte Bibliothek erstellen ■ Bibliothekselement löschen ■ Lokale Bibliothek löschen ■ Abonnierte Bibliothek löschen ■ Dateien herunterladen ■ Bibliothekselement entfernen ■ Abonnierte Bibliothek entfernen ■ Abonnementinformationen prüfen ■ Speicherinfos lesen ■ Bibliothekselement synchronisieren ■ Abonnierte Bibliothek synchronisieren ■ Selbstüberprüfung des Typs ■ Konfigurationseinstellungen aktualisieren ■ Dateien aktualisieren ■ Bibliothek aktualisieren ■ Bibliothekselement aktualisieren ■ Lokale Bibliothek aktualisieren ■ Abonnierte Bibliothek aktualisieren ■ Konfigurationseinstellungen anzeigen

Tabelle 2-4. Erforderliche Berechtigungen für vSphere-Agenten zur Verwaltung von vCenter Server-Instanzen (Fortsetzung)

Attributwert	Berechtigung
Tags	<ul style="list-style-type: none"> ■ vSphere-Tag zuweisen oder Zuweisung aufheben ■ vSphere-Tag erstellen ■ Kategorie für vSphere-Tag erstellen ■ vSphere-Tag löschen ■ Kategorie für vSphere-Tag löschen ■ vSphere-Tag bearbeiten ■ Kategorie für vSphere-Tag bearbeiten ■ UsedBy-Feld für Kategorie ändern ■ UsedBy-Feld für Tag ändern
vApp	<ul style="list-style-type: none"> ■ Importieren ■ vApp-Anwendungskonfiguration <p>Die Anwendungskonfiguration <code>vApp.Import</code> ist für OVF-Vorlagen und für die Bereitstellung von VMs aus der Inhaltsbibliothek erforderlich.</p> <p>Die Anwendungskonfiguration <code>vApp.vApp</code> ist erforderlich, wenn Sie cloud-init für Cloud-Konfigurationsskripts verwenden. Diese Einstellung ermöglicht das Ändern der internen Struktur einer vApp, wie z. B. zugehöriger Produktinformationen und Eigenschaften.</p>
Virtuelle Maschine – Bestandsliste	<ul style="list-style-type: none"> ■ Aus vorhandener erstellen ■ Neue erstellen ■ Verschieben ■ Entfernen
Virtuelle Maschine – Interaktion	<ul style="list-style-type: none"> ■ CD-Medien konfigurieren ■ Konsoleninteraktion ■ Geräteverbindung ■ Ausschalten ■ Einschalten ■ Zurücksetzen ■ Anhalten ■ Tools installieren

Tabelle 2-4. Erforderliche Berechtigungen für vSphere-Agenten zur Verwaltung von vCenter Server-Instanzen (Fortsetzung)

Attributwert	Berechtigung
Virtuelle Maschine – Konfiguration	<ul style="list-style-type: none"> ■ Vorhandene Festplatte hinzufügen ■ Neue Festplatte hinzufügen ■ Festplatte entfernen ■ Erweitert ■ CPU-Anzahl ändern ■ Ressource ändern ■ Virtuelle Festplatte erweitern ■ Festplattenänderungsverfolgung ■ Arbeitsspeicher ■ Geräteeinstellungen ändern ■ Umbenennen ■ Anmerkung festlegen ■ Einstellungen ■ Platzierung der Auslagerungsdatei
Virtuelle Maschine – Bereitstellung	<ul style="list-style-type: none"> ■ Anpassen ■ Vorlage klonen ■ Virtuelle Maschine klonen ■ Vorlage bereitstellen ■ Anpassungsspezifikationen lesen
Virtuelle Maschine – Zustand	<ul style="list-style-type: none"> ■ Snapshot erstellen ■ Snapshot entfernen ■ Snapshot wiederherstellen

Konfigurieren von Microsoft Azure für die Verwendung mit vRealize Automation Cloud Assembly

Sie müssen einige Informationen erfassen und einige Konfigurationsschritte durchführen, um ein Microsoft Azure Cloud-Konto in vRealize Automation Cloud Assembly zu erstellen.

Verfahren

- 1 Suchen und notieren Sie Ihr Microsoft Azure-Abonnement und Ihre Mandanten-IDs.
 - Abonnement-ID: Klicken Sie auf das Symbol „Abonnements“ in der linken Symbolleiste in Ihrem Azure-Portal, um die Abonnement-ID anzuzeigen.
 - Mandanten-ID: Klicken Sie auf das Hilfesymbol und wählen Sie „Diagnose anzeigen“ in Ihrem Azure-Portal aus. Suchen Sie nach einem Mandanten und notieren Sie sich dessen ID.

- 2 Sie können ein neues Speicherkonto und eine Ressourcengruppe erstellen, um zu beginnen. Alternativ können Sie diese später in Blueprints erstellen.

- Speicherkonto: Verwenden Sie das folgende Verfahren, um ein Konto zu konfigurieren.
 - 1 Suchen Sie in Ihrem Azure-Portal das Symbol „Speicherkonten“ auf der Seitenleiste. Stellen Sie sicher, dass das richtige Abonnement ausgewählt ist, und klicken Sie auf **Hinzufügen**. Sie können auch im Azure-Suchfeld nach „Speicherkonto“ suchen.
 - 2 Geben Sie die erforderlichen Informationen für das Speicherkonto ein. Sie benötigen Ihre Abonnement-ID.
 - 3 Wählen Sie aus, ob eine vorhandene Ressourcengruppe verwendet oder eine neue erstellt werden soll. Notieren Sie sich Ihren Ressourcengruppennamen, da Sie ihn später benötigen.

Hinweis Speichern Sie den Speicherort Ihres Speicherkontos, da Sie es später benötigen werden.

- 3 Erstellen Sie ein virtuelles Netzwerk. Alternativ können Sie auch ein geeignetes vorhandenes Netzwerk auswählen.

Wenn Sie ein Netzwerk erstellen, müssen Sie „Vorhandene Ressourcengruppe verwenden“ auswählen und die Gruppe angeben, die Sie im vorherigen Schritt erstellt haben. Wählen Sie außerdem denselben Speicherort aus, den Sie zuvor angegeben haben. Microsoft Azure stellt keine virtuellen Maschinen oder anderen Objekte bereit, wenn der Speicherort nicht für alle zutreffenden Komponenten übereinstimmt, die das Objekt nutzen wird.

- a Suchen Sie im linken Fensterbereich das Symbol für das virtuelle Netzwerk und klicken Sie darauf oder suchen Sie nach einem virtuellen Netzwerk. Stellen Sie sicher, dass Sie das richtige Abonnement auswählen, und klicken Sie auf **Hinzufügen**.
 - b Geben Sie einen eindeutigen Namen für Ihr neues virtuelles Netzwerk ein und notieren Sie ihn für später.
 - c Geben Sie im Feld **Adressraum** die entsprechende IP-Adresse für Ihr virtuelles Netzwerk ein.
 - d Stellen Sie sicher, dass das richtige Abonnement ausgewählt ist, und klicken Sie auf **Hinzufügen**.
 - e Geben Sie die verbleibenden grundlegenden Konfigurationsinformationen ein.
 - f Sie können die anderen Optionen nach Bedarf ändern, aber für die meisten Konfigurationen können Sie die Standardeinstellungen beibehalten.
 - g Klicken Sie auf **Erstellen**.
- 4 Richten Sie eine Azure Active Directory-Anwendung ein, damit vRA authentifiziert werden kann.
- a Suchen Sie das Active Directory-Symbol im linken Azure-Menü und klicken Sie darauf.
 - b Klicken Sie auf **App-Registrierungen** und wählen Sie **Hinzufügen** aus.

- c Geben Sie einen Namen für Ihre Anwendung ein, der mit der Validierung des Azure-Namens übereinstimmt.
 - d Belassen Sie Web-App/API als Anwendungstyp.
 - e Die Anmelde-URL kann alles sein, was für Ihre Nutzung geeignet ist.
 - f Klicken Sie auf **Erstellen**.
- 5 Erstellen Sie einen geheimen Schlüssel, um die Anwendung in Cloud Assembly zu authentifizieren.
- a Klicken Sie auf den Namen Ihrer Anwendung in Azure.
Notieren Sie sich Ihre Anwendungs-ID für die spätere Verwendung.
 - b Klicken Sie auf **Alle Einstellungen** im nächsten Fensterbereich und wählen Sie „Schlüssel“ aus der Einstellungsliste aus.
 - c Geben Sie eine Beschreibung für den neuen Schlüssel ein und wählen Sie eine Dauer aus.
 - d Klicken Sie auf **Speichern** und stellen Sie sicher, dass Sie den Schlüsselwert an einen sicheren Speicherort kopieren, da Sie ihn später nicht mehr abrufen können.
 - e Wählen Sie im linken Menü die Option **API-Berechtigungen** für die Anwendung und klicken Sie auf **Berechtigung hinzufügen**, um eine neue Berechtigung zu erstellen.
 - f Wählen Sie auf der Seite „API auswählen“ die Option „Azure Service Management“ aus.
 - g Klicken Sie auf **Delegierte Berechtigungen**.
 - h Wählen Sie unter „Berechtigungen auswählen“ die Option „user_impersonation“ und klicken Sie dann auf **Berechtigungen hinzufügen**.
- 6 Autorisieren Sie Ihre Active Directory-Anwendung für die Herstellung einer Verbindung mit Ihrem Azure-Abonnement, damit Sie virtuelle Maschinen bereitstellen und verwalten können.
- a Klicken Sie im linken Menü auf das Abonnements-Symbol und wählen Sie Ihr neues Abonnement aus.

Sie müssen möglicherweise auf den Text des Namens klicken, um das Fenster zu verschieben.
 - b Wählen Sie die Zugriffssteuerungsoption (IAM) aus, um die Berechtigungen für Ihr Abonnement anzuzeigen.
 - c Klicken Sie unter der Überschrift „Rollenzuweisung hinzufügen“ auf **Hinzufügen**.
 - d Wählen Sie in der Dropdown-Liste „Rolle“ die Option „Beitragender“ aus.
 - e Belassen Sie die Standardauswahl im Dropdown-Menü „Zugriff zuweisen“.
 - f Geben Sie den Namen Ihrer Anwendung in das Auswahlfeld ein.
 - g Klicken Sie auf **Speichern**.

- h Fügen Sie zusätzliche Rollen hinzu, damit Ihre neue Anwendung über Besitzer-, Beitragender- und Leser-Rollen verfügt.
- i Klicken Sie auf **Speichern**.

Nächste Schritte

Sie müssen die Tools der Microsoft Azure-Befehlszeilenschnittstelle installieren. Diese Tools sind für Windows- und Mac-Betriebssysteme kostenlos erhältlich. Weitere Informationen zum Herunterladen und Installieren dieser Tools finden Sie in der Microsoft-Dokumentation.

Nachdem die Befehlszeilenschnittstelle installiert ist, müssen Sie sich bei Ihrem neuen Abonnement authentifizieren.

- 1 Öffnen Sie ein Terminal-Fenster und geben Sie Ihre Microsoft Azure-Anmeldung ein. Sie erhalten eine URL und einen Kurzcode, um sich zu authentifizieren.
- 2 Geben Sie in einem Browser den Code ein, den Sie von der Anwendung auf Ihrem Gerät erhalten haben.
- 3 Geben Sie Ihren Authentifizierungscode ein und klicken Sie auf **Weiter**.
- 4 Wählen Sie Ihr Azure-Konto aus und melden Sie sich an.

Wenn Sie über mehrere Abonnements verfügen, stellen Sie mit dem Befehl `azure account set <subscription-name>` sicher, dass das richtige ausgewählt wurde.

- 5 Bevor Sie fortfahren, müssen Sie den Microsoft.Compute-Anbieter für Ihr neues Azure-Abonnement mit dem Befehl `azure provider register microsoft.compute` registrieren.

Wenn beim ersten Ausführen des Befehls eine Zeitüberschreitung auftritt und ein Fehler generiert wird, führen Sie ihn erneut aus.

Nachdem Sie die Konfiguration abgeschlossen haben, können Sie den Befehl `azure vm image list` verwenden, um die verfügbaren Image-Namen für virtuelle Maschinen abzurufen. Sie können das gewünschte Image auswählen und den dafür bereitgestellten URN aufzeichnen und später in Blueprints verwenden.

Erstellen Sie ein Microsoft Azure-Cloud-Konto in vRealize Automation.

Als Cloud-Administrator können Sie ein Microsoft Azure-Cloud-Konto für Kontoregionen erstellen, in denen Ihr Team vRealize Automation-Blueprints bereitstellt.

Ein Anwendungsbeispiel zur Funktionsweise eines Microsoft Azure-Cloud-Kontos in vRealize Automation finden Sie im [Anwendungsbeispiel: WordPress](#).

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderlichen Administratoranmeldedaten verfügen und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).

- Stellen Sie sicher, dass Sie über die erforderliche Benutzerrolle verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Konfigurieren Sie ein Microsoft Azure-Konto zur Verwendung mit vRealize Automation. Weitere Informationen hierzu finden Sie unter [Konfigurieren von Microsoft Azure für die Verwendung mit vRealize Automation Cloud Assembly](#).
- Wenn Sie nicht über externen Internetzugriff verfügen, konfigurieren Sie einen Internet-Proxyserver. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus und klicken Sie auf **Cloud-Konto hinzufügen**.
- 2 Wählen Sie den Microsoft Azure-Kontotyp aus und geben Sie Anmeldedaten und andere Werte ein.
- 3 Klicken Sie auf **Validieren**.
Die dem Konto zugeordneten Kontobereiche werden erfasst.
- 4 Wählen Sie die Regionen aus, in denen diese Ressource bereitgestellt werden soll.
- 5 Klicken Sie aus Effizienzgründen auf **Cloud-Zone für die ausgewählten Regionen erstellen**.
- 6 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags hinzufügen müssen, geben Sie Funktions-Tags ein. Weitere Informationen finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#) und [Erstellen einer Tagging-Strategie](#).
- 7 Klicken Sie auf **Speichern**.

Ergebnisse

Das Konto wird zu vRealize Automation hinzugefügt und die ausgewählten Regionen stehen für die angegebene Cloud-Zone zur Verfügung.

Nächste Schritte

Erstellen Sie Infrastrukturressourcen für dieses Cloud-Konto.

Erstellen eines Amazon Web Services-Cloud-Kontos in vRealize Automation

Als Cloud-Administrator können Sie ein Amazon Web Services-Cloud-Konto (AWS) für Kontoregionen erstellen, in denen Ihr Team vRealize Automation-Blueprints bereitstellt.

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderlichen Administratoranmeldedaten verfügen und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die erforderliche Benutzerrolle verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie über die notwendigen AWS-Administratoranmeldedaten verfügen.
- Wenn Sie nicht über externen Internetzugriff verfügen, konfigurieren Sie einen Internet-Proxyserver. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus und klicken Sie auf **Cloud-Konto hinzufügen**.
- 2 Wählen Sie den AWS-Kontotyp aus und geben Sie Anmeldedaten und andere Werte ein.
- 3 Klicken Sie auf **Validieren**.
Die dem Konto zugeordneten Kontobereiche werden erfasst.
- 4 Wählen Sie die Regionen aus, in denen diese Ressource bereitgestellt werden soll.
- 5 Klicken Sie aus Effizienzgründen auf **Cloud-Zone für die ausgewählten Regionen erstellen**.
- 6 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags hinzufügen müssen, geben Sie Funktions-Tags ein. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#) und [Erstellen einer Tagging-Strategie](#).
- 7 Klicken Sie auf **Hinzufügen**.

Ergebnisse

Das Konto wird zu vRealize Automation hinzugefügt und die ausgewählten Regionen stehen für die angegebene Cloud-Zone zur Verfügung.

Nächste Schritte

Konfigurieren Sie Infrastrukturressourcen für dieses Cloud-Konto.

Erstellen Sie ein Google Cloud Platform-Cloud-Konto in vRealize Automation.

Als Cloud-Administrator können Sie ein Google Cloud Platform-Cloud-Konto (GCP) für Kontoregionen erstellen, in denen Ihr Team vRealize Automation-Blueprints bereitstellt.

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderlichen Administratoranmeldedaten verfügen und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die erforderliche Benutzerrolle verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie Zugriff auf den JSON-Sicherheitsschlüssel von Google Cloud Platform haben.
- Stellen Sie sicher, dass Sie über die erforderlichen Sicherheitsinformationen für Ihre Google Cloud Platform-Instanz verfügen. Sie können die meisten dieser Informationen aus Ihrer Instanz oder aus der Google-Dokumentation abrufen.
- Wenn Sie nicht über externen Internetzugriff verfügen, konfigurieren Sie einen Internet-Proxyserver. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus und klicken Sie auf **Cloud-Konto hinzufügen**.
- 2 Wählen Sie den Google Cloud Platform-Kontotyp aus und geben Sie die entsprechenden Anmeldedaten und zugehörigen Informationen ein. Verwenden Sie das Dienstkonto, das beim Initialisieren der Computing-Engine des GCP-Quellkontos erstellt wurde.

Wie im obigen Abschnitt **Voraussetzungen** angegeben, stehen die Anforderungen an die Anmeldedaten unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#) zur Verfügung. Zur erfolgreichen Erstellung des Cloud-Kontos in vRealize Automation muss der Computing-Engine-Dienst für das GCP-Quellkonto aktiviert sein.

In vRealize Automation ist die Projekt-ID Teil des Google Cloud Platform-Endpoints. Sie geben sie bei der Erstellung des Cloud-Kontos an. Während der Datenerfassung projektspezifischer privater Images fragt der GCP-Adapter von vRealize Automation die Google Cloud Platform-API ab.

- 3 Klicken Sie auf **Validieren**.
Die dem Konto zugeordneten Kontobereiche werden erfasst.
- 4 Wählen Sie die Regionen aus, in denen diese Ressource bereitgestellt werden soll.
- 5 Klicken Sie aus Effizienzgründen auf **Cloud-Zone für die ausgewählten Regionen erstellen**.
- 6 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags benötigen, geben Sie Funktions-Tags ein. Weitere Informationen finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#) und [Erstellen einer Tagging-Strategie](#).

7 Klicken Sie auf **Hinzufügen**.

Ergebnisse

Das Konto wird zu vRealize Automation hinzugefügt und die ausgewählten Regionen stehen für die angegebene Cloud-Zone zur Verfügung.

Nächste Schritte

Erstellen Sie Infrastrukturressourcen für dieses Cloud-Konto.

Erstellen Sie ein vCenter-Cloud-Konto in vRealize Automation Cloud Assembly.

Sie fügen ein vCenter-Cloud-Konto für die Kontobereiche hinzu, für die Sie vRealize Automation Cloud Assembly-Blueprints bereitstellen möchten.

Zu Netzwerk- und Sicherheitszwecken können Sie ein NSX-T- oder NSX-V-Cloud-Konto mit dem vCenter-Cloud-Konto verknüpfen.

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderlichen Administratoranmeldedaten verfügen und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Ihre Ports und Protokolle ordnungsgemäß zur Unterstützung des Cloud-Kontos konfiguriert wurden. Weitere Informationen finden Sie im Thema *Ports und Protokolle für vRealize Automation* unter *Installieren von vRealize Automation mit dem vRealize Easy-Installationsprogramm* und im Thema *Portanforderungen* im *vRealize Automation-Handbuch zur Referenzarchitektur* in der [vRealize Automation-Produktdokumentation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus und klicken Sie auf **Cloud-Konto hinzufügen**.
- 2 Wählen Sie den vCenter-Kontotyp aus und geben Sie die IP-Adresse des vCenter Server-Hosts ein.
- 3 Geben Sie Ihre vCenter Server-Administratoranmeldedaten ein und klicken Sie auf **Validieren**.
Die Datacenter, die dem Konto zugeordnet sind, werden erfasst.
- 4 Wählen Sie mindestens ein verfügbares Datacenter auf dem angegebenen vCenter Server aus, um die Bereitstellung für dieses Cloud-Konto zuzulassen.

- 5 Erstellen Sie aus Effizienzgründen eine Cloud-Zone zur Bereitstellung in den ausgewählten Datencentern.

Sie können Cloud-Zonen auch in einem separaten Schritt gemäß der Cloud-Strategie Ihrer Organisation erstellen.

Informationen über Cloud-Zonen finden Sie unter [Weitere Informationen zu vRealize Automation Cloud Assembly-Cloud-Zonen](#).

- 6 Wählen Sie ein vorhandenes NSX-Cloud-Konto aus.

Sie können das NSX-Konto jetzt oder später auswählen, wenn Sie das Cloud-Konto bearbeiten.

Informationen über NSX-V-Cloud-Konten finden Sie unter [Erstellen eines NSX-V-Cloud-Kontos in vRealize Automation Cloud Assembly](#).

Informationen über NSX-T-Cloud-Konten finden Sie unter [Erstellen eines NSX-T-Cloud-Kontos in vRealize Automation Cloud Assembly](#).

- 7 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags hinzufügen möchten, geben Sie Funktions-Tags ein.

Sie können Tags jetzt oder später hinzufügen, wenn Sie das Cloud-Konto bearbeiten. Weitere Informationen zum Tagging finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#).

- 8 Klicken Sie auf **Speichern**.

Ergebnisse

Das Cloud-Konto wird hinzugefügt und die ausgewählten Datacenter stehen für die angegebene Cloud-Zone zur Verfügung. Erfasste Daten, wie z. B. Maschinen und Volumes, werden im Abschnitt „Ressourcen“ der Registerkarte „Infrastruktur“ aufgeführt.

Nächste Schritte

Konfigurieren Sie verbleibende Infrastrukturressourcen für dieses Cloud-Konto. Weitere Informationen hierzu finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#).

Erstellen eines NSX-V-Cloud-Kontos in vRealize Automation Cloud Assembly

Zu Netzwerk- und Sicherheitszwecken können Sie ein NSX-V-Cloud-Konto erstellen und mit einem vCenter-Cloud-Konto verknüpfen.

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderlichen Administratoranmeldedaten verfügen und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).

- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie über ein vCenter-Cloud-Konto für die Verwendung mit diesem NSX-Cloud-Konto verfügen. Weitere Informationen hierzu finden Sie unter [Erstellen Sie ein vCenter-Cloud-Konto in vRealize Automation Cloud Assembly](#).
- Stellen Sie sicher, dass Ihre Ports und Protokolle ordnungsgemäß zur Unterstützung des Cloud-Kontos konfiguriert wurden. Weitere Informationen finden Sie im Thema *Ports und Protokolle für vRealize Automation* unter *Installieren von vRealize Automation mit dem vRealize Easy-Installationsprogramm* und im Thema *Portanforderungen* im *vRealize Automation-Handbuch zur Referenzarchitektur* in der [vRealize Automation-Produktdokumentation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus und klicken Sie auf **Cloud-Konto hinzufügen**.
- 2 Wählen Sie den NSX-V-Kontotyp aus und geben Sie die IP-Adresse des NSX-V-Hosts ein.
- 3 Geben Sie Ihre NSX-Administratoranmeldedaten ein und klicken Sie auf **Validieren**.
Die dem Konto zugeordneten Objekte werden erfasst.
Wenn die IP-Adresse des NSX-Hosts nicht verfügbar ist, schlägt die Validierung fehl.
- 4 Wählen Sie gegebenenfalls den vCenter-Endpoint aus, der das vCenter-Cloud-Konto darstellt, das Sie diesem NSX-V-Konto zuordnen.
- 5 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags hinzufügen möchten, geben Sie Funktions-Tags ein.
Sie können Funktions-Tags auch noch später hinzufügen oder entfernen. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#).
- 6 Klicken Sie auf **Speichern**.

Nächste Schritte

Sie können ein vCenter-Cloud-Konto erstellen oder bearbeiten und es diesem NSX-Cloud-Konto zuordnen. Weitere Informationen hierzu finden Sie unter [Erstellen Sie ein vCenter-Cloud-Konto in vRealize Automation Cloud Assembly](#).

Erstellen und konfigurieren Sie eine oder mehrere Cloud-Zonen für die Verwendung mit den Datencentern, die von diesem Cloud-Konto verwendet werden. Weitere Informationen hierzu finden Sie unter [Weitere Informationen zu vRealize Automation Cloud Assembly-Cloud-Zonen](#).

Konfigurieren Sie Infrastrukturressourcen für dieses Cloud-Konto. Weitere Informationen hierzu finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#).

Erstellen eines NSX-T-Cloud-Kontos in vRealize Automation Cloud Assembly

Zu Netzwerk- und Sicherheitszwecken können Sie ein NSX-T-Cloud-Konto erstellen und mit einem vCenter-Cloud-Konto verknüpfen.

Um Fault Tolerance und Hochverfügbarkeit in Bereitstellungen zu vereinfachen, stellt jeder NSX-T-Datencenter-Endpoint ein Cluster aus drei NSX Managern dar.

- vRealize Automation kann auf einen der NSX Manager verweisen. Bei dieser Option empfängt ein NSX Manager die API-Aufrufe von vRealize Automation.
- vRealize Automation kann auf die virtuelle IP des Clusters verweisen. Bei dieser Option übernimmt ein NSX Manager die Steuerung der VIP. Dieser Manager empfängt die API-Aufrufe von vRealize Automation. Bei einem Ausfall übernimmt ein anderer Knoten im Cluster die Steuerung der VIP und empfängt die API-Aufrufe von vRealize Automation.

Weitere Informationen zur VIP-Konfiguration finden Sie unter *Konfigurieren einer VIP-Adresse (Virtual IP) für einen Cluster* im *Installationshandbuch für NSX-T Data Center* in der [Dokumentation zu VMware NSX-T Data Center](#).

- vRealize Automation kann auf eine Lastausgleichsdienst-VIP verweisen, um die Last der Aufrufe auf die drei NSX Manager zu verteilen. Bei dieser Option empfangen alle drei NSX Manager API-Anrufe von vRealize Automation.

Sie können die VIP auf einem Lastausgleichsdienst eines Drittanbieters oder auf einem NSX-T-Lastausgleichsdienst konfigurieren.

In umfangreichen Umgebungen sollten Sie diese Option verwenden, um die vRealize Automation-API-Aufrufe auf die drei NSX Manager zu verteilen.

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderlichen Administratoranmeldedaten verfügen und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie über ein vCenter-Cloud-Konto für die Verwendung mit diesem NSX-Cloud-Konto verfügen. Weitere Informationen hierzu finden Sie unter [Erstellen Sie ein vCenter-Cloud-Konto in vRealize Automation Cloud Assembly](#).
- Stellen Sie sicher, dass Ihre Ports und Protokolle ordnungsgemäß zur Unterstützung des Cloud-Kontos konfiguriert wurden. Weitere Informationen finden Sie im Thema *Ports und Protokolle für vRealize Automation* unter *Installieren von vRealize Automation mit dem vRealize Easy-Installationsprogramm* und im Thema *Portanforderungen* im *vRealize Automation-Handbuch zur Referenzarchitektur* in der [vRealize Automation-Produktdokumentation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus und klicken Sie auf **Cloud-Konto hinzufügen**.
- 2 Wählen Sie den NSX-T-Kontotyp aus und geben Sie die IP-Adresse des Hosts für die Instanz des NSX-T-Endpoint-Managers oder die VIP ein (siehe oben).
- 3 Geben Sie Ihre NSX-Administratoranmeldedaten ein und klicken Sie auf **Validieren**.
Die dem Konto zugeordneten Objekte werden erfasst.
Wenn die IP-Adresse des NSX-Hosts nicht verfügbar ist, schlägt die Validierung fehl.
- 4 Falls verfügbar, wählen Sie den vCenter-Endpoint aus, der das vCenter-Cloud-Konto darstellt, das Sie diesem NSX-T-Cloud-Konto zuordnen.
- 5 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags hinzufügen möchten, geben Sie Funktions-Tags ein.
Sie können Funktions-Tags auch noch später hinzufügen oder entfernen. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#).
- 6 Klicken Sie auf **Speichern**.

Nächste Schritte

Sie können ein vCenter-Cloud-Konto erstellen oder bearbeiten, um es diesem NSX-Cloud-Konto zuzuordnen. Weitere Informationen hierzu finden Sie unter [Erstellen Sie ein vCenter-Cloud-Konto in vRealize Automation Cloud Assembly](#).

Erstellen und konfigurieren Sie eine oder mehrere Cloud-Zonen für die Verwendung mit den Datencentern, die von diesem Cloud-Konto verwendet werden. Weitere Informationen hierzu finden Sie unter [Weitere Informationen zu vRealize Automation Cloud Assembly-Cloud-Zonen](#).

Konfigurieren Sie Infrastrukturressourcen für dieses Cloud-Konto. Weitere Informationen hierzu finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#).

Erstellen Sie ein VMware Cloud on AWS-Cloud-Konto in vRealize Automation.

Als Cloud-Administrator können Sie ein VMware Cloud on AWS-Cloud-Konto für Kontoregionen erstellen, in denen Ihr Team vRealize Automation-Blueprints bereitstellt.

VMware Cloud on AWS erfordert bestimmte eindeutige Konfigurationsverfahren in vRealize Automation. Informationen zum ordnungsgemäßen Konfigurieren von vRealize Automation für VMware Cloud on AWS, einschließlich der Festlegung von API-Token-Werten für das Cloud-Konto und der Angabe von Gateway-Firewallregeln für den zugehörigen Cloud-Proxy, finden Sie im Workflow [VMware Cloud on AWS-Anwendungsbeispiel](#).

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderlichen Administratoranmeldedaten für VMware Cloud on AWS verfügen, einschließlich VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter, und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Wenn Sie nicht über externen Internetzugriff verfügen, konfigurieren Sie einen Internet-Proxyserver. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus, klicken Sie auf **Cloud-Konto hinzufügen** und wählen Sie den Kontotyp VMware Cloud on AWS aus.

- 2 Fügen Sie das **VMC-API-Token** für Ihre Organisation hinzu, um auf die verfügbaren SDDCs zuzugreifen.

Sie können auf der verknüpften Seite **API-Token** für Ihre Organisation ein neues Token erstellen oder ein vorhandenes Token verwenden. Weitere Informationen finden Sie unter [Erstellen eines VMware Cloud on AWS-Cloud-Kontos in vRealize Automation innerhalb eines Beispiel-Workflows](#).

- 3 Wählen Sie das SDDC aus, das für Bereitstellungen verfügbar sein soll.

NSX-V-SDDCs werden nicht unterstützt und nicht in der Liste angezeigt.

Die Werte für Manager-IP-Adresse/FQDN in vCenter und NSX-T werden basierend auf dem SDDC automatisch befüllt.

- 4 Geben Sie Ihren vCenter-Benutzernamen und das zugehörige Kennwort für das angegebene SDDC ein, wenn diese sich vom Standardwert „cloudadmin@vmc.local“ unterscheiden.

- 5 Klicken Sie auf **Validieren**, um Ihre Zugriffsrechte für das angegebene vCenter zu bestätigen, und stellen Sie sicher, dass das vCenter ausgeführt wird.

Die dem Konto zugeordneten Datacenter werden erfasst.

- 6 Erstellen Sie aus Effizienzgründen eine Cloud-Zone zur Bereitstellung im ausgewählten SDDC.

Sie können Cloud-Zonen auch in einem separaten Schritt gemäß der Cloud-Strategie Ihrer Organisation erstellen.

- 7 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags hinzufügen müssen, geben Sie Funktions-Tags ein.

Sie können Funktions-Tags auch noch später hinzufügen oder entfernen. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#).

- 8 Klicken Sie auf **Speichern**.

Ergebnisse

Das Cloud-Konto wird hinzugefügt, und das ausgewählte SDDC ist für die angegebene Cloud-Zone verfügbar.

Nächste Schritte

Informationen zum ordnungsgemäßen Konfigurieren von vRealize Automation für VMware Cloud on AWS finden Sie unter [VMware Cloud on AWS-Anwendungsbeispiel](#).

Weitere Informationen zu VMware Cloud on AWS außerhalb von vRealize Automation finden Sie in der [Dokumentation zu VMware Cloud on AWS](#).

Integrieren von vRealize Automation mit anderen Anwendungen

Integrationen ermöglichen das Hinzufügen externer Systeme zu vRealize Automation.

Zu den Integrationen gehören vRealize Orchestrator, Konfigurationsverwaltung und andere externe Systeme, wie z. B. GitHub, Ansible und Puppet, sowie externe IPAM-Drittanbieter, wie z. B. Infoblox.

Hinweis Wenn Sie nicht über externen Internetzugriff verfügen, dieser von der Integration aber benötigt wird, können Sie einen Internet-Proxyserver konfigurieren. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation](#).

Vorgehensweise zum Verwenden der GitLab- und GitHub-Integration in vRealize Automation Cloud Assembly

vRealize Automation Cloud Assembly unterstützt die Integration mit GitLab- und GitHub-Repositorys, damit Sie Blueprints und Aktionsskripts in der Quellcodeverwaltung verwalten können. Diese Funktion erleichtert die Überwachung und Nachprüfbarkeit von Prozessen rund um die Bereitstellung.

Die Konfiguration eines geeigneten lokalen Git-Repositorys mit Zugriff für alle benannten Benutzer ist Voraussetzung für die Einrichtung der Git-Integration mit vRealize Automation Cloud Assembly. Außerdem müssen Sie die Blueprints in einer bestimmten Struktur speichern, damit sie von Git erkannt werden. Wählen Sie zum Erstellen einer Integration mit GitLab oder GitHub die Optionen **Infrastruktur > Verbindungen > Integrationen** in Cloud Assembly aus und treffen Sie dann die entsprechende Auswahl. Sie benötigen die URL und das Token für das Ziel-Repository.

Wenn die Git-Integration mit einem vorhandenen Repository konfiguriert ist, werden alle Blueprints, die mit den ausgewählten Projekten verknüpft sind, qualifizierten Benutzern zur Verfügung gestellt. Sie können diese Blueprints mit einer vorhandenen Bereitstellung oder als Grundlage einer neuen Bereitstellung verwenden. Wenn Sie ein Projekt hinzufügen, müssen Sie einige Eigenschaften für den Speicherort und die Art der Speicherung in Git auswählen.

Sie können Aktionen in einem Git-Repository direkt aus vRealize Automation Cloud Assembly speichern. Aktionsskripte für Versionen können Sie entweder direkt in Git erstellen, oder Sie können Versionen in vRealize Automation Cloud Assembly erstellen. Wenn Sie eine Version einer Aktion in vRealize Automation Cloud Assembly erstellen, wird diese automatisch als Version in Git gespeichert. Blueprints sind etwas komplizierter, da Sie sie nicht direkt von vRealize Automation Cloud Assembly aus zu einer Git-Integration hinzufügen können. Stattdessen müssen Sie sie direkt in einer Git-Instanz speichern. Anschließend können Sie sie von Git abrufen, wenn Sie mit der Seite „Blueprint-Verwaltung“ in vRealize Automation Cloud Assembly arbeiten.

Vorbereitungen

Sie müssen Ihre Blueprints in einer bestimmten Struktur erstellen und speichern, damit sie von GitLab oder GitHub erkannt werden.

- Konfigurieren und speichern Sie Blueprints zur ordnungsgemäßen Integration in GitLab. Nur gültige Blueprints werden in GitLab importiert.
 - Erstellen Sie einen oder mehrere Ordner für die Blueprints.
 - Alle Blueprints müssen in `blueprint.yaml`-Dateien gespeichert werden.
 - Stellen Sie sicher, dass die oberen Blueprints die Eigenschaften `name:` und `version:` enthalten.
- Extrahieren Sie einen API-Schlüssel für das entsprechende Repository. Wählen Sie in Ihrem Git-Konto oben rechts Ihre Anmeldung aus und navigieren Sie zum Menü „Einstellungen“. Wählen Sie **Zugriffstoken** aus, benennen Sie Ihr Token um und legen Sie ein Ablaufdatum fest. Wählen Sie dann API aus und erstellen Sie das Token. Kopieren Sie den Ergebniswert und speichern Sie ihn.

Die folgenden Richtlinien müssen für alle Blueprints beachtet werden, die bei der Git-Integration verwendet werden.

- Jeder Blueprint muss sich in einem separaten Ordner befinden.
- Alle Blueprints müssen den Namen `blueprint.yaml` haben.
- In allen Blueprint-YAML-Dateien müssen die Felder `name` und `version` verwendet werden.

- Es werden nur gültige Blueprints importiert.
- Wenn Sie einen von Git importierten Entwurf eines Blueprints aktualisieren und dessen Inhalt sich von dem in der höchsten Version unterscheidet, wird der Entwurf bei nachfolgenden Synchronisierungen nicht aktualisiert und es wird eine neue Version erstellt. Wenn Sie einen Blueprint aktualisieren und auch weitere Synchronisierungen von Git zulassen möchten, müssen Sie nach den abschließenden Änderungen eine neue Version erstellen.
- **Konfigurieren der GitLab-Blueprint-Integration in vRealize Automation Cloud Assembly**
In diesem Verfahren wird dargestellt, wie Sie eine GitLab-Integration in vRealize Automation Cloud Assembly so konfigurieren, dass Sie im Repository mit Blueprints arbeiten und gespeicherte Blueprints automatisch herunterladen können, die zugewiesenen Projekten zugeordnet sind. Zur Verwendung von Blueprints mit GitLab müssen Sie eine Verbindung zu einer geeigneten GitLab-Instanz herstellen und die gewünschten Blueprints dann in dieser Instanz speichern.
- **Konfigurieren der GitHub-Integration in vRealize Automation Cloud Assembly**
Sie können den cloudbasierten Repository-Hosting-Dienst von GitHub in vRealize Automation Cloud Assembly integrieren.

Konfigurieren der GitLab-Blueprint-Integration in vRealize Automation Cloud Assembly

In diesem Verfahren wird dargestellt, wie Sie eine GitLab-Integration in vRealize Automation Cloud Assembly so konfigurieren, dass Sie im Repository mit Blueprints arbeiten und gespeicherte Blueprints automatisch herunterladen können, die zugewiesenen Projekten zugeordnet sind. Zur Verwendung von Blueprints mit GitLab müssen Sie eine Verbindung zu einer geeigneten GitLab-Instanz herstellen und die gewünschten Blueprints dann in dieser Instanz speichern.

Wenn die GitLab-Integration mit einem vorhandenen Repository konfiguriert ist, werden alle Blueprints, die mit den ausgewählten Projekten verknüpft sind, qualifizierten Benutzern zur Verfügung gestellt. Sie können diese Blueprints mit einer vorhandenen Bereitstellung oder als Grundlage einer neuen Bereitstellung verwenden. Wenn Sie ein Projekt hinzufügen, müssen Sie einige Eigenschaften für den Speicherort und die Art der Speicherung in GitLab auswählen.

Hinweis Sie können keine neuen oder aktualisierten Blueprints aus vRealize Automation Cloud Assembly in das Git-Repository übertragen. Sie können auch keine neuen Blueprints aus vRealize Automation Cloud Assembly in das Repository übertragen. Um Blueprints zu einem Repository hinzuzufügen, müssen Entwickler die Git-Schnittstelle verwenden.

Wenn Sie einen von Git importierten Entwurf eines Blueprints aktualisieren und dessen Inhalt sich von dem in der höchsten Version unterscheidet, wird der Entwurf bei nachfolgenden Synchronisierungen nicht aktualisiert und es wird eine neue Version erstellt. Wenn Sie einen Blueprint aktualisieren und auch weitere Synchronisierungen von Git zulassen möchten, müssen Sie nach den abschließenden Änderungen eine neue Version erstellen.

Nachdem Sie Ihre Blueprints für die Verwendung mit GitLab eingerichtet und die erforderlichen Informationen erfasst haben, müssen Sie die GitLab-Instanz integrieren. Anschließend können Sie die vorgesehenen Blueprints in GitLab importieren. Sie können eine Videovorführung dieses Verfahrens unter <https://www.youtube.com/watch?v=h0vqo63Sdgg> anzeigen.

Voraussetzungen

- Extrahieren Sie einen API-Schlüssel für das entsprechende Repository. Wählen Sie in Ihrem GitLab-Konto oben rechts Ihre Anmeldung aus und navigieren Sie zum Menü „Einstellungen“. Wählen Sie „Zugriffstoken“ aus, benennen Sie Ihr Token und legen Sie ein Ablaufdatum fest. Wählen Sie dann API aus und erstellen Sie das Token. Kopieren Sie den Ergebniswert und speichern Sie ihn.

Die Konfiguration eines geeigneten lokalen Git-Repositorys mit Zugriff für alle benannten Benutzer ist Voraussetzung für die Einrichtung der Git-Integration mit vRealize Automation Cloud Assembly. Außerdem müssen Sie Ihre Blueprints in einer bestimmten Struktur erstellen und speichern, damit sie von GitLab erkannt werden.

- Konfigurieren und speichern Sie Blueprints zur ordnungsgemäßen Integration in GitLab. Nur gültige Blueprints werden in GitLab importiert. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwenden der GitLab- und GitHub-Integration in vRealize Automation Cloud Assembly](#).

Verfahren

- 1 Richten Sie die Integration mit Ihrer GitLab-Umgebung in vRealize Automation Cloud Assembly ein.
 - a Wählen Sie **Infrastruktur > Integrationen > Neue hinzufügen** und dann „GitLab“ aus.
 - b Geben Sie die **URL** für Ihre GitLab-Instanz ein. Für eine GitLab-SaaS-Instanz (Software as a Service) lautet diese in den meisten Fällen „gitlab.com“.
 - c Geben Sie das **Token**, das auch als API-Schlüssel bezeichnet wird, für die angegebene GitLab-Instanz ein. Informationen zum Extrahieren des Tokens aus Ihrer GitLab-Instanz finden Sie in den obigen Voraussetzungen.
 - d Fügen Sie einen geeigneten Namen und eine geeignete Beschreibung hinzu.
 - e Klicken Sie auf **Überprüfen**, um die Verbindung zu überprüfen.
 - f Fügen Sie bei Bedarf Funktions-Tags hinzu. Weitere Informationen hierzu finden Sie unter [Verwenden von Funktions-Tags in vRealize Automation Cloud Assembly](#).
 - g Klicken Sie auf **Hinzufügen**.
- 2 Konfigurieren Sie die GitLab-Verbindung so, dass Blueprints in einem geeigneten Repository akzeptiert werden.
 - a Wählen Sie **InfrastrukturIntegrationen** und dann die entsprechende GitLab-Integration aus.
 - b Wählen Sie **Projekte** aus.

- c Wählen Sie **Neues Projekt** aus und erstellen Sie einen Namen für das Projekt.
- d Geben Sie den Pfad des **Repositorys** innerhalb von GitLab ein. In der Regel ist dies der Benutzername des Hauptkontos, der an den Namen des Repositorys angehängt wird.
- e Geben Sie die entsprechende GitLab-**Verzweigung** ein, die verwendet werden soll.
- f Geben Sie gegebenenfalls unter **Ordner** einen Namen ein. Wenn Sie dieses Feld leer lassen, stehen alle Ordner zur Verfügung.
- g Geben Sie einen geeigneten **Typ** ein. Geben Sie gegebenenfalls einen Ordnernamen ein. Wenn Sie dieses Feld leer lassen, stehen alle Ordner zur Verfügung.
- h Klicken Sie auf **Weiter**, um das Hinzufügen des Repositorys abzuschließen.

Wenn Sie auf **Weiter** klicken, wird eine automatisierte Synchronisierungsaufgabe initiiert, die Blueprints in die Plattform importiert.

Nach Abschluss der Synchronisierungsaufgaben wird eine Meldung mit dem Hinweis angezeigt, dass die Blueprints importiert wurden.

Ergebnisse

Sie können jetzt Blueprints aus GitLab abrufen.

Konfigurieren der GitHub-Integration in vRealize Automation Cloud Assembly

Sie können den cloudbasierten Repository-Hosting-Dienst von GitHub in vRealize Automation Cloud Assembly integrieren.

Sie benötigen ein gültiges GitHub-Token, um die GitHub-Integration in vRealize Automation Cloud Assembly zu konfigurieren. Weitere Informationen zum Erstellen und Suchen Ihres Tokens finden Sie in der GitHub-Dokumentation.

Voraussetzungen

- Dazu müssen Sie Zugriff auf GitHub haben.
- Konfigurieren und speichern Sie Blueprints zur ordnungsgemäßen Integration in GitHub. Nur gültige Blueprints werden in GitHub importiert. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwenden der GitLab- und GitHub-Integration in vRealize Automation Cloud Assembly](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus und klicken Sie auf **Integration hinzufügen**.
- 2 Wählen Sie „GitHub“ aus.
- 3 Geben Sie auf der GitHub-Konfigurationsseite die erforderlichen Informationen ein.
- 4 Klicken Sie auf **Validieren**, um die Integration zu überprüfen.

- 5 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags hinzufügen müssen, geben Sie Funktions-Tags ein. Weitere Informationen finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#) und [Erstellen einer Tagging-Strategie](#).

- 6 Klicken Sie auf **Hinzufügen**.

Ergebnisse

GitHub ist für die Verwendung in vRealize Automation Cloud Assembly-Blueprints verfügbar.

Nächste Schritte

Sie können jetzt Blueprints aus GitHub abrufen.

Konfigurieren eines externen IPAM-Integrationspunkts in vRealize Automation

Sie können einen anbieterspezifischen externen IPAM-Integrationspunkt erstellen, um die in Ihren Blueprint-Bereitstellungen verwendeten IP-Adressen zu verwalten. Bei Verwendung eines externen IPAM-Integrationspunkts werden IP-Adressen vom benannten IPAM-Anbieter und nicht von vRealize Automation abgerufen und verwaltet.

Sie können einen anbieterspezifischen IPAM-Integrationspunkt erstellen, um IP-Adressen und DNS-Einstellungen für Blueprint-Bereitstellungen und VMs in vRealize Automation zu verwalten.

Informationen zum Konfigurieren der Voraussetzungen und ein Beispiel für die Erstellung eines anbieterspezifischen externen IPAM-Integrationspunkts im Kontext eines Beispielworkflows finden Sie unter [Hinzufügen eines externen IPAM-Integrationspunkts in vRealize Automation](#).

Informationen zum Erstellen der erforderlichen Elemente, damit externe IPAM-Partner und -Anbieter ihre IPAM-Lösung in vRealize Automation integrieren können, finden Sie unter [Vorgehensweise zum Verwenden des IPAM-SDK zur Erstellung eines anbieterspezifischen, externen IPAM-Integrationspakets für vRealize Automation](#).

Voraussetzungen

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie über ein Konto beim externen IPAM-Anbieter verfügen, z. B. [Infoblox](#) oder [BlueCat](#), und dass Sie über die korrekten Zugriffsberechtigungen für das Konto Ihrer Organisation beim IPAM-Anbieter verfügen.

- Vergewissern Sie sich, dass Sie Zugriff auf ein bereitgestelltes Integrationspaket für den IPAM-Anbieter haben, wie z. B. Infoblox oder BlueCat. Das bereitgestellte Paket wird zunächst als ZIP-Download von Ihrem IPAM-Anbieter oder vom vRealize Automation-Marketplace abgerufen und anschließend in vRealize Automation bereitgestellt.
- Vergewissern Sie sich, dass Sie auf eine konfigurierte Ausführungsumgebung für den IPAM-Anbieter zugreifen können.
- Wenn Sie eine lokale eingebettete Ausführungsumgebung mit aktionsbasierter Erweiterbarkeit (ABX) verwenden, müssen Sie sicherstellen, dass im vRealize Automation-Netzwerk ein HTTP-Proxyserver vorhanden ist, der ausgehenden Datenverkehr an externe Sites wie „gcr.io“ und „storage.googleapis.com“ weiterleiten kann. Weitere Informationen finden Sie unter [Abrufen von Docker-Images hinter einem Proxy in vRealize Automation 8.x \(75180\)](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus und klicken Sie auf **Integration hinzufügen**.
- 2 Klicken Sie auf **IPAM**.
- 3 Wählen Sie im Dropdown-Menü **Anbieter** ein konfiguriertes IPAM-Anbieterpaket aus der Liste aus.

Wenn die Liste leer ist, klicken Sie auf **Anbieterpaket importieren**, navigieren Sie zur ZIP-Datei eines vorhandenen Anbieterpakets und wählen Sie sie aus. Wenn Sie nicht über die ZIP-Datei verfügen, können Sie sie von der Website Ihres Anbieters oder über die Registerkarte **Marketplace** in vRealize Automation abrufen.

- 4 Geben Sie Ihren Administratorbenutzernamen und das zugehörige Kennwort für Ihr Konto beim externen IPAM-Anbieter sowie die Informationen für alle anderen obligatorischen Felder (sofern vorhanden) ein, z. B. den Hostnamen Ihres Anbieters.
- 5 Wählen Sie in der Dropdown-Liste **Laufende Umgebung** eine vorhandene Ausführungsumgebung aus, z. B. den lokalen aktionsbasierten Erweiterbarkeits-Integrationspunkt.

Die Ausführungsumgebung unterstützt die Kommunikation zwischen vRealize Automation und dem IPAM-Anbieter.

Das IPAM-Framework unterstützt nur eine lokale eingebettete Ausführungsumgebung mit aktionsbasierter Erweiterbarkeit (ABX).

Hinweis Wenn Sie ein Amazon Web Services- oder ein Microsoft Azure-Cloud-Konto als Ausführungsumgebung der Integration verwenden, stellen Sie sicher, dass auf die IPAM-Anbieter-Appliance über das Internet zugegriffen werden kann, dass sie sich nicht hinter einer NAT oder Firewall befindet und dass sie einen öffentlich auflösbaren DNS-Namen aufweist. Wenn auf den IPAM-Anbieter nicht zugegriffen werden kann, können die Amazon Web Services-Lambda- oder Microsoft Azure-Funktionen keine Verbindung zu ihm herstellen und die Integration schlägt fehl.

6 Klicken Sie auf **Validieren**.

7 Wenn Sie dazu aufgefordert werden, dem selbstsignierten Zertifikat vom externen IPAM-Anbieter zu vertrauen, klicken Sie auf **Akzeptieren**.

Nachdem Sie das selbstsignierte Zertifikat akzeptiert haben, kann die Validierungsaktion bis zum Abschluss fortgesetzt werden.

8 Geben Sie einen Namen für diesen IPAM-Integrationspunkt ein und klicken Sie auf **Hinzufügen**, um den neuen IPAM-Integrationspunkt zu speichern.

Eine Datenerfassungsaktion wird imitiert. Die Daten von Netzwerken und IP-Adressen werden vom externen IPAM-Anbieter erfasst.

Vorgehensweise zum Upgrade auf ein neueres-IPAM-Integrationspaket in vRealize Automation

Sie können einen vorhandenen externen IPAM-Integrationspunkt aktualisieren, um eine aktuellere Version des anbieterspezifischen IPAM-Integrationspakets zu erhalten.

Ein externer IPAM-Anbieter oder VMware kann ein IPAM-Quellintegrationspaket für einen bestimmten Anbieter aktualisieren. Das externe IPAM-Integrationspaket für Infoblox wurde beispielsweise mehrmals aktualisiert. Zur Beibehaltung aller vorhandenen vRealize Automation-Infrastruktureinstellungen, die einen benannten IPAM-Integrationspunkt verwenden, können Sie einen IPAM-Integrationspunkt bearbeiten, um das aktualisierte IPAM-Integrationspaket zu beziehen, statt einen neuen IPAM-Integrationspunkt zu erstellen.

Voraussetzungen

In diesem Verfahren wird davon ausgegangen, dass Sie bereits einen externen IPAM-Integrationspunkt erstellt haben und ein Upgrade für diesen Integrationspunkt durchführen möchten, um eine aktuellere Version des IPAM-Integrationspakets des Anbieters zu verwenden.

Informationen zum Erstellen eines externen IPAM-Integrationspunkts finden Sie unter [Hinzufügen eines externen IPAM-Integrationspunkts in vRealize Automation](#).

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).

- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie über ein Konto beim externen IPAM-Anbieter sowie über die korrekten Zugriffsberechtigungen für das Konto Ihrer Organisation bei diesem IPAM-Anbieter verfügen.
- Vergewissern Sie sich, dass Sie auf ein bereitgestelltes Integrationspaket für Ihren IPAM-Anbieter zugreifen können. Das bereitgestellte Paket wird zunächst als ZIP-Download von Ihrer IPAM-Anbieter-Website oder vom vRealize Automation-Marketplace abgerufen und anschließend in vRealize Automation bereitgestellt.

Informationen dazu, wie Sie die ZIP-Datei des Anbieterpakets herunterladen und bereitstellen und das Paket als Wert für **Anbieter** auf der Seite „IPAM-Integration“ zur Verfügung stellen, finden Sie unter [Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets zur Verwendung in vRealize Automation Cloud Assembly](#).

- Vergewissern Sie sich, dass Sie auf eine konfigurierte Ausführungsumgebung für den IPAM-Anbieter zugreifen können. Bei der Ausführungsumgebung handelt es sich in der Regel um einen lokalen eingebetteten Integrationspunkt mit aktionsbasierter Erweiterbarkeit.

Informationen zu den Merkmalen der Ausführungsumgebung finden Sie unter [Erstellen einer Ausführungsumgebung für einen IPAM-Integrationspunkt in vRealize Automation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen IPAM** aus und öffnen Sie den vorhandenen IPAM-Integrationspunkt.
- 2 Klicken Sie auf **Anbieter verwalten**.
- 3 Navigieren Sie zum aktualisierten IPAM-Integrationspaket und importieren Sie es.
- 4 Klicken Sie auf **Validieren** und anschließend auf **Speichern**.

Konfigurieren der My VMware-Integration in vRealize Automation Cloud Assembly

Sie können My VMware in vRealize Automation Cloud Assembly integrieren, um VMware-bezogene Aktionen und Funktionen, wie z. B. den Zugriff auf den VMware Marketplace für Blueprints, zu unterstützen.

Sie können für jede Organisation nur eine My VMware-Integration erstellen.

Voraussetzungen

Sie müssen über ein Benutzerkonto mit den entsprechenden Berechtigungen für My VMware verfügen.

- Informationen zum Einladen eines Benutzers zu einem My VMware-Konto finden Sie unter [KB 2070555](#).

- Informationen zum Zuweisen von Benutzerberechtigungen in einem My VMware-Konto finden Sie unter [KB 2006977](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus und klicken Sie auf **Integration hinzufügen**.
- 2 Wählen Sie „My VMware“ aus.
- 3 Geben Sie die erforderlichen Informationen auf der Konfigurationsseite von My VMware ein.
- 4 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags benötigen, geben Sie Funktions-Tags ein. Weitere Informationen finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#) und [Erstellen einer Tagging-Strategie](#).
- 5 Klicken Sie auf **Hinzufügen**.

Ergebnisse

My VMware steht für die Verwendung mit Blueprints zur Verfügung.

Nächste Schritte

Fügen Sie eine My VMware-Komponente zu den gewünschten Blueprints hinzu.

Konfigurieren der vRealize Orchestrator-Integration in Cloud Assembly

Sie können eine oder mehrere vRealize Orchestrator-Integrationen konfigurieren, um Workflows als Teil der Erweiterbarkeit zu verwenden.

vRealize Automation enthält eine vorkonfigurierte vRealize Orchestrator-Instanz, die für Erweiterbarkeitsabonnements verwendet werden kann. Sie können auch über die vRealize Automation Cloud Services-Konsole auf den Client des eingebetteten vRealize Orchestrator zugreifen.

Mit der vRealize Orchestrator-Integration in vRealize Automation Cloud Assembly können Sie eine externe vRealize Orchestrator-Instanz hinzufügen und die im Lieferumfang enthaltene Workflow-Bibliothek in Erweiterungsabonnements verwenden. Weitere Informationen finden Sie unter [Abonnements für Erweiterbarkeits-Workflows](#).

Voraussetzungen

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).

- Migrieren Sie Ihre vRealize Orchestrator 7.x-Instanz auf Version 8.0. Weitere Informationen finden Sie unter *Migrieren vom eigenständigen vRealize Orchestrator zum externen vRealize Orchestrator 8.0* in *Installieren, Konfigurieren und Migrieren von VMware vRealize Orchestrator*.

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus.
- 2 Klicken Sie auf **Integration hinzufügen**.
- 3 Wählen Sie vRealize Orchestrator aus.
- 4 Geben Sie in vRealize Automation Cloud Assembly die URL der vRealize Orchestrator-Instanz ein.
- 5 Klicken Sie zum Überprüfen der Integration auf **Validieren**.
- 6 Geben Sie einen Namen für die vRealize Orchestrator-Integration ein.
- 7 (Optional) Geben Sie eine Beschreibung für die vRealize Orchestrator-Integration ein.
- 8 (Optional) Fügen Sie Funktions-Tags hinzu. Weitere Informationen zu Funktions-Tags finden Sie unter [Verwenden von Funktions-Tags in vRealize Automation Cloud Assembly](#).

Hinweis Funktions-Tags können zum Verwalten mehrerer vRealize Orchestrator-Integrationen verwendet werden. Weitere Informationen hierzu finden Sie unter [Verwalten mehrerer vRealize Orchestrator-Integrationen mit Projekteinschränkungen](#).

- 9 Klicken Sie auf **Hinzufügen**.

Die vRealize Orchestrator-Integration wird gespeichert.

Nächste Schritte

Um zu überprüfen, ob die Integration konfiguriert ist und die Workflows hinzugefügt wurden, wählen Sie **Erweiterbarkeit > Bibliothek > Workflows** aus.

Verwalten mehrerer vRealize Orchestrator-Integrationen mit Projekteinschränkungen

Mithilfe von Projekteinschränkungen können Sie steuern, welche vRealize Orchestrator-Integrationen in Workflow-Abonnements verwendet werden.

vRealize Automation Cloud Assembly unterstützt die Integration mehrerer vRealize Orchestrator-Server, die in Workflow-Abonnements verwendet werden können. Sie können steuern, welche vRealize Orchestrator-Integrationen in von Ihrem Projekt bereitgestellten Blueprints mit flexiblen oder strengen Projekteinschränkungen verwendet werden. Weitere Informationen zu Projekteinschränkungen finden Sie unter [Verwenden von vRealize Automation Cloud Assembly-Projekt-Tags und benutzerdefinierten Eigenschaften](#).

Voraussetzungen

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Konfigurieren Sie mindestens zwei vRealize Orchestrator-Integrationen in vRealize Automation Cloud Assembly. Weitere Informationen finden Sie unter [Konfigurieren der vRealize Orchestrator-Integration in Cloud Assembly](#).
- Fügen Sie Ihren vRealize Orchestrator-Integrationen Funktions-Tags hinzu. Weitere Informationen zu Funktions-Tags finden Sie unter [Verwenden von Funktions-Tags in vRealize Automation Cloud Assembly](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Konfigurieren > Projekte** und wählen Sie Ihr Projekt aus.
- 2 Wählen Sie die Registerkarte **Bereitstellung** aus.
- 3 Geben Sie die Funktions-Tags Ihrer vRealize Orchestrator-Integrationen in das Textfeld **Erweiterbarkeitseinschränkungen** ein und legen Sie sie als flexible oder strenge Projekteinschränkungen fest.
- 4 Klicken Sie auf **Speichern**.

Ergebnisse

Wenn Sie einen Blueprint bereitstellen, steuert vRealize Automation Cloud Assembly anhand der Projekteinschränkungen, welche vRealize Orchestrator-Integrationen in Workflow-Abonnements verwendet werden.

Nächste Schritte

Alternativ können Sie Funktions-Tags verwenden, um mehrere vRealize Orchestrator-Integrationen auf der Ebene eines Cloud-Kontos zu verwalten. Weitere Informationen finden Sie unter [Verwalten mehrerer vRealize Orchestrator-Integrationen mit Cloud-Konto-Funktions-Tags](#).

Verwalten mehrerer vRealize Orchestrator-Integrationen mit Cloud-Konto-Funktions-Tags

Mithilfe von Funktions-Tags können Sie steuern, welche vRealize Orchestrator-Integrationen in Workflow-Abonnements verwendet werden.

vRealize Automation Cloud Assembly unterstützt die Integration mehrerer vRealize Orchestrator-Server, die in Workflow-Abonnements verwendet werden können. Durch Hinzufügen von Funktions-Tags zu Ihrem Cloud-Konto können Sie steuern, welche vRealize Orchestrator-Integrationen in Workflow-Abonnements verwendet werden.

Voraussetzungen

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Konfigurieren Sie mindestens zwei vRealize Orchestrator-Integrationen in vRealize Automation Cloud Assembly. Weitere Informationen finden Sie unter [Konfigurieren der vRealize Orchestrator-Integration in Cloud Assembly](#).
- Fügen Sie Ihren vRealize Orchestrator-Integrationen Funktions-Tags hinzu. Weitere Informationen zu Funktions-Tags finden Sie unter [Verwenden von Funktions-Tags in vRealize Automation Cloud Assembly](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Verbindungen > Cloud-Konten**.
- 2 Wählen Sie Ihr Cloud-Konto aus.
- 3 Geben Sie die Funktions-Tags der vRealize Orchestrator-Integrationen ein, die Sie verwenden möchten.

Die Funktions-Tags werden automatisch in flexible Einschränkungen umgewandelt. Um strenge Einschränkungen bei der Verwaltung Ihrer Integrationen zu verwenden, müssen Sie Projekteinschränkungen verwenden. Weitere Informationen finden Sie unter [Verwalten mehrerer vRealize Orchestrator-Integrationen mit Projekteinschränkungen](#).

- 4 Klicken Sie auf **Speichern**.

Ergebnisse

Wenn Sie einen Blueprint bereitstellen, verwendet vRealize Automation Cloud Assembly das Tagging im zugeordneten Cloud-Konto, um zu steuern, welche vRealize Orchestrator-Integrationen in Workflow-Abonnements verwendet werden.

Vorgehensweise zum Arbeiten mit Kubernetes in vRealize Automation Cloud Assembly

Sie können den Pivotal Container Service (PKS) oder Red Hat OpenShift in vRealize Automation Cloud Assembly integrieren, um Kubernetes-Ressourcen zu verwalten und bereitzustellen. Sie können externe Kubernetes-Ressourcen auch in vRealize Automation Cloud Assembly konfigurieren.

Nachdem Sie eine PKS- oder OpenShift-Integration erstellt haben, werden entsprechende Kubernetes-Cluster in vRealize Automation Cloud Assembly verfügbar. Sie können Kubernetes-Komponenten erstellen und zu vRealize Automation Cloud Assembly hinzufügen, um die Verwaltung von Cluster- und Containeranwendungen zu unterstützen. Diese Anwendungen bilden die Basis für Self-Service-Bereitstellungen, die im Service Broker-Katalog verfügbar sind.

- [Konfigurieren der PKS-Integration in vRealize Automation Cloud Assembly](#)

Sie können eine PKS-Ressourcenverbindung lokal und in der Cloud konfigurieren, um Integrations- und Verwaltungsfunktionen von Kubernetes in vRealize Automation Cloud Assembly zu unterstützen.

- [Arbeiten mit Kubernetes-Clustern und -Namespaces in vRealize Automation Cloud Assembly](#)

Sie können die Konfiguration von Kubernetes-Clustern und -Namespaces, die als Basis für Kubernetes-Bereitstellungen in vRealize Automation Cloud Assembly dienen, hinzufügen, anzeigen und verwalten.

- [Konfigurieren einer Kubernetes-Zone in vRealize Automation Cloud Assembly](#)

Mit Kubernetes-Zonen können Cloud-Administratoren die richtlinienbasierte Platzierung von Kubernetes-Clustern und -Namespaces definieren, die in vRealize Automation Cloud Assembly-Bereitstellungen verwendet werden. Ein Administrator kann diese Seite verwenden, um anzugeben, welche Cluster für die Bereitstellung von Kubernetes-Namespaces verfügbar sind, und zusätzlich, welche Eigenschaften für Cluster akzeptabel sind.

- [Hinzufügen von Kubernetes-Komponenten zu Blueprints in vRealize Automation Cloud Assembly](#)

Beim Hinzufügen von Kubernetes-Komponenten zu einem vRealize Automation Cloud Assembly-Blueprint können Sie Cluster hinzufügen oder Benutzern das Erstellen von Namespaces in verschiedenen Konfigurationen ermöglichen. Diese Auswahl hängt in der Regel von Ihren Anforderungen an die Zugriffssteuerung, von der Konfiguration Ihrer Kubernetes-Komponenten und von Ihren Bereitstellungsanforderungen ab.

- [Verwenden der Erweiterbarkeit von vRealize Automation Cloud Assembly mit Kubernetes](#)

vRealize Automation Cloud Assembly bietet einen Standardsatz von Ereignisthemen, die typischen Aktionen im Zusammenhang mit der Bereitstellung von Kubernetes-Clustern entsprechen. Benutzer können diese Themen nach Bedarf abonnieren und erhalten eine Benachrichtigung, wenn das Ereignis im Zusammenhang mit dem abonnierten Thema auftritt. Sie können vRO-Workflows auch so konfigurieren, dass sie auf der Grundlage von Ereignisbenachrichtigungen ausgeführt werden.

Konfigurieren der PKS-Integration in vRealize Automation Cloud Assembly

Sie können eine PKS-Ressourcenverbindung lokal und in der Cloud konfigurieren, um Integrations- und Verwaltungsfunktionen von Kubernetes in vRealize Automation Cloud Assembly zu unterstützen.

Mit den PKS-Integrationen können Sie sowohl PKS-Instanzen lokal und in der Cloud als auch auf PKS bereitgestellte Kubernetes-Cluster und externe Cluster verwalten. Sie müssen ein Kubernetes-Profil erstellen und es mit einem Projekt verknüpfen, um die richtlinienbasierte Platzierung von Ressourcen zu unterstützen.

Voraussetzungen

- Sie müssen über einen entsprechend konfigurierten PKS-Server (Pivotal Container Service) verfügen, der mit UAA-Authentifizierung eingerichtet ist.
- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus und klicken Sie auf **Integration hinzufügen**.
- 2 Wählen Sie VMware Enterprise PKS aus.
- 3 Geben Sie die IP-Adresse oder den FQDN und die PKS-Adresse für das PKS-Cloud-Konto ein, das Sie erstellen.
 - Die IP-Adresse ist der FQDN oder die IP-Adresse des PKS-Benutzerauthentifizierungsservers.
 - Die PKS-Adresse ist der FQDN oder die IP-Adresse für den Haupt-PKS-Server.
- 4 Wählen Sie aus, ob dieser PKS-Server lokal ist oder sich in der Public Cloud oder in einer Private Cloud befindet.
- 5 Geben Sie einen entsprechenden **Benutzernamen** und ein **Kennwort** für den PKS-Server und weitere zugehörige Informationen ein.
- 6 Wenn Sie zur Unterstützung einer Tagging-Strategie Tags verwenden, geben Sie Funktions-Tags ein. Weitere Informationen finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#) und [Erstellen einer Tagging-Strategie](#).
- 7 Klicken Sie auf **Hinzufügen**.

Ergebnisse

Sie können neue Kubernetes-Zonen erstellen und einem Projekt zuweisen. Alternativ können Sie externe Kubernetes-Cluster erkennen und diese Projekten zuweisen. Darüber hinaus können Sie Kubernetes-Namespace hinzufügen oder erstellen, die die Verwaltung von Clustern zwischen großen Gruppen und Organisationen vereinfachen.

Nächste Schritte

Erstellen oder wählen Sie die entsprechenden Kubernetes-Zonen aus. Wählen Sie dann einen oder mehrere Cluster oder Namespaces aus und weisen Sie sie einem Projekt zu. Anschließend können Sie Blueprints erstellen und veröffentlichen, um Benutzern die Erstellung von Self-Service-Bereitstellungen zu ermöglichen, die Kubernetes verwenden.

Arbeiten mit Kubernetes-Clustern und -Namespaces in vRealize Automation Cloud Assembly

Sie können die Konfiguration von Kubernetes-Clustern und -Namespaces, die als Basis für Kubernetes-Bereitstellungen in vRealize Automation Cloud Assembly dienen, hinzufügen, anzeigen und verwalten.

Sie können Kubernetes-Cluster und -Namespaces anzeigen, hinzufügen und verwalten, auf die Sie auf der Seite **Infrastruktur > Ressourcen > Kubernetes** zugreifen können. In der Regel erleichtert diese Seite die Verwaltung von bereitgestellten Clustern und Namespaces.

- **Cluster:** Ein Cluster ist eine Gruppe von Kubernetes-Knoten, die über eine oder mehrere physische Maschinen verteilt sind. Auf dieser Seite werden bereitgestellte und nicht bereitgestellte Cluster angezeigt, die für die Verwendung in Ihrer vRealize Automation Cloud Assembly-Instanz konfiguriert wurden. Sie können auf einen Cluster klicken, um Informationen über den aktuellen Status anzuzeigen. Wenn Sie einen Cluster bereitstellen, enthält er einen Link zu einer Kubconfig-Datei, auf die nur Cloud-Administratoren zugreifen können. Diese Datei gewährt vollständige Administratorrechte für den Cluster, einschließlich einer Liste von Namespaces.
- **Namespaces:** Namespaces sind virtuelle Cluster, die Administratoren die Möglichkeit bieten, Clusterressourcen aufzuteilen. Sie erleichtern die Verwaltung von Ressourcen in großen Gruppen von Benutzern und Organisationen. Als eine Form der rollenbasierten Zugriffssteuerung kann ein Cloud-Administrator Benutzern ermöglichen, Namespaces zu einem Projekt hinzuzufügen, wenn sie eine Bereitstellung anfordern, und diese dann später auf der Seite „Kubernetes-Cluster“ zu verwalten. Wenn Sie einen Namespace bereitstellen, enthält er einen Link zu einer Kubconfig-Datei, mit der gültige Benutzer, wie z. B. Entwickler, einige Aspekte dieses Namespaces anzeigen und verwalten können.

Wenn Sie einen neuen oder vorhandenen Cluster konfigurieren, müssen Sie auswählen, ob eine Verbindung mit einer Master-IP-Adresse oder einem Master-Hostnamen hergestellt werden soll.

Arbeiten mit Kubernetes-Clustern in vRealize Automation Cloud Assembly

Sie können mithilfe der Optionen auf dieser Seite neue, vorhandene oder externe Cluster zu vRealize Automation Cloud Assembly hinzufügen.

- 1 Wählen Sie **Infrastruktur > Ressourcen > Kubernetes** aus und bestätigen Sie, dass die Registerkarte „Cluster“ aktiv ist.

Wenn derzeit Cluster für Ihre vRealize Automation Cloud Assembly-Instanz konfiguriert sind, werden diese auf dieser Seite angezeigt.

- 2 Wenn Sie einen neuen oder vorhandenen Cluster hinzufügen oder einen Cluster bereitstellen, wählen Sie die entsprechende Option gemäß der folgenden Tabelle aus.

Option	Beschreibung	Details
Bereitstellen	Neue Cluster zu vRealize Automation Cloud Assembly hinzufügen	Sie müssen das PKS Cloud-Konto angeben, dem dieser Cluster bereitgestellt werden soll, sowie den gewünschten Plan und die Anzahl der Knoten.
Vorhandene hinzufügen	Konfigurieren Sie einen vorhandenen Cluster für Ihr Projekt.	Sie müssen das PKS Cloud-Konto, den zu verwendenden Cluster und das entsprechende Projekt für den jeweiligen Entwickler angeben. Außerdem müssen Sie den Freigabebereich angeben. Wenn Sie global freigeben möchten, müssen Sie Ihre Kubernetes-Zonen und -Namespaces entsprechend konfigurieren.
Externe hinzufügen	Fügen Sie vRealize Automation Cloud Assembly einen Vanilla-Kubernetes-Cluster hinzu, der möglicherweise nicht mit PKS verknüpft ist.	Sie müssen ein Projekt benennen, dem der Cluster zugeordnet ist, die IP-Adresse für den gewünschten Cluster eingeben und einen Cloud-Proxy und die erforderlichen Zertifikatinformationen eingeben, die zum Herstellen einer Verbindung mit diesem Cluster erforderlich sind.

- 3 Klicken Sie auf **Hinzufügen**, um den Cluster innerhalb von vRealize Automation Cloud Assembly verfügbar zu machen.

Arbeiten mit Kubernetes-Namespaces in vRealize Automation Cloud Assembly

Wenn Sie ein Cloud-Administrator sind, helfen Ihnen Namespaces dabei, Kubernetes-Clusterressourcen zu gruppieren und zu verwalten. Wenn Sie ein Benutzer sind, sind Namespaces der Bereich in Kubernetes-Clustern für Ihre Bereitstellungen. Administratoren und Benutzer können auf der Registerkarte „Namespaces“ auf der Seite **Infrastruktur > Ressourcen > Kubernetes** auf Namespaces zugreifen.

gibt es mehrere Möglichkeiten, Kubernetes-Namespaces zu Ressourcen in vRealize Automation Cloud Assembly hinzuzufügen. Im folgenden Verfahren wird eine typische Methode beschrieben.

- 1 Wählen Sie **Infrastruktur > Ressourcen > Kubernetes** aus und klicken Sie auf die Registerkarte „Namespaces“.
- 2 Um einen neuen Namespace hinzuzufügen, klicken Sie auf **Neuer Namespace**. Um einen vorhandenen Namespace hinzuzufügen, klicken Sie auf **Namespace hinzufügen**.
- 3 Geben Sie einen **Namen** und eine **Beschreibung** für den Namespace ein.
An dieser Stelle haben Sie einen Namespace für die Verwendung mit Kubernetes-Ressourcen hinzugefügt, aber er ist mit nichts verknüpft.
- 4 Geben Sie den **Cluster** an, den Sie diesem Namespace zuordnen möchten.
- 5 Klicken Sie auf **Erstellen**, um den Namespace zu vRealize Automation Cloud Assembly hinzuzufügen.

Konfigurieren einer Kubernetes-Zone in vRealize Automation Cloud Assembly

Mit Kubernetes-Zonen können Cloud-Administratoren die richtlinienbasierte Platzierung von Kubernetes-Clustern und -Namespaces definieren, die in vRealize Automation Cloud Assembly-Bereitstellungen verwendet werden. Ein Administrator kann diese Seite verwenden, um anzugeben, welche Cluster für die Bereitstellung von Kubernetes-Namespaces verfügbar sind, und zusätzlich, welche Eigenschaften für Cluster akzeptabel sind.

Cloud-Administratoren können Kubernetes-Zonen mit PKS-Cloud-Konten verknüpfen, die für Cloud Assembly konfiguriert wurden, oder mit externen Kubernetes-Clustern, die keinem Projekt zugeordnet sind.

Wenn Sie eine Kubernetes-Zone erstellen, können Sie der Zone mehrere anbieterspezifische Ressourcen zuweisen. Diese Ressourcen geben vor, welche Eigenschaften für die neu bereitgestellten Cluster in Bezug auf die Anzahl der Worker, Masters, verfügbaren CPU, Arbeitsspeicher und andere Konfigurationseinstellungen festgelegt werden können. Für PKS-Anbieter entsprechen diese den PKS-Plänen. Ein Administrator kann auch mehrere Cluster zu einer Kubernetes-Zone zuweisen, die für die Platzierung von neu bereitgestellten Kubernetes-Namespaces verwendet wird. Der Administrator kann nur Cluster zuweisen, die nicht integriert sind oder nicht von CMX verwaltet werden und die über den vorgewählten Cluster-Anbieter bereitgestellt werden. Der Administrator kann einem einzelnen Projekt mehrere Kubernetes-Zonen zuweisen, sodass sie alle für Platzierungsvorgänge verfügbar sind, die in diesem Projekt durchgeführt werden.

Ein Cloud-Administrator kann Prioritäten auf mehreren Ebenen zuweisen:

- Kubernetes-Zonenpriorität innerhalb eines Projekts.
- Ressourcenpriorität innerhalb einer Kubernetes-Zone.
- Cluster-Priorität innerhalb einer Kubernetes-Zone.

Der Cloud-Administrator kann auch Tags auf mehreren Ebenen zuweisen:

- Funktions-Tags pro Kubernetes-Zone.
- Tags pro Ressourcenzuweisung.
- Tags pro Cluster-Zuweisung.

Service Broker enthält eine Version der Seite „Kubernetes-Zone“, um Service Broker-Administratoren Zugriff auf vorhandene Kubernetes-Zonen zu ermöglichen, damit sie Platzierungsrichtlinien für Kubernetes-Namespaces und -Cluster erstellen können, die über den Katalog bereitgestellt werden.

Voraussetzungen

Konfigurieren Sie die Integration mit einer geeigneten PKS-Bereitstellung. Weitere Informationen finden Sie unter [Konfigurieren der PKS-Integration in vRealize Automation Cloud Assembly](#)

Verfahren

- 1 Wählen Sie **Infrastruktur > Konfigurieren > Kubernetes-Zone** aus und klicken Sie dann auf **Neue Kubernetes-Zone**.
- 2 Geben Sie den Namen des **Kontos** der PKS-Integration an, für das diese Zone gelten soll.
- 3 Fügen Sie einen **Namen** und eine **Beschreibung** für die Kubernetes-Zone hinzu.
- 4 Fügen Sie bei Bedarf Funktions-Tags hinzu. Weitere Informationen hierzu finden Sie unter [Verwenden von Funktions-Tags in vRealize Automation Cloud Assembly](#).
- 5 Klicken Sie auf **Speichern**.

- 6 Klicken Sie auf die Registerkarte „Bedarfsgesteuert“ und fügen Sie der Zone entsprechende PKS-Pläne hinzu, die für die Cluster-Bereitstellung verwendet werden sollen.

Sie können einen oder mehrere Pläne auswählen und ihnen Prioritäten zuweisen. Niedrigere Zahlen entsprechen höherer Priorität. Prioritätszuweisungen sind für die Tag-basierte Auswahl sekundär.

- 7 Klicken Sie auf die Registerkarte „Cluster“ und anschließend auf die Schaltfläche **Hinzufügen**, um der Zone Kubernetes-Cluster hinzuzufügen. Wenn Sie mit einem externen Cluster arbeiten, wird er automatisch in vRealize Automation Cloud Assembly integriert, wenn Sie ihn auswählen.

Sie können Kubernetes-Namespace auf der Seite „Kubernetes-Cluster“ in vRealize Automation Cloud Assembly zum Cluster hinzufügen.

Ergebnisse

Kubernetes-Zonen sind für die Verwendung mit vRealize Automation Cloud Assembly-Bereitstellungen konfiguriert.

Nächste Schritte

Weisen Sie die Kubernetes-Zone einem Projekt zu.

- 1 Wählen Sie **Infrastruktur > Konfigurieren > Projekte** aus und wählen Sie dann das Projekt aus, das Sie Ihrer Kubernetes-Zone zuordnen möchten.
- 2 Klicken Sie auf der Seite „Projekt“ auf die Registerkarte „Kubernetes-Bereitstellung“.
- 3 Klicken Sie auf **Kubernetes Zone hinzufügen** und fügen Sie die soeben erstellte Zone hinzu. Sie können bei Bedarf mehrere Zonen hinzufügen und auch die Priorität für die Zonen festlegen.
- 4 Klicken Sie auf **Speichern**.

Nachdem Sie einem Projekt eine Zone zugewiesen haben, können Sie die Seite „Blueprints“ verwenden, um eine Bereitstellung basierend auf der Kubernetes-Zone und der Projektkonfiguration bereitzustellen.

Hinzufügen von Kubernetes-Komponenten zu Blueprints in vRealize Automation Cloud Assembly

Beim Hinzufügen von Kubernetes-Komponenten zu einem vRealize Automation Cloud Assembly-Blueprint können Sie Cluster hinzufügen oder Benutzern das Erstellen von Namespaces in verschiedenen Konfigurationen ermöglichen. Diese Auswahl hängt in der Regel von Ihren Anforderungen an die Zugriffssteuerung, von der Konfiguration Ihrer Kubernetes-Komponenten und von Ihren Bereitstellungsanforderungen ab.

Um eine Kubernetes-Komponente zu einem Blueprint in vRealize Automation Cloud Assembly hinzuzufügen, klicken Sie auf „Blueprints“, wählen Sie **Neu** aus und suchen Sie dann im linken Menü die Option „Kubernetes“. Erweitern Sie die Option. Nehmen Sie dann die gewünschte Auswahl vor, entweder „Cluster“ oder „KBS-Namespace“, indem Sie sie auf die Arbeitsfläche ziehen.

Das Hinzufügen eines Kubernetes-Clusters, der einem Projekt zugeordnet ist, zu einem Blueprint ist die einfachste Methode, Kubernetes-Ressourcen für gültige Benutzer verfügbar zu machen. Sie können Tags in Clustern verwenden, um zu steuern, wo diese bereitgestellt werden, genau wie bei anderen Cloud Assembly-Ressourcen. Sie können mithilfe von Tags eine Zone und einen PKS-Plan während der Zuteilungsphase der Cluster-Bereitstellung auswählen.

Sobald Sie einen Cluster auf diese Weise hinzugefügt haben, steht er automatisch allen gültigen Benutzern zur Verfügung.

Blueprint-Beispiele

Das erste Blueprint-Beispiel zeigt einen Blueprint für eine einfache Kubernetes-Bereitstellung, die durch Tagging gesteuert wird. Eine Kubernetes-Zone wurde mit zwei Bereitstellungsplänen erstellt und auf der Seite „Neue Kubernetes-Zone“ konfiguriert. In diesem Fall wurde ein Tag mit dem Namen `placement:tag` als Funktion für die Zone hinzugefügt. Es wurde verwendet, um der analogen Einschränkung auf dem Blueprint zu entsprechen. Wenn mehr als eine Zone mit dem Tag konfiguriert ist, wird diejenige mit der niedrigsten Prioritätsnummer ausgewählt.

```
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
```

Im zweiten Blueprint-Beispiel wird gezeigt, wie ein Blueprint mit einer Variablen namens *\$ (input.hostname)* eingerichtet wird, damit Benutzer beim Anfordern einer Bereitstellung den gewünschten Cluster-Hostnamen eingeben können. Tags können auch verwendet werden, um eine Zone und einen PKS-Plan während der Ressourcenzuteilungsphase der Clusterbereitstellung auszuwählen.

```
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
```

Wenn Sie Namespaces verwenden möchten, um die Cluster-Nutzung zu verwalten, können Sie eine Variable im Blueprint namens *name: \${input.name}* einrichten, um den Namespace-Namen zu ersetzen, den ein Benutzer beim Anfordern einer Bereitstellung eingibt. Für diese Art der Bereitstellung erstellen Sie einen Blueprint ähnlich dem folgenden Beispiel:

```
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}
```

Benutzer können bereitgestellte Cluster über kubeconfig-Dateien verwalten, auf die von der Seite **Infrastruktur > Ressourcen > Kubernetes-Cluster** zugegriffen werden kann. Suchen Sie die Karte auf der Seite für den gewünschten Cluster und klicken Sie auf **Kubeconfig**.

Verwenden der Erweiterbarkeit von vRealize Automation Cloud Assembly mit Kubernetes

vRealize Automation Cloud Assembly bietet einen Standardsatz von Ereignisthemen, die typischen Aktionen im Zusammenhang mit der Bereitstellung von Kubernetes-Clustern entsprechen. Benutzer können diese Themen nach Bedarf abonnieren und erhalten eine Benachrichtigung, wenn das Ereignis im Zusammenhang mit dem abonnierten Thema auftritt. Sie können vRO-Workflows auch so konfigurieren, dass sie auf der Grundlage von Ereignisbenachrichtigungen ausgeführt werden.

Die folgenden Themen sind zum Abonnieren auf der Seite **Erweiterbarkeit > Bibliothek > Ereignisthemen** in vRealize Automation Cloud Assembly verfügbar. Um diese Themen anzuzeigen, suchen Sie im Textfeld „Ereignisthemen suchen“ nach Kubernetes.

- Kubernetes-Cluster-Zuteilung
- Kubernetes-Cluster nach Bereitstellung
- Kubernetes-Cluster nach Entfernung
- Kubernetes-Cluster-Bereitstellung
- Kubernetes-Cluster-Entfernung

Klicken Sie auf eines der Themen, um das Schema für dieses Thema anzuzeigen, das alle erfassten und übermittelten Informationen anzeigt. Sie können alle dieser Schemainformationen verwenden, um verschiedene Benachrichtigungen und Verwaltungs- sowie Berichterstellungsaufgaben einzurichten.

Sie können Aktionsskripts für CMX-bezogene Aktionen auf der Seite **Erweiterbarkeit > Bibliothek > Aktionen** einrichten. Aktionsskripts können für verschiedene Zwecke verwendet werden: z. B. zum Erstellen eines DNS-Eintrags für die Kubernetes-Cluster-Bereitstellung. Wenn Sie einen DNS-Datensatz erstellen, können Sie das Feld `masternodeips` aus dem Thema „Kubernetes-Cluster nach Bereitstellung“ mit einem Rest-Befehl in einem Aktionsskript zum Erstellen eines DNS-Datensatzes verwenden.

Auf der Seite „Abonnements“ wird die Beziehung zwischen den Ereignisthemen und Aktionsskripten definiert. Sie können diese Komponenten auf der Seite „Abonnements“ in vRealize Automation Cloud Assembly anzeigen und verwalten.

Definition der Konfigurationsverwaltung in vRealize Automation Cloud Assembly

vRealize Automation Cloud Assembly unterstützt die Integration mit Puppet Enterprise und Ansible Open Source, wodurch Bereitstellungen für Konfigurationsabweichungen verwaltet werden können.

Puppet-Integration

Zur Integration der Puppet-basierten Konfigurationsverwaltung muss eine gültige Instanz von Puppet Enterprise in einer Public oder Private Cloud mit einer vSphere-Arbeitslast installiert sein. Sie müssen eine Verbindung zwischen diesem externen System und der vRealize Automation Cloud Assembly-Instanz erstellen. Anschließend können Sie die Puppet-Konfigurationsverwaltung in vRealize Automation Cloud Assembly zur Verfügung stellen, indem Sie sie zu entsprechenden Blueprints hinzufügen.

Der Puppet-Anbieter des vRealize Automation Cloud Assembly-Blueprint-Diensts installiert, konfiguriert und führt den Puppet-Agent in einer bereitgestellten Computing-Ressource aus. Der Puppet-Anbieter unterstützt sowohl SSH- als auch WinRM-Verbindungen mit den folgenden Voraussetzungen:

- SSH-Verbindungen:
 - Zur Ausführung von Befehlen mit `NOPASSWD` muss als Benutzername ein Superuser oder ein Benutzer mit `sudo`-Berechtigungen angegeben werden.
 - Deaktivieren Sie `requiretty` für den angegebenen Benutzer.
 - `cURL` muss in der Computing-Ressource der Bereitstellung verfügbar sein.
- WinRM-Verbindungen:
 - PowerShell 2.0 muss in der Computing-Ressource der Bereitstellung verfügbar sein.
 - Konfigurieren Sie die Windows-Vorlage gemäß der Beschreibung in der vRealize Orchestrator-Dokumentation.

Der DevOps-Administrator ist für die Verwaltung der Verbindungen mit einem Puppet Master und für die Anwendung von Puppet-Rollen oder Konfigurationsregeln auf bestimmte Bereitstellungen zuständig. Nach der Bereitstellung werden virtuelle Maschinen, die zur Unterstützung der Konfigurationsverwaltung konfiguriert wurden, beim zugewiesenen Puppet-Master registriert.

Wenn virtuelle Maschinen bereitgestellt werden, können Benutzer einen Puppet-Master als externes System hinzufügen bzw. löschen oder Projekte aktualisieren, die dem Puppet-Master zugewiesen sind. Schließlich können entsprechende Benutzer die Registrierung bereitgestellter virtueller Maschinen mithilfe des Puppet-Masters aufheben, wenn die Maschinen außer Betrieb genommen werden.

Ansible Open Source-Integration

Beim Einrichten einer Ansible-Integration installieren Sie Ansible Open Source gemäß den Installationsanweisungen für Ansible. Weitere Informationen zur Installation finden Sie in der Ansible-Dokumentation.

Ansible aktiviert standardmäßig die Überprüfung von Host-Schlüsseln. Wenn ein Host mit einem anderen Schlüssel in der `known_hosts`-Datei neu installiert wird, wird eine Fehlermeldung angezeigt. Wenn ein Host nicht in der `known_hosts`-Datei aufgeführt ist, müssen Sie den Schlüssel beim Start angeben. Mit der folgenden Einstellung in der Datei `/etc/ansible/ansible.cfg` oder `~/.ansible.cfg` können Sie die Hostschlüsselüberprüfung deaktivieren:

```
[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
```

Um Fehler bei der Überprüfung der Hostschlüssel zu vermeiden, legen Sie `host_key_checking` und `record_host_keys` auf „False“ fest und fügen eine zusätzliche Option `UserKnownHostsFile=/dev/null` hinzu, die in `ssh_args` festgelegt wird. Wenn die Bestandsliste anfänglich leer ist, warnt Ansible außerdem, dass die Hostliste leer ist. Dies führt dazu, dass die Überprüfung der Playbook-Syntax fehlschlägt.

Im Ansible-Tresor können Sie vertrauliche Informationen, wie z. B. Kennwörter oder Schlüssel, in verschlüsselten Dateien statt in Form von Klartext speichern. Der Tresor ist mit einem Kennwort verschlüsselt. In vRealize Automation Cloud Assembly werden Daten von Ansible wie SSH-Kennwörter für Host-Maschinen im Tresor verschlüsselt. Ansible setzt dabei voraus, dass der Pfad zum Tresor-Kennwort festgelegt wurde.

Sie können die Datei `ansible.cfg` ändern, um den Speicherort der Kennwortdatei anzugeben. Verwenden Sie folgendes Format.

```
vault_password_file = /path to/file.txt
```

Außerdem können Sie die Umgebungsvariable `ANSIBLE_VAULT_PASSWORD_FILE` so festlegen, dass Ansible automatisch nach dem Kennwort sucht. Beispiel:

```
ANSIBLE_VAULT_PASSWORD_FILE=~/.vault_pass.txt.
```

Da die Ansible-Bestandslistendatei von vRealize Automation Cloud Assembly verwaltet wird, müssen Sie sicherstellen, dass der vRealize Automation Cloud Assembly-Benutzer über `rwX`-Zugriff auf die Bestandslistendatei verfügt.

```
cat ~/var/tmp/vmware/provider/user_defined_script/${ls -t ~/var/tmp/vmware/provider/
user_defined_script/ | head -1}/log.txt
```

Wenn Sie einen Nicht-Root-Benutzer mit vRealize Automation Cloud Assembly-Open Source-Integration verwenden möchten, benötigen die Benutzer eine Reihe von Berechtigungen zum Ausführen der Befehle, die vom Open Source-Anbieter von vRealize Automation Cloud Assembly verwendet werden. Die folgenden Befehle müssen in der Sudoers-Datei des Benutzers festgelegt werden.

```
Defaults:myuser !requiretty
```

Wenn der Benutzer nicht zu einer Admin-Gruppe gehört, für die keine `askpass`-Anwendung angegeben ist, legen Sie den folgenden Befehl in der Sudoers-Datei des Benutzers fest:

```
myuser ALL=(ALL) NOPASSWD: ALL
```

Wenn Sie beim Einrichten der Ansible-Integration auf Fehler oder andere Probleme stoßen, finden Sie weitere Informationen in der `log.txt`-Datei auf der Ansible-Steuerungsmaschine. Diese Datei befindet sich im Verzeichnis

```
'cat~/var/tmp/vmware/provider/user_defined_script/$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ | head -1)/'.
```

Konfigurieren der Puppet Enterprise-Integration in vRealize Automation Cloud Assembly

vRealize Automation Cloud Assembly unterstützt die Integration mit der Puppet Enterprise-Konfigurationsverwaltung.

Wenn Sie Puppet Enterprise als externes System zu Cloud Assembly hinzufügen, steht es standardmäßig in allen Projekten zur Verfügung. Sie können Puppet auf bestimmte Projekte beschränken.

Zum Hinzufügen einer Puppet Enterprise-Integration müssen Sie über den Namen des Puppet-Masters und den Hostnamen oder die IP-Adresse des Masters verfügen.

Sie finden Puppet-Protokolle am folgenden Speicherort, falls Sie sie auf Fehler überprüfen oder zu Informationszwecken ansehen möchten.

Beschreibung	Speicherort des Protokolls
Protokoll zum Erstellen und Installieren von verwandten Ereignissen	Die Protokolle befinden sich auf der bereitgestellten Maschine im Pfad <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ head -1)/</code> . Die vollständigen Protokolle finden Sie in der Datei log.txt . Detaillierte Informationen zu den Puppet-Agentprotokollen finden Sie unter https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging .
Protokoll für Aufgaben in Bezug auf das Löschen und Ausführen von Puppet	Die Protokolle befinden sich auf der PE unter dem Pfad <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/ head -1)/</code> . Die vollständigen Protokolle finden Sie in der Datei log.txt .

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus und klicken Sie auf **Integration hinzufügen**.
- 2 Wählen Sie „Puppet“ aus.
- 3 Geben Sie die erforderlichen Informationen auf der Konfigurationsseite von Puppet ein.
- 4 Klicken Sie auf **Validieren**, um die Integration zu überprüfen.
- 5 Klicken Sie auf **Hinzufügen**.

Ergebnisse

Puppet steht für die Verwendung mit Blueprints zur Verfügung.

Nächste Schritte

Fügen Sie Puppet-Komponenten zu den gewünschten Blueprints hinzu.

- 1 Wählen Sie Puppet unter der Überschrift „Inhaltsverwaltung“ im Blueprint-Menü aus und ziehen Sie die Puppet-Komponente auf die Arbeitsfläche.
- 2 Geben Sie Puppet-Eigenschaften im Fensterbereich auf der rechten Seite ein.

Eigenschaft	Beschreibung
Master	Geben Sie den Namen der primären Puppet-Maschine ein, der mit diesem Blueprint verwendet werden soll.
Umgebung	Wählen Sie die Umgebung für die primäre Puppet-Maschine aus.
Rolle	Wählen Sie die Puppet-Rolle aus, die mit diesem Blueprint verwendet werden soll.
Agent-Ausführungsintervall	Die Häufigkeit, mit der der Puppet-Agent die primäre Puppet-Maschine bezüglich Konfigurationsdetails abfragen soll, die auf bereitgestellte und mit diesem Blueprint verknüpfte virtuelle Maschinen angewendet werden sollen.

- 3 Klicken Sie im rechten Fensterbereich auf die Registerkarte „Code“, um den YAML-Code für die Eigenschaften der Puppet-Konfiguration anzuzeigen.

Konfigurieren der Ansible Open Source-Integration in vRealize Automation Cloud Assembly

vRealize Automation Cloud Assembly unterstützt die Integration mit der Ansible Open Source-Konfigurationsverwaltung. Nachdem Sie die Integration konfiguriert haben, können Sie Ansible-Komponenten zu neuen oder vorhandenen Bereitstellungen hinzufügen.

Wenn Sie Ansible Open Source mit vRealize Automation Cloud Assembly integrieren, können Sie es so konfigurieren, dass ein oder mehrere Ansible-Playbooks in einer bestimmten Reihenfolge ausgeführt werden, wenn eine neue Maschine zur Automatisierung der Konfigurationsverwaltung bereitgestellt wird. Sie geben die gewünschten Playbooks im Blueprint für eine Bereitstellung an.

Beim Einrichten einer Ansible-Integration müssen Sie die Ansible Open Source-Hostmaschine sowie den Pfad der Bestandslistendatei angeben, in der Informationen für die Ressourcenverwaltung definiert sind. Darüber hinaus müssen Sie einen Namen und ein Kennwort für den Zugriff auf die Ansible Open Source-Instanz angeben. Wenn Sie eine Ansible-Komponente später zu einer Bereitstellung hinzufügen, können Sie die Verbindung aktualisieren, um die schlüsselbasierte Authentifizierung zu verwenden.

Standardmäßig verwendet Ansible SSH zum Herstellen einer Verbindung mit den physischen Maschinen. Falls Sie Windows-Maschinen wie im Blueprint mit der Windows-Eigenschaft „osType“ angegeben verwenden, wird die Variable „connection_type“ automatisch auf „winm“ festgelegt.

Die Ansible-Integration unterstützt physische Maschinen, die keine IP-Adresse verwenden. Bei Maschinen, die in Public Clouds wie AWS, Azure und GCP bereitgestellt werden, wird die Adresseigenschaft in der erstellten Ressource erst dann mit der öffentlichen IP-Adresse der Maschine versehen, wenn die Maschine mit einem öffentlichen Netzwerk verbunden ist. Für Maschinen, die nicht mit einem öffentlichen Netzwerk verbunden sind, sucht die Ansible-Integration nach der IP-Adresse im Netzwerk, das mit der Maschine verbunden ist. Wenn mehrere Netzwerke angeschlossen sind, sucht die Ansible-Integration nach dem Netzwerk mit dem niedrigsten deviceIndex, d. h., dem Index der Netzwerkschnittstellenkarte (NIC), die mit der Maschine verbunden ist. Wenn die Eigenschaft „deviceIndex“ nicht im Blueprint angegeben ist, verwendet die Integration das erste angeschlossene Netzwerk.

Unter [Definition der Konfigurationsverwaltung in vRealize Automation Cloud Assembly](#) erhalten Sie weitere Informationen zum Konfigurieren von Ansible Open Source für die Integration in vRealize Automation Cloud Assembly.

Voraussetzungen

- Die Ansible-Steuerungsmaschine muss Ansible Version 2.6.0 oder höher verwenden.
- Der Benutzer muss über Lese-/Schreibzugriff auf das Verzeichnis verfügen, in dem sich die Ansible-Bestandslistendatei befindet. Darüber hinaus muss der Benutzer über Lese-/Schreibzugriff auf die Bestandslistendatei verfügen, sofern sie bereits vorhanden ist.
- Stellen Sie bei Verwendung eines Nicht-Root-Benutzers mit der Option „sudo“ sicher, dass Folgendes in der Sudoers-Datei festgelegt ist:

```
Defaults:user_name !requiretty
```

und

```
username ALL=(ALL) NOPASSD: ALL
```

- Stellen Sie sicher, dass die Überprüfung des Hostschlüssels deaktiviert ist, indem Sie `host_key_checking = False` unter `/etc/ansible/ansible.cfg` oder `~/.ansible.cfg` festlegen.
- Stellen Sie sicher, dass das Tresor-Kennwort festgelegt ist, indem Sie der Datei `/etc/ansible/ansible.cfg` oder `~/.ansible.cfg` die folgende Zeile hinzufügen:

```
vault_password_file = /path/to/password_file
```

Die Tresor-Kennwortdatei enthält das Kennwort im Klartext und wird nur verwendet, wenn Blueprints oder Bereitstellungen eine Kombination aus Benutzername und Kennwort zur Verwendung zwischen ACM und dem Knoten bereitstellen. Siehe hierzu folgendes Beispiel.

```
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
```

- Um bei der Ausführung von Playbooks Ausfälle des Hostschlüssels zu vermeiden, empfiehlt sich die Aufnahme der folgenden Einstellungen in die Datei `/etc/ansible/ansible`.

```
[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.
```

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus und klicken Sie auf **Integration hinzufügen**.
- 2 Klicken Sie auf „Ansible“.
Die Seite zum Konfigurieren von Ansible wird angezeigt.
- 3 Geben Sie den Hostnamen, den Pfad der Bestandslistendatei und andere erforderliche Informationen für die Ansible Open Source-Instanz ein.
- 4 Klicken Sie auf **Validieren**, um die Integration zu überprüfen.
- 5 Klicken Sie auf **Hinzufügen**.

Ergebnisse

Ansible ist für die Verwendung mit Blueprints verfügbar.

Nächste Schritte

Fügen Sie Ansible-Komponenten zu den gewünschten Blueprints hinzu.

- 1 Wählen Sie auf der Seite mit der Blueprint-Arbeitsfläche im Menü der Blueprint-Optionen unter der Überschrift „Konfigurationsverwaltung“ den Eintrag „Ansible“ aus und ziehen Sie die Ansible-Komponente auf die Arbeitsfläche.
- 2 Konfigurieren Sie im Bereich auf der rechten Seite die entsprechenden Ansible-Eigenschaften. Geben Sie beispielsweise die auszuführenden Playbooks an.

In Ansible können Benutzer einem einzelnen Host eine Variable zuweisen und diese dann später in Playbooks verwenden. Mit der Ansible Open Source-Integration können Sie diese Host-Variable in Blueprints angeben. Die `hostVariables`-Eigenschaft muss im korrekten YAML-Format vorliegen, wie von der Ansible-Steuerungsmaschine erwartet, und diese Inhalte werden an folgendem Speicherort platziert:

```
parent_directory_of_inventory_file/host_vars/host_ip_address/vra_user_host_vars.yml
```

Der Standardspeicherort der Ansible-Bestandslistendatei wird im Ansible-Konto definiert, wie auf der Seite „Integrationen“ in Cloud Assembly hinzugefügt. Die Ansible-Integration validiert die YAML-Syntax für hostVariable im Blueprint nicht, die Ansible-Steuerungsmaschine löst eine Ausnahme aus, sobald Sie ein Playbook im Falle eines falschen Formats oder einer falschen Syntax ausführen.

Der folgende Blueprint-YAML-Ausschnitt zeigt ein Beispiel für die Nutzung der hostVariables-Eigenschaft.

```
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}
```

Vorgehensweise zum Erstellen einer Active Directory-Integration in vRealize Automation Cloud Assembly

vRealize Automation Cloud Assembly unterstützt die Integration mit Active Directory-Servern, um die sofortige Erstellung von Computerkonten in einer bestimmten Organisationseinheit (OE) innerhalb eines Active Directory-Servers vor der Bereitstellung einer virtuellen Maschine zu ermöglichen.

Die Active Directory-Integration unterstützt nur eine LDAP-Verbindung zum Active Directory-Server.

Voraussetzungen

- Wenn Sie eine Active Directory-Integration mit einem vCenter lokal konfigurieren, müssen Sie einen Erweiterbarkeits-Cloud-Proxy für die Active Directory-Integration konfigurieren. Wählen Sie **Erweiterbarkeit > Aktivität > Integrationen** und anschließend **Lokale Erweiterbarkeitsaktionen** aus.
- Wenn Sie eine Integration mit Active Directory in der Cloud konfigurieren, müssen Sie über ein Microsoft Azure- oder Amazon Web Services-Konto verfügen.
- Der Active Directory-Server muss eine LDAP-Serververbindung verwenden.

- Sie müssen über ein mit den entsprechenden Cloud-Zonen konfiguriertes Projekt sowie über Image- und Typzuordnungen für die Verwendung mit der Active Directory-Integration verfügen.
- Die gewünschte OE in Ihrem Active Directory muss vorab erstellt werden, bevor Sie Ihre Active Directory-Integration mit einem Projekt verknüpfen.

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** und dann **Neue Integration** aus.
- 2 Klicken Sie auf **Active Directory**.
- 3 Geben Sie auf der Registerkarte **Übersicht** die entsprechenden Namen für LDAP-Host und Umgebung ein.
- 4 Geben Sie den Benutzernamen und das Kennwort für den LDAP-Server an.
- 5 Geben Sie den entsprechenden Basis-DN für die gewünschten Benutzer und Gruppen in Ihrem Active Directory ein.

Hinweis Sie können pro Active Directory-Integration nur einen DN angeben.

- 6 Klicken Sie auf **Validieren**, um sicherzustellen, dass die Integration funktioniert.
- 7 Geben Sie einen Namen und eine Beschreibung für diese Integration ein.
- 8 Klicken Sie auf **Speichern**.
- 9 Klicken Sie auf die Registerkarte **Projekt**, um der Active Directory-Integration ein Projekt hinzuzufügen.

Im Dialogfeld **Projekte hinzufügen** müssen Sie einen Projektnamen und einen relativen DN auswählen. Dieser DN befindet sich innerhalb des auf der Registerkarte „Übersicht“ angegebenen Basis-DN.

- 10 Klicken Sie auf **Speichern**.

Ergebnisse

Sie können das Projekt nun mit der Active Directory-Integration zu einem Blueprint verknüpfen. Wenn eine Maschine mithilfe dieses Blueprints vorab bereitgestellt wird, wird sie auch im angegebenen Active Directory und in der angegebenen Organisationseinheit vorab bereitgestellt.

Definition von Onboarding-Plänen in vRealize Automation Cloud Assembly

Sie verwenden den Onboarding-Plan einer Arbeitslast zur Angabe von Maschinen, deren Daten aus einem Cloud-Kontotyp in einer Zielregion oder einem Datencenter erfasst wurden, die aber noch nicht von einem vRealize Automation Cloud Assembly-Projekt verwaltet werden.

Beim Hinzufügen eines Cloud-Kontos mit Maschinen, die außerhalb von vRealize Automation Cloud Assembly bereitgestellt wurden, werden die Maschinen erst nach ihrer Integration von Cloud Assembly verwaltet. Verwenden Sie einen Onboarding-Plan, um nicht verwaltete Maschinen in die vRealize Automation Cloud Assembly-Verwaltung einzubinden. Sie erstellen einen Plan, befüllen ihn mit Maschinen und führen ihn dann aus, um die Maschinen zu importieren. Mithilfe des Onboarding-Plans können Sie einen Blueprint sowie eine oder mehrere Bereitstellungen erstellen.

Sie können eine oder mehrere nicht verwaltete Maschinen in einen einzelnen Plan einbinden. Sie können Maschinen manuell oder mithilfe einer Filterregel auswählen. Filterregeln wählen Maschinen für das Onboarding basierend auf Kriterien wie Maschinenname, Status, IP-Adresse und Tags aus.

- Sie können pro Stunde bis zu 3.500 nicht verwaltete Maschinen innerhalb eines einzelnen Onboarding-Plans einbinden.
- Sie können pro Stunde bis zu 17.000 nicht verwaltete Maschinen innerhalb mehrerer Onboarding-Pläne einbinden.

Maschinen, die für das Arbeitslast-Onboarding verfügbar sind, werden auf der Seite **Ressourcen > Maschinen** für einen bestimmten Cloud-Kontotyp und eine bestimmte Cloud-Region aufgelistet und in der Spalte „Ursprung“ als *Discovered* bezeichnet. Nur Maschinen, für die Daten erfasst wurden, werden aufgelistet. Nach dem Onboarding der Maschinen werden diese in der Spalte „Ursprung“ als *Deployed* angezeigt.

Der Benutzer, der den Onboarding-Plan für die Arbeitslast ausführt, wird automatisch als Maschinenbesitzer zugewiesen.

Beispiele für Onboarding

Beispiele für Onboarding-Techniken finden Sie unter [Beispiel: Integrieren ausgewählter Maschinen als Einzelbereitstellung in vRealize Automation Cloud Assembly](#) und [Beispiel: Integrieren von durch Regeln gefilterten Maschinen als separate Bereitstellungen in vRealize Automation Cloud Assembly](#).

Onboarding-Ereignisabonnements

Ein *Deployment Onboarded*-Ereignis wird erstellt, wenn Sie den Plan ausführen. Mithilfe der Optionen auf der Registerkarte „Erweiterbarkeit“ können Sie diese Bereitstellungsereignisse abonnieren und Aktionen für sie durchführen.

Beispiel: Integrieren ausgewählter Maschinen als Einzelbereitstellung in vRealize Automation Cloud Assembly

In diesem Beispiel können Sie zwei nicht verwaltete Maschinen als vRealize Automation Cloud Assembly-Einzelbereitstellung einbinden und einen einzelnen Blueprint für alle Maschinen im Plan erstellen.

Wenn Sie ein Cloud-Konto erstellen, werden die Daten aller Maschinen, die diesem Konto zugeordnet sind, erfasst und anschließend auf der Seite **Infrastruktur > Ressourcen > Maschinen** angezeigt. Wenn das Cloud-Konto über Maschinen verfügt, die außerhalb von vRealize Automation Cloud Assembly bereitgestellt wurden, können Sie einen Onboarding-Plan verwenden, der zulässt, dass die Maschinen von vRealize Automation Cloud Assembly verwaltet werden.

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderliche Benutzerrolle verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Weitere Informationen finden Sie unter [Definition von Onboarding-Plänen in vRealize Automation Cloud Assembly](#).
- Erstellen und bereiten Sie ein vRealize Automation Cloud Assembly-Projekt vor.

Dieses Verfahren enthält einige der Schritte aus dem grundlegenden WordPress-Anwendungsfall. Weitere Informationen hierzu finden Sie unter [Anwendungsbeispiel: WordPress](#).

- Erstellen Sie ein Projekt, fügen Sie Benutzer hinzu und weisen Sie Benutzerrollen im Projekt zu. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Erstellen eines Projekts](#).
- Erstellen Sie ein Amazon Web Services-Cloud-Konto für das Projekt. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Konten](#).

Das Amazon Web Services-Cloud-Konto in diesem Verfahren enthält Maschinen, die vor dem Hinzufügen des Cloud-Kontos zu vRealize Automation Cloud Assembly von einer anderen Anwendung als vRealize Automation Cloud Assembly bereitgestellt wurden.

- Stellen Sie sicher, dass die Seite **Maschinen** zu integrierende Maschinen enthält. Weitere Informationen hierzu finden Sie unter [Maschinenressourcen](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Onboarding**.
- 2 Klicken Sie auf **Neuer Onboarding-Plan** und geben Sie Beispielwerte ein.

Einstellung	Beispielwert
Name des Plans	VC-sqa-Bereitstellungen
Beschreibung	Beispiel eines Onboarding-Plans für AWS-Maschine für OurCo-AWS-Cloud-Konto
Cloud-Konto	OurCo-AWS
Standardprojekt	WordPress

- 3 Klicken Sie auf **Erstellen**.
- 4 Klicken Sie auf der Registerkarte **Bereitstellungen** des Plans auf **Maschinen auswählen**, wählen Sie eine oder mehrere Maschinen aus und klicken Sie auf **OK**.

Maschinen auswählen

Filtern...

<input type="checkbox"/>	Name	Status
<input checked="" type="checkbox"/>	Application_VM-mcm261184-101051524497	▶ Ein
<input checked="" type="checkbox"/>	autoscale1033231	▶ Ein

- 5 Wählen Sie **Eine Bereitstellung erstellen, die alle Maschinen enthält** aus und klicken Sie auf **Erstellen**.
- 6 Aktivieren Sie das Kontrollkästchen neben dem Namen der neuen Bereitstellung und klicken Sie auf **Blueprint...**
- 7 Klicken Sie auf **Blueprint im Cloud Assembly-Format erstellen**.

8 Geben Sie einen Namen für den Blueprint ein und klicken Sie auf **Speichern**.

Blueprint-Konfiguration

Bereitstellung Deployment-c8a6e0f9-790a-411b-b0e9-4d2e79ce118e

☐ Keine (Laufzeit-Snapshot verwenden)

☒ Blueprint im Cloud Assembly-Format erstellen

Blueprint-Name BP_Sample_1

Blueprint-Vorschau

```

1 ---
2 resources:
3   VMware-vRO-Appliance-SAAS-1127:
4     type: "Cloud.vSphere.Machine"
5     properties:
6       imageRef: "no_image_available"
7       cpuCount: 2
8       totalMemoryMB: 6144
9   VMware-Cloud-Services-Data-Collector-7.2.0.25668-11138205_OVF10:
10    type: "Cloud.vSphere.Machine"
11    properties:
12      imageRef: "no_image_available"
13      cpuCount: 4
14      totalMemoryMB: 12288
  
```

ABBRECHEN **SPEICHERN**

Hinweis Wenn Ihr Onboarding-Plan eine vSphere-Maschine verwendet, müssen Sie den Blueprint bearbeiten, nachdem der Onboarding-Vorgang abgeschlossen ist. Der Onboarding-Vorgang kann die vSphere-Quellmaschine und die zugehörige Maschinenvorlage nicht verknüpfen, und der resultierende Blueprint enthält den `imageRef: "no image available"`-Eintrag im Blueprint-Code. Der Blueprint kann erst bereitgestellt werden, wenn Sie den korrekten Vorlagennamen im Feld `imageRef:` angeben. Um das Auffinden und Aktualisieren des Blueprints nach Abschluss des Onboarding-Vorgangs zu vereinfachen, verwenden Sie die Option **Blueprint-Name** auf der Seite **Blueprint-Konfiguration** der Bereitstellung. Notieren Sie den automatisch generierten Blueprint-Namen oder geben Sie einen Blueprint-Namen Ihrer Wahl ein und notieren Sie ihn. Wenn die Onboarding-Funktion abgeschlossen ist, suchen und öffnen Sie den Blueprint und ersetzen Sie den `"no image available"`-Eintrag im Feld `imageRef:` durch den korrekten Vorlagennamen.

9 Aktivieren Sie das Kontrollkästchen neben dem Namen der Bereitstellung, klicken Sie auf **Ausführen** und dann auf der Seite **Plan ausführen** erneut auf **Ausführen**.

Die ausgewählten Amazon Web Services-Maschinen werden als Einzelbereitstellung mit einem begleitenden Blueprint integriert.

10 Öffnen und untersuchen Sie den Blueprint, indem Sie auf die Registerkarte **Blueprints** und dann auf den Namen des Blueprints klicken.

11 Öffnen und untersuchen Sie die Bereitstellung, indem Sie auf die Registerkarte **Bereitstellungen** und dann auf den Namen der Bereitstellung klicken.

Beispiel: Integrieren von durch Regeln gefilterten Maschinen als separate Bereitstellungen in vRealize Automation Cloud Assembly

In diesem Beispiel verwenden Sie eine Filterregel zur Einbindung von Maschinen mit dem Status „Eingeschaltet“, deren Name mit den Buchstaben „BG“ beginnt. Sie erstellen auch einen separaten vRealize Automation Cloud Assembly-Blueprint und eine Bereitstellung für jede Maschine im Plan.

Wenn Sie ein Cloud-Konto erstellen, werden die Daten aller Maschinen, die diesem Konto zugeordnet sind, erfasst und anschließend auf der Seite **Infrastruktur > Ressourcen > Maschinen** angezeigt. Wenn das Cloud-Konto über Maschinen verfügt, die außerhalb von vRealize Automation Cloud Assembly bereitgestellt wurden, können Sie einen Onboarding-Plan verwenden, der zulässt, dass die Maschinen von vRealize Automation Cloud Assembly verwaltet werden.

Voraussetzungen

- Stellen Sie sicher, dass Sie über die erforderliche Benutzerrolle verfügen. Weitere Informationen finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Weitere Informationen finden Sie unter [Definition von Onboarding-Plänen in vRealize Automation Cloud Assembly](#).
- Erstellen und bereiten Sie ein vRealize Automation Cloud Assembly-Projekt vor und befüllen Sie es mit einem oder mehreren Cloud-Konten.

Dies umfasst einige grundlegende Schritte im geführten Setup-Verfahren.

- Erstellen Sie ein Projekt, fügen Sie Benutzer hinzu und weisen Sie Benutzerrollen im Projekt zu. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Erstellen eines Projekts](#).
- Erstellen Sie ein oder mehrere Cloud-Konten in festgelegten Regionen für das Projekt. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Konten](#).
- Stellen Sie sicher, dass die Seite **Maschinen** zu integrierende Maschinen enthält. Weitere Informationen hierzu finden Sie unter [Maschinenressourcen](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Onboarding**.
- 2 Klicken Sie auf **Neuer Onboarding-Plan** und geben Sie Werte ein.

Einstellung	Beispielwert
Name des Plans	ob_rules_1
Beschreibung	Integration von Maschinen mit rules1

Einstellung	Beispielwert
Cloud-Konto	rs-aws
Standardprojekt	rs-project

Neuer Onboarding-Plan ×

Planname

Beschreibung

Voraussetzung

Fügen Sie das Cloud-Konto hinzu und erstellen Sie Cloud-Zonen für Computing-Ressourcen, in denen sich die einzubindenden Maschinen befinden. Erstellen Sie ein Projekt mit mindestens einem Benutzer und erteilen Sie dem Projekt Zugriff auf die Cloud-Zonen.

Cloud-Konto × ?

Standardprojekt × ?

ABBRECHEN

ERSTELLEN

3 Klicken Sie auf **Erstellen**.

ob_rules_1

Übersicht Regeln Maschinen Bereitstellungen

Planname

Beschreibung

Planstatus ✓

Letzte Ausführung Niemals

Quellinformationen

Cloud-Konto ?

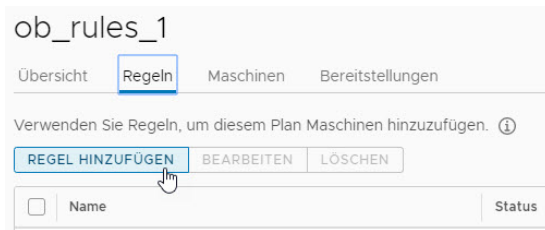
Tag-Schlüssel der Bereitstellung ?

Zielkonfiguration

Standardprojekt × ?

4 Klicken Sie auf die Registerkarte **Regeln** und dann auf **Regel hinzufügen**.

Sie können eine oder mehrere Regeln erstellen, um eine Gruppe von Maschinen für das Onboarding auf Basis bestimmter Maschineneigenschaften auszuwählen.



5 Geben Sie einen Regelnamen ein, wie z. B. **ob_rules_1**.

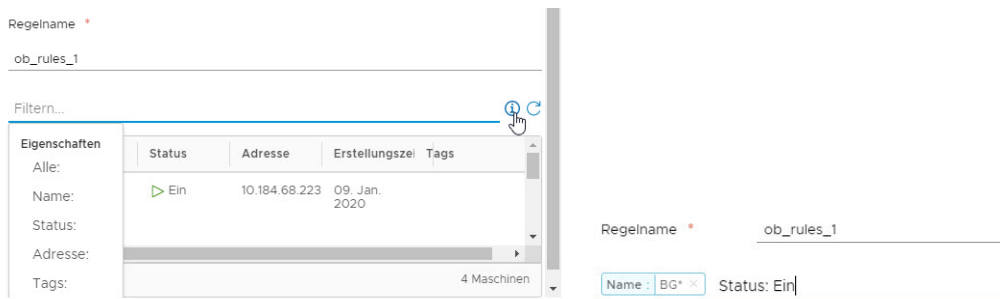
Hinzufügen Regel

Erstellen Eine filterbasierte Regel, die zum Befüllen von Maschinen in diesem Plan verwendet wird.

Regelname * ob_rules_1

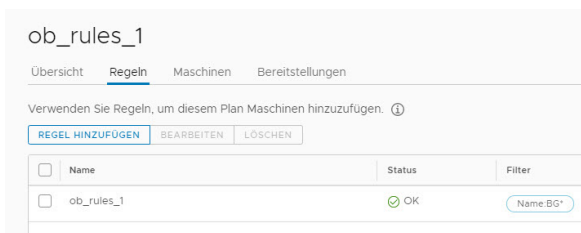
6 Erstellen Sie die Regel durch Hinzufügen von Filtern.

Für dieses Beispiel verwenden Sie die Filter **Status** und **Name** im Dropdown-Menü **Filter**, um alle Maschinen anzugeben, deren Name BG* enthält und deren Status On lautet.



7 Klicken Sie auf **Speichern**.

In diesem Beispiel wird eine einzige Regel verwendet. Sie können jedoch zusätzliche Regeln festlegen.



- 8 Klicken Sie auf die Registerkarte **Maschinen**. In diesem Beispiel werden vier Maschinen ausgewählt. Der Name dreier Maschinen beginnt mit den Buchstaben BG, und der Name einer Maschine enthält die Buchstaben BG.

ob_rules_1

Übersicht Regeln **Maschinen** Bereitstellungen

Die hier aufgeführten Maschinen werden beim Ausführen des Plans eingebunden. Planregeln werden alle 24 Stunden ausgewertet und neue Maschinen werden unter Umständen zum Plan hinzugefügt.

MASCHINEN HINZUFÜGEN BEIBEHALTEN AUSSCHLIESSEN ENTFERNEN Filtern...

<input type="checkbox"/>	Name	Status	Leistung	Adresse	Bereitstellung	Regel	Tags
<input type="checkbox"/>	tf-machine-mcm333-124160625552	Ausstehend	Ein	10.196.157.207	Deployment-a46900d1-2f11-440b...	ob_rules_1	Us:Ubuntu too bar
<input type="checkbox"/>	Terraform_Provider-003-mcm423-124577107709	Ausstehend	Ein	10.196.157.229	Deployment-2468b529-b7b1-4bfc...	ob_rules_1	Us:Ubuntu too bar
<input type="checkbox"/>	vm345-mcm593-124945691582	Ausstehend	Ein	10.196.157.161	Deployment-56791dfe-cb55-4276...	ob_rules_1	vm345key vm345
<input type="checkbox"/>	tf-machine-mcm333-124160625585	Ausstehend	Ein	10.196.157.213	Deployment-57da746-8847-44fa...	ob_rules_1	Us:Ubuntu too bar

4 Maschinen

- 9 Entfernen Sie die Maschine, deren Name nicht mit BG beginnt, indem Sie das zugehörige Kontrollkästchen aktivieren und dann auf **Ausschließen** klicken.

ob_rules_1

Übersicht Regeln **Maschinen** Bereitstellungen

Die hier aufgeführten Maschinen werden beim Ausführen des Plans eingebunden. Planregeln werden alle 24 Stunden ausgewertet und neue Maschinen werden unter Umständen zum Plan hinzugefügt.

MASCHINEN HINZUFÜGEN BEIBEHALTEN AUSSCHLIESSEN ENTFERNEN Filtern...

<input type="checkbox"/>	Name	Status	Leistung	Adresse	Bereitstellung	Regel	Tags
<input type="checkbox"/>	tf-machine-mcm333-124160625552	Ausstehend	Ein	10.196.157.207	Deployment-a46900d1-2f11-440b...	ob_rules_1	Us:Ubuntu too bar
<input type="checkbox"/>	Terraform_Provider-003-mcm423-124577107709	Ausstehend	Ein	10.196.157.229	Deployment-2468b529-b7b1-4bfc...	ob_rules_1	Us:Ubuntu too bar
<input checked="" type="checkbox"/>	vm345-mcm593-124945691582	Ausstehend	Ein	10.196.157.161	Deployment-56791dfe-cb55-4276...	ob_rules_1	vm345key vm345
<input type="checkbox"/>	tf-machine-mcm333-124160625585	Ausstehend	Ein	10.196.157.213	Deployment-57da746-8847-44fa...	ob_rules_1	Us:Ubuntu too bar

4 Maschinen

- 10 Klicken Sie auf die Registerkarte **Bereitstellungen**.

Die drei Maschinen, die mit den Buchstaben BG beginnen und On sind, können bereitgestellt werden. Standardmäßig werden für jede Maschine ein eigener Blueprint und eine eigene Bereitstellung erstellt.

ob_rules_1

ÜbersichtRegelnMaschinenBereitstellungen

Diese Bereitstellungen werden während der Planausführung erstellt oder aktualisiert. Standardmäßig wird jede hinzugefügte Maschine in einer eigenen Cloud Assembly-Bereitstellung platziert.

UMBENENNENBLUEPRINTENTFERNEN

<input type="checkbox"/>	Name der Bereitstellung	Status	Blueprint erstellen	Komponenten									
<input type="checkbox"/>	<div><div><input checked="" type="checkbox"/></div>Deployment-2468b529-b7b1-4bfc-a0ff-a3e6074a6d35</div>	<div><div></div><div>✓</div></div>		1									
	<table><tr><th>Komponentenname</th><th>Status</th><th>Typ</th><th>Adresse</th><th>Tags</th></tr><tr><td>Terraform_Provider-003-mcm423-124577107709</td><td><div><div></div><div>✓</div></div></td><td>Maschine</td><td></td><td><div>Us:Ubuntu</div><div>Too Bar</div></td></tr></table>	Komponentenname	Status	Typ	Adresse	Tags	Terraform_Provider-003-mcm423-124577107709	<div><div></div><div>✓</div></div>	Maschine		<div>Us:Ubuntu</div> <div>Too Bar</div>		
Komponentenname	Status	Typ	Adresse	Tags									
Terraform_Provider-003-mcm423-124577107709	<div><div></div><div>✓</div></div>	Maschine		<div>Us:Ubuntu</div> <div>Too Bar</div>									

<input type="checkbox"/>	<div><div><input checked="" type="checkbox"/></div>Deployment-57da746-8847-44fa-86c4-4f36079fb362</div>	<div><div></div><div>✓</div></div>		1									
	<table><tr><th>Komponentenname</th><th>Status</th><th>Typ</th><th>Adresse</th><th>Tags</th></tr><tr><td>tf-machine-mcm333-124160625585</td><td><div><div></div><div>✓</div></div></td><td>Maschine</td><td></td><td><div>Us:Ubuntu</div><div>Too Bar</div></td></tr></table>	Komponentenname	Status	Typ	Adresse	Tags	tf-machine-mcm333-124160625585	<div><div></div><div>✓</div></div>	Maschine		<div>Us:Ubuntu</div> <div>Too Bar</div>		
Komponentenname	Status	Typ	Adresse	Tags									
tf-machine-mcm333-124160625585	<div><div></div><div>✓</div></div>	Maschine		<div>Us:Ubuntu</div> <div>Too Bar</div>									

<input type="checkbox"/>	<div><div><input checked="" type="checkbox"/></div>Deployment-a46900d1-2f11-440b-8906-9202f85d1854</div>	<div><div></div><div>✓</div></div>		1									
	<table><tr><th>Komponentenname</th><th>Status</th><th>Typ</th><th>Adresse</th><th>Tags</th></tr><tr><td>tf-machine-mcm332-124160625552</td><td><div><div></div><div>✓</div></div></td><td>Maschine</td><td></td><td><div>Us:Ubuntu</div><div>Too Bar</div></td></tr></table>	Komponentenname	Status	Typ	Adresse	Tags	tf-machine-mcm332-124160625552	<div><div></div><div>✓</div></div>	Maschine		<div>Us:Ubuntu</div> <div>Too Bar</div>		
Komponentenname	Status	Typ	Adresse	Tags									
tf-machine-mcm332-124160625552	<div><div></div><div>✓</div></div>	Maschine		<div>Us:Ubuntu</div> <div>Too Bar</div>									

☐

3 Bereitstellungen

SPEICHERN

AUSFÜHREN

ABBRECHEN

- 11 Aktivieren Sie das Kontrollkästchen neben den drei Bereitstellungsamen, klicken Sie auf **Blueprints**, dann auf **Blueprint im Cloud Assembly-Format erstellen** und anschließend auf **Speichern**.

Blueprint-Konfiguration ×

3 Ausgewählte Bereitstellungen

☐ Keine (Laufzeit-Snapshot verwenden)

☒ Blueprint im Cloud Assembly-Format erstellen

Blueprint-Vorschau

Mehrere Bereitstellungen ausgewählt

ABBRECHEN
SPEICHERN

Hinweis Wenn Ihr Onboarding-Plan eine vSphere-Maschine verwendet, müssen Sie den Blueprint bearbeiten, nachdem der Onboarding-Vorgang abgeschlossen ist. Der Onboarding-Vorgang kann die vSphere-Quellmaschine und die zugehörige Maschinenvorlage nicht verknüpfen, und der resultierende Blueprint enthält den `imageRef: "no image available"`-Eintrag im Blueprint-Code. Der Blueprint kann erst bereitgestellt werden, wenn Sie den korrekten Vorlagennamen im Feld `imageRef:` angeben. Um das Auffinden und Aktualisieren des Blueprints nach Abschluss des Onboarding-Vorgangs zu vereinfachen, verwenden Sie die Option **Blueprint-Name** auf der Seite **Blueprint-Konfiguration** der Bereitstellung. Notieren Sie den automatisch generierten Blueprint-Namen oder geben Sie einen Blueprint-Namen Ihrer Wahl ein und notieren Sie ihn. Wenn die Onboarding-Funktion abgeschlossen ist, suchen und öffnen Sie den Blueprint und ersetzen Sie den `"no image available"`-Eintrag im Feld `imageRef:` durch den korrekten Vorlagennamen.

- 12 Aktivieren Sie auf der Seite **Bereitstellungen** das Kontrollkästchen neben den drei Bereitstellungsamen und klicken Sie auf **Ausführen**.

ob_rules_1

Übersicht Regeln Maschinen Bereitstellungen

Diese Bereitstellungen werden während der Planausführung erstellt oder aktualisiert. Standardmäßig wird jede hinzugefügte Maschine in einer eigenen Cloud Assembly-Bereitstellung platziert.

UMBENENNEN BLUEPRINT ... ENTFERNEN

<input checked="" type="checkbox"/>	Name der Bereitstellung	Status	Blueprint erstellen	Komponenten	
<input checked="" type="checkbox"/>	Deployment-2468b529-e7b1-4bfc-a0ff-a3e6074a6d35	✓	✓	1	
	Komponentenname	Status	Typ	Adresse	Tags
	Terraform_Provider-003-mcm423-124577107709	✓		Maschine	Os: Ubuntu too bar
<input checked="" type="checkbox"/>	Deployment-57da746-8847-44fa-86c4-4f36079fb362	✓	✓	1	
	Komponentenname	Status	Typ	Adresse	Tags
	tf-machine-mcm333-124160625585	✓		Maschine	Os: Ubuntu too bar
<input checked="" type="checkbox"/>	Deployment-a46900d1-12f1-44a6-8906-9202f85d854	✓	✓	1	
	Komponentenname	Status	Typ	Adresse	Tags
	tf-machine-mcm332-124160625552	✓		Maschine	Os: Ubuntu too bar

3 Bereitstellungen

SPEICHERN AUSFÜHREN ABBRECHEN

- 13 Wenn Sie zur Bestätigung aufgefordert werden, klicken Sie auf **Ausführen**, um die Maschinen zu integrieren.

Plan ausführen ×

Planname	ob_rules_1
Beschreibung	Machine onboarding with rules1
Cloud-Konto	346test_vc_account
Standardprojekt	123
Bereitstellungen	3
Letzte Ausführung	Niemals

ABBRECHEN
AUSFÜHREN

Der Plan wird ausgeführt, und die Maschinen werden in vRealize Automation Cloud Assembly verwaltet. Für jede Maschine wird ein eigener Blueprint und eine eigene Bereitstellung erstellt.

Erweiterte Konfiguration für vRealize Automation Cloud Assembly-Umgebung

Sie können Ihre vRealize Automation Cloud Assembly-Umgebung so konfigurieren, dass Projektkonfiguration, -integration und -bereitstellung weiter unterstützt werden.

Weitere Informationen zu Verwaltungsmethoden, wie z. B. die Verwendung von Benutzern und Protokollen sowie der Beitritt zum Programm zur Verbesserung der Benutzerfreundlichkeit bzw. das Verlassen des Programms, finden Sie in der Hilfe unter [Verwalten von vRealize Automation](#).

Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation

Für vRealize Automation 8.0.1-Installationen und höher in isolierten Netzwerken ohne direkten Internetzugriff können Sie einen Internet-Proxyserver so verwenden, dass das Internet gemäß Proxy-Funktionen freigegeben wird. Der Internet-Proxyserver unterstützt HTTP und HTTPS.

Um Public Cloud-Anbieter wie Amazon Web Services (AWS), Microsoft Azure und Google Cloud Platform (GCP) sowie externe Integrationspunkte wie IPAM, Ansible und Puppet mit vRealize Automation zu konfigurieren und zu verwenden, müssen Sie einen Internet-Proxyserver für den Zugriff auf einen internen vRealize Automation-Internet-Proxyserver konfigurieren.

vRealize Automation enthält einen internen Proxyserver, der mit Ihrem Internet-Proxyserver kommuniziert. Dieser Server kommuniziert mit Ihrem Proxyserver, wenn er mit dem Befehl `vracli proxy set ...` konfiguriert wurde. Wenn Sie keinen Internet-Proxyserver für Ihre Organisation konfiguriert haben, versucht der interne vRealize Automation-Proxyserver, eine direkte Verbindung mit dem Internet herzustellen.

Sie können vRealize Automation so einrichten, dass ein Internet-Proxyserver mithilfe des bereitgestellten Befehlszeilen-Dienstprogramms `vracli` verwendet wird. Informationen zur Verwendung der `vracli`-API finden Sie unter Verwendung des `--help`-Arguments in der `vracli`-Befehlszeile, z. B. `vracli proxy --help`.

Der Zugriff auf den Internet-Proxyserver erfordert die Verwendung von lokalen eingebetteten Steuerelementen mit aktionsbasierter Erweiterbarkeit (ABX), die in vRealize Automation integriert sind.

Hinweis Der Zugriff auf Workspace ONE Access (zuvor VMware Identity Manager) über den Internet-Proxy wird nicht unterstützt. Sie können den Befehl `vracli set vidm` nicht für den Zugriff auf Workspace ONE Access über den Internet-Proxyserver verwenden.

Der interne Proxyserver erfordert IPv4 als Standard-IP-Format. Es sind keine Internet-Protokolleinschränkungen, Authentifizierungs- oder Man-in-the-Middle-Aktionen für den TLS (HTTPS)-zertifizierten Datenverkehr erforderlich.

Voraussetzungen

- Stellen Sie sicher, dass Sie über einen vorhandenen als Internet-Proxyserver zu verwendenden HTTP- oder HTTPS-Server in dem vRealize Automation-Netzwerk verfügen, das den ausgehenden Datenverkehr an externe Sites weiterleiten kann. Die Verbindung muss für IPv4 konfiguriert werden.
- Stellen Sie sicher, dass der Ziel-Internet-Proxyserver für die Unterstützung von IPv4 statt IPv6 als IP-Standardformat konfiguriert ist.
- Wenn der Internet-Proxyserver TLS verwendet und eine HTTPS-Verbindung mit seinen Clients benötigt, müssen Sie das Serverzertifikat mithilfe eines der folgenden Befehle importieren, bevor Sie die Proxy-Konfiguration festlegen.
 - `vracli certificate proxy --set path_to_proxy_certificate.pem`
 - `vracli certificate proxy --set stdin`

Verwenden Sie den Parameter `stdin` für die interaktive Eingabe.

Verfahren

- 1 Erstellen Sie eine Proxy-Konfiguration für die Pods oder Container, die von den Kubernetes verwendet werden. In diesem Beispiel wird über das HTTP-Schema auf den Proxyserver zugegriffen.

```
vracli proxy set --host http://proxy.vmware.com:3128
```

- 2 Zeigen Sie die Proxy-Konfiguration an.

```
vracli proxy show
```

Das Ergebnis ähnelt Folgendem:

```
{
  "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"
}
```

Hinweis Wenn Sie einen Internet-Proxyserver für Ihre Organisation konfiguriert haben, wird im obigen Beispiel "internal.proxy.config.type": "non-default" anstelle von 'default' angezeigt. Aus Sicherheitsgründen wird das Kennwort nicht angezeigt.

Hinweis Wenn Sie den Parameter `--proxy-exclude` verwenden, müssen Sie die Standardwerte bearbeiten. Wenn Sie z. B. `acme.com` als Domäne hinzufügen möchten, auf die über den Internet-Proxyserver nicht zugegriffen werden kann, führen Sie die folgenden Schritte aus:

- a Geben Sie `vracli proxy default-no-proxy` ein, um die standardmäßigen `proxy-exclude`-Einstellungen zu erhalten. Dies ist eine Liste der automatisch erstellten Domänen und Netzwerke.
- b Bearbeiten Sie den Wert, der `.acme.com` hinzugefügt werden soll.
- c Geben Sie `vracli proxy set --proxy-exclude ...` ein, um die aktuellen Konfigurationseinstellungen zu aktualisieren.
- d Führen Sie den Befehl `/opt/scripts/deploy.sh` aus, um die Umgebung erneut bereitzustellen.

- 3 (Optional) Schließen Sie DNS-Domänen, FQDNs und IP-Adressen vom Zugriff durch den Internet-Proxyserver aus.

Ändern Sie immer die Standardwerte der `proxy-exclude`-Variablen mithilfe von parameter `--proxy-exclude`. Um die Domäne `exclude.vmware.com` hinzuzufügen, verwenden Sie zuerst den Befehl `vracli proxy show`, kopieren Sie dann die Variable `proxy-exclude` und fügen Sie den Domänenwert mithilfe des Befehls `vracli proxy set ...` wie folgt hinzu:

```
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"
```

Hinweis Fügen Sie `proxy-exclude` Elemente hinzu, anstatt Werte zu ersetzen.

Wenn Sie `proxy-exclude`-Standardwerte löschen, funktioniert vRealize Automation nicht ordnungsgemäß. Sollte dies geschehen, löschen Sie die Proxy-Konfiguration und beginnen Sie von vorn.

- 4 Nachdem Sie den Internet-Proxyserver mit dem Befehl `vracli proxy set ...` festgelegt haben, können Sie mithilfe des Befehls `vracli proxy apply` die Konfiguration des Internet-Proxyservers aktualisieren und die neuesten Proxy-Einstellungen aktivieren.

- 5 Sollten Sie dies noch nicht getan haben, aktivieren Sie die Skriptänderungen, indem Sie den folgenden Befehl ausführen:

```
/opt/scripts/deploy.sh
```

- 6 (Optional) Konfigurieren Sie bei Bedarf den Proxyserver, um den externen Zugriff auf Port 22 zu unterstützen.

Um Integrationen wie Puppet und Ansible zu unterstützen, muss der Proxyserver zulassen, dass Port 22 auf die relevanten Hosts zugreift.

Beispiel: Beispiel für Squid-Konfiguration

In Bezug auf Schritt 1 können Sie, falls Sie einen Squid-Proxy einrichten, Ihre Konfiguration in `/etc/squid/squid.conf` optimieren, indem Sie sie an das folgende Beispiel anpassen:

```
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
```

Vorgehensweise zum Einrichten einer Windows-Vorlage mit cloud-init oder cloudbase-init in vRealize Automation

Sie können Ihre vRealize Automation-Umgebung so konfigurieren, dass die cloud-init- und cloudbase-init-Skripterstellung für das Erstellen und Verwenden von benutzerdefinierten Images in Windows-Bereitstellungen unterstützt wird.

Cloud-Konfigurationsskripts werden in der Image-Zuordnung und im Blueprint-Code unterstützt. Bei der Skripterstellung für die Cloud-Konfiguration werden Cloud-init- und Cloudbase-init-Regeln und -Formate verwendet. Weitere Informationen zu cloud-init und cloudbase-init finden Sie unter <https://cloudbase.it/cloudbase-init>.

Hinweis Informationen zum Konfigurieren von cloud-init oder cloud-init für Windows finden Sie in den folgenden VMware-Blogs:

- Blog-Artikel [Windows Cloud-Init solution](#)
- Blog-Artikel [Windows guest initialization with Cloudbase-Init in vCenter](#)

Weitere Informationen zum Konfigurieren von cloud-init für Linux finden Sie im Blogbeitrag [Erstellen einer vRealize Automation Cloud-fähigen Ubuntu-Vorlage für vSphere](#).

Informationen zur Verwendung von cloud-init und Cloud-Konfigurationsskripts in vRealize Automation finden Sie unter:

- [Weitere Informationen zu Image-Zuordnungen in vRealize Automation Cloud Assembly](#)
- [Vorgehensweise zum automatischen Initialisieren einer Maschine in einem vRealize Automation Cloud Assembly-Blueprint](#)

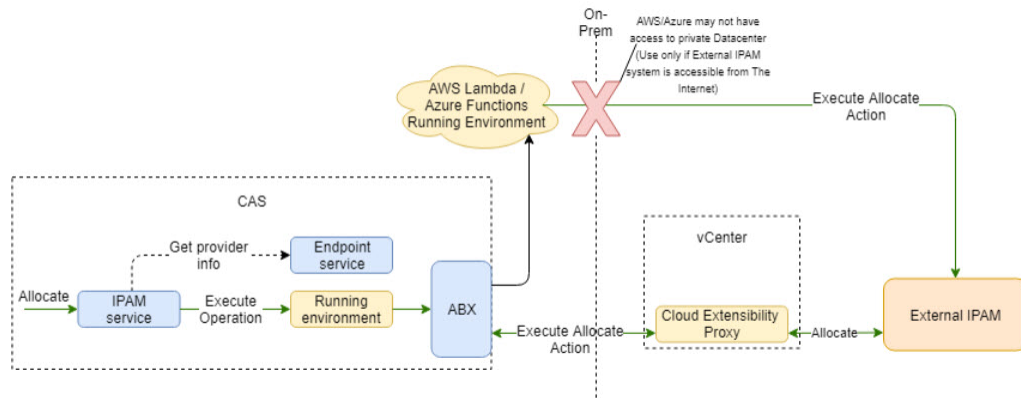
Weitere Informationen finden Sie auch auf den folgenden Anbieterseiten zu cloud-init und cloudbase-init:

- <https://cloud-init.io>
- <https://cloudinit.readthedocs.io/en/latest/index.html>
- <https://cloudinit.readthedocs.io/en/latest/topics/examples.html#yaml-examples>
- <https://cloudbase.it/cloudbase-init>
- <https://cloudbase-init.readthedocs.io/en/latest/services.html>
- <https://cloudbase-init.readthedocs.io/en/latest/userdata.html#userdata>

Vorgehensweise zum Verwenden des IPAM-SDK zur Erstellung eines anbieterspezifischen, externen IPAM-Integrationspakets für vRealize Automation

Externe IPAM-Anbieter und -Partner können das IPAM-SDK herunterladen und verwenden, um ein IPAM-Integrationspaket zu erstellen, mit dem vRealize Automation die anbieterspezifische IPAM-Lösung unterstützen kann.

Der Vorgang zum Erstellen und Bereitstellen eines benutzerdefinierten IPAM-Pakets für vRealize Automation mithilfe des bereitgestellten IPAM-SDK wird im Dokument [Erstellen und Bereitstellen eines anbieterspezifischen IPAM-Integrationspakets für VMware Cloud Assembly](#) beschrieben. Wie im Dokument beschrieben, können Sie das *VMware vRealize Automation-Drittanbieter-IPAM-SDK* über den [VMware Solutions Exchange Marketplace](#) herunterladen.



Überprüfen Sie vor dem Erstellen eines anbieterspezifischen IPAM-Integrationspakets mithilfe des IPAM-SDK, ob bereits ein Paket für vRealize Automation vorhanden ist. Sie können auf der Website des IPAM-Anbieters im [VMware Solutions Exchange Marketplace](#) und auf der Registerkarte vRealize Automation **Marketplace** nach einem anbieterspezifischen IPAM-Integrationspaket suchen.

Im anbieterspezifischen Beispiel [Anwendungsfall für eine anbieterspezifische externe IPAM-Integration](#) sind hilfreiche Referenzinformationen enthalten.

vRealize Automation Cloud Assembly-Anwendungsfälle

3

Diese Anwendungsfälle zeigen Beispiele für die Erstellung einer Ressourceninfrastruktur in vRealize Automation Cloud Assembly und den anschließenden Entwurf und die Bereitstellung von Anwendungen in dieser Infrastruktur.

Die Anwendungsfälle stellen nur Beispielwerte dar. Ihre Umgebungsstruktur und die Benennungskonventionen sind unterschiedlich.

Dieses Kapitel enthält die folgenden Themen:

- [Anwendungsbeispiel: WordPress](#)
- [VMware Cloud on AWS-Anwendungsbeispiel](#)
- [Anwendungsfall für eine anbieterspezifische externe IPAM-Integration](#)

Anwendungsbeispiel: WordPress

Dieser End-to-End-Anwendungsfall von vRealize Automation Cloud Assembly zeigt ein Beispiel für das Erstellen einer Infrastruktur und für die Bereitstellung einer WordPress-Site in dieser Infrastruktur.

Sehen Sie sich die schrittweise Einrichtung an, um den Vorgang zu verstehen, der eine WordPress-Site zum Abschluss bringt.

Beachten Sie, dass es sich bei den angezeigten Werten nur um Anwendungsbeispiele handelt. Sie können diese nicht eins zu eins auf Ihre Umgebung übertragen.

Überlegen Sie sich, wo Sie Ihre eigenen Ersetzungen vornehmen würden, oder übernehmen Sie eine Hochrechnung aus den Beispielwerten, um Ihre eigene Cloud-Infrastruktur einzurichten und Ihre Bereitstellungsanforderungen zu erfüllen.

Verfahren

1 [WordPress-Anwendungsbeispiel: Erstellen der Infrastruktur](#)

Als Cloud-Administrator müssen Sie zunächst die Ressourcen konfigurieren, mit denen die Entwickler später eine WordPress-Site entwickeln, testen und in Betrieb nehmen können.

2 [WordPress-Anwendungsbeispiel: Erstellen eines Projekts](#)

Ein Projekt aktiviert die Benutzer mit Bereitstellungsfunktion und konfiguriert die Bereitstellungsmöglichkeiten.

3 WordPress-Anwendungsbeispiel: Erstellen und Erweitern eines Blueprints

Als Entwickler definieren Sie die WordPress-Site in Form eines generischen vRealize Automation Cloud Assembly-Blueprints, der für jeden Cloud-Anbieter bereitgestellt werden kann.

WordPress-Anwendungsbeispiel: Erstellen der Infrastruktur

Als Cloud-Administrator müssen Sie zunächst die Ressourcen konfigurieren, mit denen die Entwickler später eine WordPress-Site entwickeln, testen und in Betrieb nehmen können.

Die Infrastruktur umfasst Cloud-Ziele und Definitionen rund um die Maschinen, Netzwerke und Speicher, die für die WordPress-Site erforderlich sind.

Verfahren

1 WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Konten

In diesem Schritt fügt der Cloud-Administrator zwei Cloud-Konten hinzu. Laut Projektanforderung sollen Entwicklung und Test auf AWS durchgeführt werden, während die Produktion auf Azure stattfinden soll.

2 WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen

In diesem Schritt fügt der Cloud-Administrator drei Cloud-Zonen hinzu, jeweils eine für die Entwicklung, die Tests und die Produktion.

3 WordPress-Anwendungsbeispiel: Hinzufügen von Konfigurationszuordnungen

In diesem Schritt fügt der Cloud-Administrator Konfigurationszuordnungen hinzu, um die Kapazitätsanforderungen zu berücksichtigen, die je nach Bereitstellung variieren können.

4 WordPress-Anwendungsbeispiel: Hinzufügen von Image-Zuordnungen

In diesem Schritt fügt der Cloud-Administrator eine Image-Zuordnung für Ubuntu hinzu, den Host für den WordPress-Server und den zugehörigen MySQL-Datenbankserver.

5 WordPress-Anwendungsbeispiel: Hinzufügen von Netzwerkprofilen

In diesem Schritt fügt der Cloud-Administrator jeder Cloud-Zone ein Netzwerkprofil hinzu.

6 WordPress-Anwendungsbeispiel: Hinzufügen von Speicherprofilen

In diesem Schritt fügt der Cloud-Administrator jeder Cloud-Zone ein Speicherprofil hinzu.

WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Konten

In diesem Schritt fügt der Cloud-Administrator zwei Cloud-Konten hinzu. Laut Projektanforderung sollen Entwicklung und Test auf AWS durchgeführt werden, während die Produktion auf Azure stattfinden soll.

Verfahren

1 Navigieren Sie zu **Infrastruktur > Verbindungen > Cloud-Konten**.

- 2 Klicken Sie auf **Cloud-Konto hinzufügen**, wählen Sie Amazon Web Services aus und geben Sie Werte ein.

Einstellung	Beispielwert
Zugriffsschlüssel-ID	R5SDR3PXVV2ZW8B7YNSM
Geheimer Zugriffsschlüssel	SZXAINXU4UHNQA1E156S
Name	OurCo-AWS
Beschreibung	WordPress
Funktionen	cloud:aws

Beachten Sie, dass es sich bei allen Werten nur um Anwendungsbeispiele handelt. Die Kontospezifikationen sind unterschiedlich.

- 3 Klicken Sie zum Überprüfen der Anmeldedaten auf **Überprüfen**.
- 4 Klicken Sie auf **Hinzufügen**.
- 5 Bearbeiten Sie das neu hinzugefügte Konto **Konfiguration** und ermöglichen Sie die Bereitstellung für die Regionen „us-east-1“ und „us-west-2“.
- 6 Klicken Sie auf **Cloud-Konto hinzufügen**, wählen Sie Microsoft Azure aus und geben Sie Werte ein.

Einstellung	Beispielwert
Abonnement-ID	ef2avpf-dfdv-zxlugui1i-g4h0-i8ep2jwp4c9arbfe
Mandanten-ID	dso9wv3-4zgc-5nrcy5h3m-4skf-nnovp40wfxsro22r
Client-Anwendungs-ID	bg224oq-3ptp-mbhi6aa05-q511-uflyjr2sttyik6bs
Geheimer Schlüssel der Client-Anwendung	7uqxi57-0wtn-kymgf9wcj-t2l7-e52e4nu5fig4pmdd
Name	OurCo-Azure
Beschreibung	WordPress
Funktionen	cloud:az

- 7 Klicken Sie zum Überprüfen der Anmeldedaten auf **Überprüfen**.
- 8 Klicken Sie auf **Hinzufügen**.
- 9 Bearbeiten Sie das neu hinzugefügte Konto **Konfiguration** und ermöglichen Sie die Bereitstellung für die Region „East US“.

Nächste Schritte

Fügen Sie Cloud-Zonen hinzu, in denen das Projekt die WordPress-Site bereitstellt. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#).

WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen

In diesem Schritt fügt der Cloud-Administrator drei Cloud-Zonen hinzu, jeweils eine für die Entwicklung, die Tests und die Produktion.

Cloud-Zonen sind die Ressourcen, auf denen das Projekt die Maschinen, Netzwerke und Speicher zur Unterstützung der WordPress-Site bereitstellt.

Voraussetzungen

Hinzufügen von Cloud-Konten. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Konten](#).

Verfahren

- 1 Wechseln Sie zu **Infrastruktur > Konfigurieren > Cloud-Zonen**.
- 2 Klicken Sie auf **Neue Cloud-Zone** und geben Sie Werte für die Entwicklungsumgebung ein.

Einstellungen für die Cloud-Zone	Beispielwert
Konto/Region	OurCo-AWS/us-east-1
Name	OurCo-AWS-US-East
Beschreibung	WordPress
Platzierungsrichtlinie	Standard
Funktions-Tags	env:dev

Beachten Sie, dass es sich bei allen Werten nur um Anwendungsbeispiele handelt. Ihre Zonen-spezifischen Besonderheiten sind unterschiedlich.

- 3 Klicken Sie auf **Computing** und stellen Sie sicher, dass die erwarteten Zonen vorhanden sind.
- 4 Klicken Sie auf **Erstellen**.
- 5 Wiederholen Sie den Vorgang zweimal mit Werten für die Test- und die Produktionsumgebung.

Einstellungen für die Cloud-Zone	Beispielwert
Konto/Region	OurCo-AWS/us-west-2
Name	OurCo-AWS-US-West
Beschreibung	WordPress
Platzierungsrichtlinie	Standard
Funktions-Tags	env:test

Einstellungen für die Cloud-Zone	Beispielwert
Konto/Region	OurCo-Azure/East US
Name	OurCo-Azure-East-US
Beschreibung	WordPress
Platzierungsrichtlinie	Standard
Funktions-Tags	env:prod

Nächste Schritte

Berücksichtigen Sie die Bereitstellung von Maschinen verschiedener Größen durch Hinzufügen von Konfigurationszuordnungen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Konfigurationszuordnungen](#).

WordPress-Anwendungsbeispiel: Hinzufügen von Konfigurationszuordnungen

In diesem Schritt fügt der Cloud-Administrator Konfigurationszuordnungen hinzu, um die Kapazitätsanforderungen zu berücksichtigen, die je nach Bereitstellung variieren können.

Die Konfigurationszuordnung wird informell als „T-Shirt-Sizing“ bezeichnet.

Voraussetzungen

Hinzufügen von Cloud-Zonen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Konfigurieren > Konfigurationszuordnungen**. Jede Cloud-Zone muss für kleine, mittlere und große Konfigurationen geeignet sein.
- 2 Klicken Sie auf **Neue Konfigurationszuordnung** und geben Sie Werte für die Entwicklungs-Cloud-Zone ein.

Einstellung	Beispielwert
Konfigurationsname	small
Konto/Region	OurCo-AWS/us-east-1
Wert	t2.micro
Konto/Region	OurCo-AWS/us-west-2
Wert	t2.micro
Konto/Region	OurCo-Azure/East US
Wert	Standard_A0

Beachten Sie, dass es sich bei allen Werten nur um Anwendungsbeispiele handelt. Ihre Konfigurationen werden variieren.

- 3 Klicken Sie auf **Erstellen**.

4 Wiederholen Sie den Vorgang zweimal mit Werten für mittlere und große Konfigurationen.

Einstellung	Beispielwert
Konfigurationsname	medium
Konto/Region	OurCo-AWS/us-east-1
Wert	t2.medium
Konto/Region	OurCo-AWS/us-west-2
Wert	t2.medium
Konto/Region	OurCo-Azure/East US
Wert	Standard_A3

Einstellung	Beispielwert
Konfigurationsname	large
Konto/Region	OurCo-AWS/us-east-1
Wert	t2.large
Konto/Region	OurCo-AWS/us-west-2
Wert	t2.large
Konto/Region	OurCo-Azure/East US
Wert	Standard_A7

Nächste Schritte

Planen Sie das Betriebssystem, indem Sie Image-Zuordnungen hinzufügen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Image-Zuordnungen](#).

WordPress-Anwendungsbeispiel: Hinzufügen von Image-Zuordnungen

In diesem Schritt fügt der Cloud-Administrator eine Image-Zuordnung für Ubuntu hinzu, den Host für den WordPress-Server und den zugehörigen MySQL-Datenbankserver.

Jede Cloud-Zone benötigt eine Ubuntu-Image-Zuordnung.

Voraussetzungen

Hinzufügen von Cloud-Zonen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Konfigurieren > Image-Zuordnungen**.

- 2 Klicken Sie auf **Neue Image-Zuordnung** und geben Sie Werte für Ubuntu-Server ein.

Einstellung	Beispielwert
Image-Name	ubuntu-16
Konto/Region	OurCo-AWS/us-east-1
Wert	ubuntu-16.04-server-cloudimg-amd64
Konto/Region	OurCo-AWS/us-west-2
Wert	ubuntu-16.04-server-cloudimg-amd64
Konto/Region	OurCo-Azure/East US
Wert	azul-zulu-ubuntu-1604-923eng

Beachten Sie, dass es sich bei allen Werten nur um Anwendungsbeispiele handelt. Ihre Images sind unterschiedlich.

- 3 Klicken Sie auf **Erstellen**.

Nächste Schritte

Fügen Sie Netzwerke hinzu. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Netzwerkprofilen](#).

WordPress-Anwendungsbeispiel: Hinzufügen von Netzwerkprofilen

In diesem Schritt fügt der Cloud-Administrator jeder Cloud-Zone ein Netzwerkprofil hinzu.

In jedem Profil fügt der Administrator ein Netzwerk für die WordPress-Maschinen sowie ein zweites Netzwerk hinzu, das sich auf der anderen Seite eines möglichen Lastausgleichsdiensts befindet. Bei dem zweiten Netzwerk handelt es sich um das Netzwerk, über das die Benutzer schließlich eine Verbindung herstellen.

Voraussetzungen

Hinzufügen von Cloud-Zonen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Konfigurieren > Netzwerkprofile**.
- 2 Klicken Sie auf **Neues Netzwerkprofil** und erstellen Sie ein Profil für die Cloud-Entwicklungszone.

Einstellung „Netzwerkprofiltyp“	Beispielwert
Konto/Region	OurCo-AWS/us-east-1
Name	devnets
Beschreibung	WordPress
Funktions-Tags	env:dev

3 Klicken Sie auf **Netzwerke** und dann auf **Netzwerk hinzufügen**.

4 Wählen Sie „wpnet“, „appnet-public“ aus und klicken Sie auf **Hinzufügen**.

Beachten Sie, dass es sich bei allen Werten nur um Anwendungsbeispiele handelt. Ihre Netzwerknamen sind unterschiedlich.

5 Klicken Sie auf **Erstellen**.

In diesem WordPress-Beispiel ist es nicht notwendig, dass Sie Einstellungen für Netzwerkrichtlinien oder Netzwerksicherheit angeben.

6 Wiederholen Sie den Vorgang zweimal, um ein Netzwerkprofil für die Cloud-Test- und Cloud-Produktionszonen des WordPress-Beispiels zu erstellen. Fügen Sie in jedem Fall die Netzwerke „wpnet“ und „appnet-puplic“ hinzu.

Einstellung „Netzwerkprofiltyp“	Beispielwert
Konto/Region	OurCo-AWS/us-west-2
Name	testnets
Beschreibung	WordPress
Funktions-Tags	env:test
Einstellung „Netzwerkprofiltyp“	Wert
Konto/Region	OurCo-Azure/East US
Name	prodnets
Beschreibung	WordPress
Funktions-Tags	env:prod

Nächste Schritte

Fügen Sie Speicher hinzu. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Speicherprofilen](#).

WordPress-Anwendungsbeispiel: Hinzufügen von Speicherprofilen

In diesem Schritt fügt der Cloud-Administrator jeder Cloud-Zone ein Speicherprofil hinzu.

Der Administrator platziert ein schnelles Speichergerät in der Produktionszone und ein allgemeines Speichergerät bei der Entwicklung und beim Test.

Voraussetzungen

Hinzufügen von Cloud-Zonen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Konfigurieren > Speicherprofile**.
- 2 Klicken Sie auf **Neues Speicherprofil** und erstellen Sie ein Profil für die Cloud-Entwicklungszone.

Nach der Auswahl des Kontos/der Region werden zusätzliche Felder angezeigt.

Einstellung für das Speicherprofil	Beispielwert
Konto/Region	OurCo-AWS/us-east-1
Name	OurCo-AWS-US-East-Disk
Beschreibung	WordPress
Gerätetyp	EBS
Volume-Typ	Universelle SSD
Funktions-Tags	usage:general

Beachten Sie, dass es sich bei allen Werten nur um Anwendungsbeispiele handelt.

- 3 Klicken Sie auf **Erstellen**.
- 4 Wiederholen Sie den Vorgang, um ein Profil für die Cloud-Testzone zu erstellen.

Einstellung für das Speicherprofil	Beispielwert
Konto/Region	OurCo-AWS/us-west-2
Name	OurCo-AWS-US-West-Disk
Beschreibung	WordPress
Gerätetyp	EBS
Volume-Typ	Universelle SSD
Funktions-Tags	usage:general

- 5 Wiederholen Sie den Vorgang zum Erstellen eines Profils für die Cloud-Produktionszone, die andere Einstellungen aufweist, da es sich um eine Azure-Zone handelt.

Einstellung für das Speicherprofil	Beispielwert
Konto/Region	OurCo-Azure/East US
Name	OurCo-Azure-East-US-Disk
Beschreibung	WordPress
Speichertyp	Verwaltete Festplatten
Datenträgertyp	Premium-LRS

Einstellung für das Speicherprofil	Beispielwert
Caching der Betriebssystemfestplatte	Schreibgeschützt
Caching des Datenträgers	Schreibgeschützt
Funktions-Tags	usage:fast

Nächste Schritte

Erstellen Sie ein Projekt zur Angabe von Benutzern und zur Definition von Bereitstellungseinstellungen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Erstellen eines Projekts](#).

WordPress-Anwendungsbeispiel: Erstellen eines Projekts

Ein Projekt aktiviert die Benutzer mit Bereitstellungsfunktion und konfiguriert die Bereitstellungsmöglichkeiten.

In Projekten werden die Einstellungen für den Benutzer und die Bereitstellung definiert.

- Benutzer und deren Berechtigungsrollenebene
- Priorität für Bereitstellungen, da sie in einer Cloud-Zone bereitgestellt werden
- Maximale Anzahl der Bereitstellungsinstanzen pro Cloud-Zone

Voraussetzungen

Hinzufügen von Cloud-Zonen. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#).

Verfahren

- 1 Navigieren Sie zu **Infrastruktur > Verwaltung > Projekte**.
- 2 Klicken Sie auf **Neues Projekt** und geben Sie den Namen „WordPress“ ein.
- 3 Klicken Sie auf **Benutzer** und dann auf **Benutzer hinzufügen**.
- 4 Fügen Sie E-Mail-Adressen und Rollen für die Benutzer hinzu.

Zum erfolgreichen Hinzufügen eines Benutzers muss ein VMware Cloud Services-Administrator Zugriff auf vRealize Automation Cloud Assembly für den Benutzer aktiviert haben.

Beachten Sie, dass es sich bei den angezeigten Adressen nur um Fallbeispiele handelt.

- chris.ladd@ourco.com, Mitglied
- kerry.mott@ourco.com, Mitglied
- pat.tubb@ourco.com, Administrator

- 5 Klicken Sie auf **Bereitstellung** und dann auf **Cloud-Zone hinzufügen**.

- 6 Fügen Sie die Cloud-Zonen hinzu, in denen die Benutzer eine Bereitstellung durchführen können.

Einstellung „Cloud-Projektzone“	Beispielwert
Cloud-Zone	OurCo-AWS-US-East
Bereitstellungspriorität	1
Grenzwert der Instanzen	5
Cloud-Zone	OurCo-AWS-US-West
Bereitstellungspriorität	1
Grenzwert der Instanzen	5
Cloud-Zone	OurCo-Azure-East-US
Bereitstellungspriorität	0
Grenzwert der Instanzen	1

- 7 Klicken Sie auf **Erstellen**.
- 8 Navigieren Sie zu **Infrastruktur > Konfigurieren > Cloud-Zonen** und öffnen Sie eine Zone, die unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#) erstellt wurde.
- 9 Klicken Sie auf **Projekte** und stellen Sie sicher, dass es sich bei WordPress um ein Projekt handelt, das zur Bereitstellung in der Zone berechtigt ist.
- 10 Überprüfen Sie die anderen Zonen, die unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#) erstellt wurden.

Nächste Schritte

Erstellen Sie einen einfachen Blueprint.

WordPress-Anwendungsbeispiel: Erstellen und Erweitern eines Blueprints

Als Entwickler definieren Sie die WordPress-Site in Form eines generischen vRealize Automation Cloud Assembly-Blueprints, der für jeden Cloud-Anbieter bereitgestellt werden kann.

Der Anwendungsfall-Blueprint besteht aus einem WordPress-Anwendungsserver, einem MySQL-Datenbankserver und unterstützenden Komponenten, die für AWS-, Azure- oder vSphere-basierte Clouds bereitgestellt werden können. Der Blueprint beginnt mit einigen Komponenten und nimmt zu, wenn Sie vorhandene Komponenten ändern und weitere Komponenten hinzufügen.

Die Beispiele von [WordPress-Anwendungsbeispiel: Erstellen der Infrastruktur](#) enthielten eine Infrastruktur, die von einem Cloud-Administrator festgelegt wurde:

- Zwei Cloud-Konten, AWS und Azure.
- Drei Cloud-Zonen-Umgebungen:
 - Entwicklung – OurCo-AWS-US-East
 - Test – OurCo-AWS-US-West

- Produktion – OurCo-Azure-East-US
- Konfigurationszuordnungen mit kleinen, mittleren und großen Computing-Ressourcen für jede Zone.
- In jeder Zone konfigurierte Image-Zuordnungen für Ubuntu 16.
- Netzwerkprofile mit internen und externen Subnetzen für jede Zone: devnets, testnets, prodnets.
- Speicher zur Unterstützung einer Archivierungsfestplatte, allgemeiner Speicher für Entwicklung und Test, mit schnellem Speicher für die Produktion.
- Das WordPress-Projekt umfasst alle drei Cloud-Zonen-Umgebungen sowie Benutzer, die den Anwendungsfall ausprobieren können.

Voraussetzungen

Machen Sie sich mit den Werten Ihrer Infrastruktur vertraut. So verwendet beispielsweise das Anwendungsbeispiel AWS für die Entwicklung und für Tests und Azure für die Produktion. Wenn Sie einen eigenen Blueprint erstellen, ersetzen Sie Ihre eigenen Werte, die in der Regel von Ihrem Cloud-Administrator festgelegt werden.

Verfahren

1 [WordPress-Anwendungsbeispiel: Erstellen eines einfachen Blueprints](#)

Als Entwickler beginnen Sie mit einem vRealize Automation Cloud Assembly-Blueprint, der nur ein Minimum an WordPress-Komponenten enthält, zum Beispiel nur einen Anwendungsserver.

2 [WordPress-Anwendungsbeispiel: Testen eines einfachen Blueprints](#)

Während der Entwicklung erstellen Sie in der Regel einen vRealize Automation Cloud Assembly-Blueprint, indem Sie mit den Grundlagen beginnen und dann den Blueprint bereitstellen und testen, wenn er an Größe zunimmt.

3 [WordPress-Anwendungsbeispiel: Erweitern eines Blueprints](#)

Nachdem Sie einen einfachen vRealize Automation Cloud Assembly-Blueprint erstellt und getestet haben, erweitern Sie ihn in eine mehrschichtige Anwendung, die für die Entwicklung, den Test und schließlich die Produktion bereitgestellt werden kann.

WordPress-Anwendungsbeispiel: Erstellen eines einfachen Blueprints

Als Entwickler beginnen Sie mit einem vRealize Automation Cloud Assembly-Blueprint, der nur ein Minimum an WordPress-Komponenten enthält, zum Beispiel nur einen Anwendungsserver.

vRealize Automation Cloud Assembly ist ein „Infrastruktur-als-Code“-Tool. Sie können Komponenten auf die Design-Arbeitsfläche ziehen, um den Vorgang zu starten. Anschließend vervollständigen Sie die Details mit dem Code-Editor rechts neben der Arbeitsfläche.

Mit dem Code-Editor können Sie Code direkt eingeben, ausschneiden und einfügen. Wenn Sie nicht gern Code bearbeiten, können Sie eine Ressource in der Arbeitsfläche auswählen, auf die Registerkarte **Eigenschaften** im Code-Editor klicken und die Werte dort eingeben. Die von Ihnen eingegebenen Werte werden im Code so angezeigt, als hätten Sie sie direkt eingegeben.

Voraussetzungen

Machen Sie sich mit Ihrer Infrastruktur vertraut. Die hier gezeigten Beispiele verwenden die Infrastrukturwerte von [WordPress-Anwendungsbeispiel: Erstellen der Infrastruktur](#), aber Sie würden diese durch Ihre eigene ersetzen.

Verfahren

- 1 Wechseln Sie zu **Blueprints** und klicken Sie auf **Neu**.
- 2 Geben Sie dem Blueprint den Namen **Wordpress-BP**.
- 3 Wählen Sie das **WordPress**-Projekt aus und klicken Sie auf **Erstellen**.
- 4 Ziehen Sie aus den Komponenten auf der linken Seite der Blueprint-Design-Seite zwei Cloud-unabhängige Maschinen auf die Arbeitsfläche.

Die Maschinen dienen als WordPress-Anwendungsserver (WebTier) und MySQL-Datenbankserver (DBTier).

- 5 Bearbeiten Sie auf der rechten Seite den Maschinen-YAML-Code, um Namen, Images, Konfigurationen und Einschränkungs-Tags hinzuzufügen:

```
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: 'small'
      constraints: - tag: env:dev
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: 'ubuntu-16'
      flavor: 'small'
      constraints: - tag: env:dev
```

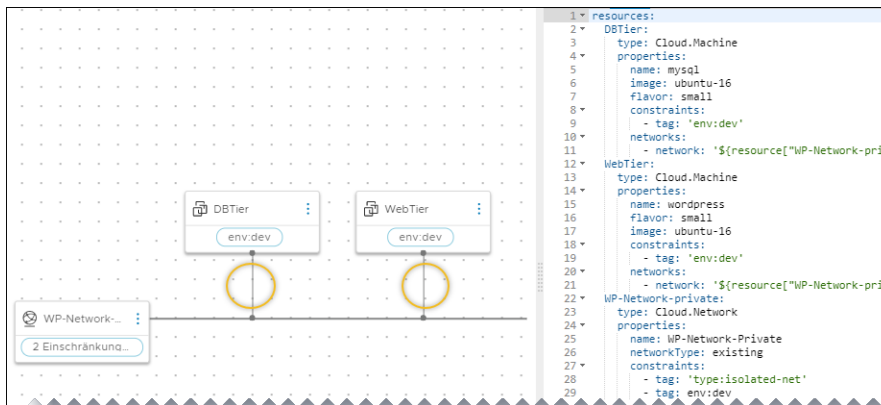
- 6 Ziehen Sie ein Cloud-unabhängiges Netzwerk auf die Arbeitsfläche und bearbeiten Sie den Code:

```
WP-Network-Private:
  type: Cloud.Network
```

```
properties:
  name: WP-Network-Private
  networkType: existing
  constraints: - tag: 'type:isolated-net' - tag: 'env:dev'
```

7 Verbinden Sie die Maschinen mit dem Netzwerk:

Klicken und halten Sie die Maustaste an der Stelle gedrückt, wo die Linie den Netzwerkblock berührt, ziehen Sie den Cursor zu einem Maschinenblock und lassen Sie ihn los.



Beachten Sie im Editor, dass der Netzwerkcode zu den beiden Maschinen hinzugefügt wird:

```
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: 'small'
      constraints:
        - tag: env:dev
      networks: - network: '${resource["WP-Network-Private"].id}'
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: 'ubuntu-16'
      flavor: 'small'
      constraints:
        - tag: env:dev
      networks: - network: '${resource["WP-Network-Private"].id}'
```

8 Fügen Sie eine Benutzereingabeaufforderung hinzu.

An einigen Stellen wurde die Infrastruktur des Anwendungsfalls für mehrere Optionen eingerichtet. Beispiel:

- Cloud-Zonen-Umgebungen für Entwicklung, Tests und Produktion

- Konfigurationszuordnungen für kleine, mittlere und große Maschinen
- Speicherlaufwerksgeschwindigkeiten für allgemeine und schnelle Nutzung

Sie können eine bestimmte Option direkt im Blueprint festlegen. Ein besserer Ansatz ist jedoch, dass der Benutzer die Option zur Blueprint-Bereitstellungszeit auswählen kann. Durch die Eingabeaufforderung für die Benutzereingabe können Sie einen Blueprint erstellen, der auf viele Arten bereitgestellt werden kann, anstatt mit vielen hartcodierten Blueprints zu arbeiten.

- a Erstellen Sie einen `inputs`-Abschnitt im Code, damit Benutzer die Maschinengröße und die Zielumgebung zur Bereitstellungszeit auswählen können. Definieren Sie die auswählbaren Werte:

```
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
```

- b Fügen Sie im `resources`-Abschnitt des Codes den Code `${input.input-name}` hinzu, um die Benutzerauswahl zu bestätigen:

```
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: 'ubuntu-16'
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: 'ubuntu-16'
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
```

```
networks:
  - network: '${resource["WP-Network-Private"].id}'
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
    constraints:
      - tag: 'type:isolated-net'
      - tag: '${input.env}'
```

- 9 Erweitern Sie schließlich den `WebTier`- und den `DBTier`-Code anhand der folgenden Beispiele. Der `WP-Network-Private`-Code benötigt keine zusätzlichen Änderungen.

Beachten Sie, dass die Verbesserungen den Anmeldezugriff auf den Datenbankserver, eine Datenbankfestplatte sowie `deployment-time`- und `cloudConfig`-Initialisierungsskripts umfassen.

Komponente	Beispiel
Zusätzliche DBTier- Eingaben	<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 databaseDiskSize: type: number default: 4 maximum: 10 title: MySQL Data Disk Size description: Database Disk Size </pre>
DBTier- Ressource	<pre> DBTier: type: Cloud.Machine properties: name: mysql image: ubuntu-16 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/ mysqlld.cnf - service mysql restart - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';" - mysql -e "FLUSH PRIVILEGES;" attachedDisks: [] </pre>
WebTier- Ressource	<pre> WebTier: type: Cloud.Machine properties: name: wordpress flavor: '\${input.size}' </pre>

Komponente	Beispiel
	<pre> image: ubuntu-16 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 - 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 </pre>

Beispiel: Beispiel für abgeschlossenen einfachen Blueprint-Code

```

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@#&$]+'
  encrypted: true
  title: Database Password
  description: Database Password
databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Database Disk Size
resources:
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu-16
      flavor: '${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 "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
      - mysql -e "FLUSH PRIVILEGES;"
    attachedDisks: []
  WebTier:

```

```

type: Cloud.Machine
properties:
  name: wordpress
  flavor: '${input.size}'
  image: ubuntu-16
  constraints:
    - tag: '${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
    - 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
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing
    constraints:
      - tag: 'type:isolated-net'
      - tag: '${input.env}'

```

Nächste Schritte

Testen Sie den Blueprint, indem Sie die Syntax überprüfen und den Blueprint bereitstellen.

WordPress-Anwendungsbeispiel: Testen eines einfachen Blueprints

Während der Entwicklung erstellen Sie in der Regel einen vRealize Automation Cloud Assembly-Blueprint, indem Sie mit den Grundlagen beginnen und dann den Blueprint bereitstellen und testen, wenn er an Größe zunimmt.

Wenn Sie sicherstellen möchten, dass eine Bereitstellung Ihren Anforderungen entsprechend funktioniert, können Sie den Blueprint mehrmals testen und bereitstellen. Schrittweise fügen Sie jeweils weitere Komponenten hinzu, testen den Blueprint dann erneut und stellen ihn erneut bereit.

Voraussetzungen

Erstellen Sie den einfachen Blueprint. Weitere Informationen hierzu finden Sie unter [WordPress-Anwendungsbeispiel: Erstellen eines einfachen Blueprints](#).

Verfahren

- 1 Klicken Sie auf **Blueprints** und öffnen Sie den WordPress-BP-Blueprint.

Der einfache Blueprint wird in der Design-Arbeitsfläche und im Code-Editor angezeigt.

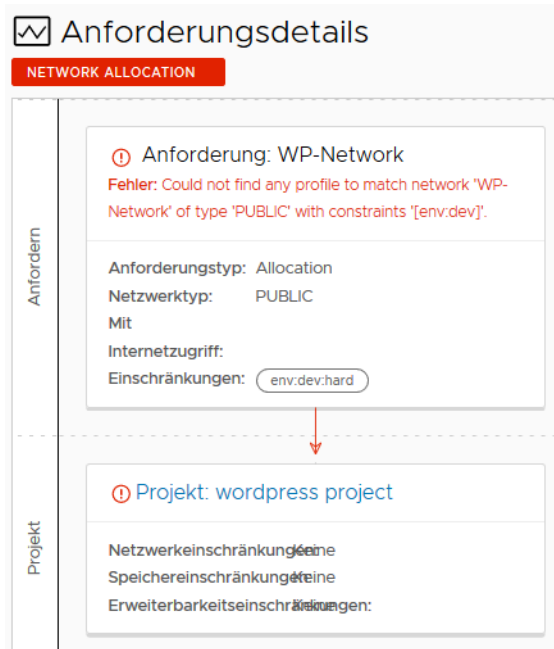
- 2 Um die Blueprint-Syntax, die Platzierung und die grundlegende Gültigkeit zu überprüfen, klicken Sie unten links auf **Test**.

- 3 Wählen Sie Eingabewerte aus und klicken Sie auf **Test**.

Environment	env.dev	▼	ⓘ
Database Tier Size *	small	▼	ⓘ
Database Username *	ouradmin		
Database Password *	•••••		
MySQL Data Disk Size	4	⬆️⬆️⬆️⬆️⬆️⬆️⬆️⬆️⬆️	ⓘ

Der Test ist nur eine Simulation und virtuelle Maschinen oder sonstige Ressourcen werden dabei nicht wirklich bereitgestellt. Die Simulation fördert potenzielle Probleme zutage, z. B. wenn keine Ressourcenkapazitäten definiert sind, die mit den harten Einschränkungen im Blueprint übereinstimmen.

Der Test enthält einen Link zu einem **Bereitstellungsdiagramm**, in dem Sie den simulierten Bereitstellungsablauf überprüfen und die aufgetretenen Fehler anzeigen können.



Eine erfolgreiche Simulation garantiert allerdings nicht, dass Sie den Blueprint ohne Fehler bereitstellen können.

- 4 Nachdem der Blueprint die Simulation erfolgreich durchlaufen hat, klicken Sie unten links auf **Bereitstellen**.
- 5 Wählen Sie **Neue Bereitstellung erstellen** aus.
- 6 Geben Sie der Bereitstellung den Namen **WordPress for OurCo** und klicken Sie auf **Weiter**.
- 7 Wählen Sie Eingabewerte aus und klicken Sie auf **Bereitstellen**.
- 8 Um zu überprüfen, ob der Blueprint erfolgreich bereitgestellt wurde, sehen Sie sich die Angaben unter **Bereitstellungen** an.

Wenn eine Bereitstellung fehlschlägt, klicken Sie auf Ihren Namen und klicken Sie auf die Registerkarte **Verlauf**, um Nachrichten anzuzeigen, die Ihnen bei der Fehlerbehebung helfen können.

Zeitstempel	Status	Ressourcentyp	Ressourcenname	Details
21. Jan. 2020, 09:41:32	REQUEST_FINISHED			
21. Jan. 2020, 09:41:31	COMPLETION_FINISHED			
21. Jan. 2020, 09:41:14	COMPLETION_IN_PROGRESS			
21. Jan. 2020, 09:40:51	CREATE_FINISHED	Cloud.Machine	Cloud_vSphere_Machine_1[...	
21. Jan. 2020, 09:33:05	CREATE_IN_PROGRESS	Cloud.Machine	Cloud_vSphere_Machine_1[...	Request is in stage STARTED and substage RESOURCE_COUNTED
21. Jan. 2020, 09:31:05	CREATE_IN_PROGRESS	Cloud.Machine	Cloud_vSphere_Machine_1[...	

Einige Verlaufeinträge verfügen unter Umständen ganz rechts über einen Link **Bereitstellungsdiagramm**. Das Diagramm ähnelt dem simulierten, in dem Sie das Flussdiagramm von vRealize Automation Cloud Assembly-Entscheidungspunkten im Bereitstellungsprozess überprüfen.

Weitere Flussdiagramme sind unter **Infrastruktur > Aktivität > Anforderungen** verfügbar.

- 9 Um zu überprüfen, ob die Anwendung funktioniert, öffnen Sie die WordPress-Startseite in einem Browser.
 - a Warten Sie, bis die WordPress-Server vollständig erstellt und initialisiert wurden.
Die Initialisierung kann je nach Umgebung 30 Minuten oder länger dauern.
 - b Um den FQDN oder die IP-Adresse der Site zu finden, wechseln Sie zu **Bereitstellungen > Topologie**.
 - c Klicken Sie auf der Arbeitsfläche auf die Webebene (WebTier) und suchen Sie die IP-Adresse im Bereich auf der rechten Seite.
 - d Geben Sie die IP-Adresse als Teil der vollständigen URL zur WordPress-Startseite ein.
In diesem Anwendungsfall lautet die vollständige URL:
`http://{IP-address}/mywordpresssite`
oder
`http://{IP-address}/mywordpresssite/wp-admin/install.php`
- 10 Nachdem Sie WordPress in einem Browser überprüft haben, nehmen Sie Blueprint-Änderungen vor und stellen Sie sie mit der Option **Vorhandene Bereitstellung aktualisieren** erneut bereit, falls die Anwendung einen höheren Aufwand erfordert.
- 11 Beachten Sie die Versionsverwaltung des Blueprints. Sie können eine funktionierende Version wiederherstellen, wenn eine Änderung dazu führt, dass die Bereitstellung fehlschlägt.
 - a Klicken Sie auf der Seite „Blueprint-Design“ auf **Version**.
 - b Geben Sie auf der Seite „Version erstellen“ **WP-1.0** ein.
Geben Sie in Versionsnamen keine Leerzeichen ein.
 - c Klicken Sie auf **Erstellen**.
Um eine Version zu überprüfen oder wiederherzustellen, klicken Sie auf der Seite „Design“ auf die Registerkarte **Versionsverlauf**.
- 12 Wenn jetzt eine einfache Bereitstellung möglich ist, versuchen Sie, Ihre erste Bereitstellungszeit zu verbessern, indem Sie CPU und Arbeitsspeicher auf den Anwendungs- und Datenbankservern erhöhen.

Führen Sie eine Aktualisierung auf eine mittlere Knotengröße für beide Komponenten durch. Wählen Sie unter Verwendung desselben Blueprints **medium** zur Bereitstellungszeit aus, führen Sie die Bereitstellung erneut durch und überprüfen Sie die Anwendung erneut.

Nächste Schritte

Erweitern Sie den Blueprint in eine produktionsfähige Anwendung, indem Sie weitere Komponenten hinzufügen.

WordPress-Anwendungsbeispiel: Erweitern eines Blueprints

Nachdem Sie einen einfachen vRealize Automation Cloud Assembly-Blueprint erstellt und getestet haben, erweitern Sie ihn in eine mehrschichtige Anwendung, die für die Entwicklung, den Test und schließlich die Produktion bereitgestellt werden kann.

Um den Blueprint zu erweitern, fügen Sie die folgenden Verbesserungen hinzu.

- Eine Option zum Clustern von Anwendungsservern für eine größere Kapazität
- Ein öffentliches Netzwerk und ein Lastausgleichsdienst vor den Anwendungsservern
- Ein Sicherungsserver mit Archivspeicher

Voraussetzungen

Erstellen Sie den einfachen Blueprint und testen Sie ihn. Weitere Informationen finden Sie unter [WordPress-Anwendungsbeispiel: Erstellen eines einfachen Blueprints](#) und [WordPress-Anwendungsbeispiel: Testen eines einfachen Blueprints](#).

Verfahren

- 1 Klicken Sie auf **Blueprints** und öffnen Sie den WordPress-BP-Blueprint.

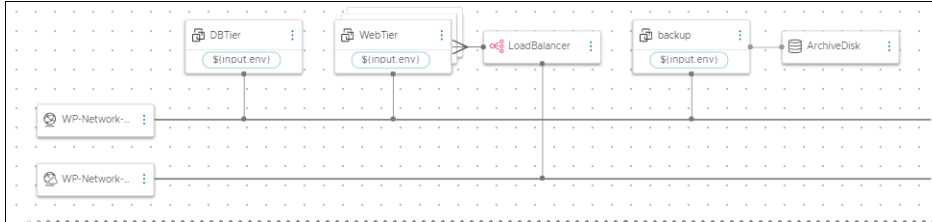
Der einfache Blueprint wird in der Design-Arbeitsfläche und im Code-Editor angezeigt.

- 2 Nehmen Sie Ergänzungen und Änderungen vor, indem Sie das Codebeispiel und die Abbildung für die Anleitung verwenden.

Sie verwenden die Benutzeroberfläche, um neue Ressourcen auf die Arbeitsfläche (zum Beispiel den Lastausgleichsdienst) zu ziehen, und schließen dann die Konfiguration im Code-Editor ab.

- a Fügen Sie eine Eingabeaufforderung vom Typ `count` hinzu, um den WordPress-Anwendungsserver in einen Cluster umzuwandeln.
- b Fügen Sie einen Cloud-unabhängigen Lastausgleichsdienst hinzu.
- c Verbinden Sie den Lastausgleichsdienst mit dem Cluster des WordPress-Anwendungsservers.
- d Fügen Sie eine Cloud-unabhängige Sicherungsmaschine hinzu.
- e Verbinden Sie die Sicherungsmaschine mit dem privaten/internen Netzwerk.
- f Fügen Sie ein Cloud-unabhängiges öffentliches/externes Netzwerk hinzu.
- g Verbinden Sie den Lastausgleichsdienst mit dem öffentlichen Netzwerk.
- h Fügen Sie einen Cloud-unabhängigen Speicherdatenträger zwecks Verwendung als Archivierungsdatenträger hinzu.
- i Verbinden Sie die Archivfestplatte mit der Sicherungsmaschine.

- j Fügen Sie eine Eingabeaufforderung `archiveusage` für die Speicherfestplattengeschwindigkeit hinzu.
- k Fügen Sie eine Eingabeaufforderung `archiveDiskSize` für die Größe der Speicherfestplatte hinzu.



- 3 Führen Sie die Bereitstellung, die Tests und Änderungen auf dieselbe Weise wie beim einfachen Blueprint durch.

Sie können vorhandene Bereitstellungen aktualisieren oder sogar neue Instanzen bereitstellen, damit Sie Bereitstellungen vergleichen können.

Ziel ist es, einen soliden, wiederholbaren Blueprint zu erreichen, der für Produktionsbereitstellungen verwendet werden kann.

Beispiel: Beispiel für vollständigen erweiterten Blueprint-Code

```
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
  databaseDiskSize:
    type: number
    default: 4
    maximum: 10
    title: MySQL Data Disk Size
    description: Database Disk Size
  count: type: integer default: 2 maximum: 5 minimum: 2 title: WordPress Cluster Size
  description: WordPress Cluster Size (Number of Nodes) archiveDiskSize: type: number default:
  4 maximum: 10 title: WordPress Archive Disk Size description: Archive Storage Disk Speed
  archiveusage: type: string enum: - 'usage:general' - 'usage:fast' description: Archive
  Storage Disk Speed title: Archive Disk Speed
  resources:
    DBTier:
      type: Cloud.Machine
      properties:
        name: mysql
        image: ubuntu-16
        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 "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
          - mysql -e "FLUSH PRIVILEGES;"
        attachedDisks: []
    WebTier:
      type: Cloud.Machine
      properties:
        name: wordpress
        flavor: '${input.size}'
        image: 'ubuntu-16'
        count: '${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
    - 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 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 ${DBTier.networks[0].address} -e "create database
wordpress_blog;"
    - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
    - sed -i -e s/"define( 'DB_NAME', 'database_name_here' );"/"define( 'DB_NAME',
'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER',
'root' );"/ /var/www/html/mywordpresssite/wp-config.php && sed -i
-e s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD',
'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php && sed
-i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '$
{DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-config.php &&
sed -i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '$
{DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-config.php
    - service apache2 reload

  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
      constraints:
        - tag: 'type:isolated-net'
        - tag: '${input.env}'

  WP-Network-Public: type: Cloud.Network properties: name: WP-Network-Public networkType:
public constraints: - tag: 'type:public-net' - tag: '${input.env}' backup: type:
Cloud.Machine properties: name: backup flavor: '${input.size}' image: 'ubuntu-16' networks:
- network: '${resource["WP-Network-Private"].id}' constraints: - tag: '${input.env}'
attachedDisks: - source: '${ArchiveDisk.id}' ArchiveDisk: type: Cloud.Volume properties:

```

```
name: ArchiveDisk capacityGb: 5 constraints: - tag: '${input.archiveusage}' - tag: '${input.env}'
```

Nächste Schritte

Definieren Sie Ihre eigene Infrastruktur und erstellen Sie Ihre eigenen Blueprints.

Weitere Informationen finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#) und [Kapitel 6 Entwerfen Ihrer vRealize Automation Cloud Assembly-Bereitstellungen](#).

VMware Cloud on AWS-Anwendungsbeispiel

In diesem vRealize Automation Cloud Assembly-Anwendungsfall wird der Vorgang zum Definieren von Ressourceninfrastruktur- und Blueprint-Einstellungen für die Bereitstellung in einer VMware Cloud on AWS-Umgebung veranschaulicht.

Dieser Vorgang setzt voraus, dass ein Cloud-Administrator das VMware Cloud on AWS-SDDC Ihrer Organisation gemäß der Beschreibung unter *Bereitstellen und Verwalten eines SDDC (Software-Defined Data Center)* im Dokument [Erste Schritte mit VMware Cloud on AWS](#) bereits konfiguriert hat.

Schauen Sie sich die schrittweise Einrichtung genau an, um den Vorgang zum Konfigurieren Ihrer Umgebung für VMware Cloud on AWS zu verstehen. Beachten Sie, dass es sich bei den angezeigten Werten nur um Anwendungsfallbeispiele handelt. Überlegen Sie sich, wo Sie Ihre eigenen Ersetzungen vornehmen würden, oder übernehmen Sie eine Hochrechnung aus den Beispielwerten, um Ihre eigene Cloud-Infrastruktur einzurichten und Ihre Bereitstellungsanforderungen zu erfüllen.

Ein detailliertes Video eines ähnlichen Workflows hat *VMware Cloud Management Technical Marketing* unter dem Titel [How to Configure VMware Cloud on AWS for Cloud Assembly](#) (Konfigurieren von VMware Cloud on AWS für Cloud Assembly) bereitgestellt.

Verfahren

1 [Konfigurieren eines grundlegenden VMware Cloud on AWS-Workflows in vRealize Automation Cloud Assembly](#)

Dieser vRealize Automation Cloud Assembly-Anwendungsfall zeigt den Prozess, in dem die Ressourceninfrastruktur und ein entsprechender Blueprint für die Bereitstellung in einer VMware Cloud on AWS-Umgebung definiert werden.

2 [Konfigurieren eines isolierten Netzwerks im VMware Cloud on AWS-Workflow in vRealize Automation Cloud Assembly](#)

In diesem Verfahren fügen Sie ein isoliertes Netzwerk für Ihre VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly hinzu.

Konfigurieren eines grundlegenden VMware Cloud on AWS-Workflows in vRealize Automation Cloud Assembly

Dieser vRealize Automation Cloud Assembly-Anwendungsfall zeigt den Prozess, in dem die Ressourceninfrastruktur und ein entsprechender Blueprint für die Bereitstellung in einer VMware Cloud on AWS-Umgebung definiert werden.

In diesem Verfahren konfigurieren Sie Infrastruktur, die die Blueprint-Bereitstellung für Ressourcen in Ihrer vorhandenen VMware Cloud on AWS-Umgebung unterstützt.

Voraussetzungen

- Bevor Sie ein VMware Cloud on AWS-Cloud-Konto in vRealize Automation Cloud Assembly erstellen und konfigurieren können, müssen Sie Teil einer Organisation in einer vorhandenen VMware Cloud on AWS-SDDC-Umgebung sein. Informationen zum Konfigurieren des VMware Cloud on AWS-Diensts finden Sie in der [Dokumentation zu VMware Cloud on AWS](#).
- Um die erforderliche Verbindung zwischen Ihrem vorhandenen VMware Cloud on AWS-Host-SDDC in vCenter und einem VMware Cloud on AWS-Cloud-Konto in vRealize Automation Cloud Assembly zu erleichtern, müssen Sie eine Netzwerkverbindung bereitstellen und Firewallregeln hinzufügen, indem Sie ein VPN oder ein ähnliches Netzwerk verwenden. Weitere Informationen finden Sie unter [Vorbereiten des VMware Cloud on AWS-SDDC für die Verbindung mit VMware Cloud on AWS-Cloud-Konten in vRealize Automation](#).

Verfahren

- 1 [Vorbereiten des VMware Cloud on AWS-SDDC für die Verbindung mit VMware Cloud on AWS-Cloud-Konten in vRealize Automation](#)

Bei Verwendung von VMware Cloud on AWS-Cloud-Konten in Ihrer lokalen vRealize Automation Cloud Assembly-Umgebung müssen Sie eine Netzwerkverbindung erstellen, um die Kommunikation zwischen Ihrem SDDC in vCenter und allen VMware Cloud on AWS-Cloud-Konten in vRealize Automation zu unterstützen.

- 2 [Erstellen eines VMware Cloud on AWS-Cloud-Kontos in vRealize Automation innerhalb eines Beispiel-Workflows](#)

In diesem Verfahren erstellen Sie ein VMware Cloud on AWS-Cloud-Konto in vRealize Automation.

- 3 [Erstellen einer Cloud-Zone für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly](#)

In diesem Schritt erstellen Sie eine Cloud-Zone zur Angabe einer Computing-Ressource, auf die der CloudAdmin-Benutzer beim Arbeiten mit VMware Cloud on AWS in vRealize Automation Cloud Assembly zugreifen kann.

4 Konfigurieren von Netzwerk- und Speicherprofilen für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly

In diesem Schritt konfigurieren Sie ein Netzwerk- und Speicherprofil zur Angabe von Ressourcen, die einem VMware Cloud on AWS-CloudAdmin-Benutzer in vRealize Automation Cloud Assembly zur Verfügung stehen.

5 Erstellen eines Projekts zur Unterstützung von VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly

In diesem Schritt definieren Sie ein vRealize Automation Cloud Assembly-Projekt, das zum Steuern der Ressourcen verwendet werden kann, die für VMware Cloud on AWS-Bereitstellungen verfügbar sind.

6 Definieren einer vCenter-Maschinenressource in einem Blueprint-Design zur Unterstützung der VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly

In diesem Schritt ziehen Sie eine vCenter-Maschinenressource auf die Design-Arbeitsfläche und fügen Einstellungen für eine VMware Cloud on AWS-Bereitstellung hinzu.

Vorbereiten des VMware Cloud on AWS-SDDC für die Verbindung mit VMware Cloud on AWS-Cloud-Konten in vRealize Automation

Bei Verwendung von VMware Cloud on AWS-Cloud-Konten in Ihrer lokalen vRealize Automation Cloud Assembly-Umgebung müssen Sie eine Netzwerkverbindung erstellen, um die Kommunikation zwischen Ihrem SDDC in vCenter und allen VMware Cloud on AWS-Cloud-Konten in vRealize Automation zu unterstützen.

Um die erforderliche Verbindung zwischen Ihrem vorhandenen VMware Cloud on AWS-Host-SDDC in vCenter und einem VMware Cloud on AWS-Cloud-Konto in vRealize Automation zu erleichtern, müssen Sie eine Netzwerkverbindung zwischen den beiden Elementen mithilfe eines VPN oder ähnlicher Netzwerkmittel bereitstellen.

Verfahren

- 1 Konfigurieren Sie eine VPN-Verbindung über das öffentliche Internet oder AWS Direct Connect.

Weitere Informationen finden Sie unter *VMware Cloud on AWS – Netzwerk und Sicherheit* in der [VMware Cloud on AWS-Dokumentation](#).

- 2 Stellen Sie sicher, dass der vCenter Server-FQDN unter einer privaten IP-Adresse im Verwaltungsnetzwerk aufgelöst werden kann.

Weitere Informationen finden Sie unter *VMware Cloud on AWS – Netzwerk und Sicherheit* in der [VMware Cloud on AWS-Dokumentation](#).

3 Konfigurieren Sie erforderliche Firewallregeln.

Sie müssen Firewallregeln für das Verwaltungs-Gateway in der VMware Cloud on AWS-Konsole des SDDC konfigurieren, um die Kommunikation zu unterstützen. Die Regeln müssen sich im Abschnitt mit Firewallregeln für das **Verwaltungs-Gateway** befinden. Erstellen Sie die Firewallregeln mithilfe der Optionen auf der Registerkarte **Netzwerke und Sicherheit** in der SDDC-Konsole.

- Begrenzen Sie den Netzwerkdatenverkehr zu ESXi für HTTPS (TCP 443)-Dienste auf die ermittelte IP-Adresse der/des vRealize Automation-Appliance/Servers oder die Lastausgleichsdienst-VIP für vRealize Automation.
- Begrenzen Sie den Netzwerkdatenverkehr zu vCenter für ICMP (Alle ICMP-), SSO (TCP 7444)- und HTTPS (TCP 443)-Dienste auf die ermittelte IP-Adresse der/des vRealize Automation-Appliance/Servers oder die Lastausgleichsdienst-VIP für vRealize Automation.
- Begrenzen Sie den Netzwerkdatenverkehr zu NSX-T Manager für HTTPS (TCP 443)-Dienste auf die ermittelte IP-Adresse der/des vRealize Automation-Appliances/Servers oder die Lastausgleichsdienst-VIP für vRealize Automation.

Die erforderlichen Firewallregeln werden in der folgenden Tabelle zusammengefasst.

Tabelle 3-1. Erforderliche Firewallregeln für das Verwaltungs-Gateway – Übersicht

Name	Quelle	Ziel	Dienst
vCenter	CIDR-Block des lokalen Datencenters	vCenter	Beliebig (Gesamter Datenverkehr)
vCenter-Ping	Alle	vCenter	ICMP (Gesamtes ICMP)
NSX Manager	CIDR-Block des lokalen Datencenters	NSX Manager	Beliebig (Gesamter Datenverkehr)
Lokal zu ESXi-Ping	CIDR-Block des lokalen Datencenters	Nur ESXi-Verwaltung	ICMP (Gesamtes ICMP)
Lokal zu ESXi Remote Console und Bereitstellung	CIDR-Block des lokalen Datencenters	Nur ESXi-Verwaltung	TCP 902
Lokal zu SDDC-VM	CIDR-Block des lokalen Datencenters	CIDR-Block des logischen SDDC-Netzwerks	Beliebig (Gesamter Datenverkehr)
SDDC-VM zu lokal	CIDR-Block des logischen SDDC-Netzwerks	CIDR-Block des lokalen Datencenters	Beliebig (Gesamter Datenverkehr)

Zugehörige Informationen finden Sie unter *Netzwerk und Sicherheit von VMware Cloud on AWS* und im *Betriebshandbuch für VMware Cloud on AWS* in der [VMware Cloud on AWS-Dokumentation](#).

Ergebnisse

Nachdem Sie den erforderlichen Zugriff auf Gateways und die Firewallregeln konfiguriert haben, können Sie mit dem Vorgang zum Erstellen eines VMware Cloud on AWS-Cloud-Kontos fortfahren.

Erstellen eines VMware Cloud on AWS-Cloud-Kontos in vRealize Automation innerhalb eines Beispiel-Workflows

In diesem Verfahren erstellen Sie ein VMware Cloud on AWS-Cloud-Konto in vRealize Automation.

Weitere Informationen finden Sie in der [Dokumentation zu VMware Cloud on AWS](#).

Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.


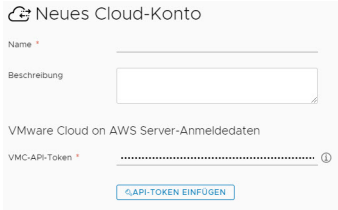
Voraussetzungen

- Bei diesem Verfahren wird davon ausgegangen, dass Sie über die erforderlichen Administratoranmeldedaten verfügen, einschließlich VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter, und dass der HTTPS-Zugriff auf Port 443 aktiviert wurde. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Um die erforderliche Verbindung zwischen Ihrem vorhandenen VMware Cloud on AWS-Host-SDDC in vCenter und einem VMware Cloud on AWS-Cloud-Konto in vRealize Automation zu erleichtern, müssen Sie eine Netzwerkverbindung und Firewallregeln bereitstellen, indem Sie ein VPN oder ein ähnliches Netzwerk verwenden. Weitere Informationen hierzu finden Sie unter [Vorbereiten des VMware Cloud on AWS-SDDC für die Verbindung mit VMware Cloud on AWS-Cloud-Konten in vRealize Automation](#). Wenn Sie einen externen HTTP-Internet-Proxy verwenden, muss dieser für IPv4 konfiguriert sein.
- Wenn Sie nicht über externen Internetzugriff verfügen, konfigurieren Sie einen Internet-Proxyserver. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Cloud-Konten** aus.
- 2 Klicken Sie auf **Cloud-Konto hinzufügen**, wählen Sie VMware Cloud on AWS aus und geben Sie Werte ein.

Beispielwerte und zusätzliche Informationen werden in der folgenden Tabelle bereitgestellt.

Einstellung	Beispielwert und Anweisung	Beschreibung
VMC-API-Token	<ol style="list-style-type: none"> Klicken Sie auf das <i>i</i>-Hilfensymbol am Ende der Zeile VMC-API-Token und klicken Sie auf die Seite API-Token im Hilfetextfeld, um die Registerkarte API-Token auf der Mein Konto-Seite Ihrer Organisation zu öffnen. Klicken Sie auf Token generieren, um die Optionen für Neues API-Token generieren anzuzeigen. Geben Sie einen Namen für das neue Token ein, z. B. myinitials_mytoken. Legen Sie die Token-TTL auf nie ablaufen fest. Wenn Sie ein Token erstellen, das auf „ablaufen“ festgelegt ist, funktionieren die VMware Cloud on AWS-Vorgänge von vRealize Automation nicht mehr, wenn das Token abläuft, und funktionieren erst wieder, nachdem Sie das Cloud-Konto mit einem neuen Token aktualisiert haben. Wählen Sie im Abschnitt Geltungsbereiche definieren die Option Alle Rollen.  Klicken Sie auf Generieren. Klicken Sie auf der Seite des generierten Tokens auf Kopieren und dann auf Weiter. Kehren Sie zur Seite Neues Cloud-Konto zurück, fügen Sie das kopierte Token in die Zeile VMC-API-Token ein und klicken Sie auf API-Token einfügen.  	<p>Sie können auf der verknüpften Seite API-Token für Ihre Organisation ein neues Token erstellen oder ein vorhandenes Token verwenden.</p> <p>Im Abschnitt Geltungsbereiche definieren müssen mindestens die folgenden Rollen für das API-Token festgelegt werden:</p> <ul style="list-style-type: none"> ■ Organisationsrollen <ul style="list-style-type: none"> ■ Organisationsmitglied ■ Organisationsbesitzer ■ Dienstrollen – VMware Cloud on AWS <ul style="list-style-type: none"> ■ Administrator ■ NSX Cloud-Administrator ■ NSX Cloud-Auditor <p>Hinweis Sie können das generierte Token kopieren, herunterladen oder drucken. Wenn Sie diese Seite verlassen, können Sie das generierte Token nicht mehr abrufen.</p> <p>Fügen Sie das generierte oder bereitgestellte Token ein, um eine Verbindung zur verfügbaren SDDC-Umgebung im VMware Cloud on AWS-Abonnement Ihrer Organisation herzustellen, und füllen Sie die Liste der SDDC-Namen auf.</p> <p>Wenn sich die vRealize Automation- und VMware Cloud on AWS-Dienste in unterschiedlichen Organisationen befinden, sollten Sie zur VMware Cloud on AWS-Organisation wechseln und dann das Token generieren.</p> <p>Weitere Informationen zu API-Token finden Sie unter Generieren von API-Token.</p>
SDDC-Name	Wählen Sie für dieses Beispiel „Datacenter:Datacenter-abz“ aus.	Treffen Sie eine Auswahl in der Liste der verfügbaren SDDCs in Ihrem VMware Cloud on AWS-Abonnement. Die Liste der SDDCs

Einstellung	Beispielwert und Anweisung	Beschreibung
	Die Einträge für den vCenter- und NSX-T-FQDN werden automatisch mit dem SDDC-Namen befüllt. Wenn bereits ein Cloud-Proxy für das SDDC bereitgestellt wurde, wird der Wert für den Cloud-Proxy ebenfalls automatisch befüllt.	<p>basiert auf dem VMware Cloud on AWS-API-Token.</p> <p>NSX-V-SDDCs werden nicht mit vRealize Automation unterstützt und in der Liste der verfügbaren SDDCs nicht angezeigt.</p>
vCenter-IP-Adresse/-FQDN	Die Adresse wird automatisch basierend auf der SDDC-Auswahl angegeben.	<p>Geben Sie die IP-Adresse oder den FQDN des vCenter Server im angegebenen SDDC ein.</p> <p>Als IP-Adresse wird standardmäßig die private IP-Adresse verwendet. Je nach Typ der Netzwerkverbindung, die für den Zugriff auf Ihr SDDC verwendet wird, kann sich die Standardadresse von der IP-Adresse des NSX Manager-Servers im angegebenen SDDC unterscheiden.</p>
IP-Adresse/FQDN in NSX Manager	Die Adresse wird automatisch basierend auf der SDDC-Auswahl angegeben.	<p>Gibt die IP-Adresse oder den FQDN des NSX Managers im angegebenen SDDC ein.</p> <p>Als IP-Adresse wird standardmäßig die private IP-Adresse verwendet. Je nach Typ der Netzwerkverbindung, die für den Zugriff auf Ihr SDDC verwendet wird, kann sich die Standardadresse von der IP-Adresse des NSX Manager-Servers im angegebenen SDDC unterscheiden.</p> <p>VMware Cloud on AWS-Cloud-Konten bieten Unterstützung für NSX-T.</p>
Benutzername und Kennwort in vCenter	Als Benutzername wird automatisch „cloudadmin@vmc.local“ angegeben.	<p>Geben Sie Ihren vCenter-Benutzernamen für das angegebene SDDC ein, wenn er sich von der Standardeinstellung unterscheidet.</p> <p>Der angegebene Benutzer benötigt CloudAdmin-Anmeldedaten. Der Benutzer benötigt keine CloudGlobalAdmin-Anmeldedaten.</p> <p>Geben Sie das Benutzerkennwort ein.</p>
Validieren	Klicken Sie auf Validieren .	Mit „Validieren“ werden Ihre Zugriffsrechte für das angegebene vCenter bestätigt und es wird überprüft, ob das vCenter ausgeführt wird.
Name und Beschreibung	<p>Geben Sie OurCo-VMC als Namen für das Cloud-Konto ein.</p> <p>Geben Sie Beispielbereitstellung für VMC als Beschreibung für das Cloud-Konto ein.</p>	
Bereitstellung für diese Datencenter zulassen	Diese Informationen sind schreibgeschützt.	Listet verfügbare Datencenter in der angegebenen VMware Cloud on AWS-SDDC-Umgebung auf.

Einstellung	Beispielwert und Anweisung	Beschreibung
Cloud-Zone erstellen	Deaktivieren Sie das Kontrollkästchen. In diesem Beispiel erstellen Sie später im Workflow eine Cloud-Zone.	Weitere Informationen hierzu finden Sie unter Weitere Informationen zu vRealize Automation Cloud Assembly-Cloud-Zonen .
Funktions-Tags	Lassen Sie dieses Feld leer. In diesem Workflow werden keine Funktions-Tags verwendet.	Verwenden Sie Tags gemäß der Tag-Strategie Ihrer Organisation. Weitere Informationen finden Sie unter Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen und Erstellen einer Tagging-Strategie .

3 Klicken Sie auf **Hinzufügen**.

Ergebnisse

Ressourcen, wie z. B. Maschinen und Volumes, werden vom VMware Cloud on AWS-SDDC-Datencenter erfasst und im Abschnitt **Ressourcen** der Registerkarte vRealize Automation **Infrastruktur** aufgelistet.

Nächste Schritte

[Erstellen einer Cloud-Zone für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly](#).

Erstellen einer Cloud-Zone für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly

In diesem Schritt erstellen Sie eine Cloud-Zone zur Angabe einer Computing-Ressource, auf die der CloudAdmin-Benutzer beim Arbeiten mit VMware Cloud on AWS in vRealize Automation Cloud Assembly zugreifen kann.

In VMware Cloud on AWS lauten die beiden wichtigsten Administratoranmeldedaten „CloudGlobalAdmin“ und „CloudAdmin“. vRealize Automation Cloud Assembly wurde für die Unterstützung des CloudAdmin-Benutzers entwickelt. Stellen Sie die Ressourcen bereit, die für einen VMware Cloud on AWS CloudAdmin-Benutzer verfügbar sind. Führen Sie keine Bereitstellung für Ressourcen durch, die VMware Cloud on AWS CloudGlobalAdmin-Anmeldedaten benötigen.

Cloud-Zonen geben die Computing-Ressourcen an, auf denen ein Projekt-Blueprint Maschinen, Netzwerke und Speicher bereitstellt. Weitere Informationen hierzu finden Sie unter [Weitere Informationen zu vRealize Automation Cloud Assembly-Cloud-Zonen](#).

Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.

Voraussetzungen

- Schließen Sie das Verfahren [Erstellen eines VMware Cloud on AWS-Cloud-Kontos in vRealize Automation innerhalb eines Beispiel-Workflows](#) ab.

- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die erforderlichen Administratoranmeldedaten verfügen, einschließlich der VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Konfigurieren > Cloud-Zonen** aus.
- 2 Klicken Sie auf **Neue Cloud-Zone** und geben Sie Werte für die VMware Cloud on AWS-Umgebung ein.

Einstellung	Beispielwert
Konto/Region	OurCo-VMC / Datacenter:Datacenter-abz Dies ist das Cloud-Konto und die zugehörige Region, die Sie im vorherigen Schritt, Erstellen eines VMware Cloud on AWS-Cloud-Kontos in vRealize Automation innerhalb eines Beispiel-Workflows , definiert haben.
Name	VMC_cloud_zone-1
Beschreibung	Nur VMware Cloud on AWS-Ressourcen
Platzierungsrichtlinie	Standard
Funktions-Tags	Lassen Sie dieses Feld leer. In diesem Workflow werden keine Funktions-Tags verwendet.

- 3 Klicken Sie auf die Registerkarte **Berechnen**.
- 4 Wie in Bereich 1 unten dargestellt, suchen und wählen Sie eine Computing-Ressource aus, die dem CloudAdmin-Benutzer zur Verfügung steht. Verwenden Sie für dieses Beispiel die Ressource mit dem Namen Cluster 1/ Compute-ResourcePool.

Cluster 1/ Compute-ResourcePool ist die standardmäßige Computing-Ressource für VMware Cloud on AWS.



- 5 Wie in Bereich 2 oben dargestellt, fügen Sie den Tag-Namen `vmc_placements_abz` hinzu.

Tags

1 Objekt(e) ausgewählt

Tags hinzufügen

vmc_placements_abz X

 Neues Tag eingeben

Tags entfernen

keine Tags ⓘ

- 6 Filtern Sie die Computing-Ressourcen, die in dieser Cloud-Zone verwendet werden, indem Sie `vmc_placements_abz` im Abschnitt **Filter-Tags** eingeben.
- 7 Klicken Sie auf **Speichern**.

<input type="checkbox"/>	Name	Konto/Region	Typ	Tags
<input type="checkbox"/>	ComputeClusterA	LK-TEST 测试资源池A中正在运行的Ubuntu / NSX62-Scale-DC	common.title cluster	Cluster ComputeClusterA
<input checked="" type="checkbox"/>	ComputeClusterA-New	nsx-vm 测试资源池A中正在运行的Ubuntu / NSX621-DataCenter	common.title cluster	ComputeClusterA
<input type="checkbox"/>	ComputeClusterA / Scale	270_VC_account 测试资源池A中正在运行的Ubuntu / NSX62-Scale-DC	ResourcePool	ComputeClusterA

In diesem Beispiel steht dem CloudAdmin-Benutzer nur die Computing-Ressource mit dem Namen Cluster 1/ Compute-ResourcePool zur Verfügung.

Nächste Schritte

[Konfigurieren von Netzwerk- und Speicherprofilen für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly.](#)

Konfigurieren von Netzwerk- und Speicherprofilen für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly

In diesem Schritt konfigurieren Sie ein Netzwerk- und Speicherprofil zur Angabe von Ressourcen, die einem VMware Cloud on AWS-CloudAdmin-Benutzer in vRealize Automation Cloud Assembly zur Verfügung stehen.

Obwohl auch ein Image- und Konfigurationswert benötigt werden, gibt es keine eindeutigen spezifischen Informationen zu den VMware Cloud on AWS-Benutzeranmeldedaten. In diesem Beispiel verwenden Sie einen Konfigurationswert vom Typ `small` und einen Image-Wert vom Typ `ubuntu-16`, wenn Sie den Blueprint definieren.

Allgemeine Informationen zu Zuordnungen und Profilen finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#).

Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.

Voraussetzungen

- Erstellen Sie eine Cloud-Zone. Weitere Informationen hierzu finden Sie unter [Erstellen einer Cloud-Zone für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die erforderlichen Administratoranmeldedaten verfügen, einschließlich der VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).

Verfahren

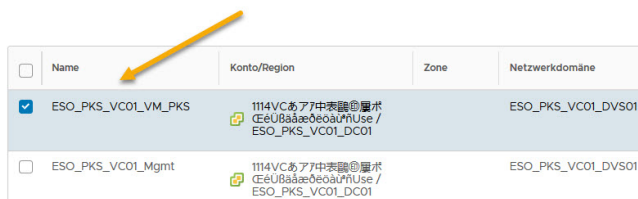
- 1 Definieren Sie ein Netzwerkprofil für VMware Cloud on AWS-Bereitstellungen.

- a Wählen Sie **Infrastruktur > Konfigurieren > Netzwerkprofile** aus und klicken Sie auf **Neues Netzwerkprofil**.

Einstellung	Beispielwert
Konto/Region	OurCo-VMC / Datacenter:Datacenter-abz Hinweis Wählen Sie das VMware Cloud on AWS-Cloud-Konto und das zugehörige SDDC aus, das Sie in Erstellen eines VMware Cloud on AWS-Cloud-Kontos in vRealize Automation innerhalb eines Beispiel-Workflows erstellt haben.
Name	vmc-network1
Beschreibung	Enthält Netzwerke, auf die Blueprint-Administratoren mit VMware Cloud on AWS-CloudAdmin-Anmeldedaten zugreifen können.

- b Klicken Sie auf die Registerkarte **Netzwerk** und dann auf **Netzwerk hinzufügen**.
- c Wählen Sie ein Netzwerk aus, auf dem ein VMware Cloud on AWS-Benutzer mit CloudAdmin-Anmeldedaten eine Bereitstellung durchführen kann, wie z. B. `sddc-cgw-network-1`.

Netzwerk hinzufügen



<input type="checkbox"/>	Name	Konto/Region	Zone	Netzwerkdomäne
<input checked="" type="checkbox"/>	ESO_PKS_VCO1_VM_PKS	114VCアア7中表監創量ホ CeU8aaæbeoauHUse / ESO_PKS_VCO1_DC01		ESO_PKS_VCO1_DVS01
<input type="checkbox"/>	ESO_PKS_VCO1_Mgmt	114VCアア7中表監創量ホ CeU8aaæbeoauHUse / ESO_PKS_VCO1_DC01		ESO_PKS_VCO1_DVS01

- 2 Speichern Sie das Netzwerkprofil.

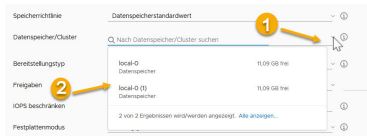
3 Definieren Sie ein Speicherprofil für VMware Cloud on AWS-Bereitstellungen.

Konfigurieren Sie ein Speicherprofil für einen Datenspeicher/Cluster, auf den der CloudAdmin-Benutzer zugreifen kann.

- a Wählen Sie **Infrastruktur > Konfigurieren > Speicherprofile** aus und klicken Sie auf **Neues Speicherprofil**.

Einstellung	Beispielwert
Konto/Region	OurCo-VMC / Datacenter:Datacenter-abz Wählen Sie das VMware Cloud on AWS-Cloud-Konto und das zugehörige SDDC aus, das Sie in Erstellen eines VMware Cloud on AWS-Cloud-Kontos in vRealize Automation innerhalb eines Beispiel-Workflows erstellt haben.
Name	vmc-storage1
Beschreibung	Enthält den Datenspeicher-Cluster, auf dem Blueprint-Administratoren mit VMware Cloud on AWS-CloudAdmin-Anmeldedaten Bereitstellungen durchführen können.

- b Wählen Sie im Dropdown-Menü **Datenspeicher/Cluster** den Datenspeicher **WorkloadDatastore** aus.



Für VMware Cloud on AWS in vRealize Automation Cloud Assembly muss die Speicherrichtlinie den Datenspeicher **WorkloadDatastore** verwenden, um die VMware Cloud on AWS-Bereitstellung zu unterstützen.

4 Speichern Sie das Speicherprofil.

Nächste Schritte

[Erstellen eines Projekts zur Unterstützung von VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly.](#)

Erstellen eines Projekts zur Unterstützung von VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly

In diesem Schritt definieren Sie ein vRealize Automation Cloud Assembly-Projekt, das zum Steuern der Ressourcen verwendet werden kann, die für VMware Cloud on AWS-Bereitstellungen verfügbar sind.

Weitere Informationen zu Projekten finden Sie unter [Funktionsweise von vRealize Automation Cloud Assembly-Projekten zur Bereitstellungszeit](#).

Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.

Voraussetzungen

- Schließen Sie das Verfahren [Konfigurieren von Netzwerk- und Speicherprofilen für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly](#) ab.
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die erforderlichen Administratoranmeldedaten verfügen, einschließlich der VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).

Verfahren

- 1 Klicken Sie auf **Infrastruktur > Konfigurieren > Projekte**.
- 2 Klicken Sie auf **Neues Projekt** und geben Sie den Projektnamen `VMC_proj-1_abz` ein.
- 3 Klicken Sie auf **Benutzer** und dann auf **Benutzer hinzufügen**.

Die Benutzer benötigen CloudAdmin-Anmeldedaten für das VMware Cloud on AWS-Abonnement ihrer Organisation.

- `chris.gray@ourco.com`, Administrator
- `kerry.white@ourco.com`, Mitglied

- 4 Klicken Sie auf **Bereitstellung** und dann auf **Cloud-Zone hinzufügen**.
- 5 Fügen Sie die Cloud-Zone hinzu, die Sie im vorherigen Schritt konfiguriert haben.

Einstellung	Beispielwert
Cloud-Zone	VMC_cloud_zone-1 Sie haben diese Cloud-Zone im vorherigen Schritt, Erstellen einer Cloud-Zone für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly , erstellt.
Bereitstellungspriorität	1
Grenzwert der Instanzen	3

- 6 Ignorieren Sie in diesem Beispiel die anderen Optionen.

Nächste Schritte

Erstellen Sie einen Blueprint, der in Ihrer VMware Cloud on AWS-Umgebung bereitgestellt werden soll. Weitere Informationen hierzu finden Sie unter [Definieren einer vCenter-Maschinenressource in einem Blueprint-Design zur Unterstützung der VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly](#).

Definieren einer vCenter-Maschinenressource in einem Blueprint-Design zur Unterstützung der VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly

In diesem Schritt ziehen Sie eine vCenter-Maschinenressource auf die Design-Arbeitsfläche und fügen Einstellungen für eine VMware Cloud on AWS-Bereitstellung hinzu.

Erstellen Sie ein Blueprint-Design, das Sie verfügbaren VMware Cloud on AWS-Ressourcen bereitstellen können.

Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.

Voraussetzungen

- Dieses Verfahren setzt voraus, dass Sie über Anmeldedaten als Blueprint-Designer verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- In diesem Verfahren wird davon ausgegangen, dass Sie über VMware Cloud on AWS CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter (Datacenter:Datacenter-abz) verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Konfigurieren Sie die Ressourceninfrastruktur und das Projekt gemäß der Beschreibung in den vorherigen Abschnitten.

Verfahren

- 1 Klicken Sie auf die Registerkarte **Design** und dann auf **Neu**.

Einstellung	Beispielwert
Name	vmc-bp_abz
Beschreibung	1
Projekt	VMC_proj-1_abz Dies ist das zuvor erstellte Projekt, das die ebenfalls bereits erstellte Cloud-Zone unterstützt. Das Projekt ist nun mit der Cloud-Zone verknüpft, die wiederum mit dem/der VMware Cloud on AWS-Cloud-Konto/-Region verknüpft ist, die Sie zuvor erstellt haben.

- 2 Ziehen Sie eine vSphere-Maschinenressource auf die Arbeitsfläche.
- 3 Bearbeiten Sie den folgenden (fett gedruckten) Blueprint-Ressourcencode in der Maschinenressource.

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

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

Bei `image` kann es sich um einen beliebigen Wert handeln, der für Ihre Bereitstellungsanforderungen geeignet ist.

Sie müssen die `folderName: Workloads`-Anweisung zum Code des Blueprint-Designs hinzufügen, um die VMware Cloud on AWS-Bereitstellung zu unterstützen. Die Einstellung `folderName: Workloads` unterstützt die CloudAdmin-Anmeldedaten in der VMware Cloud on AWS-SDDC-Umgebung und ist obligatorisch.

Hinweis: Obwohl die im obigen Codebeispiel angezeigte Einstellung `folderName: Workloads` notwendig ist, können Sie sie direkt in den Code des Blueprint-Designs einfügen (wie oben gezeigt) oder zur verknüpften Cloud-Zone bzw. zum verknüpften Objekt hinzufügen. Wenn die Einstellung an mehr als einer dieser drei Positionen angegeben ist, lautet die Reihenfolge folgendermaßen:

- Die Projekteinstellung überschreibt die Blueprint-Design- und die Cloud-Zoneneinstellung.
- Die Blueprint-Design-Einstellung überschreibt die Cloud-Zoneneinstellung.

Hinweis: Sie können die Einstellungen für `cpuCount` und `totalMemoryMB` optional durch einen Eintrag vom Typ `flavor` (Größenänderung) ersetzen, wie nachfolgend dargestellt:

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

Wenn für die Cloud-Zone der Ordnerwert auf **Workloads** festgelegt ist, müssen Sie die Eigenschaft `folderName` im Blueprint-Design nicht festlegen, es sei denn, Sie möchten den Wert für den Cloud-Zonenordner überschreiben.

Nächste Schritte

Erweitern Sie diesen grundlegenden VMware Cloud on AWS-Workflow, indem Sie Netzwerkisolierung hinzufügen. Weitere Informationen hierzu finden Sie unter [Konfigurieren eines isolierten Netzwerks im VMware Cloud on AWS-Workflow in vRealize Automation Cloud Assembly](#).

Konfigurieren eines isolierten Netzwerks im VMware Cloud on AWS-Workflow in vRealize Automation Cloud Assembly

In diesem Verfahren fügen Sie ein isoliertes Netzwerk für Ihre VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly hinzu.

Beim Festlegen Ihres VMware Cloud on AWS-Cloud-Kontos stehen die NSX-T-Einstellungen zur Verfügung, die in Ihrem VMware Cloud on AWS-Dienst konfiguriert sind. Informationen zum Konfigurieren von NSX-T-Einstellungen in Ihrem VMware Cloud on AWS-Dienst finden Sie in der VMware Cloud on AWS-[Produktdokumentation](#).

vRealize Automation Cloud Assembly unterstützt VMware Cloud on AWS mit NSX-T. VMware Cloud on AWS wird mit NSX-V nicht unterstützt.

vRealize Automation Cloud Assembly unterstützt Netzwerkisolierung für VMware Cloud on AWS-Bereitstellungen. Für VMware Cloud on AWS werden keine anderen Netzwerkmethoden unterstützt.

Diese Erweiterung des grundlegenden VMware Cloud on AWS-Workflows beschreibt die folgenden Methoden zum Erstellen eines isolierten Netzwerks für die Verwendung in Ihrem vRealize Automation Cloud Assembly-Blueprint:

- Konfigurieren Sie auf einem bedarfsgesteuerten Netzwerk basierende Isolierung.
- Konfigurieren Sie auf bedarfsgesteuerten Sicherheitsgruppen basierende Isolierung.

Voraussetzungen

Mit diesem Verfahren wird der grundlegende VMware Cloud on AWS-Workflow erweitert. Der Workflow verwendet dasselbe Cloud-Konto und dieselbe Region, dieselbe Cloud-Zone, dasselbe Projekt und Netzwerkprofil, die Sie im Workflow [VMware Cloud on AWS-Anwendungsbeispiel](#) konfiguriert haben.

Verfahren

1 [Definieren eines isolierten Netzwerks für eine VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly](#)

Sie können die Netzwerkisolierung für eine VMware Cloud on AWS-Bereitstellung mithilfe einer der folgenden Vorgehensweisen konfigurieren:

2 [Definieren einer Netzwerkkomponente in einem Blueprint zur Unterstützung der Netzwerkisolierung für VMware Cloud on AWS in vRealize Automation Cloud Assembly](#)

In diesem Schritt ziehen Sie die Komponente einer Netzwerkmaschine auf die Arbeitsfläche eines vRealize Automation Cloud Assembly-Blueprints und fügen Einstellungen für eine isolierte Netzwerkbereitstellung zu Ihrer VMware Cloud on AWS-Zielumgebung hinzu.

Definieren eines isolierten Netzwerks für eine VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly

Sie können die Netzwerkisolierung für eine VMware Cloud on AWS-Bereitstellung mithilfe einer der folgenden Vorgehensweisen konfigurieren:

- [Konfigurieren bedarfsgesteuerter netzwerkbasierter Isolierung in vRealize Automation Cloud Assembly](#)
- [Konfigurieren der auf bedarfsgesteuerten Sicherheitsgruppen basierenden Isolierung in vRealize Automation Cloud Assembly](#)

Konfigurieren bedarfsgesteuerter netzwerkbasierter Isolierung in vRealize Automation Cloud Assembly

Sie können Netzwerkisolierung für die Anforderungen Ihrer VMware Cloud on AWS-Bereitstellung konfigurieren, indem Sie bedarfsgesteuerte Netzwerkeinstellungen in einem vRealize Automation Cloud Assembly-Netzwerkprofil angeben und verwenden.

Sie können ein isoliertes Netzwerk mithilfe einer Sicherheitsgruppe oder mithilfe bedarfsgesteuerter Netzwerkeinstellungen angeben. In diesem Beispiel konfigurieren Sie Netzwerkisolierung durch Angabe bedarfsgesteuerter Netzwerkeinstellungen im Netzwerkprofil. Zu einem späteren Zeitpunkt greifen Sie auf das Netzwerk in einem Blueprint zu und verwenden den Blueprint in einer VMware Cloud on AWS-Bereitstellung.

Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.

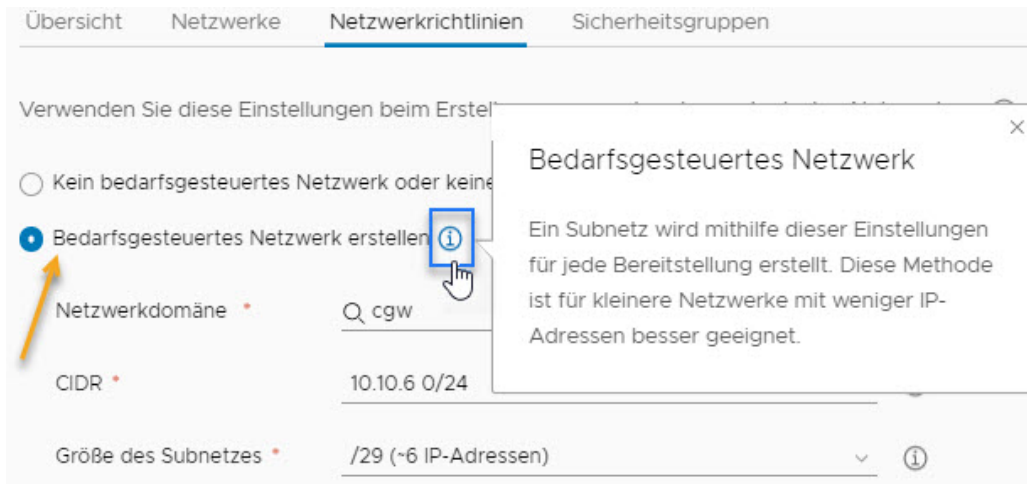
Voraussetzungen

- Schließen Sie den unter [Konfigurieren eines grundlegenden VMware Cloud on AWS-Workflows in vRealize Automation Cloud Assembly](#) beschriebenen Workflow ab.
- Weitere Informationen finden Sie unter [Konfigurieren eines isolierten Netzwerks im VMware Cloud on AWS-Workflow in vRealize Automation Cloud Assembly](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die erforderlichen Administratoranmeldedaten verfügen, einschließlich der VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).

Verfahren

- 1 Öffnen Sie das Netzwerkprofil, das Sie im grundlegenden VMware Cloud on AWS-Workflow verwendet haben, z. B. `vmc-network1`. Weitere Informationen hierzu finden Sie unter [Konfigurieren von Netzwerk- und Speicherprofilen für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly](#).

- 2 Auf der Registerkarte **Netzwerke** muss keine Auswahl vorgenommen werden.
- 3 Klicken Sie auf die Registerkarte **Netzwerkrichtlinien**.
- 4 Wählen Sie die Option **Bedarfsgesteuertes Netzwerk erstellen** und die `cgw`-Standardnetzwerkdomäne aus. Geben Sie eine geeignete Größe für CIDR und Subnetz an.



- 5 Klicken Sie auf **Speichern**.

Bei Verwendung dieses Netzwerkprofils werden Maschinen in einem Netzwerk in der Standardnetzwerkdomäne bereitgestellt. Das Netzwerk wird von anderen Netzwerken isoliert, indem privater oder ausgehender Netzwerkzugriff verwendet wird.

Nächste Schritte

Konfigurieren Sie eine Netzwerkkomponente in Ihrem Blueprint. Weitere Informationen finden Sie unter [Definieren einer Netzwerkkomponente in einem Blueprint zur Unterstützung der Netzwerkisolierung für VMware Cloud on AWS in vRealize Automation Cloud Assembly](#)

Konfigurieren der auf bedarfsgesteuerten Sicherheitsgruppen basierenden Isolierung in vRealize Automation Cloud Assembly

Sie können die Netzwerkisolierung für Ihre VMware Cloud on AWS-Bereitstellungsanforderungen konfigurieren, indem Sie eine bedarfsgesteuerte Sicherheitsgruppe in einem vRealize Automation Cloud Assembly-Netzwerkprofil angeben und verwenden.

Sie können ein isoliertes Netzwerk mithilfe einer Sicherheitsgruppe oder mithilfe bedarfsgesteuerter Netzwerkeinstellungen angeben. In diesem Beispiel konfigurieren Sie Netzwerkisolierung durch Angabe einer bedarfsgesteuerten Sicherheitsgruppe im Netzwerkprofil. Zu einem späteren Zeitpunkt geben Sie das Netzwerk in einem Blueprint an und verwenden den Blueprint in einer VMware Cloud on AWS-Bereitstellung.

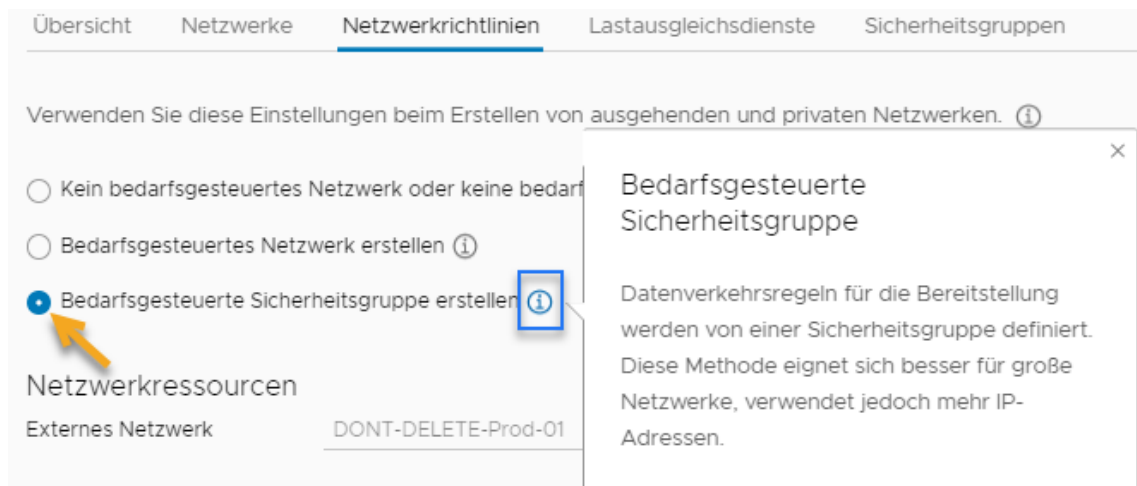
Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.

Voraussetzungen

- Schließen Sie den unter [Konfigurieren eines grundlegenden VMware Cloud on AWS-Workflows in vRealize Automation Cloud Assembly](#) beschriebenen Workflow ab.
- Weitere Informationen finden Sie unter [Konfigurieren eines isolierten Netzwerks im VMware Cloud on AWS-Workflow in vRealize Automation Cloud Assembly](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die erforderlichen Administratoranmeldedaten verfügen, einschließlich der VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Bei diesem Verfahren wird vorausgesetzt, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).

Verfahren

- 1 Öffnen Sie das Netzwerkprofil, das Sie im grundlegenden VMware Cloud on AWS-Workflow verwendet haben, z. B. `vmc-network1`. Weitere Informationen finden Sie unter [Konfigurieren von Netzwerk- und Speicherprofilen für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly](#).
- 2 Wählen Sie das vorhandene Netzwerk aus, das Sie im grundlegenden VMware Cloud on AWS-Workflow verwendet haben, z. B. `sddc-cgw-network-1`. Weitere Informationen hierzu finden Sie unter [Konfigurieren von Netzwerk- und Speicherprofilen für VMware Cloud on AWS-Bereitstellungen in vRealize Automation Cloud Assembly](#).
- 3 Klicken Sie auf die Registerkarte **Netzwerkrichtlinien**.
- 4 Wählen Sie die Option **Bedarfsgesteuerte Sicherheitsgruppe erstellen** aus.



5 Klicken Sie auf **Speichern**.

Bei Verwendung dieses Netzwerkprofils werden Maschinen im ausgewählten Netzwerk bereitgestellt und durch eine neue Sicherheitsgruppenrichtlinie isoliert. Die neue Sicherheitsrichtlinie lässt privaten oder ausgehenden Netzwerkzugriff zu.

Nächste Schritte

Konfigurieren Sie eine Netzwerkkomponente in Ihrem Blueprint. Weitere Informationen finden Sie unter [Definieren einer Netzwerkkomponente in einem Blueprint zur Unterstützung der Netzwerkisolierung für VMware Cloud on AWS in vRealize Automation Cloud Assembly](#)

Definieren einer Netzwerkkomponente in einem Blueprint zur Unterstützung der Netzwerkisolierung für VMware Cloud on AWS in vRealize Automation Cloud Assembly

In diesem Schritt ziehen Sie die Komponente einer Netzwerkmaschine auf die Arbeitsfläche eines vRealize Automation Cloud Assembly-Blueprints und fügen Einstellungen für eine isolierte Netzwerkbereitstellung zu Ihrer VMware Cloud on AWS-Zielumgebung hinzu.

Fügen Sie dem zuvor erstellten Blueprint die Netzwerkisolierung hinzu. Der Blueprint ist bereits mit einem Projekt und einer Cloud-Zone verknüpft, die die Bereitstellung in Ihrer VMware Cloud on AWS-Umgebung unterstützen, sowie mit dem Netzwerkprofil und dem Netzwerk, die Sie für die Isolierung konfiguriert haben.

Sofern nicht anders angegeben, gelten die in diesem Verfahren eingegebenen Schrittwerte nur für diesen Beispiel-Workflow.

Voraussetzungen

- Schließen Sie das Verfahren [Konfigurieren der auf bedarfsgesteuerten Sicherheitsgruppen basierenden Isolierung in vRealize Automation Cloud Assembly](#) oder [Konfigurieren bedarfsgesteuerter netzwerkbasierter Isolierung in vRealize Automation Cloud Assembly](#) ab.
- Dieses Verfahren setzt voraus, dass Sie über Anmeldedaten als Blueprint-Designer verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Bei diesem Verfahren wird davon ausgegangen, dass Sie über VMware Cloud on AWS-CloudAdmin-Anmeldedaten für das Ziel-SDDC in vCenter verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).

Verfahren

- 1 Öffnen Sie den Blueprint, den Sie im vorherigen Workflow erstellt haben. Weitere Informationen hierzu finden Sie unter [Definieren einer vCenter-Maschinenressource in einem Blueprint-Design zur Unterstützung der VMware Cloud on AWS-Bereitstellung in vRealize Automation Cloud Assembly](#).

- 2 Ziehen Sie aus den Komponenten links auf der Entwurfsseite des Blueprints eine Netzwerkkomponente auf die Arbeitsfläche.
- 3 Bearbeiten Sie den YAML-Code der Netzwerkkomponente, um den Netzwerktyp `private` oder `outbound` anzugeben (Fettformatierung).

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

ODER

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

Nächste Schritte

Sie können den Blueprint nun bereitstellen oder schließen.

Anwendungsfall für eine anbieterspezifische externe IPAM-Integration

Sie können einen externen IPAM-Anbieter zum Verwalten von IP-Adresszuweisungen für Ihre Blueprint-Bereitstellungen verwenden. In diesem Beispielverfahren wird beschrieben, wie Sie die externe IPAM-Integration in vRealize Automation Cloud Assembly konfigurieren und Blueprints bereitstellen, die ihre IP-Adresszuteilung vom externen IPAM-Anbieter beziehen. In diesem Beispiel wird beschrieben, wie Sie eine IPAM-Integration mithilfe von Infoblox als externen IPAM-Anbieter konfigurieren.

In diesem Verfahren verwenden Sie ein vorhandenes IPAM-Anbieterpaket, in diesem Fall ein Infoblox-Paket, und eine vorhandene Ausführungsumgebung, um einen anbieterspezifischen IPAM-Integrationspunkt zu erstellen. Sie konfigurieren ein vorhandenes Netzwerk und erstellen ein Netzwerkprofil, um die IP-Adresszuteilung aus dem externen IPAM-Anbieter zu unterstützen. Schließlich erstellen Sie einen Blueprint, der mit dem Netzwerk und Netzwerkprofil abgeglichen wird, und stellen vernetzte Maschinen mithilfe von IP-Werten bereit, die vom externen IPAM-Anbieter bezogen werden.

Informationen zum Erhalten und Konfigurieren des IPAM-Anbieterpakets sowie zum Konfigurieren einer Ausführungsumgebung, die zur Unterstützung der IPAM-Anbieterintegration auf einen Cloud-Erweiterbarkeits-Proxy zugreift, sind zu Referenzzwecken enthalten.

Beachten Sie, dass es sich bei den angezeigten Werten lediglich um Beispielwerte handelt. Sie können diese nicht eins zu eins auf Ihre Umgebung übertragen.

Überlegen Sie, wo Sie Werte austauschen würden, oder führen Sie eine Extrapolation anhand der Beispielergebnisse durch, um die Anforderungen Ihres Unternehmens zu erfüllen.

Verfahren

1 Hinzufügen erforderlicher erweiterbarer Attribute in der Infoblox-Anwendung für die Integration mit vRealize Automation

Bevor Sie das Infoblox-Anbieterpaket (`infoblox.zip`) für die Integration mit vRealize Automation von der Infoblox-Website oder vom VMware Marketplace herunterladen und bereitstellen können, müssen Sie in Infoblox erforderliche Erweiterbarkeitsattribute hinzufügen.

2 Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets zur Verwendung in vRealize Automation Cloud Assembly

Bevor Sie einen externen IPAM-Integrationspunkt in vRealize Automation Cloud Assembly definieren können, benötigen Sie ein konfiguriertes IPAM-Anbieterpaket. Sie können ein Anbieterpaket von der Website Ihres IPAM-Anbieters oder vom vRealize Automation Cloud Assembly-Marketplace herunterladen.

3 Erstellen einer Ausführungsumgebung für einen IPAM-Integrationspunkt in vRealize Automation

Bevor Sie einen externen IPAM-Integrationspunkt in vRealize Automation definieren können, müssen Sie eine Ausführungsumgebung erstellen bzw. auf eine vorhandene Ausführungsumgebung zugreifen, die als Vermittler zwischen dem IPAM-Anbieter und vRealize Automation fungiert. Die Ausführungsumgebung ist üblicherweise ein Amazon Web Services- oder Microsoft Azure-Cloud-Konto, das einem lokalen Integrationspunkt mit aktionsbasierter Erweiterbarkeit zugeordnet ist, der wiederum einem Cloud-Erweiterbarkeits-Proxy zugeordnet ist.

4 Hinzufügen eines externen IPAM-Integrationspunkts in vRealize Automation

vRealize Automation unterstützt die Integration mit externen IPAM-Anbietern. Sie können einen anbieterspezifischen IPAM-Integrationspunkt verwenden, um IP-Adressen und zugehörige Netzwerkeigenschaften für Blueprint-Bereitstellungen abzurufen und zu verwalten.

5 Konfigurieren eines Netzwerks und eines Netzwerkprofils in vRealize Automation Cloud Assembly zur Verwendung von IPAM-Anbieterwerten

Sie können ein Netzwerk so definieren, dass es IP-Adresswerte verwendet, die von einem externen IPAM-Anbieter anstatt intern von vRealize Automation Cloud Assembly abgerufen und verwaltet werden.

6 Definieren und Bereitstellen eines vRealize Automation Cloud Assembly-Blueprints, der die Bereichszuweisung des IPAM-Anbieters verwendet

Sie können einen Blueprint definieren, um die Zuweisungen von IP-Adressen von Ihrem externen IPAM-Anbieter abzurufen und zu verwalten.

7 Verwenden Infoblox-spezifischer Eigenschaften und erweiterbarer Attribute für IPAM-Integrationen in vRealize Automation

Sie können für vRealize Automation-Projekte, die externe IPAM-Integrationen für Infoblox enthalten, Infoblox-spezifische Eigenschaften verwenden.

Hinzufügen erforderlicher erweiterbarer Attribute in der Infoblox-Anwendung für die Integration mit vRealize Automation

Bevor Sie das Infoblox-Anbieterpaket (`infoblox.zip`) für die Integration mit vRealize Automation von der Infoblox-Website oder vom VMware Marketplace herunterladen und bereitstellen können, müssen Sie in Infoblox erforderliche Erweiterbarkeitsattribute hinzufügen.

Dieses Verfahren ist anwendbar, wenn Sie für die Infoblox-Integration in vRealize Automation Cloud Assembly einen externen IPAM-Integrationspunkt erstellen.

Bevor Sie `Infoblox.zip` herunterladen können, müssen Sie sich mit den Administratoranmeldedaten des Kontos Ihrer Organisation bei Ihrem Infoblox-Konto anmelden und die folgenden erweiterbaren Infoblox-Attribute vorab erstellen:

- `VMware NIC index`
- `VMware resource ID`
- `Tenant ID`
- `CMP Type`
- `VM ID`
- `VM Name`

Voraussetzungen

- Stellen Sie sicher, dass Sie über ein Konto bei [Infoblox](#) verfügen und dass Sie über die korrekten Zugriffsberechtigungen für das Infoblox-Konto Ihres Unternehmens verfügen.
- Bestätigen Sie, dass die WAPI-Version von Infoblox unterstützt wird. Die IPAM-Integration mit Infoblox beruht auf Infoblox WAPI v2.7. Alle Infoblox-Appliances mit Unterstützung für WAPI v2.7 werden unterstützt.
- Weitere Informationen finden Sie unter [Verwenden Infoblox-spezifischer Eigenschaften und erweiterbarer Attribute für IPAM-Integrationen in vRealize Automation](#).

Verfahren

- 1 Melden Sie sich mit Administratoranmeldedaten bei Ihrem Infoblox-Konto an.

Bei diesen Anmeldedaten handelt es sich um denselben Administratorbenutzernamen und das zugehörige Kennwort, die Sie beim Erstellen eines externen IPAM-Integrationspunkts in vRealize Automation Cloud Assembly über die Menüfolge **Infrastruktur > Verbindungen > Integrationen** > verwenden.

2 Verwenden Sie das in der Infoblox-Dokumentation beschriebene Verfahren, um die folgenden erforderlichen erweiterbaren Attribute in Ihrer Infoblox-Anwendung zu erstellen.

- VMware NIC index –Typ Ganzzahl
- VMware resource ID –Typ-Zeichenfolge
- Tenant ID –Typ-Zeichenfolge
- CMP Type –Typ-Zeichenfolge
- VM ID –Typ-Zeichenfolge
- VM Name –Typ-Zeichenfolge

Der Vorgang wird im Abschnitt *Hinzufügen von erweiterbaren Attributen* des Infoblox-Dokumentationsthemas [Informationen zu erweiterbaren Attributen](#) beschrieben. Weitere Informationen finden Sie unter [Verwalten von erweiterbaren Attributen](#).

Nächste Schritte

Nachdem Sie die erforderlichen Attribute hinzugefügt haben, können Sie den Vorgang zum Herunterladen und Bereitstellen des Infoblox-Pakets wie unter [Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets zur Verwendung in vRealize Automation Cloud Assembly](#) beschrieben fortsetzen.

Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets zur Verwendung in vRealize Automation Cloud Assembly

Bevor Sie einen externen IPAM-Integrationspunkt in vRealize Automation Cloud Assembly definieren können, benötigen Sie ein konfiguriertes IPAM-Anbieterpaket. Sie können ein Anbieterpaket von der Website Ihres IPAM-Anbieters oder vom vRealize Automation Cloud Assembly-Marketplace herunterladen.

Sie können ein anbieterspezifisches Integrationspaket von der Website des IPAM-Anbieters, über den [VMware Solution Exchange Marketplace](#) oder, sofern verfügbar, über die vRealize Automation Cloud Assembly-Registerkarte **Marketplace** abrufen.

Hinweis In diesem Beispiel wird das von VMware bereitgestellte Infoblox-Paket `Infoblox.zip` verwendet, das folgendermaßen über den VMware Solution Exchange Marketplace heruntergeladen werden kann.

- [vRA Cloud Infoblox Plug-In Version 1.1](#) – unterstützt vRealize Automation 8.1
- [vRA Cloud Infoblox Plug-In Version 1.0](#) – unterstützt vRealize Automation 8.0.1
- [vRA Cloud Infoblox Plug-In Version 0.1](#) – unterstützt vRealize Automation 8.0

Die IPAM-Integration mit Infoblox beruht auf Infoblox WAPI v2.7. Alle Infoblox-Appliances mit Unterstützung für WAPI v2.7 werden unterstützt.

Informationen zum Erstellen eines IPAM-Integrationspakets für andere IPAM-Anbieter finden Sie unter [Vorgehensweise zum Verwenden des IPAM-SDK zur Erstellung eines anbieterspezifischen, externen IPAM-Integrationspakets für vRealize Automation](#), wenn noch keins im Download-Center vorhanden ist.

Das IPAM-Anbieterpaket enthält Skripts in einem Paket mit Metadaten und anderen Konfigurationen. Die Skripts enthalten den Quellcode, der für die Vorgänge verwendet wird, die von vRealize Automation Cloud Assembly in Abstimmung mit dem externen IPAM-Anbieter ausgeführt werden. Zu den Beispielvorgängen gehören `Allocate an IP address for a virtual machine`, `Fetch a list of IP ranges from the provider` und `Update the MAC address of a host record in the provider`.

Voraussetzungen

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Vergewissern Sie sich, dass Sie über ein Konto beim externen IPAM-Anbieter, z. B. [Infoblox](#) oder [BlueCat](#), und über die korrekten Zugriffsberechtigungen für das Konto Ihrer Organisation beim IPAM-Anbieter verfügen.
- Wenn Sie Infoblox als Ihren externen IPAM-Anbieter verwenden, vergewissern Sie sich, dass Sie Ihrem Infoblox-Konto die erforderlichen erweiterbaren Attribute hinzugefügt haben, bevor Sie fortfahren. Weitere Informationen hierzu finden Sie unter [Hinzufügen erforderlicher erweiterbarer Attribute in der Infoblox-Anwendung für die Integration mit vRealize Automation](#).

Verfahren

- 1 Navigieren Sie zur Seite des Pakets [vRA Cloud Infoblox Plug-In Version 0.1](#) (vRealize Automation 8.0) oder [vRA Cloud Infoblox Plug-In Version 1.0](#) (vRealize Automation 8.0.1) VMware Solutions Exchange Marketplace.
- 2 Melden Sie sich an und laden Sie das Plug-In-Paket herunter.
- 3 Wenn Sie dies noch nicht getan haben, fügen Sie die erforderlichen erweiterbaren Eigenschaften in Infoblox hinzu. Weitere Informationen hierzu finden Sie unter [Hinzufügen erforderlicher erweiterbarer Attribute in der Infoblox-Anwendung für die Integration mit vRealize Automation](#).

Ergebnisse

Das Paket steht nun für die Bereitstellung über den Menüpfad **Integrationen > Integration hinzufügen > IPAM > Anbieter verwalten > Paket importieren** zur Verfügung. Siehe hierzu die Beschreibung in [Hinzufügen eines externen IPAM-Integrationspunkts in vRealize Automation](#).

Erstellen einer Ausführungsumgebung für einen IPAM-Integrationspunkt in vRealize Automation

Bevor Sie einen externen IPAM-Integrationspunkt in vRealize Automation definieren können, müssen Sie eine Ausführungsumgebung erstellen bzw. auf eine vorhandene Ausführungsumgebung zugreifen, die als Vermittler zwischen dem IPAM-Anbieter und vRealize Automation fungiert. Die Ausführungsumgebung ist üblicherweise ein Amazon Web Services- oder Microsoft Azure-Cloud-Konto, das einem lokalen Integrationspunkt mit aktionsbasierter Erweiterbarkeit zugeordnet ist, der wiederum einem Cloud-Erweiterbarkeits-Proxy zugeordnet ist.

Für die externe IPAM-Integration ist eine Ausführungsumgebung erforderlich. Wenn Sie den IPAM-Integrationspunkt definieren, erstellen Sie eine Verbindung zwischen vRealize Automation Cloud Assembly und Ihrem IPAM-Anbieter, indem Sie eine verfügbare Ausführungsumgebung angeben.

Die IPAM-Integration verwendet eine Reihe von heruntergeladenen anbieterspezifischen Skripts oder Plug-Ins in einer Ausführungsumgebung, die von einem FaaS-Anbieter (Feature-as-a-Service) wie z. B. Amazon Web Services Lambda, Microsoft Azure-Funktionen oder einem lokalen eingebetteten Integrationspunkt mit aktionsbasierter Erweiterbarkeit (ABX) bereitgestellt wird. Die Ausführungsumgebung wird zum Herstellen einer Verbindung mit dem externen IPAM-Anbieter verwendet, z. B. Infoblox.

Hinweis Ein IPAM-Integrationspunkt von Infoblox erfordert einen lokalen eingebetteten Integrationspunkt mit aktionsbasierter Erweiterbarkeit.

Jeder Typ von Laufzeitumgebung hat Vor- und Nachteile:

- Integrationspunkt mit aktionsbasierter Erweiterbarkeit (ABX)
 - kostenlos, keine zusätzlichen Kosten für Anbieternutzung
 - kann eine Verbindung zu IPAM-Anbieter-Appliances herstellen, die sich in einem lokalen Datacenter hinter einer NAT/Firewall befinden, das nicht öffentlich zugänglich ist, z. B. Infoblox
 - langsamere und etwas weniger zuverlässige Leistung als kommerzielle Cloud-Anbieter
- Amazon Web Services
 - hat entsprechende Kosten für die Anbieter-FaaS-Verbindung/Nutzung
 - es kann keine Verbindung zu IPAM-Anbieter-Appliances hergestellt werden, die sich in einem lokalen Datacenter hinter einer NAT/Firewall befinden, das nicht öffentlich zugänglich ist
 - schnelle und äußerst zuverlässige Leistung
- Microsoft Azure
 - hat entsprechende Kosten für die Anbieter-FaaS-Verbindung/Nutzung

- es kann keine Verbindung zu IPAM-Anbieter-Appliances hergestellt werden, die sich in einem lokalen Datacenter hinter einer NAT/Firewall befinden, das nicht öffentlich zugänglich ist
- schnelle und äußerst zuverlässige Leistung

Voraussetzungen

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Vergewissern Sie sich, dass Sie über ein Konto beim externen IPAM-Anbieter, z. B. [Infoblox](#) oder [BlueCat](#), und über die korrekten Zugriffsberechtigungen für das Konto Ihrer Organisation beim IPAM-Anbieter verfügen.
- Vergewissern Sie sich, dass Sie auf ein bereitgestelltes Integrationspaket für Ihren IPAM-Anbieter zugreifen können, wie z. B. Infoblox oder BlueCat. Das bereitgestellte Paket wird zunächst als ZIP-Download von Ihrer IPAM-Anbieter-Website oder vom vRealize Automation Cloud Assembly-Marketplace abgerufen und anschließend in vRealize Automation Cloud Assembly bereitgestellt.

Informationen zum Bereitstellen der ZIP-Datei des Anbieterpakets und dessen Verfügbarmachung als Wert für **Anbieter** auf der Seite „IPAM-Integration“ finden Sie unter [Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets zur Verwendung in vRealize Automation Cloud Assembly](#).

Verfahren

- 1 Um eine lokale FaaS-basierte Erweiterbarkeitsaktion zu erstellen, die als Ausführungsumgebung der IPAM-Integration verwendet werden soll, wählen Sie **Erweiterbarkeit > Bibliothek > Aktionen** aus.
- 2 Klicken Sie auf **Neue Aktion**, geben Sie einen Aktionsnamen und eine Beschreibung ein und geben Sie ein Projekt an.
- 3 Wählen Sie im Dropdown-Menü **FaaS-Anbieter** die Option **Lokal** aus.
- 4 Füllen Sie das Formular aus, um die Erweiterbarkeitsaktion zu definieren.



Verwandte Informationen zur ausgeführten Umgebung finden Sie im Blog-Video [Infoblox IPAM Plug-in 1.1-Integration](#) ab Minute 24.

Hinzufügen eines externen IPAM-Integrationspunkts in vRealize Automation

vRealize Automation unterstützt die Integration mit externen IPAM-Anbietern. Sie können einen anbieterspezifischen IPAM-Integrationspunkt verwenden, um IP-Adressen und zugehörige Netzwerkeigenschaften für Blueprint-Bereitstellungen abzurufen und zu verwalten.

In diesem Beispiel erstellen Sie einen externen IPAM-Integrationspunkt, um den Zugriff auf das Konto Ihrer Organisation über einen externen IPAM-Anbieter zu unterstützen. In diesem Beispiel-Workflow ist der IPAM-Anbieter Infoblox, und das anbieterspezifische Integrationspaket ist bereits vorhanden. Obwohl diese Anweisungen für eine Infoblox-Integration spezifisch sind, können sie als Referenz verwendet werden, wenn Sie eine IPAM-Integration für einen anderen externen IPAM-Anbieter erstellen.

Sie können ein anbieterspezifisches Integrationspaket von der Website des IPAM-Anbieters, über den [VMware Solution Exchange Marketplace](#) oder, sofern verfügbar, über die vRealize Automation Cloud Assembly-Registrierkarte **Marketplace** abrufen.

In diesem Beispiel wird das von VMware bereitgestellte Infoblox-Paket `Infoblox.zip` verwendet, das folgendermaßen über den VMware Solution Exchange Marketplace heruntergeladen werden kann.

- [vRA Cloud Infoblox Plug-In Version 1.1](#) – unterstützt vRealize Automation 8.1
- [vRA Cloud Infoblox Plug-In Version 1.0](#) – unterstützt vRealize Automation 8.0.1
- [vRA Cloud Infoblox Plug-In Version 0.1](#) – unterstützt vRealize Automation 8.0

Voraussetzungen

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie über ein Konto beim externen IPAM-Anbieter sowie über die korrekten Zugriffsberechtigungen für das Konto Ihrer Organisation beim IPAM-Anbieter verfügen.
- Vergewissern Sie sich, dass Sie auf ein bereitgestelltes Integrationspaket für Ihren IPAM-Anbieter zugreifen können. Das bereitgestellte Paket wird zunächst als ZIP-Download von der Website Ihres IPAM-Anbieters oder aus dem VMware Solutions Exchange Marketplace abgerufen und anschließend in vRealize Automation bereitgestellt.

Informationen dazu, wie Sie die ZIP-Datei des Anbieterpakets herunterladen und bereitstellen und das Paket als Wert für **Anbieter** auf der Seite „IPAM-Integration“ zur Verfügung stellen, finden Sie unter [Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets zur Verwendung in vRealize Automation Cloud Assembly](#).

- Vergewissern Sie sich, dass Sie auf eine konfigurierte Ausführungsumgebung für den IPAM-Anbieter zugreifen können. Bei der Ausführungsumgebung handelt es sich in der Regel um einen lokalen eingebetteten Integrationspunkt mit aktionsbasierter Erweiterbarkeit.
Informationen zu den Merkmalen der Ausführungsumgebung finden Sie unter [Erstellen einer Ausführungsumgebung für einen IPAM-Integrationspunkt in vRealize Automation](#).
- Aktivieren Sie die erforderlichen erweiterbaren Attribute in Ihrer Infoblox-Anwendung. Weitere Informationen hierzu finden Sie unter [Hinzufügen erforderlicher erweiterbarer Attribute in der Infoblox-Anwendung für die Integration mit vRealize Automation](#).
- Wenn Sie nicht über externen Internetzugriff verfügen, können Sie einen Internet-Proxyserver konfigurieren. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Integrieren eines Internet-Proxyservers für vRealize Automation](#).

Verfahren

- 1 Wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus und klicken Sie auf **Integration hinzufügen**.
- 2 Klicken Sie auf **IPAM**.
- 3 Wählen Sie im Dropdown-Menü **Anbieter** ein konfiguriertes IPAM-Anbieterpaket in der Liste aus, z. B. *Infoblox_hrg*.

Wenn die Liste leer ist, klicken Sie auf **Anbieterpaket importieren**, navigieren Sie zur ZIP-Datei eines vorhandenen Anbieterpakets und wählen Sie sie aus. Wenn Sie nicht über die Anbieter-ZIP-Datei verfügen, können Sie sie von der Website Ihres IPAM-Anbieters oder über die Registerkarte **Marketplace** in vRealize Automation Cloud Assembly abrufen.

Informationen zum Bereitstellen der ZIP-Datei des Anbieterpakets in vCenter und als **Anbieter** auf der Seite „Integration“ finden Sie unter [Herunterladen und Bereitstellen eines externen IPAM-Anbieterpakets zur Verwendung in vRealize Automation Cloud Assembly](#).

Weitere Informationen zum Upgrade einer vorhandenen IPAM-Integration zur Verwendung einer aktuelleren Version des IPAM-Integrationspakets eines Anbieters finden Sie unter [Vorgehensweise zum Upgrade auf ein neueres-IPAM-Integrationspaket in vRealize Automation](#).

- 4 Geben Sie Ihren Administratorbenutzernamen und das zugehörige Kennwort für Ihr Konto beim externen IPAM-Anbieter sowie die Informationen für alle anderen obligatorischen Felder (sofern vorhanden) ein, z. B. den Hostnamen Ihres Anbieters.

In diesem Beispiel erhalten Sie den Hostnamen Ihres Infoblox-IPAM-Anbieters mit den folgenden Schritten:

- a Melden Sie sich auf einer separaten Browser-Registerkarte mit Ihren Infoblox-Administratoranmeldedaten beim IPAM-Anbieterkonto an.
- b Kopieren Sie die URL des Hostnamens.
- c Fügen Sie die URL des Hostnamens im Feld **Hostname** auf der Seite „IPAM-Integration“ ein.

- 5 Wählen Sie in der Dropdown-Liste **Ausführungsumgebung** einen vorhandenen lokalen Integrationspunkt mit aktionsbasierter Erweiterbarkeit aus, z. B. *Infoblox_abx_intg*.

Die Ausführungsumgebung unterstützt die Kommunikation zwischen vRealize Automation und dem externen IPAM-Anbieter.

Hinweis Wenn Sie ein Amazon Web Services- oder ein Microsoft Azure-Cloud-Konto als Ausführungsumgebung der Integration verwenden, stellen Sie sicher, dass auf die IPAM-Anbieter-Appliance über das Internet zugegriffen werden kann, dass sie sich nicht hinter einer NAT oder Firewall befindet und dass sie einen öffentlich auflösbaren DNS-Namen aufweist. Wenn auf den IPAM-Anbieter nicht zugegriffen werden kann, können die Amazon Web Services-Lambda- oder Microsoft Azure-Funktionen keine Verbindung zu ihm herstellen und die Integration schlägt fehl. Informationen hierzu finden Sie unter [Erstellen einer Ausführungsumgebung für einen IPAM-Integrationspunkt in vRealize Automation](#).

Das IPAM-Framework unterstützt nur eine lokale eingebettete Ausführungsumgebung mit aktionsbasierter Erweiterbarkeit (ABX).

Hinweis Ein IPAM-Integrationspunkt von Infoblox erfordert einen lokalen eingebetteten Integrationspunkt mit aktionsbasierter Erweiterbarkeit.

Das konfigurierte Cloud-Konto bzw. der Integrationspunkt ermöglicht die Kommunikation zwischen vRealize Automation und dem IPAM-Anbieter, in diesem Beispiel Infoblox, über einen zugeordneten Cloud-Erweiterbarkeits-Proxy. Sie können einen bereits erstellten Anbieter auswählen oder einen Anbieter erstellen.

Informationen zum Erstellen einer Ausführungsumgebung finden Sie unter [Erstellen einer Ausführungsumgebung für einen IPAM-Integrationspunkt in vRealize Automation](#).

- 6 Klicken Sie auf **Validieren**.

Da in diesem Beispiel die lokale aktionsbasierte Erweiterbarkeitsintegration für die Ausführungsumgebung verwendet wird, können Sie die Validierungsaktion anzeigen.

- a Klicken Sie auf die Registerkarte **Erweiterbarkeit**.
- b Klicken Sie auf **Aktivität > Aktionsausführungen** und wählen Sie im Filter entweder **Alle Ausführungen** oder **Integrationsausführungen** aus. Sie sehen, dass eine Endpoint-Validierungsaktion initiiert ist und ausgeführt wird.

- 7 Wenn Sie dazu aufgefordert werden, dem selbstsignierten Zertifikat vom IPAM-Anbieter zu vertrauen, klicken Sie auf **Akzeptieren**.

Nachdem Sie das selbstsignierte Zertifikat akzeptiert haben, kann die Validierungsaktion bis zum Abschluss fortgesetzt werden.

- 8 Geben Sie einen **Namen** für diesen IPAM-Integrationspunkt (z. B. *Infoblox_Integration*) und eine **Beschreibung** (z. B. *Infoblox IPAM with ABX integration for team HRG*) ein.

- 9 Klicken Sie auf **Hinzufügen**, um den neuen externen IPAM-Integrationspunkt zu speichern.

Eine Datenerfassungsaktion wird initiiert. Die Daten zu Netzwerken und IP-Bereichen werden vom IPAM-Anbieter erfasst. Sie können die Datenerfassungsaktion folgendermaßen anzeigen:

- a Klicken Sie auf die Registerkarte **Erweiterbarkeit**.
- b Klicken Sie auf **Aktivität > Aktionsausführungen** und beachten Sie, dass eine Datenerfassungsaktion initiiert und ausgeführt wird. Sie können den Inhalt der Aktionsausführung öffnen und anzeigen.

Ergebnisse

Die anbieterspezifische externe IPAM-Integration steht nun zur Verwendung mit Netzwerken und Netzwerkprofilen zur Verfügung.

Konfigurieren eines Netzwerks und eines Netzwerkprofils in vRealize Automation Cloud Assembly zur Verwendung von IPAM-Anbieterwerten

Sie können ein Netzwerk so definieren, dass es IP-Adresswerte verwendet, die von einem externen IPAM-Anbieter anstatt intern von vRealize Automation Cloud Assembly abgerufen und verwaltet werden.

Sie können ein Netzwerk so definieren, dass es auf vorhandene IP-Einstellungen zugreift, die Sie im externen IPAM-Anbieterkonto Ihrer Organisation definiert haben. Dieser Schritt stellt eine Erweiterung der von Ihnen im vorherigen Schritt erstellten Infoblox-Anbieterintegration dar.

In diesem Beispiel konfigurieren Sie ein Netzwerkprofil mit vorhandenen Netzwerken, deren Daten von vCenter erfasst wurden. Anschließend konfigurieren Sie diese Netzwerke, um IP-Informationen von einem externen IPAM-Anbieter zu erhalten, in diesem Fall Infoblox. Virtuelle Maschinen, die Sie über vRealize Automation Cloud Assembly bereitstellen und die mit diesem Netzwerkprofil abgeglichen werden können, erhalten Ihre IP-Adresse und andere TCP/IP-bezogene Einstellungen vom externen IPAM-Anbieter.

Weitere Informationen zu Netzwerken finden Sie unter [Netzwerkressourcen](#). Weitere Informationen zu Netzwerkprofilen finden Sie unter [Hinzufügen von vRealize Automation Cloud Assembly-Netzwerkprofilen](#) und [Weitere Informationen zu Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

Voraussetzungen

Diese Abfolge von Schritten wird im Kontext eines Workflows für die IPAM-Anbieterintegration angezeigt. Weitere Informationen hierzu finden Sie unter [Anwendungsfall für eine anbieterspezifische externe IPAM-Integration](#).

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).

- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Stellen Sie sicher, dass Sie über ein Konto beim externen IPAM-Anbieter verfügen, z. B. [Infoblox](#) oder [BlueCat](#), und dass Sie über die korrekten Zugriffsberechtigungen für das Konto Ihrer Organisation beim IPAM-Anbieter verfügen. In diesem Beispiel-Workflow ist der IPAM-Anbieter Infoblox.
- Vergewissern Sie sich, dass Sie über einen IPAM-Integrationspunkt für den IPAM-Anbieter verfügen. Weitere Informationen hierzu finden Sie unter [Hinzufügen eines externen IPAM-Integrationspunkts in vRealize Automation](#).

Verfahren

- 1 Um ein Netzwerk zu konfigurieren, klicken Sie auf **Infrastruktur > Ressourcen > Netzwerke**.
- 2 Wählen Sie auf der Registerkarte **Netzwerke** ein vorhandenes Netzwerk aus, das mit dem IPAM-Anbieter-Integrationspunkt verwendet werden soll. In diesem Beispiel lautet der Netzwerkname *net.23.117-only-IPAM*.

Die Daten der aufgelisteten Netzwerke wurden von vRealize Automation Cloud Assembly von einem vCenter in Ihrer Organisation erfasst.

- 3 Um Werte vom externen IPAM-Anbieter abzurufen, stellen Sie sicher, dass außer **Konto/Region, Name** und **Netzwerkdomäne** alle anderen Netzwerkeinstellungen leer sind, einschließlich der folgenden:
 - Domäne (siehe Hinweis in Schritt 8)
 - CIDR
 - Standard-Gateway
 - DNS-Server
 - DNS-Suchdomänen
- 4 Klicken Sie auf die Registerkarte **IP-Bereiche** und dann auf **IPAM-IP-Bereich hinzufügen**.
- 5 Wählen Sie im Menü **Netzwerk** das Netzwerk aus, das Sie gerade konfiguriert haben, z. B. *net.23.117-only-IPAM*.
- 6 Wählen Sie im Menü **Anbieter** den IPAM-Integrationspunkt *Infoblox_Integration* aus, den Sie zuvor im Workflow erstellt haben.
- 7 Wählen Sie im jetzt sichtbaren Dropdown-Menü **Adressraum** eine der aufgelisteten Netzwerkansichten aus.

Ein Adressraum in Infoblox wird als Netzwerkansicht bezeichnet.

Die Netzwerkansichten werden von Ihrem IPAM-Anbieterkonto abgerufen. In diesem Beispiel werden das Netzwerk-Subnetz, das Sie gerade konfiguriert haben, z. B. *net.23.117-only-IPAM*, der *Infoblox_Integration*-Integrationspunkt, den Sie zuvor im Workflow erstellt haben, und ein Adressraum mit dem Namen *default* verwendet.

Die aufgelisteten Adressraumwerte werden vom externen IPAM-Anbieter abgerufen.

- 8 Wählen Sie aus der Liste der angezeigten Netzwerke, die für den ausgewählten Adressraum verfügbar sind, ein oder mehrere Netzwerke aus. Wählen Sie beispielsweise 10.23.117.0/24 aus.

In diesem Beispiel enthalten die Spaltenwerte von **Domänen** und **DNS-Server** für das ausgewählte Netzwerk Werte aus Infoblox.

Hinweis Wenn Sie in Schritt 3 ein Netzwerk auswählen, für das eine Domäne für vRealize Automation Cloud Assembly angegeben wurde, und dann ein Netzwerk aus dem Adressraum des externen IPAM-Anbieters auswählen, der einen Domänenwert enthält, hat der Domänenwert im externen IPAM-Anbietwork Vorrang vor der in vRealize Automation Cloud Assembly angegebenen Domäne. Wenn die Einstellung für den IPAM-IP-Bereich keinen Domänenwert aufweist, der wie oben beschrieben entweder in Cloud Assembly oder im externen IPAM-Anbieter angegeben ist, schlägt die Bereitstellung fehl.

- 9 Klicken Sie auf **Hinzufügen**, um den IPAM-IP-Bereich für das Netzwerk zu speichern.

Der Bereich wird in der Tabelle **IP-Bereiche** angezeigt.

- 10 Klicken Sie auf die Registerkarte **IP-Adressen**.

Nachdem Sie eine Maschine mithilfe des neuen Adressbereichs des externen IPAM-Anbieters bereitgestellt haben, wird ein neuer Datensatz in der Tabelle **IP-Adressen** angezeigt.

- 11 Um ein Netzwerkprofil für die Verwendung des Netzwerks zu konfigurieren, klicken Sie auf **Infrastruktur > Konfigurieren > Netzwerkprofile**.

- 12 Benennen Sie das Netzwerkprofil, z. B. als *Infoblox-NP*, und fügen Sie die folgenden Beispieleinstellungen hinzu.

- Registerkarte „Übersicht“

- Geben Sie ein vSphere-Cloud-Konto/eine -Cloud-Region an.
- Fügen Sie ein Funktions-Tag für das Netzwerkprofil hinzu, z. B. mit dem Namen *infoblox_abx*.

Notieren Sie sich das Funktions-Tag, da Sie es auch als Blueprint-Einschränkungs-Tag verwenden müssen, um die Bereitstellungszuordnung im Blueprint herzustellen.

- Registerkarte „Netzwerke“

- Fügen Sie das zuvor erstellte Netzwerk hinzu, beispielsweise *net.23.117-only-IPAM*.

- 13 Klicken Sie auf **Speichern**, um das Netzwerkprofil mit diesen Einstellungen zu speichern.

Ergebnisse

Die Einstellung für das Netzwerk und das Netzwerkprofil ist nun für die Unterstützung der externen IPAM-Integration konfiguriert und kann in einem Blueprint verwendet werden.

Definieren und Bereitstellen eines vRealize Automation Cloud Assembly-Blueprints, der die Bereichszuweisung des IPAM-Anbieters verwendet

Sie können einen Blueprint definieren, um die Zuweisungen von IP-Adressen von Ihrem externen IPAM-Anbieter abzurufen und zu verwalten.

In diesem letzten Schritt im Workflow zur externen IPAM-Integration nehmen Sie die Definition und die Bereitstellung eines Blueprints vor, der Ihr zuvor definiertes Netzwerk und Netzwerkprofil mit dem Infoblox-Konto Ihrer Organisation verbindet, um IP-Adresszuweisungen für bereitgestellte VMs vom externen IPAM-Anbieter anstatt von vRealize Automation Cloud Assembly abzurufen und zu verwalten.

Der Workflow verwendet Infoblox als externen IPAM-Anbieter, und in einigen Schritten gelten die Beispielwerte nur für Infoblox. Aber es besteht die Absicht, dass der Vorgang auf andere externe IPAM-Integrationen angewendet werden kann.

Nachdem Sie den Blueprint bereitgestellt haben und die virtuelle Maschine gestartet wurde, wird die für jede VM in der Bereitstellung verwendete IP-Adresse als Netzwerkeintrag auf der Seite **Ressourcen > Netzwerke** als neuer Hosteintrag im IPAM-Anbietwork im Konto Ihres IPAM-Anbieters und im vSphere Web Client-Datensatz für jede bereitgestellte VM im Host-vCenter angezeigt.

Voraussetzungen

Diese Abfolge von Schritten wird im Kontext eines Workflows für die Integration eines externen IPAM-Anbieters gezeigt. Weitere Informationen hierzu finden Sie unter [Anwendungsfall für eine anbieterspezifische externe IPAM-Integration](#).

- Vergewissern Sie sich, dass Sie über Cloud-Administratoranmeldedaten verfügen. Weitere Informationen hierzu finden Sie unter [Zum Arbeiten mit Cloud-Konten in vRealize Automation sind Anmeldedaten erforderlich](#).
- Stellen Sie sicher, dass Sie über die Benutzerrolle des Cloud-Administrators verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Vergewissern Sie sich, dass Sie über ein Konto beim externen IPAM-Anbieter, z. B. Infoblox oder BlueCat, und über die korrekten Zugriffsberechtigungen für das Konto Ihrer Organisation beim IPAM-Anbieter verfügen.
- Vergewissern Sie sich, dass Sie über Administratorzugriff auf das Hostkonto und alle Rollenanforderungen verfügen, die zum Anzeigen von Statusdatensätzen im vSphere Web Client-Datensatz für Ihre bereitgestellten VMs auf dem Host vCenter erforderlich sind.

- Stellen Sie sicher, dass Sie über einen IPAM-Integrationspunkt für den externen IPAM-Anbieter verfügen. Weitere Informationen hierzu finden Sie unter [Hinzufügen eines externen IPAM-Integrationspunkts in vRealize Automation](#).
- Vergewissern Sie sich, dass Sie ein vRealize Automation Cloud Assembly-Netzwerk und -Netzwerkprofil konfiguriert haben, das die externe IPAM-Integration für Ihren geplanten IPAM-Integrationspunkt unterstützt. Weitere Informationen hierzu finden Sie unter [Konfigurieren eines Netzwerks und eines Netzwerkprofils in vRealize Automation Cloud Assembly zur Verwendung von IPAM-Anbieterwerten](#).
- Vergewissern Sie sich, dass das Projekt und die Cloud-Zone übereinstimmend mit den Tags im IPAM-Integrationspunkt und im Netzwerk oder Netzwerkprofil getaggt sind. Konfigurieren Sie optional das Projekt so, dass es die benutzerdefinierte Ressourcenbenennung unterstützt.

Über die gemachten Angaben hinausgehende Informationen zur Rolle eines Projekts und einer Cloud-Zone sowie der Rolle anderer Infrastrukturelemente in Ihrem Blueprint finden Sie unter [Anwendungsbeispiel: WordPress](#). Weitere Informationen zum Tagging finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#).

Informationen zur benutzerdefinierten Benennung von VMs mithilfe von Einstellungen in Ihrem Projekt finden Sie unter [Anpassen der Namen von bereitgestellten Ressourcen mithilfe von vRealize Automation Cloud Assembly](#).

Verfahren

- 1 Klicken Sie auf **Blueprints > Neu**, geben Sie die folgenden Informationen auf der Seite **Neuer Blueprint** ein und klicken Sie auf **Erstellen**.
 - **Name** = ipam-bpa
 - **Beschreibung** = Blueprint, der die IPAM-Integration Infoblox verwendet
 - **Projekt** = 123VC
- 2 Fügen Sie in diesem Beispiel der Blueprint-Arbeitsfläche eine Cloud-unabhängige Maschinenkomponente und eine Cloud-unabhängige Netzwerkkomponente hinzu und verbinden Sie die beiden Komponenten.
- 3 Bearbeiten Sie den Blueprint-Code, um der Netzwerkkomponente ein Einschränkungs-Tag hinzuzufügen, das mit dem Funktions-Tag, das Sie dem Netzwerkprofil hinzugefügt haben, übereinstimmt. In diesem Beispiel lautet der Tag-Wert *infoblox_abx*.
- 4 Bearbeiten Sie den Blueprint-Code, um anzugeben, dass der Netzwerkzuweisungstyp *static* lautet.

In diesem Beispiel wird als bekannt vorausgesetzt, dass die angegebene IP-Adresse 10.23.117.4 derzeit im externen IPAM-Adressraum, den wir für das Netzwerk im zugeordneten Netzwerkprofil ausgewählt haben, verfügbar ist. Die Zuweisungseinstellung *static* ist

erforderlich, der Wert *address* ist dies hingegen nicht. Wenn Sie die Auswahl einer externen IP-Adresse bei einer bestimmten Adresse starten möchten, ist dies zwar möglich, aber es ist nicht erforderlich. Wenn Sie keinen Wert für *address* angeben, wählt der externe IPAM-Anbieter die nächste verfügbare Adresse im externen IPAM-Netzwerk aus.

5 Überprüfen Sie den Blueprint-Code anhand des folgenden Beispiels.

```
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}'
```

Hinweis Standardmäßig erstellt die Infoblox-Integration einen DNS-Host-Datensatz in der *standardmäßigen* DNS-Ansicht in Infoblox.

Wenn Ihr Infoblox-Administrator *benutzerdefinierte* DNS-Ansichten erstellt hat, können Sie das Standardintegrationsverhalten überschreiben und mithilfe der Eigenschaft `Infoblox.IPAM.Network.dnsView` in der Maschinenkomponente eine benannte Ansicht angeben. Sie können beispielsweise der Komponente `Cloud_Machine_1` die folgende Eigenschaft hinzufügen, um eine benannte DNS-Ansicht in Infoblox anzugeben.

```
Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ubuntu
    flavor: small
    Infoblox.IPAM.Network.dnsView:<dns-view-name>
```

Informationen zum Konfigurieren und Verwenden von DNS-Ansichten finden Sie unter [DNS-Ansichten](#) in der Infoblox-Produktdokumentation.

6 Klicken Sie auf der Blueprint-Seite auf **Bereitstellen**, geben Sie der Bereitstellung den Namen *Infoblox-1* und klicken Sie auf der Seite **Bereitstellungstyp** auf **Bereitstellen**.

- 7 Klicken Sie bei der Bereitstellung des Blueprints auf die Registerkarte **Erweiterbarkeit** und wählen Sie **Aktivität > Aktionsausführungen** aus, um die derzeit ausgeführte *Infoblox_AllocateIP_n*-Erweiterbarkeitsaktion anzuzeigen.

Nachdem die Erweiterbarkeitsaktion abgeschlossen ist und die Maschine bereitgestellt wurde, wird die MAC-Adresse von der Aktion *Infoblox_Update_n* an Infoblox weitergegeben.

- 8 Sie können sich bei Ihrem Infoblox-Konto anmelden und das Konto öffnen, um den neuen Hosteintrag für die IPAM-Adresse im zugehörigen Netzwerk 10.23.117.0/24 anzuzeigen. Sie können auch die Registerkarte „DNS“ in Infoblox öffnen, um den neuen DNS-Hostdatensatz anzuzeigen.
- 9 Um zu überprüfen, ob die VM bereitgestellt wird, melden Sie sich bei Ihrem Host-vCenter und vSphere Web Client an, um die bereitgestellte Maschine zu finden und den DNS-Namen und die IP-Adresse anzuzeigen.

Nach dem Starten der bereitgestellten virtuellen Maschine wird die MAC-Adresse durch eine *Infoblox_AllocateIP*-Erweiterbarkeitsaktion an Infoblox weitergegeben.

- 10 Um den neuen Netzwerkdatensatz in vRealize Automation Cloud Assembly anzuzeigen, wählen Sie **Infrastruktur > Ressourcen > Netzwerke** aus und klicken Sie, um die Registerkarte **IP-Adressen** zu öffnen.
- 11 Wenn Sie die Bereitstellung löschen, wird die IPAM-Adresse der VMs in der Bereitstellung freigegeben, und die IP-Adressen stehen dem externen IPAM-Anbieter für andere Zuteilungen erneut zur Verfügung. Die Erweiterbarkeitsaktion für dieses Ereignis in vRealize Automation Cloud Assembly lautet *Infoblox_Deallocate*.

Verwenden Infoblox-spezifischer Eigenschaften und erweiterbarer Attribute für IPAM-Integrationen in vRealize Automation

Sie können für vRealize Automation-Projekte, die externe IPAM-Integrationen für Infoblox enthalten, Infoblox-spezifische Eigenschaften verwenden.

Die folgenden Infoblox-Eigenschaften sind für die Verwendung mit Ihren Infoblox-IPAM-Integrationen verfügbar. Sie können sie in vRealize Automation verwenden, um die Zuteilung der IP-Adressen während der Blueprint-Bereitstellung weiter zu steuern. Die Verwendung dieser Eigenschaften ist optional.

Während alle der folgenden Infoblox-Eigenschaften für die Verwendung mit dem Paket [vRA Cloud Infoblox Plug-In Version 0.1](#) für vRealize Automation 8.0 verfügbar sind, steht die Eigenschaft `Infoblox.IPAM.Network.dnsView` nur für die Verwendung mit dem Paket [vRA Cloud Infoblox Plug-In Version 1.0](#) für vRealize Automation 8.0.1 und dem Paket [vRA Cloud Infoblox Plug-In Version 1.1](#) für vRealize Automation 8.1 zur Verfügung.

Hinweis Die Verwendung dieser Eigenschaften ist nicht im [Anwendungsfall für eine anbieterspezifische externe IPAM-Integration](#)-Beispiel-Workflow enthalten.

- `Infoblox.IPAM.createFixedAddress`

Diese Eigenschaft ermöglicht es Ihnen, einen festen Adressdatensatz in Infoblox zu erstellen. Mögliche Werte sind „True“ und „False“. Standardmäßig wird ein Hostdatensatz erstellt. Standardwert: False.

- `Infoblox.IPAM.Network.dnsView`

Mit dieser Eigenschaft können Sie eine DNS-Ansicht beim Erstellen eines Hostdatensatzes in Infoblox verwenden. Standardwert: Standard.

- `Infoblox.IPAM.Network.enableDns`

Wenn Sie eine IP in Infoblox zuweisen, können Sie mit dieser Eigenschaft auch einen DNS-Datensatz erstellen. Mögliche Werte sind „True“ und „False“. Standardwert: True.

- `Infoblox.IPAM.Network.dnsSuffix`

Mit dieser Eigenschaft können Sie die DHCP-Option *domain* eines Infoblox-Netzwerks mit einer neuen überschreiben. Diese Funktion ist nützlich, wenn für das Infoblox-Netzwerk die DHCP-Option *domain* nicht festgelegt ist oder wenn die DHCP-Option *domain* überschrieben werden muss. Standardwert: None (leere Zeichenfolge).

`Infoblox.IPAM.Network.dnsSuffix` ist nur anwendbar, wenn `Infoblox.IPAM.Network.enableDns` auf „True“ festgelegt ist.

Sie können eine Infoblox-Eigenschaft mithilfe einer der folgenden Methoden in vRealize Automation Cloud Assembly angeben:

- Sie können Eigenschaften in einem Projekt mithilfe des Abschnitts **Benutzerdefinierte Eigenschaften** auf der Seite **Infrastruktur > Konfigurieren > Projekte** angeben. Mit dieser Methode werden die angegebenen Eigenschaften auf alle Maschinen angewendet, die im Rahmen dieses Projekts bereitgestellt werden.
- Sie können die Eigenschaften für jede Maschinenkomponente in einem Blueprint angeben. Der folgende Blueprint-Beispielcode veranschaulicht die Verwendung der `Infoblox.IPAM.Network.dnsView`-Eigenschaft:

```
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
```

- Sie können Eigenschaften mithilfe eines Erweiterbarkeitsabonnements angeben.

Verwandte Informationen zu den erweiterbaren Infoblox-Attributen in Bezug auf diesen Anwendungsfall finden Sie unter [Hinzufügen erforderlicher erweiterbarer Attribute in der Infoblox-Anwendung für die Integration mit vRealize Automation](#).

Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur

4

In der vRealize Automation Cloud Assembly-Ressourceninfrastruktur definieren Sie Cloud-Kontobereiche als Zonen, in denen Blueprints und deren Arbeitslasten bereitgestellt werden können.

Darüber hinaus beinhaltet die Ressourceninfrastruktur die Erstellung allgemeiner Zuordnungen von Images und Maschinengrößen sowie von Profilen, die Netzwerk- und Speicherfunktionen über Cloud-Kontoregionen oder Datacenter hinweg definieren.

Dieses Kapitel enthält die folgenden Themen:

- Vorgehensweise zum Hinzufügen von Cloud-Zonen, die Regionen oder Datacenter für die Platzierung des vRealize Automation Cloud Assembly-Ziels definieren
- Vorgehensweise zum Hinzufügen von vRealize Automation Cloud Assembly-Konfigurationszuordnungen für die Erstellung allgemeiner Maschinengrößen in vRealize Automation Cloud Assembly
- Vorgehensweise zum Hinzufügen von vRealize Automation Cloud Assembly-Image-Zuordnungen zum Erstellen allgemeiner Betriebssysteme
- Hinzufügen von vRealize Automation Cloud Assembly-Netzwerkprofilen
- Vorgehensweise zum Hinzufügen von vRealize Automation Cloud Assembly-Speicherprofilen, die verschiedenen Anforderungen entsprechen
- Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen
- Vorgehensweise zum Arbeiten mit Ressourcen in vRealize Automation Cloud Assembly

Vorgehensweise zum Hinzufügen von Cloud-Zonen, die Regionen oder Datacenter für die Platzierung des vRealize Automation Cloud Assembly-Ziels definieren

Eine vRealize Automation Cloud Assembly-Cloud-Zone ist eine Gruppe von Ressourcen innerhalb eines Cloud-Kontotyps, wie z. B. AWS oder vSphere.

Arbeitslasten werden von Blueprints in Cloud-Zonen in einer bestimmten Kontoregion bereitgestellt. Jede Cloud-Zone ist mit einem vRealize Automation Cloud Assembly-Projekt verknüpft.

Wählen Sie **Infrastruktur > Konfigurieren > Cloud-Zonen** aus und klicken Sie auf **Neue Zone hinzufügen**.

Weitere Informationen zu vRealize Automation Cloud Assembly-Cloud-Zonen

Bei vRealize Automation Cloud Assembly-Cloud-Zonen handelt es sich um Abschnitte innerhalb Ihres Cloud-Kontotyps, z. B. AWS oder vSphere. Cloud-Zonen sind projektspezifisch.

Zu den zusätzlichen Platzierungssteuerelementen gehören Optionen für Platzierungsrichtlinien, Funktions- und Computing-Tags.

■ Platzierungsrichtlinie

Mit Platzierungsrichtlinien wird die Hostauswahl innerhalb der angegebenen Region gesteuert.

- Standard – Computing-Ressourcen werden nach dem Zufallsprinzip auf Hosts platziert.
- Binpack – Computing-Ressourcen werden auf dem am stärksten ausgelasteten Host platziert, der über genügend Ressourcen zum Ausführen der betreffenden Berechnung verfügt.
- Verteilung – Computing-Ressourcen werden gleichmäßig auf alle verfügbaren Hosts verteilt.

■ Funktions-Tags

Blueprints enthalten Einschränkungs-Tags, die bei der Bestimmung der Bereitstellungsplatzierung hilfreich sind. Während der Bereitstellung werden Einschränkungs-Tags des Blueprints passenden Funktions-Tags in Cloud-Zonen zugewiesen und somit die Cloud-Zonen festgelegt, die für die Platzierung der Computing-Ressourcen verfügbar sind.

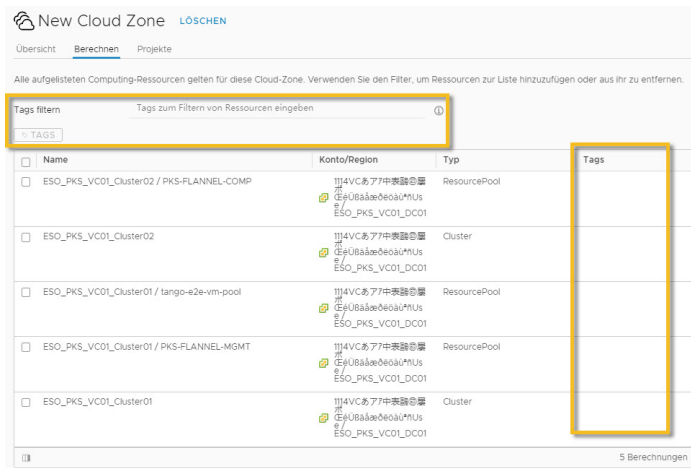
■ Berechnungen

Sie können die Berechnungen anzeigen und verwalten, die für diese Cloud-Zone bereitgestellt werden können.

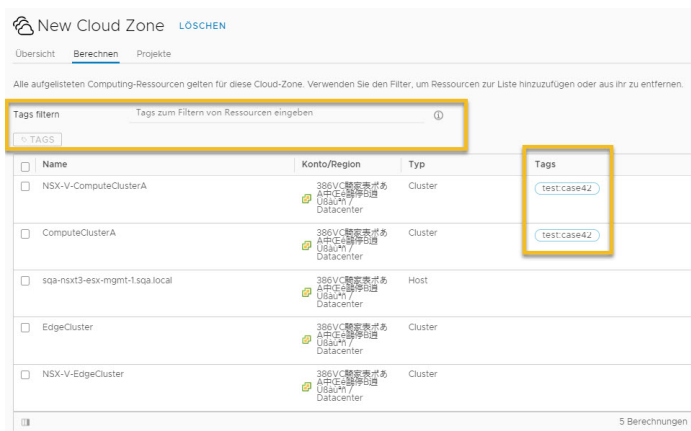
Wenn ein vCenter-Computing-Cluster DRS-fähig ist, wird in der Cloud-Zone nur der Cluster in der Liste der Berechnungen angezeigt und die untergeordneten Hosts werden nicht angezeigt. Wenn ein vCenter-Computing-Cluster nicht DRS-fähig ist, zeigt die Cloud-Zone nur eigenständige ESXi-Hosts an, sofern vorhanden.

Computing-Tags helfen bei der weiteren Steuerung der Platzierung. Mithilfe von Tags können Sie verfügbare Computing-Ressourcen so filtern, dass sie nur einem oder mehreren Tags entsprechen (siehe hierzu folgende Beispiele).

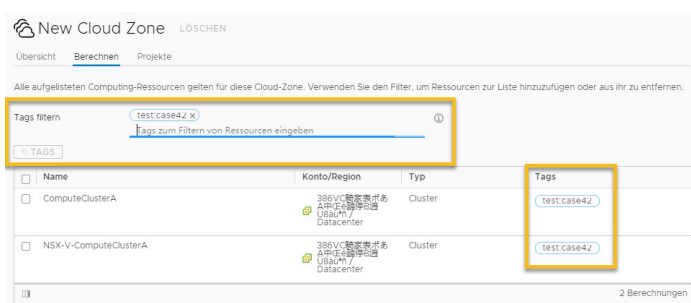
- Berechnungen enthalten keine Tags und es werden keine Filter verwendet.



- Zwei Berechnungen enthalten dasselbe Tag. Filter werden nicht verwendet.



- Zwei Berechnungen enthalten dasselbe Tag und der Tag-Filter entspricht dem Tag, das in den beiden Berechnungen verwendet wird.



■ Projekte

Sie können die Projekte anzeigen, die zur Unterstützung der Arbeitslastbereitstellung in dieser Cloud-Zone konfiguriert wurden.

Nach der Erstellung einer Cloud-Zone können Sie deren Konfiguration validieren.

Vorgehensweise zum Hinzufügen von vRealize Automation Cloud Assembly-Konfigurationszuordnungen für die Erstellung allgemeiner Maschinengrößen in vRealize Automation Cloud Assembly

Sie verwenden natürliche Sprache in einer vRealize Automation Cloud Assembly-Konfigurationszuordnung, um Zielbereitstellungsgrößen für ein bestimmtes Konto/eine bestimmte Region festzulegen.

Mit Konfigurationszuordnungen werden die Bereitstellungsgrößen angegeben, die für Ihre Umgebung sinnvoll sind. Beispiel: *Klein* für 1 CPU und 2 GB Arbeitsspeicher und *Groß* für 2 CPUs und 8 GB für ein vCenter-Konto in einem benannten Datacenter und t2.nano für ein Amazon Web Services-Konto in einer benannten Region.

Wählen Sie **Infrastruktur > Konfigurieren > Konfigurationszuordnungen** aus und klicken Sie auf **Neue Konfigurationszuordnung**.

Weitere Informationen zu Konfigurationszuordnungen in vRealize Automation Cloud Assembly

In einer Konfigurationszuordnung werden mehrere Zielbereitstellungsgrößen für ein bestimmtes Cloud-Konto und/oder eine bestimmte Region in vRealize Automation Cloud Assembly mithilfe natürlicher Sprachbezeichnungen zusammengefasst.

Mit der Konfigurationszuordnung können Sie eine benannte Zuordnung erstellen, die ähnliche Konfigurationsgrößen für Ihre Kontoregionen enthält. Eine Konfigurationszuordnung mit der Bezeichnung `standard_small` kann beispielsweise eine ähnliche Konfigurationsgröße (z. B. 1 CPU, 2 GB RAM) für bestimmte oder alle verfügbaren Konten/Regionen in Ihrem Projekt aufweisen. Wählen Sie beim Erstellen eines Blueprints eine verfügbare Konfiguration aus, die Ihren Anforderungen entspricht.

Verwalten Sie Konfigurationszuordnungen für Ihr Projekt nach Bereitstellungsabsicht.

Zum Vereinfachen der Blueprint-Erstellung können Sie eine Option zur Vorabkonfiguration auswählen, wenn Sie ein neues Cloud-Konto hinzufügen. Wenn Sie die Option zur Vorabkonfiguration auswählen, werden die beliebteste Konfigurations- und Image-Zuordnung in Ihrer Organisation für die angegebene Region verwendet.

In Bezug auf die Image-Zuordnung in Blueprints, die vSphere-Ressourcen enthalten, können Sie, sofern keine Konfigurationszuordnungen für eine vSphere-Cloud-Zone festgelegt sind, unbegrenzten Arbeitsspeicher und CPU konfigurieren, indem Sie die vSphere-spezifischen Einstellungen im Blueprint verwenden. Wenn für eine vSphere-Cloud-Zone Konfigurationszuordnungen definiert sind, dient die Konfigurationszuordnung als Grenzwert für vSphere-spezifische Konfigurationen im Blueprint.

Ein Beispiel für eine einfache Konfigurationszuordnung finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Konfigurationszuordnungen](#).

Vorgehensweise zum Hinzufügen von vRealize Automation Cloud Assembly-Image-Zuordnungen zum Erstellen allgemeiner Betriebssysteme

Sie verwenden natürliche Sprache in einer vRealize Automation Cloud Assembly-Image-Zuordnung, um Betriebssysteme als Zielbereitstellung für ein bestimmtes Konto/eine bestimmte Region festzulegen.

Wählen Sie **Infrastruktur > Konfigurieren > Image-Zuordnungen** aus und klicken Sie auf **Neue Image-Zuordnung**.

Weitere Informationen zu Image-Zuordnungen in vRealize Automation Cloud Assembly

In einer Image-Zuordnung werden mehrere vordefinierte Zielbetriebssystemspezifikationen für ein bestimmtes Cloud-Konto und/oder eine bestimmte Cloud-Region in vRealize Automation Cloud Assembly mithilfe natürlicher Sprachbezeichnungen zusammengefasst.

Cloud-Anbieterkonten, wie z. B. Microsoft Azure und Amazon Web Services, verwenden Images, um mehrere Zielbereitstellungsbedingungen zusammenzufassen, einschließlich Betriebssystem und zugehöriger Konfigurationseinstellungen. vCenter- und NSX-basierte Umgebungen, einschließlich VMware Cloud on AWS, verwenden einen ähnlichen Gruppierungsmechanismus, um mehrere Bereitstellungsbedingungen für Betriebssysteme zu definieren. Wenn Sie einen Blueprint erstellen und schließlich bereitstellen und durchlaufen, wählen Sie ein verfügbares Image aus, das Ihren Anforderungen entspricht.

Verwalten Sie Image-Zuordnungen für ein Projekt anhand ähnlicher Betriebssystemeinstellungen, der Kennzeichnungsstrategie und des Zwecks der funktionalen Bereitstellung.

Ein Beispiel über das Erstellen einer grundlegenden Image-Zuordnung finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Image-Zuordnungen](#).

Zum Vereinfachen der Blueprint-Erstellung können Sie eine Option zur Vorabkonfiguration auswählen, wenn Sie ein neues Cloud-Konto hinzufügen. Wenn Sie die Option zur Vorabkonfiguration auswählen, werden die beliebteste Konfigurations- und Image-Zuordnung in Ihrer Organisation für die angegebene Region verwendet.

Wenn Sie Image-Informationen zu einem Blueprint hinzufügen, verwenden Sie entweder den Eintrag `image` oder `imageRef` im Abschnitt `properties` der Maschinenkomponente. Wenn Sie beispielsweise einen Klon anhand eines Snapshots erstellen möchten, verwenden Sie die Eigenschaft `imageRef`.

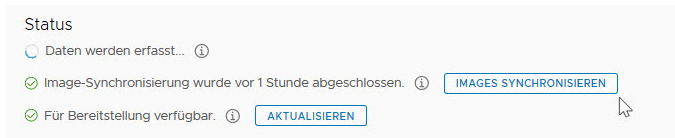
Beispiele für `image`- und `imageRef`-Einträge im Blueprint-Code finden Sie unter [Kapitel 6 Entwerfen Ihrer vRealize Automation Cloud Assembly-Bereitstellungen](#).

Um eine Berechtigung für eine Inhaltsbibliothek zuzuweisen, muss ein Administrator dem Benutzer die Berechtigung als globale Berechtigung erteilen. Weitere Informationen finden Sie im Abschnitt [Hierarchische Vererbung von Berechtigungen für Inhaltsbibliotheken](#) unter *Verwaltung virtueller vSphere-Maschinen* in der [VMware vSphere-Dokumentation](#).

Synchronisieren von Images für das Cloud-Konto/die Cloud-Region

Sie können die Image-Synchronisierung ausführen, um sicherzustellen, dass die Images, die Sie für ein bestimmtes Cloud-Konto/eine bestimmte Cloud-Region auf der Seite **Infrastruktur > Konfigurieren > Image-Zuordnung**, aktuell sind.

- 1 Öffnen Sie das/die zugeordnete **Cloud-Konto/Cloud-Region**, indem Sie **Infrastruktur > Verbindungen > Cloud-Konten** auswählen. Wählen Sie das vorhandene Cloud-Konto/die vorhandene Cloud-Region aus.
- 2 Klicken Sie auf die Schaltfläche **Images synchronisieren** und warten Sie, bis die Aktion abgeschlossen ist.



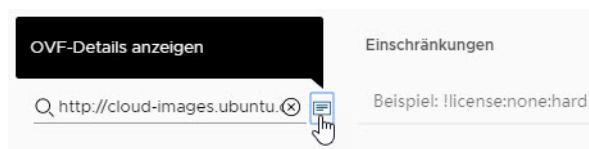
- 3 Wenn die Aktion abgeschlossen ist, klicken Sie auf **Infrastruktur > Konfigurieren > Image-Zuordnung**. Definieren Sie eine neue Image-Zuordnung oder bearbeiten Sie eine vorhandene und wählen Sie das Cloud-Konto/die Cloud-Region aus Schritt 1 aus.
- 4 Klicken Sie auf der Seite **Image-Zuordnung** auf das Symbol für die Image-Synchronisierung.



- 5 Konfigurieren Sie die Image-Zuordnungseinstellungen für das angegebene Cloud-Konto/die angegebene Cloud-Region auf der Seite **Image-Zuordnung**.

Anzeigen von OVF-Details

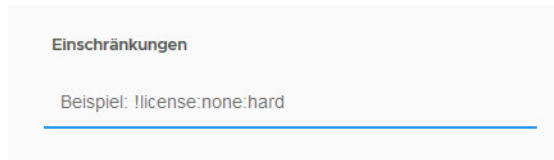
Sie können OVF-Spezifikationen in vRealize Automation Cloud Assembly-Blueprint-Objekten, z. B. vCenter-Maschinenkomponenten und -Image-Zuordnungen, einschließen. Wenn Ihr Image eine OVF-Datei enthält, können Sie deren Inhalt ermitteln, ohne die Datei öffnen zu müssen. Bewegen Sie den Mauszeiger über die OVF, um OVF-Details anzuzeigen, einschließlich des Namens und des Speicherorts. Weitere Informationen zum OVF-Dateiformat finden Sie unter [vCenter-OVF: Eigenschaft](#).



Verwenden von Einschränkungen und Tags zur Optimierung der Image-Auswahl

Zur weiteren Optimierung der Image-Auswahl in einem Blueprint können Sie eine oder mehrere Einschränkungen hinzufügen, um Tag-basierte Beschränkungen für den Typ des bereitzustellenden Images festzulegen. Das angegebene Beispiel für **Einschränkung**, das beim Erstellen oder Bearbeiten einer Image-Zuordnungskonfiguration angezeigt wird, ist !

`license:none:hard`. Im Beispiel wird eine Tag-basierte Einschränkung gezeigt, bei der das Image nur verwendet werden kann, wenn das `license:none`-Tag *nicht* im Blueprint vorhanden ist. Wenn Sie Tags wie `license:88` und `license:92` hinzufügen, kann das angegebene Image nur verwendet werden, wenn die Tags `license:88` und `license:92` *im* Blueprint vorhanden sind.



Verwenden eines Cloud-Konfigurationsskripts zur Steuerung der Bereitstellung

Sie können ein Cloud-Konfigurationsskript in einer Image-Zuordnung und/oder einem Blueprint verwenden, um benutzerdefinierte Betriebssystemmerkmale zu definieren, die in einer vRealize Automation Cloud Assembly-Bereitstellung verwendet werden sollen. Je nachdem, ob Sie einen Blueprint für eine Public Cloud oder Private Cloud bereitstellen, können Sie beispielsweise bestimmte Benutzerberechtigungen, Betriebssystemberechtigungen oder andere Bedingungen auf das Image anwenden. Ein Cloud-Konfigurationsskript verwendet ein `cloud-init`-Format für Linux-basierte Images oder ein `cloudbase-init`-Format für Windows-basierte Images. vRealize Automation Cloud Assembly unterstützt das Tool [cloud-init](#) für Linux-Systeme und das Tool [cloudbase-init](#) für Windows.

Auf Windows-Maschinen können Sie ein beliebiges Format für das Cloud-Konfigurationsskript verwenden, das von `cloudbase-init` unterstützt wird.

Die Maschinenressource im folgenden Beispiel-Blueprint-Code verwendet ein Image mit einem Cloud-Konfigurationsskript, dessen Inhalt im Eintrag `image` angezeigt wird.

```
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}
```

Zu erwartende Auswirkungen, wenn Image-Zuordnung und Blueprint ein Cloud-Konfigurationsskript enthalten

Wenn ein Blueprint mit einem Cloud-Konfigurationsskript eine Image-Zuordnung mit einem Cloud-Konfigurationsskript verwendet, werden beide Skripts kombiniert. Bei der Zusammenführungsaktion werden zuerst die Inhalte des Image-Zuordnungsskripts und anschließend die Inhalte des Blueprint-Skripts verarbeitet. Dabei wird berücksichtigt, ob die Skripts im `#cloud-config`-Format vorliegen oder nicht.

- Bei Skripts im `#cloud-config`-Format werden bei der Zusammenführung die Inhalte jedes Moduls (zum Beispiel `runcmd`, `users` und `write_files`) wie folgt kombiniert:
 - Bei Modulen, deren Inhalt eine Liste ist, werden die Listen der Befehle aus der Image-Zuordnung und aus dem Blueprint zusammengeführt, wobei Befehle, die in beiden Listen identisch sind, ausgeschlossen werden.
 - Bei Modulen, deren Inhalt ein Wörterbuch ist, werden die Befehle zusammengeführt, und das Ergebnis ist eine Kombination aus beiden Wörterbüchern. Wenn in beiden Wörterbüchern der gleiche Schlüssel vorhanden ist, bleibt der Schlüssel aus dem Wörterbuch für das Image-Zuordnungsskript erhalten, und der Schlüssel aus dem Wörterbuch des Blueprint-Skripts wird ignoriert.
 - Bei Modulen, deren Inhalt eine Zeichenfolge ist, werden die Inhaltswerte aus dem Image-Zuordnungsskript beibehalten, und die Inhaltswerte aus dem Blueprint-Skript werden ignoriert.
- Bei Skripts, die in einem anderen Format als `#cloud-config` vorliegen, oder wenn ein Skript das `#cloud-config`-Format aufweist und das andere nicht, werden beide Skripts so kombiniert, dass das Image-Zuordnungsskript zuerst ausgeführt wird und das Blueprint-Skript ausgeführt wird, nachdem das Image-Zuordnungsskript beendet wurde.

Weitere Informationen hierzu finden Sie unter [Zusammenführen von Abschnitten mit Benutzerdaten](#).

Weitere Informationen zum Konfigurieren und Verwenden von Cloud-Konfigurationsskripten

Informationen zum Konfigurieren von cloud-init für Windows-Bereitstellungen in vRealize Automation Cloud Assembly finden Sie unter [Vorgehensweise zum Einrichten einer Windows-Vorlage mit cloud-init oder cloudbase-init in vRealize Automation](#).

Informationen zur Verwendung von Cloud-Konfigurationsskripten finden Sie unter [Vorgehensweise zum automatischen Initialisieren einer Maschine in einem vRealize Automation Cloud Assembly-Blueprint](#) und im VMware-Blogger-Artikel [Customizing Cloud Assembly Deployments with Cloud-Init](#).

Hinzufügen von vRealize Automation Cloud Assembly-Netzwerkprofilen

Ein vRealize Automation Cloud Assembly-Netzwerkprofil beschreibt das Verhalten des bereitzustellenden Netzwerks.

Ein Netzwerk muss beispielsweise internetfähig sein und darf nicht nur intern verwendet werden. Netzwerke und ihre Profile sind Cloud-spezifisch.

Wählen Sie **Infrastruktur > Konfigurieren > Netzwerkprofile** aus und klicken Sie auf **Neues Netzwerkprofil**.

Weitere Informationen zu Netzwerkprofilen in vRealize Automation Cloud Assembly

Ein Netzwerkprofil definiert eine Gruppe von Netzwerken und Netzwerkeinstellungen, die für ein Cloud-Konto in einer bestimmten Region oder einem bestimmten Datacenter in vRealize Automation Cloud Assembly verfügbar sind.

In der Regel definieren Sie Netzwerkprofile zur Unterstützung einer Zielbereitstellungsumgebung, zum Beispiel eine kleine Testumgebung, in der ein vorhandenes Netzwerk nur über ausgehenden Zugriff oder über eine hohe Produktionsumgebung mit Lastausgleich verfügt, die eine Reihe von Sicherheitsrichtlinien benötigt. Stellen Sie sich ein Netzwerkprofil als eine Sammlung von arbeitslastspezifischen Netzwerkmerkmalen vor:

Funktionen in einem Netzwerkprofil

Ein Netzwerkprofil enthält spezifische Informationen für einen benannten Cloud-Kontotyp und eine Region in vRealize Automation Cloud Assembly, einschließlich der folgenden Einstellungen:

- Benanntes Cloud-Konto/benannte Region und optionale Funktions-Tags für das Netzwerkprofil.
- Benannte vorhandene Netzwerke und zugehörige Einstellungen.
- Netzwerkrichtlinien, die bedarfsgesteuerte und andere Aspekte des Netzwerkprofils definieren.
- Optionale Einbeziehung vorhandener Lastausgleichsdienste.
- Optionale Einbeziehung vorhandener Sicherheitsgruppen.

Sie bestimmen die Verwaltungsfunktionen für die Netzwerk-IP basierend auf dem Netzwerkprofil.

Funktions-Tags des Netzwerkprofils werden mit Einschränkungs-Tags in Blueprints abgeglichen, um die Netzwerkauswahl zu steuern. Darüber hinaus werden alle Tags, die den vom Netzwerkprofil erfassten Netzwerken zugewiesen sind, ebenfalls mit Tags im Blueprint abgeglichen, um die Netzwerkauswahl bei der Bereitstellung des Blueprints zu steuern.

Funktions-Tags sind optional. Funktions-Tags werden auf alle Netzwerke im Netzwerkprofil angewendet, allerdings nur dann, wenn die Netzwerke als Teil dieses Netzwerkprofils verwendet werden. Für Netzwerkprofile, die keine Funktions-Tags enthalten, findet der Tag-Abgleich nur auf den Netzwerk-Tags statt. Die Netzwerk- und Sicherheitseinstellungen, die im übereinstimmenden Netzwerkprofil definiert sind, werden bei der Bereitstellung des Blueprints angewendet.

Bei der Verwendung einer statischen IP wird der Adressbereich von vRealize Automation verwaltet. Für DHCP werden die IP-Start- und -Endadressen vom unabhängigen DHCP-Server verwaltet, und nicht von vRealize Automation. Bei Verwendung von DHCP oder einer gemischten Netzwerkadresszuteilung wird der Wert für die Netzwerknutzung auf null gesetzt. Ein bedarfsgesteuerter Netzwerkbereich basiert auf der im Netzwerkprofil angegebenen CIDR- und Subnetzgröße. Um sowohl statische als auch dynamische Zuweisungen in der Bereitstellung zu unterstützen, wird der zugeteilte Bereich in zwei Bereiche aufgeteilt: einen für die statische Zuteilung und einen weiteren für die dynamische Zuteilung.

Netzwerke

Netzwerke, die auch als Subnetze bezeichnet werden, sind logische Unterteilungen eines IP-Netzwerks. Ein Netzwerk gruppiert ein Cloud-Konto, eine IP-Adresse oder einen Bereich sowie Netzwerk-Tags, um zu steuern, wie und wo eine Blueprint-Bereitstellung stattfinden soll. Netzwerkparameter im Profil definieren, wie Maschinen in der Bereitstellung über IP-Ebene 3 miteinander kommunizieren können. Netzwerke können Tags aufweisen.

Sie können Netzwerke zum Netzwerkprofil hinzufügen, vom Netzwerkprofil verwendete Netzwerkaspekte bearbeiten und Netzwerke aus dem Netzwerkprofil entfernen.

■ Netzwerkdomäne oder Transportzone

Die Netzwerkdomäne oder Transportzone ist der Distributed Virtual Switch (dvSwitch) für die vSphere vNetwork Distributed PortGroups (dvPortGroup). Eine *Transportzone* ist ein vorhandenes NSX-Konzept, das mit den Begriffen *dvSwitch* oder *dvPortGroup* vergleichbar ist.

Wenn Sie ein Cloud-Konto von NSX verwenden, lautet der Elementname auf der Seite **Transportzone**, andernfalls **Netzwerkdomäne**.

Bei Standard-Switches entspricht die Netzwerkdomäne oder Transportzone dem Switch. Die Netzwerkdomäne oder Transportzone definiert die Begrenzungen der Subnetze innerhalb von vCenter.

Eine Transportzone steuert, welche Hosts ein logischer NSX-Switch erreichen kann. Sie kann sich auf einen oder mehrere vSphere-Cluster erstrecken. Transportzonen steuern, welche Cluster und welche virtuellen Maschinen an der Verwendung eines bestimmten Netzwerks teilnehmen können. Subnetze, die zur selben NSX-Transportzone gehören, können für dieselben Maschinenhosts verwendet werden.

■ Domäne

Die vCenter Single Sign-On-Domäne für eine virtuelle Zielmaschine. Domänen werden während der vSphere-Konfiguration von einem vCenter-Administrator konfiguriert. Die Domäne bestimmt den Bereich für die lokale Authentifizierung in vCenter.

- **IPv4-CIDR- und IPv4-Standard-Gateway**

vSphere-Cloud-Konten und vSphere-Maschinenkomponenten im Blueprint unterstützen Methoden für duales IPv6 und IPv4. Beispiel: 192.168.100.14/24 stellt die IPv4-Adresse 192.168.100.14 und das zugehörige Routing-Präfix 192.168.100.0 oder entsprechend die zugehörige Subnetzmaske 255.255.255.0 dar, die 24 führende 1-Bit aufweist. Der IPv4-Block 192.168.100.0/22 stellt die 1024 IP-Adressen von 192.168.100.0 bis 192.168.103.255 dar.

- **IPv6-CIDR- und IPv6-Standard-Gateway**

vSphere-Cloud-Konten und vSphere-Maschinenkomponenten im Blueprint unterstützen Methoden für duales IPv6 und IPv4. Beispiel: Bei 2001:db8::/48 handelt es sich um den IPv6-Adressblock von 2001:db8:0:0:0:0:0:0 bis 2001:db8:0:ffff:ffff:ffff:ffff:ffff.

Das IPv6-Format wird für bedarfsgesteuerte Netzwerke nicht unterstützt.

- **DNS-Server und DNS-Suchdomänen**

- **Öffentliche IP unterstützen**

Wählen Sie diese Option aus, um das Netzwerk als öffentlich zu kennzeichnen.

Netzwerkkomponenten in einem Blueprint, die eine Eigenschaft vom Typ `network type: public` aufweisen, werden mit Netzwerken abgeglichen, die als öffentlich gekennzeichnet sind. Weitere Übereinstimmungen treten während der Blueprint-Bereitstellung zur Ermittlung der Netzwerkauswahl auf.

- **Standardwert für Zone**

Wählen Sie diese Option aus, um das Netzwerk als Standardeinstellung für die Cloud-Zone zu kennzeichnen. Während der Blueprint-Bereitstellung werden Standardnetzwerke gegenüber anderen Netzwerken bevorzugt.

- **Herkunft**

Gibt die Netzwerkquelle an.

- **Tags**

Gibt ein oder mehrere Tags an, die dem Netzwerk zugewiesen sind. Tags sind optional. Mit dem Tag-Abgleich werden die Netzwerke bestimmt, die für Blueprint-Bereitstellungen verfügbar sind.

Netzwerk-Tags sind unabhängig vom Netzwerkprofil auf dem Netzwerkelement selbst vorhanden. Netzwerk-Tags gelten für jedes Vorkommen des Netzwerks, dem sie hinzugefügt wurden, und für alle Netzwerkprofile, die dieses Netzwerk enthalten. Netzwerke können in beliebig viele Netzwerkprofile eingeteilt werden. Unabhängig davon, wo sich das Netzwerkprofil befindet, ist ein Netzwerk-Tag mit diesem Netzwerk verknüpft, wenn das Netzwerk verwendet wird.

Wenn Sie einen Blueprint bereitstellen, werden Einschränkungs-Tags in den Netzwerkkomponenten eines Blueprints mit Netzwerk-Tags abgeglichen, einschließlich Funktions-Tags für Netzwerkprofile. Bei Netzwerkprofilen, die Funktions-Tags enthalten, werden die Funktions-Tags auf alle Netzwerke angewendet, die für dieses Netzwerkprofil verfügbar sind. Die Netzwerk- und Sicherheitseinstellungen, die im übereinstimmenden Netzwerkprofil definiert sind, werden bei der Bereitstellung des Blueprints angewendet.

Netzwerkrichtlinien

Je nach zugehörigem Cloud-Konto können Sie Netzwerkrichtlinien verwenden, um Einstellungen für die Netzwerktypen `outbound`, `private` und `routed` sowie für bedarfsgesteuerte Sicherheitsgruppen zu definieren. Sie können auch Netzwerkrichtlinien zum Steuern von `existing`-Netzwerken verwenden, wenn ein Lastausgleichsdienst mit diesem Netzwerk verknüpft ist.

Informationen zu Netzwerktypen finden Sie unter [Verwenden von Netzwerken und Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

Optionen für die folgenden bedarfsgesteuerten Auswahlen werden in der Hilfe zu **Netzwerkprofilen** beschrieben und nachfolgend zusammengefasst.

- **Kein bedarfsgesteuertes Netzwerk oder keine bedarfsgesteuerte Sicherheitsgruppe erstellen**

Sie können diese Option verwenden, wenn Sie den Netzwerktyp `existing` oder `public` angeben. Blueprints, die ein `outbound`, `private` oder `routed` Netzwerk benötigen, sind diesem Profil nicht zugeordnet.

- **Bedarfsgesteuertes Netzwerk erstellen**

Sie können diese Option verwenden, wenn Sie den Netzwerktyp `outbound`, `private` oder `routed` angeben.

Amazon Web Services, Microsoft Azure, NSX, vSphere und VMware Cloud on AWS unterstützen diese Option.

- **Bedarfsgesteuerte Sicherheitsgruppe erstellen**

Sie können diese Option verwenden, wenn Sie den Netzwerktyp `outbound` oder `private` angeben.

Eine neue Sicherheitsgruppe wird für abgeglichene Blueprints erstellt, wenn der Netzwerktyp `outbound` oder `private` lautet.

Amazon Web Services, Microsoft Azure, NSX und VMware Cloud on AWS unterstützen diese Option.

Netzwerkrichtlinieneinstellungen können spezifisch für den Cloud-Kontotyp sein. Diese Einstellungen werden in der Wegweiser-Hilfe beschrieben und nachfolgend zusammengefasst:

- **Netzwerkdomäne oder Transportzone**

Die Netzwerkdomäne oder Transportzone ist der Distributed Virtual Switch (dvSwitch) für die vSphere vNetwork Distributed PortGroups (dvPortGroup). Eine *Transportzone* ist ein vorhandenes NSX-Konzept, das mit den Begriffen *dvSwitch* oder *dvPortGroup* vergleichbar ist.

Wenn Sie ein Cloud-Konto von NSX verwenden, lautet der Elementname auf der Seite **Transportzone**, andernfalls **Netzwerkdomäne**.

Bei Standard-Switches entspricht die Netzwerkdomäne oder Transportzone dem Switch. Die Netzwerkdomäne oder Transportzone definiert die Begrenzungen der Subnetze innerhalb von vCenter.

Eine Transportzone steuert, welche Hosts ein logischer NSX-Switch erreichen kann. Sie kann sich auf einen oder mehrere vSphere-Cluster erstrecken. Transportzonen steuern, welche Cluster und welche virtuellen Maschinen an der Verwendung eines bestimmten Netzwerks teilnehmen können. Subnetze, die zur selben NSX-Transportzone gehören, können für dieselben Maschinenhosts verwendet werden.

■ Externes Subnetz

Ein bedarfsgesteuertes Netzwerk mit ausgehendem Zugriff erfordert ein externes Subnetz, das über ausgehenden Zugriff verfügt. Das externe Subnetz wird zur Bereitstellung von ausgehendem Zugriff verwendet, wenn es im Blueprint angefordert wird – die Netzwerkplatzierung wird dadurch nicht gesteuert. Das externe Subnetz hat beispielsweise keinen Einfluss auf die Platzierung eines privaten Netzwerks.

■ CIDR

Bei einer CIDR-Notation handelt es sich um eine komprimierte Darstellung einer IP-Adresse und des zugehörigen Routing-Präfixes. Der CIDR-Wert gibt den bei der Bereitstellung zu verwendenden Netzwerkadressbereich für die Erstellung von Subnetzen an. Diese CIDR-Einstellung auf der Registerkarte **Netzwerkrichtlinien** akzeptiert IPv4-Notationen, die auf „in/nn“ enden und Werte zwischen 0 und 32 enthalten.

■ Subnetzgröße

Mit dieser Option wird die Größe des bedarfsgesteuerten Netzwerks unter Verwendung der IPv4-Notation für jedes isolierte Netzwerk in einer Bereitstellung festgelegt, die dieses Netzwerkprofil verwendet. Die Einstellung „Subnetzgröße“ steht für die Verwaltung interner und externer IP-Adressen zur Verfügung.

Das IPv6-Format wird für bedarfsgesteuerte Netzwerke nicht unterstützt.

■ Distributed Logical Router

Für ein geroutetes bedarfsgesteuertes Netzwerk müssen Sie beispielsweise ein verteiltes logisches Netzwerk angeben, wenn Sie ein NSX-V-Cloud-Konto verwenden.

Ein Distributed Logical Router (DLR) wird verwendet, um den horizontalen Datenverkehr zwischen bedarfsgesteuerten gerouteten Netzwerken auf NSX-V weiterzuleiten. Diese Option wird nur angezeigt, wenn der Konto-/Regionswert für das Netzwerkprofil einem NSX-V-Cloud-Konto zugeordnet ist.

■ IP-Bereichszuweisung

Die Option ist für Cloud-Konten verfügbar, die NSX oder VMware Cloud on AWS unterstützen, einschließlich vSphere.

Sie können eine der folgenden drei Optionen auswählen, um einen IP-Bereichszuweisungstyp für das Bereitstellungsnetzwerk anzugeben:

■ Statisch und DHCP

Standardwert wird empfohlen. Diese Mischoption verwendet die zugeteilten Einstellungen für **CIDR** und **Subnetzbereich**, um den DHCP-Serverpool so zu konfigurieren, dass er die Hälfte der Adressraumzuteilung mithilfe der (dynamischen) DHCP-Methode und die Hälfte der IP-Adressraumzuteilung mithilfe der statischen Methode unterstützt. Verwenden Sie diese Option, wenn einige der mit einem bedarfsgesteuerten Netzwerk verbundenen Maschinen zugewiesene, statische IP-Adressen und einige andere Maschinen dynamische IP-Adressen benötigen. Zwei IP-Bereiche werden erstellt.

Diese Option ist äußerst wirksam bei Bereitstellungen mit Maschinen, die mit einem bedarfsgesteuerten Netzwerk verbunden sind, das Maschinen mit statischen IPs und Maschinen mit dynamisch über einen NSX-DHCP-Server zugewiesenen IPs sowie Bereitstellungen enthält, bei denen die Lastausgleichsdienst-VIP statisch ist.

■ DHCP (dynamisch)

Diese Option verwendet die zugewiesene CIDR-Adresse, um einen IP-Pool auf einem DHCP-Server zu konfigurieren. Alle IP-Adressen für dieses Netzwerk werden dynamisch zugewiesen. Für jede zugewiesene CIDR-Instanz wird ein einzelner IP-Bereich erstellt.

■ Statisch

Diese Option verwendet die zugewiesene CIDR-Adresse, um IP-Adressen statisch zuzuteilen. Verwenden Sie diese Option, wenn kein konfigurierter DHCP-Server für dieses Netzwerk benötigt wird. Für jede zugewiesene CIDR-Instanz wird ein einzelner IP-Bereich erstellt.

■ Netzwerkressourcen – Externes Netzwerk

Externe Netzwerke werden auch als vorhandene Netzwerke bezeichnet. Diese Netzwerke werden datentechnisch erfasst und zur Auswahl bereitgestellt.

■ Netzwerkressourcen – Logischer Tier-0 Router

NSX-T verwendet den logischen Tier-0-Router als Gateway zu Netzwerken, die sich außerhalb der NSX-Bereitstellung befinden. Der logische Tier-0-Router konfiguriert den ausgehenden Zugriff für bedarfsgesteuerte Netzwerke.

■ Netzwerkressourcen – Edge-Cluster

Der angegebene Edge-Cluster stellt Routing-Dienste bereit. Der Edge-Cluster wird zum Konfigurieren des ausgehenden Zugriffs für bedarfsgesteuerte Netzwerke und Lastausgleichsdienste verwendet. Er erkennt den Edge-Cluster oder Ressourcenpool, in dem die Edge-Appliance bereitgestellt werden soll.

■ **Netzwerkressourcen – Edge-Datenspeicher**

Der angegebene Edge-Datenspeicher, der für die Bereitstellung der Edge-Appliance verwendet wird. Diese Eigenschaft gilt nur für NSX-V.

Lastausgleichsdienste

Sie können Lastausgleichsdienste zum Netzwerkprofil hinzufügen. Aufgelistete Lastausgleichsdienste sind basierend auf Informationen verfügbar, die vom Cloud-Quellkonto erfasst wurden.

Wenn ein Tag in einem der Lastausgleichsdienste im Netzwerkprofil einem Tag entspricht, das in einer Lastausgleichsdienstkomponente im Blueprint verwendet wird, wird der Lastausgleichsdienst während der Bereitstellung berücksichtigt. Lastausgleichsdienste in einem übereinstimmenden Netzwerkprofil werden verwendet, wenn ein Blueprint bereitgestellt wird.

Weitere Informationen finden Sie unter [Verwenden der Einstellungen des Lastausgleichsdiensts in Netzwerkprofilen und Blueprint-Designs in vRealize Automation Cloud Assembly](#) und [Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in vRealize Automation Cloud Assembly-Blueprints](#).

Sicherheitsgruppen

Wenn ein Blueprint bereitgestellt wird, werden die Sicherheitsgruppen im zugehörigen Netzwerkprofil auf die bereitgestellten Netzwerkkarten der Maschinen angewendet. Für ein Amazon Web Services-spezifisches Netzwerkprofil stehen die Sicherheitsgruppen im Netzwerkprofil in derselben Netzwerkdomeäne (VPC) bereit wie die Netzwerke, die auf der Registerkarte „Netzwerke“ aufgelistet sind. Wenn auf der Registerkarte „Netzwerke“ im Netzwerkprofil keine Netzwerke aufgeführt sind, werden alle verfügbaren Sicherheitsgruppen angezeigt.

Sie können eine Sicherheitsgruppe verwenden, um die Isolierungseinstellungen für ein bedarfsgesteuertes `private-` oder `outbound-`Netzwerk weiter zu definieren. Sicherheitsgruppen werden auch auf `existing-`Netzwerke angewendet.

Sicherheitsgruppen werden auf alle Maschinen in der Bereitstellung angewendet, die mit dem Netzwerk verbunden sind, das mit dem Netzwerkprofil übereinstimmt. Da möglicherweise mehrere Netzwerke in einem Blueprint vorhanden sind, die jeweils einem anderen Netzwerkprofil entsprechen, können Sie verschiedene Sicherheitsgruppen für verschiedene Netzwerke verwenden.

Durch das Hinzufügen eines Tags zu einer vorhandenen Sicherheitsgruppe können Sie die Sicherheitsgruppe in einer Blueprint-`Cloud.SecurityGroup`-Komponente verwenden. Eine Sicherheitsgruppe muss über mindestens ein Tag verfügen, ansonsten kann sie nicht in einem Blueprint verwendet werden. Weitere Informationen finden Sie unter [Sicherheitsressourcen und Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in vRealize Automation Cloud Assembly-Blueprints](#).

Weitere Informationen zu Netzwerkprofilen, Netzwerken, Blueprints und Tags

Weitere Informationen zu Netzwerkprofilen finden Sie in anderen Themen in diesem Abschnitt der Hilfe sowie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Netzwerkprofilen](#).

Weitere Informationen zu Netzwerken finden Sie unter [Netzwerkressourcen](#).

Beispiele für Netzwerkkomponentencode in einem Blueprint finden Sie unter [Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in vRealize Automation Cloud Assembly-Blueprints](#).

Beispiele für Netzwerkautomatisierungs-Workflows finden Sie unter [Netzwerkautomatisierung mit Cloud Assembly und NSX](#).

Weitere Informationen zu Tags und Tag-Strategien finden Sie unter [Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen](#).

Arbeiten mit IP-Adressen in Netzwerken und Netzwerkprofilen in vRealize Automation Cloud Assembly

Über die Netzwerk- und Netzwerkprofileinstellungen können Sie steuern, wie Netzwerk-IP-Adressen in vRealize Automation Cloud Assembly-Blueprints und -Bereitstellungen verwendet werden.

Über Netzwerkprofile können Sie Subnetze für vorhandene Netzwerkdomeinen definieren, die Einstellungen für statische IP-Adressen, DHCP oder eine Mischung aus statischen und DHCP-IP-Adresseinstellungen enthalten.

Auf der Registerkarte **Netzwerkrichtlinien** können Sie Subnetze definieren und IP-Adresseinstellungen angeben. Weitere Informationen finden Sie unter [Weitere Informationen zu Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

IPv4- und IPv6-Unterstützung in vRealize Automation Cloud Assembly-Netzwerken

vRealize Automation Cloud Assembly unterstützt reines IPv4 oder Dual-Stack-IPv4 und -IPv6. Reines IPv6 wird derzeit nicht unterstützt.

Während reines IPv4 für alle Cloud-Konto- und Integrationstypen unterstützt wird, werden Dual-Stack-IPv4 und -IPv6 nur für vSphere Cloud-Konten und deren Endpoints unterstützt.

IPv6 wird derzeit nicht für die Verwendung mit Lastausgleichsdiensten, bedarfsgesteuerten NSX-Netzwerken oder externen IPAM-Drittanbietern unterstützt.

Unterstützung des externen IPAM-Anbieters

Zusätzlich zu der bereitgestellten internen IPAM-Unterstützung können Sie einen externen IPAM-Anbieter verwenden, um IP-Adressen für die Netzwerke in Ihren Blueprints und Bereitstellungen dynamisch oder statisch zuzuweisen.

Die Unterstützung von externen IPAM-Anbietern, wie z. B. Infoblox, ist für anbieterspezifische IPAM-Integrationspunkte verfügbar, die Sie über die Menüfolge **Infrastruktur > Verbindungen > Integration hinzufügen > IPAM** hinzufügen.

Optionen zum Definieren von IPAM-Drittanbieter-Adressinformationen sind über die Option **IPAM-IP-Bereich hinzufügen** auf der Seite **Netzwerkrichtlinien > IPAM-IP-Bereich hinzufügen** verfügbar.

Informationen zum Erstellen eines externen IPAM-Integrationspunkts finden Sie unter [Konfigurieren eines externen IPAM-Integrationspunkts in vRealize Automation](#). Ein Beispiel für die Erstellung eines IPAM-Integrationspunkts für einen bestimmten IPAM-Anbieter finden Sie unter [Anwendungsfall für eine anbieterspezifische externe IPAM-Integration](#).

Verwenden von Netzwerken und Netzwerkprofilen in vRealize Automation Cloud Assembly

Sie verwenden Netzwerke und Netzwerkprofile in vRealize Automation Cloud Assembly, um das Verhalten der Netzwerkbereitstellung für Ihre Bereitstellungen zu definieren.

In vRealize Automation Cloud Assembly können Sie Cloud-spezifische Netzwerkprofile definieren. Weitere Informationen hierzu finden Sie unter [Weitere Informationen zu Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

Netzwerktypen

Eine Netzwerkkomponente in einem Blueprint ist als einer der folgenden `networkType`-Typen definiert.

Netzwerktyp	Definition
<code>existing</code>	Wählt ein vorhandenes Netzwerk aus, das auf dem zugrunde liegenden Cloud-Anbieter konfiguriert ist, wie z. B. vCenter, Amazon Web Services und Microsoft Azure. Ein vorhandenes Netzwerk ist für das bedarfsgesteuerte <code>outbound</code> -Netzwerk erforderlich. Sie können einen Bereich von statischen IP-Adressen in einem vorhandenen Netzwerk definieren.
<code>public</code>	Auf Computer in einem öffentlichen Netzwerk kann über das Internet zugegriffen werden. Diese Netzwerke werden von einem IT-Administrator definiert. Die Definition eines <code>public</code> -Netzwerks ist identisch mit der eines <code>existing</code> -Netzwerks bei Netzwerken, die Netzwerkdatenverkehr in öffentlichen Netzwerken zulassen.
<code>private</code>	Ein bedarfsgesteuerter Netzwerktyp. Schränkt den Netzwerkdatenverkehr so ein, dass er nur zwischen Ressourcen im bereitgestellten Netzwerk erfolgt. Eingehender und ausgehender Datenverkehr werden verhindert. In NSX kann er mit bedarfsgesteuerter 1:n-NAT gleichgesetzt werden.

Netzwerktyp	Definition
outbound	<p>Ein bedarfsgesteuerter Netzwerktyp.</p> <p>Schränkt den Netzwerkdatenverkehr zwischen den Computing-Ressourcen in der Bereitstellung ein, ermöglicht aber auch unidirektionalen ausgehenden Netzwerkdatenverkehr. In NSX kann er mit bedarfsgesteuerter 1:n-NAT und externer IP-Adresse gleichgesetzt werden.</p>
routed	<p>Ein bedarfsgesteuerter Netzwerktyp.</p> <p>Geroutete Netzwerke enthalten einen routingfähigen IP-Adressbereich, der auf verfügbare miteinander verknüpfte Subnetze aufgeteilt ist. Die virtuellen Maschinen, die mit gerouteten Netzwerken bereitgestellt werden und die dasselbe geroutete Netzwerkprofil aufweisen, können miteinander und mit einem externen Netzwerk kommunizieren.</p> <p>Bei gerouteten Netzwerken handelt es sich um einen bedarfsgesteuerten Netzwerktyp, der für NSX-V- und NSX-T-Netzwerke verfügbar ist. Microsoft Azure und Amazon Web Services stellen standardmäßig diese Konnektivität bereit.</p> <p>Ein <code>routed</code>-Netzwerk ist nur für die Blueprint-Spezifikation in einer <code>Cloud.NSX.Network</code>-Netzwerkkomponente verfügbar.</p>

Beispiele für aufgefüllte Blueprints, die Netzwerkkomponentendaten enthalten, finden Sie unter [Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in vRealize Automation Cloud Assembly-Blueprints](#).

Netzwerkszenarien

Sie können folgendes Verhalten erwarten, wenn Sie einen Blueprint bereitstellen, der die folgende Netzwerkprofilkonfiguration verwendet.

Tabelle 4-1. Netzwerkszenarien

Netzwerktyp oder Szenario	Keine Netzwerkprofile für Cloud-Zone verfügbar	Für die Cloud-Zone verfügbare Netzwerkprofile
Kein Netzwerk	<p>Wenn im Blueprint kein Netzwerk angegeben ist, wird ein zufälliges Netzwerk aus derselben Bereitstellungsregion wie die Berechnung ausgewählt.</p> <p>Bevorzugt werden Netzwerke, die als Standard gekennzeichnet sind.</p> <p>Wenn in einer verfügbaren Bereitstellungsregion keine Netzwerke vorhanden sind, schlägt die Bereitstellung fehl.</p>	<p>Ein Netzwerk wird aus einem passenden Netzwerkprofil ausgewählt.</p> <p>Bevorzugt werden Netzwerke, die als Standard gekennzeichnet sind.</p> <p>Wenn keines der Netzwerkprofile die Kriterien erfüllt, schlägt die Bereitstellung fehl.</p>
Vorhandenes Netzwerk	<p>Wenn die Netzwerkkomponente im Blueprint Einschränkungs-Tags enthält, werden diese Einschränkungen zum Filtern der Liste der verfügbaren Netzwerke verwendet. Einschränkungs-Tags in der Netzwerkkomponente des Blueprints werden mit Netzwerk-Tags und gegebenenfalls mit den Einschränkungs-Tags des Netzwerkprofils abgeglichen.</p> <p>In der gefilterten Liste der Netzwerke wird ein einzelnes Netzwerk aus derselben Bereitstellungsregion wie die Berechnung ausgewählt.</p> <p>Bevorzugt werden Netzwerke, die als Standard gekennzeichnet sind.</p> <p>Wenn nach der auf Einschränkungen basierenden Filterung keine Netzwerke in der Bereitstellungsregion vorhanden sind, schlägt die Bereitstellung fehl.</p>	<p>Ein Netzwerk wird aus einem passenden Netzwerkprofil ausgewählt.</p> <p>Bevorzugt werden Netzwerke, die als Standard gekennzeichnet sind.</p> <p>Wenn keines der Netzwerkprofile die Kriterien erfüllt, schlägt die Bereitstellung fehl.</p> <p>Netzwerkeinschränkungen können verwendet werden, um vorhandene Netzwerke im Profil basierend auf ihren vorab zugewiesenen Tags zu filtern.</p>

Tabelle 4-1. Netzwerkszenarien (Fortsetzung)

Netzwerktyp oder Szenario	Keine Netzwerkprofile für Cloud-Zone verfügbar	Für die Cloud-Zone verfügbare Netzwerkprofile
Öffentliches Netzwerk	<p>Verfügt das Netzwerk über Einschränkungen, werden diese Einschränkungen zum Filtern der Liste der verfügbaren Netzwerke verwendet, für die das Attribut <code>supports public IP</code> festgelegt wurde.</p> <p>In der gefilterten Liste der Netzwerke wird ein zufälliges Netzwerk aus derselben Bereitstellungsregion wie die Berechnung ausgewählt.</p> <p>Bevorzugt werden Netzwerke, die als Standard gekennzeichnet sind.</p> <p>Wenn nach der auf Einschränkungen basierenden Filterung keine öffentlichen Netzwerke in der Bereitstellungsregion vorhanden sind, schlägt die Bereitstellung fehl.</p>	<p>Ein Netzwerk mit dem Attribut <code>supports public IP</code> wird aus einem passenden Netzwerkprofil ausgewählt.</p> <p>Bevorzugt werden Netzwerke, die als Standard gekennzeichnet sind.</p> <p>Netzwerkeinschränkungen können verwendet werden, um vorhandene öffentliche Netzwerke im Profil basierend auf vorab zugewiesenen Tags zu filtern.</p>
Privates Netzwerk	Die Bereitstellung schlägt fehl, da private Netzwerke Informationen aus einem Netzwerkprofil benötigen.	<p>Ein neues Netzwerk oder eine neue Sicherheitsgruppe wird basierend auf den Einstellungen im zugeordneten Netzwerkprofil erstellt.</p> <p>Netzwerkeinschränkungs-Tags können zum Filtern von Netzwerkprofilen und Netzwerken verwendet werden.</p>
Ausgehendes Netzwerk	Die Bereitstellung schlägt fehl, da ausgehende Netzwerke Informationen aus einem Netzwerkprofil benötigen.	<p>Ein neues Netzwerk oder eine neue Sicherheitsgruppe wird basierend auf den Einstellungen im zugeordneten Netzwerkprofil erstellt.</p> <p>Netzwerkeinschränkungs-Tags können zum Filtern von Netzwerkprofilen und Netzwerken verwendet werden.</p>
Geroutetes On-Demand-Netzwerk	Die Bereitstellung schlägt fehl, da geroutete Netzwerke Informationen aus einem Netzwerkprofil benötigen.	<p>Für NSX-V ist die Auswahl des DLR (Distributed Logical Router) erforderlich.</p> <p>Für NSX-T und VMware Cloud on AWS werden ähnliche On-Demand-Einstellungen wie „privat“ und „ausgehend“ benötigt.</p>
Beispielhafter WordPress-Anwendungsfall mit vorhandenen oder öffentlichen Netzwerken	Die Bereitstellung erfolgt gemäß der Beschreibung für ein vorhandenes oder öffentliches Netzwerk.	<p>Oben finden Sie Beschreibungen zum Verhalten vorhandener und öffentlicher Netzwerke.</p> <p>Weitere Informationen hierzu finden Sie unter Anwendungsbeispiel: WordPress.</p>

Tabelle 4-1. Netzwerkszenarien (Fortsetzung)

Netzwerktyp oder Szenario	Keine Netzwerkprofile für Cloud-Zone verfügbar	Für die Cloud-Zone verfügbare Netzwerkprofile
Beispielhafter WordPress-Anwendungsfall mit vorhandenen oder öffentlichen Netzwerken und privaten oder ausgehenden Netzwerken	Die Bereitstellung schlägt fehl, da das Netzwerk Informationen aus einem Netzwerkprofil benötigt.	Oben finden Sie Beschreibungen für ein privates und ein ausgehendes Netzwerk. Weitere Informationen hierzu finden Sie unter Anwendungsbeispiel: WordPress .
Beispielhafter WordPress-Anwendungsfall mit Lastausgleichsdienst	Die Bereitstellung schlägt fehl, da ein Lastausgleichsdienst Informationen aus einem Netzwerkprofil benötigt. Die Bereitstellung kann stattfinden, wenn vorhandene Lastausgleichsdienste zur Verfügung stehen.	Ein neuer Lastausgleichsdienst wird basierend auf der Konfiguration des Netzwerkprofils erstellt. Sie können einen vorhandenen Lastausgleichsdienst angeben, der im Netzwerkprofil aktiviert wurde. Die Bereitstellung schlägt fehl, wenn Sie einen vorhandenen Lastausgleichsdienst anfordern, der die Einschränkungen im Netzwerkprofil nicht erfüllt. Weitere Informationen hierzu finden Sie unter Anwendungsbeispiel: WordPress .

Verwenden der Einstellungen des Lastausgleichsdiensts in Netzwerkprofilen und Blueprint-Designs in vRealize Automation Cloud Assembly

Sie können Lastausgleichsdienste im vRealize Automation Cloud Assembly-Blueprint basierend auf der Konfiguration Ihres NSX-V- oder NSX-T-Netzwerkprofils konfigurieren.

Die Optionen des Lastausgleichsdiensts richten sich nach den Einstellungen im Netzwerkprofil des Blueprints und nach den Einstellungen in der NSX-T- oder NSX-V-Quellanwendung.

NSX-T-Netzwerke und Optionen des Lastausgleichsdiensts

NSX-T unterstützt einen Lastausgleichsdienst für jeden logischen Tier-1-Router. Die Optionen des Lastausgleichsdiensts hängen von dem Netzwerk ab, mit dem die Lastausgleichsdienstkomponente in Ihrem Blueprint verknüpft ist.

■ Ausgehendes bedarfsgesteuertes Netzwerk

Wenn die Lastausgleichsdienst-Berechnungen mit einem bedarfsgesteuerten ausgehenden Netzwerk verbunden sind, wird ein Lastausgleichsdienst für den Tier-1-Router im bedarfsgesteuerten Netzwerk erstellt.

■ Privates bedarfsgesteuertes Netzwerk

Die VIP des bedarfsgesteuerten Netzwerks muss mit der VIP des externen Quellnetzwerks verknüpft sein.

Wenn die Lastausgleichsdienst-Berechnungen mit einem bedarfsgesteuerten privaten Netzwerk verbunden sind, wird ein neuer Tier-1-Router erstellt und mit dem im Netzwerkprofil angegebenen Tier-0-Router verbunden. Der Lastausgleichsdienst wird dann an den Tier-1-Router angehängt. Die VIP-Ankündigung des Tier-1-Routers ist aktiviert, wenn sich die VIP in einem externen Netzwerk befindet.

■ Vorhandenes Netzwerk

Wenn der Lastausgleichsdienst mit einem vorhandenen Netzwerk verbunden ist, wird der Lastausgleichsdienst mit dem Tier-1-Router des vorhandenen Netzwerks erstellt. Ein neuer kleiner Lastausgleichsdienst wird nur erstellt, wenn kein Lastausgleichsdienst an den Tier-1-Router angehängt ist. Wenn der Lastausgleichsdienst bereits vorhanden ist, werden ihm neue virtuelle Server angehängt. Wenn das vorhandene Netzwerk nicht mit einem Tier-1-Router verbunden ist, wird ein neuer Tier-1-Router erstellt und an einen Tier-0-Router angehängt, der im Netzwerkprofil definiert ist. Die VIP-Ankündigung des Tier-1-Routers ist nicht aktiviert.

■ Bedarfsgesteuertes Netzwerk mit neuer Sicherheitsgruppe

Da Maschinen mit einem vorhandenen Netzwerk verbunden sind und die Isolierung mithilfe einer Sicherheitsgruppe erstellt wird, ähnelt diese Option einem Lastausgleichsdienst, der in einem vorhandenen Netzwerk erstellt wurde. Der Unterschied besteht darin, dass zum Aktivieren des Datenpfads die IP des Tier-1-Uplink-Ports zur Sicherheitsgruppe der Isolierung hinzugefügt wird.

Ein für ein bedarfsgesteuertes Netzwerk erstellter Lastausgleichsdienst wird gelöscht, wenn er nicht mehr verwendet wird. Monitore, Pools, Anwendungsprofile und VIPs werden gelöscht, wenn der Lastausgleichsdienst gelöscht wird. Ein für ein externes Netzwerk erstellter Lastausgleichsdienst wird nicht gelöscht.

Sie können NSX-T-Einstellungen für den Lastausgleichsdienst im Blueprint mithilfe der Cloud-unabhängigen Lastausgleichsdienstkomponente angeben, die im Komponentenbereich des Blueprints verfügbar ist.

Weitere Informationen zur Verwendung eines Lastausgleichsdiensts in einer NSX-T-Netzwerkbereitstellung finden Sie im VMware-Blogbeitrag [vRA Cloud Assembly-Lastausgleichsdienst mit NSX-T Deep Dive](#).

NSX-V-Netzwerke und Optionen des Lastausgleichsdiensts

Sie können einen Lastausgleichsdienst zweigliedrig oder eingliedrig konfigurieren, je nachdem, ob Sie ein isoliertes Netzwerk mithilfe eines neuen Netzwerksubnetzes oder einer Sicherheitsgruppe erstellen.

Sie können NSX-V-Einstellungen für den Lastausgleichsdienst im Blueprint angeben, indem Sie die Lastausgleichsdienstkomponente verwenden, die im Abschnitt NSX des Blueprint-Komponentenbereichs verfügbar ist.

Der Workflow zum Erstellen eines zweigliedrigen Lastausgleichsdiensts lautet:

- 1 Erstellen Sie einen Edge-Dienst.

- 2 Verbinden Sie die Uplink-Schnittstelle des Edge-Diensts mit dem öffentlichen Netzwerk
- 3 Verbinden Sie die Downlink-Schnittstelle mit dem isolierten (ausgehenden) Netzwerk.
- 4 Teilen Sie eine statische IP-Adresse für den Lastausgleichsdienst aus dem statischen IP-Bereich des Netzwerkprofils zu.
- 5 Konfigurieren Sie den Lastausgleichsdienst.
- 6 Konfigurieren Sie die Firewall.
- 7 Konfigurieren Sie das Standard-Gateway.

Der Workflow zum Erstellen eines eingliedrigen Lastausgleichsdiensts lautet:

- 1 Erstellen Sie einen Edge-Dienst.
- 2 Verbinden Sie das von der Reservierung ausgewählte Netzwerk als Uplink.
- 3 Teilen Sie eine statische IP-Adresse für den Lastausgleichsdienst aus dem statischen IP-Bereich des Netzwerkprofils zu.
- 4 Konfigurieren Sie den Lastausgleichsdienst.
- 5 Konfigurieren Sie die Firewall
- 6 Konfigurieren Sie das Standard-Gateway.

Vorgehensweise zum Hinzufügen von vRealize Automation Cloud Assembly-Speicherprofilen, die verschiedenen Anforderungen entsprechen

Ein vRealize Automation Cloud Assembly-Speicherprofil beschreibt die Art des bereitzustellenden Speichers.

Für Speicher wird in der Regel ein Profil anhand von Merkmalen wie Dienstebene oder Kosten, Leistung oder Zweck (z. B. Sicherung) erstellt.

Wählen Sie **Infrastruktur > Konfigurieren > Speicherprofile** aus und klicken Sie auf **Neues Speicherprofil**.

Weitere Informationen zu vRealize Automation Cloud Assembly-Speicherprofilen

Ein Cloud-Kontobereich enthält Speicherprofile, mit denen der Cloud-Administrator Speicher für die Region festlegen kann.

Speicherprofile enthalten Festplattenanpassungen sowie eine Möglichkeit zur Angabe des Speichertyps anhand von Funktions-Tags. Tags werden dann mit den Anforderungseinschränkungen des Bereitstellungsdiensts abgeglichen, um den gewünschten Speicher zur Bereitstellungszeit zu erstellen.

Speicherprofile sind nach Cloud-spezifischen Regionen organisiert. Ein Cloud-Konto kann aus verschiedenen Regionen mit mehreren Speicherprofilen pro Region bestehen.

Anbieterunabhängige Platzierung ist möglich. Stellen Sie sich beispielsweise drei verschiedene Anbieterkonten mit einer Region pro Konto vor. Jede Region enthält ein Speicherprofil mit dem Funktions-Tag *fast*. Zur Bereitstellungszeit sucht eine Anforderung mit einem Einschränkungs-Tag vom Typ *fast* nach einer übereinstimmenden Funktion vom Typ *fast*. Dabei spielt es keine Rolle, welche Anbieter-Cloud die Ressourcen bereitstellt. Im Fall einer Übereinstimmung werden die Einstellungen aus dem verknüpften Speicherprofil während der Erstellung des bereitgestellten Speicherelements angewendet.

Hinweis Verschiedene Cloud-Speicher können unterschiedliche Leistungsmerkmale aufweisen, werden aber dennoch als das *fast*-Angebot des Administrators betrachtet, der sie markiert hat.

Funktions-Tags, die Sie zu Speicherprofilen hinzufügen, sollten keine tatsächlichen Ressourcenziele angeben. Stattdessen sollten sie Speichertypen beschreiben. Weitere Informationen zu tatsächlichen Ressourcen finden Sie unter [Speicherressourcen](#).

Vorgehensweise zum Verwenden von Tags zur Verwaltung von vRealize Automation Cloud Assembly-Ressourcen und -Bereitstellungen

Tags sind eine kritische Komponente von vRealize Automation Cloud Assembly, die die Platzierung von Bereitstellungen durch den Abgleich von Funktionen und Einschränkungen steuern. Sie müssen Tags verstehen und effektiv implementieren, um vRealize Automation Cloud Assembly optimal zu nutzen.

Grundsätzlich sind Tags Beschriftungen, die Sie zu vRealize Automation Cloud Assembly-Elementen hinzufügen. Sie können alle Tags erstellen, die für Ihre Organisation und Implementierung geeignet sind. Im Vergleich zu Beschriftungen haben Tags jedoch einen weitaus größeren Funktionsumfang, da Sie steuern, wie und wo vRealize Automation Cloud Assembly Ressourcen und Infrastruktur zum Erstellen von bereitzustellenden Diensten verwendet. Tags unterstützen auch die Steuerung innerhalb von Cloud Assembly.

Tag-Struktur

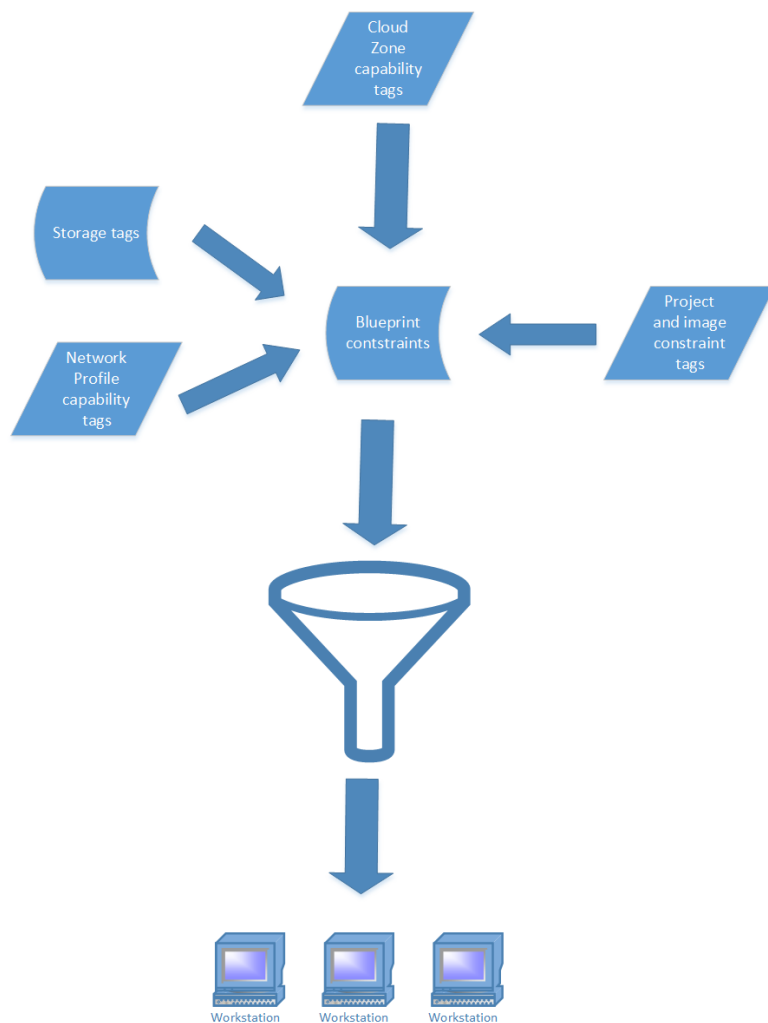
Strukturell müssen Tags der `name:value`-Paar-Konvention folgen, aber ansonsten ist ihre Konstruktion weitgehend frei. In der gesamten vRealize Automation Cloud Assembly erscheinen alle Tags gleich und die Tag-Funktionalität wird durch den Kontext bestimmt

Beispielsweise funktionieren Tags für Infrastrukturressourcen primär als Funktions-Tags, da sie von vRealize Automation Cloud Assembly verwendet werden, um Ressourcen mit Bereitstellungen abzugleichen. In zweiter Linie bezeichnen sie auch die Ressourcen.

Tag-Funktion

Die primäre Funktion von Tags innerhalb von vRealize Automation Cloud Assembly besteht darin, Bereitstellungen mithilfe von Funktionen und Einschränkungen zu konfigurieren. Funktions-Tags, die in Cloud-Zonen, Netzwerk- und Speicherprofilen und einzelnen Infrastrukturressourcen platziert werden, definieren die gewünschten Funktionen für Bereitstellungen. Einschränkungs-Tags, die von Cloud-Administratoren in Projekten abgelegt werden, ermöglichen es Ihnen, eine Form der Kontrolle für diese Projekte auszuüben. Diese Einschränkungs-Tags werden anderen in Blueprints ausgedrückten Einschränkungen hinzugefügt.

Während der Bereitstellung vergleicht vRealize Automation Cloud Assembly diese Funktionen mit Einschränkungen, die auch als Tags ausgedrückt werden, in Blueprints, um die Bereitstellungskonfiguration zu definieren. Diese Tag-basierten Funktionen und Einschränkungen dienen als Grundlage für die Bereitstellungskonfiguration in vRealize Automation Cloud Assembly. Sie können beispielsweise Tags verwenden, um die Infrastruktur nur auf PCI-Ressourcen in einer bestimmten Region verfügbar zu machen.



Auf einer sekundären Ebene erleichtern Tags auch die Suche und Identifizierung von Speicher- und Netzwerkelementen und anderen Infrastrukturressourcen.

Angenommen Sie richten Cloud-Zonen ein und es stehen Ihnen viele Computing-Ressourcen zur Verfügung. Wenn Sie Ihre Computing-Ressourcen entsprechend gekennzeichnet haben, können Sie die Suchfunktion auf der Registerkarte „Computing“ auf der Seite „Cloud-Zone“ verwenden, um die Ressourcen zu filtern, die mit dieser bestimmten Cloud-Zone verknüpft sind.

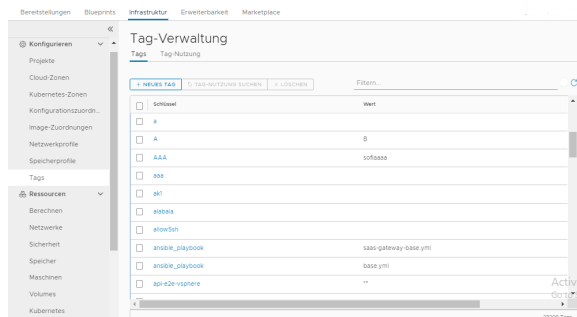
Außerdem enthalten die Seite „Tags verwalten“ und die Seiten für die Ressourcenkonfiguration Suchfunktionen, mit denen Sie Elemente nach Tag-Namen suchen können. Die Verwendung von logischen und lesbaren Tags für diese Elemente ist der Schlüssel zur Erleichterung dieser Such- und Identifizierungsfunktion.

Externe Tags

vRealize Automation Cloud Assembly kann auch externe Tags enthalten. Diese Tags werden automatisch aus Cloud-Konten importiert, die Sie mit einer vRealize Automation Cloud Assembly-Instanz verknüpfen. Diese Tags können aus vSphere, AWS, Azure oder anderen externen Softwareprodukten importiert werden. Wenn diese Tags importiert werden, stehen sie auf dieselbe Weise zur Verfügung wie die von Benutzern erstellten Tags.

Verwalten von Tags

Sie können die Seite „Tags verwalten“ in vRealize Automation Cloud Assembly verwenden, um Ihre Tags-Bibliothek zu überwachen und zu verwalten. Sie können auf dieser Seite auch Tags erstellen. Darüber hinaus ist die Seite „Tags verwalten“ die einzige Seite, auf der Sie externe Tags anzeigen und identifizieren können.



Tag-Strategie

Um die Verwirrung zu minimieren, sollten Sie vor dem Erstellen von Tags in vRealize Automation Cloud Assembly eine geeignete Tag-Strategie und Konventionen für die Kennzeichnung entwickeln, damit alle Benutzer, die Tags erstellen und verwenden, verstehen, was sie bedeuten und wie sie verwendet werden sollen. Weitere Informationen hierzu finden Sie unter [Erstellen einer Tagging-Strategie](#).

Erstellen einer Tagging-Strategie

Eine geeignete Tagging-Strategie muss unter Berücksichtigung der IT-Struktur und der Ziele Ihres Unternehmens sorgfältig geplant und implementiert werden, um die Cloud Assembly-Funktionen zu optimieren und Verwechslungen möglichst zu vermeiden.

Während das Tagging verschiedenen allgemeinen Zwecken dient, muss Ihre Tagging-Strategie auf die Bedürfnisse, die Struktur und die Ziele Ihres Unternehmens zugeschnitten werden.

Empfehlungen für Tagging

Einige allgemeine Merkmale für eine effektive Tagging-Strategie:

- Entwerfen und implementieren Sie einen einheitlichen Tagging-Plan, der die Struktur Ihres Unternehmens abbildet, und kommunizieren Sie diesen Plan an alle betreffenden Benutzer. Ein Plan muss Ihre Bereitstellungsanforderungen unterstützen, eine klar verständliche Sprache verwenden und für alle betreffenden Benutzer schlüssig sein.
- Verwenden Sie einfache, eindeutige und aussagekräftige Namen und Werte für Tags. Beispielsweise sollten die Tag-Namen für Speicher- und Netzwerkelemente klar und verständlich sein, damit Benutzer leicht nachvollziehen können, welche Tag-Zuweisungen für eine bereitgestellte Ressource auszuwählen oder zu überprüfen sind.
- Obwohl Sie Tags mit einem Namen ohne Wert erstellen können, empfiehlt sich die Erstellung eines entsprechenden Werts für jeden Tag-Namen, da dies die Verwendung von Tags für andere Benutzer erkennbar macht.

Tagging-Implementierung

Skizzieren Sie Ihre wichtigsten Überlegungen für eine grundlegende Tagging-Strategie. Die folgende Liste enthält typische Überlegungen, die beim Planen Ihrer Strategie berücksichtigt werden sollten. Beachten Sie, dass diese Überlegungen eher als Vorschlag und nicht als Weisung dienen. Unter Umständen müssen Sie bei Ihren Überlegungen andere Prioritäten für Ihre Anwendungsfälle setzen. Ihre Strategie muss für Ihre speziellen Anwendungsfälle geeignet sein.

- Anzahl der verschiedenen Umgebungen, für die eine Bereitstellung erfolgen soll. In der Regel erstellen Sie Tags für jede Umgebung.
- Struktur der Computing-Ressourcen und deren Verwendung zur Unterstützung von Bereitstellungen.
- Anzahl der verschiedenen Regionen oder Standorte, für die eine Bereitstellung erfolgen soll. In der Regel erstellen Sie Tags für diese verschiedenen Regionen und Standorte auf Profilebene.
- Anzahl der verschiedenen Speicheroptionen für Bereitstellungen und deren Eigenschaften. Diese Optionen sollten durch Tags dargestellt werden.
- Kategorisieren Sie Ihre Netzwerkooptionen und erstellen Sie Tags, um alle anwendbaren Optionen zu berücksichtigen.

- Typische Bereitstellungsvariablen. Beispielsweise die Anzahl der verschiedenen Umgebungen, für die eine Bereitstellung erfolgen soll. In der Regel verfügen die meisten Unternehmen über Test-, Entwicklungs- und Produktionsumgebungen. Sie möchten übereinstimmende Einschränkungstags für Blueprints und Funktionstags für Cloud-Zonen erstellen und aufeinander abstimmen, um Bereitstellungen problemlos in einer oder mehreren dieser Umgebungen einzurichten.
- Stimmen Sie Tags in Netzwerk- und Speicherressourcen so aufeinander ab, dass sie hinsichtlich der Netzwerk- und Speicherprofile, in denen sie verwendet werden, logisch und nachvollziehbar sind. Mithilfe der Ressourcentags kann die Ressourcenbereitstellung präziser gesteuert werden.
- Stimmen Sie Funktionstags für Cloud-Zonen und Netzwerkprofile sowie andere Funktionstags auf Einschränkungstags für Blueprints ab. Im Allgemeinen erstellt Ihr Administrator zuerst Funktionstags für Cloud-Zonen und Netzwerkprofile. Anschließend können andere Benutzer Blueprints mit Einschränkungen entwerfen, die diesen Funktionstags entsprechen.

Nachdem Sie die wichtigsten Überlegungen für Ihr Unternehmen zusammengefasst haben, können Sie geeignete Tag-Namen erstellen, die diese Überlegungen in logischer Weise widerspiegeln. Erstellen Sie dann eine Übersicht über Ihre Strategie und stellen Sie sie allen Benutzern mit Berechtigungen zum Erstellen oder Bearbeiten von Tags zur Verfügung.

Ein sinnvoller Implementierungsansatz besteht darin, alle Computing-Infrastrukturressourcen einzeln mit Tags zu versehen. Verwenden Sie logische Kategorien für Tag-Namen, die sich auf die jeweilige Ressource beziehen. Sie können Speicherressourcen beispielsweise als Tier1, Tier2 usw. kennzeichnen. Sie können Computing-Ressourcen auch basierend auf dem Betriebssystem kennzeichnen, wie z. B. Windows, Linux usw.

Nach dem Taggen der Ressourcen können Sie sich eine Methode zum Erstellen von Tags für Cloud-Zonen sowie für Speicher- und Netzwerkprofile überlegen, die Ihren Anforderungen entspricht.

Verwenden von Funktionstags in vRealize Automation Cloud Assembly

In vRealize Automation Cloud Assembly können Sie mit Funktionstags die Platzierungslogik für die Bereitstellung von Infrastrukturkomponenten definieren. Sie stellen eine leistungstärkere und prägnantere Möglichkeit zur harten Codierung solcher Platzierungen dar.

Sie können Funktionstags für Computing-Ressourcen, Cloud-Zonen, Images und Image-Zuordnungen sowie für Netzwerke und Netzwerkprofile erstellen. Die Seiten zum Erstellen dieser Ressourcen enthalten Optionen zum Erstellen von Funktionstags. Alternativ können Sie die Seite „Tags verwalten“ in vRealize Automation Cloud Assembly verwenden, um Funktionstags zu erstellen. Funktionstags in Cloud-Zonen und Netzwerkprofilen wirken sich auf alle Ressourcen in diesen Zonen oder Profilen aus. Funktionstags auf Speicher- oder Netzwerkkomponenten wirken sich nur auf die Komponenten aus, auf die sie angewendet werden.

In der Regel können Funktions-Tags Elemente wie den Speicherort für eine Computing-Ressource, den Adaptertyp für ein Netzwerk oder die Ebene für eine Speicherressource definieren. Sie können auch den Umgebungsstandort oder -typ und andere geschäftliche Aspekte definieren. Sie sollten Ihre Funktions-Tags genauso wie Ihre gesamte Tagging-Strategie logisch organisieren.

vRealize Automation Cloud Assembly gleicht Funktions-Tags mit Einschränkungen aus Cloud-Zonen und auf Blueprints zur Bereitstellungszeit ab. Daher müssen Sie beim Erstellen und Verwenden von Funktions-Tags verstehen und planen, wie sie entsprechende Blueprint-Einschränkungen erstellen, damit der Abgleich erwartungsgemäß stattfindet.

Im Thema „Hinzufügen von Cloud-Zonen“ im WordPress-Beispiel haben Sie beispielsweise dev- und test-Tags für die Zonen „OurCo-AWS-US-East“ und „OurCo AWS-US-West“ erstellt. Dies lässt erkennen, dass es sich bei der Zone „OurCo-AWS-US-East“ um eine Entwicklungsumgebung handelt und die Zone „OurCo AWS-US-West“ eine Testumgebung ist. In Verbindung mit den entsprechenden Einschränkungs-Tags ermöglichen diese Funktions-Tags die direkte Bereitstellung in der gewünschten Umgebung.

Verwenden von Einschränkungs-Tags in vRealize Automation Cloud Assembly

Sie fügen Blueprints und verschiedenen anderen Komponenten innerhalb von vRealize Automation Cloud Assembly Einschränkungs-Tags hinzu, um die für Ressourcen, Cloud-Zonen und Profile definierten Funktionen zwecks Generierung von entsprechenden Bereitstellungen abzugleichen.

Es gibt zwei Hauptbereiche in vRealize Automation Cloud Assembly, in denen Einschränkungs-Tags angewendet werden können. Der erste befindet sich auf der Konfigurationsseite in Projekten und Images. Der zweite befindet sich auf der Bereitstellungsseite in Blueprints. In beiden Bereichen angewendete Einschränkungen werden in Blueprints zusammengeführt, sodass sie eine Reihe von Bereitstellungsanforderungen bilden.

Funktionsweise von Einschränkungs-Tags in Projekten

Bei der Konfiguration von Cloud Assembly können Cloud-Administratoren Einschränkungs-Tags für Projekte und Image-Zuordnungen anwenden. Auf diese Weise können Cloud-Administratoren Kontrolleinschränkungen direkt auf Projektebene anwenden. Alle auf dieser Ebene hinzugefügten Einschränkungen werden auf jeden für das entsprechende Projekt angeforderten Blueprint angewendet.

Wenn Tags im Projekt mit Tags im Blueprint in Konflikt stehen, haben die Projekt-Tags Vorrang, sodass der Cloud-Administrator Verwaltungsregeln erzwingen kann. Wenn beispielsweise die Cloud-Administratoren ein `location:london`-Tag für das Projekt erstellen, aber ein Entwickler ein `location:boston`-Tag auf dem Blueprint platziert, hat das erste Tag Vorrang und die Ressource wird in der Infrastruktur bereitgestellt, die das `location:london`-Tag enthält.

Sie können bis zu drei Einschränkungen für Projekte anwenden. Projekteinschränkungen können hart oder weich sein. Standardmäßig sind sie hart. Harte Einschränkungen ermöglichen es Ihnen, Bereitstellungseinschränkungen strikt durchzusetzen. Wenn eine oder mehrere harte Einschränkungen nicht erfüllt werden, schlägt die Bereitstellung fehl. Weiche Einschränkungen bieten eine Möglichkeit zum Ausdrücken von Einstellungen, die bei Verfügbarkeit ausgewählt werden. Die Bereitstellung schlägt jedoch nicht fehl, wenn weiche Einschränkungen nicht erfüllt werden.

Funktionsweise von Einschränkungs-Tags in Blueprints

In Blueprints fügen Sie Einschränkungs-Tags zu Ressourcen als YAML-Code hinzu, um die entsprechenden Funktions-Tags zu erfüllen, die Ihr Cloud-Administrator für Ressourcen, Cloud-Zonen und für Speicher- und Netzwerkprofile erstellt hat. Darüber hinaus gibt es weitere komplexere Optionen für die Implementierung von Einschränkungs-Tags. Sie können beispielsweise eine Variable verwenden, um ein oder mehrere Tags für eine Anforderung aufzufüllen. Auf diese Weise können Sie einen oder mehrere Tags zur Anforderungszeit angeben.

Erstellen Sie Einschränkungs-Tags mithilfe der `tag`-Bezeichnung im Blueprint-YAML-Code. Einschränkungs-Tags aus Projekten werden den in Blueprints erstellten Einschränkungs-Tags hinzugefügt.

vRealize Automation Cloud Assembly unterstützt eine einfache Zeichenfolgenformatierung, um die Verwendung von Einschränkungen in YAML-Dateien zu vereinfachen:

```
[!]tag_key[:tag_value][:hard|:soft]
```

Standardmäßig erstellt vRealize Automation Cloud Assembly eine positive Einschränkung mit harter Erzwingung. Der Tag-Wert ist optional, wird aber wie im Rest der Anwendung empfohlen.

Das folgende WordPress-mit-MySQL-Beispiel zeigt YAML-Einschränkungs-Tags, die bestimmte Standortinformationen für Computing-Ressourcen aufweisen.

```
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 ...
```

Weitere Informationen zum Arbeiten mit Blueprints finden Sie unter [WordPress-Anwendungsbeispiel: Erstellen und Erweitern eines Blueprints](#).

Funktionsweise von harten und weichen Einschränkungen in Projekten und Blueprints

Einschränkungen in Projekten und Blueprints können hart oder weich sein. Der vorangehende Codeausschnitt zeigt Beispiele für harte und weiche Einschränkungen. Standardmäßig sind alle Einschränkungen hart. Harte Einschränkungen ermöglichen es Ihnen, Bereitstellungseinschränkungen strikt durchzusetzen. Wenn eine oder mehrere harte Einschränkungen nicht erfüllt werden, schlägt die Bereitstellung fehl. Weiche Einschränkungen drücken Einstellungen aus, die gelten, wenn sie verfügbar sind. Sie schlagen jedoch nicht fehl, wenn sie nicht erfüllt werden.

Wenn Sie über eine Reihe von harten und weichen Einschränkungen für einen bestimmten Ressourcentyp verfügen, können die weichen Einschränkungen auch als Trennung dienen. Das heißt, wenn mehrere Ressourcen eine harte Einschränkung erfüllen, werden die weichen Einschränkungen verwendet, um die tatsächlich in der Bereitstellung verwendete Ressource auszuwählen.

Sie können beispielsweise bis zu drei Einschränkungen für ein Projekt in einer beliebigen Kombination aus Netzwerk-, Speicher- und Erweiterbarkeitselementen angeben. Darüber hinaus können Sie für jede Einschränkung auswählen, ob sie hart oder weich ist. Angenommen, Sie erstellen eine Einschränkung des Festplattenspeichers mit einem `location:boston`-Tag. Wenn kein Speicher im Projekt mit dieser Einschränkung übereinstimmt, schlagen alle zugehörigen Bereitstellungen fehl.

Hinweis In Projekten und Blueprints ändert das `failOnConstraintMergeConflict`-Flag das Verhalten von Einschränkungen. Wenn dieses Flag auf „true“ festgelegt ist, schlägt die Anforderung fehl, wenn ein Konflikt zwischen Projekteinschränkungen und Blueprint-Einschränkungen besteht. Wenn das Flag nicht vorhanden ist oder auf „false“ festgelegt ist, haben die Projekteinschränkungen Vorrang vor den Blueprint-Einschränkungen.

Standard-Tags

vRealize Automation Cloud Assembly wendet Standard-Tags auf bestimmte Bereitstellungen an, um die Analyse, Überwachung und Gruppierung von bereitgestellten Ressourcen zu unterstützen.

Standard-Tags sind in vRealize Automation Cloud Assembly eindeutig. Im Gegensatz zu anderen Tags werden diese während der Bereitstellungskonfiguration nicht von Benutzern verwendet und es gelten keine Einschränkungen. Diese Tags werden automatisch während der Bereitstellung auf AWS-, Azure- und vSphere-Bereitstellungen angewendet. Diese Tags werden als benutzerdefinierte Systemeigenschaften gespeichert und den Bereitstellungen nach dem Bereitstellen hinzugefügt.

Die Liste der Standard-Tags wird im Folgenden angezeigt.

Tabelle 4-2. Standard-Tags

Beschreibung	Tag
Organisation	org:orgID
Projekt	project:projectID
Anforderer	requester:username
Bereitstellung	deployment:deploymentID
Blueprint-Referenz (falls zutreffend)	blueprint:blueprintID
Komponentenname in Blueprint	blueprintResourceName:CloudMachine_1
Platzierungseinschränkungen: in Blueprint, Anforderungsparametern oder über IT-Richtlinie angewendet	constraints:key:value:soft
Cloud-Konto	cloudAccount:accountID
Zone oder Profil (falls zutreffend)	zone:zoneID, networkProfile:profileID, storageProfile:profileID

Wie vRealize Automation Cloud Assembly Tags verarbeitet

In vRealize Automation Cloud Assembly drücken Tags Funktionen und Beschränkungen aus, die darüber bestimmen, wie und wo Ressourcen Bereitstellungen während des Bereitstellungsprozesses zugeteilt werden.

vRealize Automation Cloud Assembly verwendet eine bestimmte Reihenfolge und Hierarchie in der Auflösung von Tags zum Erstellen von Bereitstellungen. Wenn Sie die Grundlagen dieses Prozesses verstehen, können Sie Tags effizient einsetzen, um berechenbare Bereitstellungen zu erstellen.

Die folgende Liste fasst die allgemeinen Vorgänge und Abfolge der Verarbeitung von Funktions- und Beschränkungs-Tags zusammen.

- Cloud-Zonen werden nach diversen Kriterien wie Verfügbarkeit und Profilen gefiltert. An dieser Stelle werden Tags in Profilen für die Region, zu der die Zone gehört, abgeglichen.
- Zonen- und Computing-Funktions-Tags dienen zum Filtern der übrigen Cloud-Zonen nach harten Einschränkungen.
- Aus den gefilterten Zonen wird eine Cloud-Zone anhand der Priorität ausgewählt. Wenn mehrere Cloud-Zonen mit der gleichen Priorität vorhanden sind, werden diese durch Abgleich von weichen Beschränkungen sortiert. Dabei wird eine Kombination aus Cloud-Zonen- und Computing-Funktionen angewendet.
- Nach der Auswahl einer Cloud-Zone wird ein Host ausgewählt. Dies geschieht durch Abgleich einer Reihe von Filtern, darunter auch harte und weiche Einschränkungen, wie in Blueprints ausgedrückt.

Vorgehensweise zum Einrichten einer einfachen Tagging-Struktur

Unter diesem Thema werden grundlegende Vorgehensweisen und Optionen für die logische Tagging-Strategie in vRealize Automation Cloud Assembly beschrieben. Sie können diese Beispiele als Ausgangsbasis für eine reale Bereitstellung verwenden, oder Sie können eine andere Strategie verfolgen, die besser auf Ihre Anforderungen zugeschnitten ist.

Der Cloud-Administrator ist in der Regel der Hauptverantwortliche für das Erstellen und Pflegen von Tags.

Dieses Thema bezieht sich auf das an anderer Stelle in der vRealize Automation Cloud Assembly-Dokumentation beschriebene WordPress-Anwendungsbeispiel, bei dem veranschaulicht wird, wie Tags zu einigen wesentlichen Elementen hinzugefügt werden können. Außerdem werden mögliche Alternativen und Erweiterungen für die Tagging-Beispiele aus dem WordPress-Anwendungsbeispiel beschrieben.

Weitere Informationen zum WordPress-Anwendungsbeispiel finden Sie unter [Anwendungsbeispiel: WordPress](#).

Im WordPress-Anwendungsbeispiel wird beschrieben, wie Tags in Cloud-Zonen und Speicher- und Netzwerkprofilen platziert werden. Diese Profile sind ähnlich wie organisierte Ressourcenpakete. In Profilen platzierte Tags gelten für alle Elemente im Profil. Sie können Tags auch bei Speicherressourcen und einzelnen Netzwerkelementen sowie bei Computing-Ressourcen platzieren. Diese Tags gelten jedoch nur für die spezifischen Ressourcen, bei denen sie platziert werden. Beim Einrichten von Tags ist es in der Regel das Beste, mit dem Taggen der Computing-Ressourcen zu beginnen. Anschließend können Sie dann Tags zu Profilen und Cloud-Zonen hinzufügen. Mithilfe dieser Tags können Sie auch die Liste der Computing-Ressourcen für eine Cloud-Zone filtern.

Sie können z. B. Tags in Speicherprofilen wie in diesem Anwendungsbeispiel dargestellt platzieren. Sie haben aber auch die Möglichkeit, Tags in einzelnen Speicherrichtlinien, Datenspeichern und Speicherkonten zu platzieren. Mit Tags in diesen Ressourcen können Sie genauer steuern, wie die Speicherressourcen bereitgestellt werden. Während der Verarbeitung zur Vorbereitung für die Bereitstellung werden diese Tags als nächste Verarbeitungsebene nach den Profil-Tags aufgelöst.

Als Beispiel dafür, wie Sie ein typisches Kundenszenario konfigurieren könnten, könnten Sie ein Tag von `region: eastern` auf einem Netzwerkprofil platzieren. Dieses Tag würde dann für alle Ressourcen in diesem Profil gelten. Anschließend können Sie ein Tag von `networktype:pci` auf einer PCI-Netzwerkressource innerhalb dieses Profils platzieren. Ein Blueprint mit den Einschränkungen „eastern“ und „pci“ würde Bereitstellungen erstellen, die dieses PCI-Netzwerk für die Region „East“ verwenden.

Verfahren

- 1 Taggen Sie Ihre Computing-Infrastrukturressourcen in logischer und angemessener Weise.

Es ist besonders wichtig, dass Sie Computing-Ressourcen in logischer Weise taggen, sodass Sie sie mit der Suchfunktion auf der Registerkarte „Computing“ der Seite „Cloud-Zone

erstellen“ finden können. Mithilfe dieser Suchfunktion können Sie die Computing-Ressourcen, die mit einer Cloud-Zone verknüpft sind, schnell filtern. Wenn Sie Speicher und Netzwerke auf der Profilebene taggen, brauchen Sie die einzelnen Speicher- und Netzwerkressourcen unter Umständen nicht mehr zu taggen.

- a Wählen Sie **Ressourcen > Computing** aus, um die Computing-Ressourcen anzuzeigen, die für Ihre vRealize Automation Cloud Assembly-Instanz importiert wurden.
- b Wählen Sie die einzelnen Computing-Ressourcen nach Bedarf aus und klicken Sie auf **Tags**, um ein Tag zur Ressource hinzuzufügen. Sie können jeder Ressource bei Bedarf mehrere Tags hinzufügen.
- c Wiederholen Sie den vorherigen Schritt je nach Bedarf für Speicher- und Netzwerkressourcen.

2 Erstellen Sie Funktions-Tags für Cloud-Zonen und Netzwerkprofile.

Sie können dieselben Tags für Cloud-Zonen und Netzwerkprofile verwenden. Stattdessen können Sie aber auch eindeutige Tags für jedes Objekt erstellen, wenn Ihnen das für Ihre Implementierung sinnvoller erscheint.

In Netzwerkprofilen können Sie Tags auf dem gesamten Profil platzieren. Dasselbe gilt auch für Subnetze innerhalb des Profils. Auf der Profilebene angewendete Tags gelten für alle Komponenten innerhalb des Profils, z. B. für Subnetze. Tags in Subnetzen gelten nur für das spezifische Subnetz, in dem sie platziert wurden. Bei der Tag-Verarbeitung haben die Tags der Profilebene Vorrang gegenüber den Tags der Subnetzebene.

Unter [WordPress-Anwendungsbeispiel: Hinzufügen von Cloud-Zonen](#) und [WordPress-Anwendungsbeispiel: Hinzufügen von Netzwerkprofilen](#) finden Sie weitere Informationen über das Hinzufügen von Tags zu Cloud-Zonen oder Netzwerkprofilen.

In diesem Beispiel erstellen wir drei einfache Tags, die in der gesamten Dokumentation des Anwendungsbeispiels für vRealize Automation Cloud Assembly-Cloud-Zone- und Netzwerkprofil-Tags erscheinen. Diese Tags identifizieren die Umgebung für die Profilkomponenten.

- `zone:test`
- `zone:dev`
- `zone:prod`

3 Erstellen Sie Speicherprofil-Tags für Ihre Speicherkomponenten.

In der Regel bezeichnen Speicher-Tags die Leistungsstufe von Speicherelementen, z. B. tier1 oder tier2, oder sie bezeichnen die Art der Speicherelemente, z. B. pci.

Weitere Informationen zum Hinzufügen von Tags zu Speicherprofilen finden Sie unter [WordPress-Anwendungsbeispiel: Hinzufügen von Speicherprofilen](#).

- `usage:general`
- `usage:fast`

Ergebnisse

Nachdem Sie eine grundlegende Tagging-Struktur erstellt haben, können Sie damit beginnen, mit ihr zu arbeiten, und Tags entsprechend hinzufügen oder bearbeiten, um Ihre Tagging-Funktionen zu verfeinern und zu erweitern.

Vorgehensweise zum Arbeiten mit Ressourcen in vRealize Automation Cloud Assembly

Ein Cloud-Administrator kann vRealize Automation Cloud Assembly-Ressourcen überprüfen, die über die Datenerfassung offengelegt werden. Der Cloud-Administrator kann Ressourcen mit Funktions-Tags kennzeichnen, um den Bereitstellungsort von vRealize Automation Cloud Assembly-Blueprints festzulegen.

Computing-Ressourcen

Ein Cloud-Administrator kann Computing-Ressourcen überprüfen, die über die Datenerfassung angezeigt werden. Der Cloud-Administrator kann Tags direkt auf die Ressourcen anwenden, um Funktionen zu Übereinstimmungszwecken in der vRealize Automation Cloud Assembly-Bereitstellung zu kennzeichnen.

Netzwerkressourcen

In vRealize Automation Cloud Assembly können Cloud-Administratoren die Netzwerkressourcen anzeigen und bearbeiten, für die Daten aus den Cloud-Konten und -Integrationen des Projekts erfasst wurden.

Nachdem Sie ein Cloud-Konto hinzugefügt haben, erkennt die Datenerfassung die Netzwerk- und Sicherheitsinformationen des Cloud-Kontos und stellt diese Informationen zur Verwendung in Netzwerkprofilen und für andere Optionen bereit.

Netzwerke sind die IP-spezifischen Komponenten einer verfügbaren Netzwerkdomäne oder Transportzone. Stellen Sie sich als Amazon Web Services- oder Microsoft Azure-Benutzer Netzwerke wie Subnetze vor.

Die vRealize Automation Cloud Assembly-Seite **Netzwerke** enthält Informationen, wie z. B.:

- Netzwerke und Lastausgleichsdienste, die extern in der Netzwerkdomäne Ihres Cloud-Kontos definiert sind, zum Beispiel in vCenter, NSX-V oder Amazon Web Services.
- Netzwerke und Lastausgleichsdienste, die vom Cloud-Administrator bereitgestellt wurden.
- IP-Bereiche und andere Netzwerkmerkmale, die von Ihrem Cloud-Administrator definiert oder geändert wurden.
- IP-Bereiche des externen IPAM-Anbieters für einen bestimmten Adressraum in einer anbieterspezifischen externen IPAM-Integration.

Weitere Informationen zu Netzwerken finden Sie in der Wegweiser-Hilfe für verschiedene Einstellungen auf der Seite **Netzwerke** und unter [Weitere Informationen zu Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

Netzwerke

Sie können Netzwerke und deren Eigenschaften anzeigen und bearbeiten, zum Beispiel zum Hinzufügen von Tags oder zum Entfernen der Unterstützung für den öffentlichen IP-Zugriff. Sie können auch neue und vorhandene IP-Bereiche in Ihren verfügbaren Netzwerken definieren.

Für vorhandene Netzwerke können Sie den IP-Bereich und die Tag-Einstellungen ändern, indem Sie das Kontrollkästchen des Netzwerks aktivieren und entweder **IP-Bereiche verwalten** oder **Tags** auswählen. Andernfalls können Sie das Netzwerk selbst auswählen, um die jeweiligen Informationen zu bearbeiten.

Tags bieten eine Möglichkeit zur Anpassung geeigneter Netzwerke und optionaler Netzwerkprofile an Netzwerkkomponenten in Blueprints. Netzwerk-Tags werden unabhängig von den Netzwerkprofilen, in denen sich das Netzwerk befinden kann, auf jede Instanz dieses Netzwerks angewendet. Netzwerke können in beliebig viele Netzwerkprofile eingeteilt werden. Unabhängig davon, wo sich das Netzwerkprofil befindet, ist ein Netzwerk-Tag mit diesem Netzwerk verknüpft, wenn das Netzwerk verwendet wird. Der Netzwerk-Tag-Abgleich tritt mit anderen Komponenten im Blueprint auf, nachdem der Blueprint mit einem oder mehreren Netzwerkprofilen abgeglichen wurde.

IP-Bereiche

Wählen Sie einen IP-Bereich aus, um die Start- und End-IP-Adresse für ein bestimmtes Netzwerk in Ihrer Organisation zu definieren oder Änderungen daran vorzunehmen.

Das Standard-Gateway kann nicht in einen IP-Bereich aufgenommen werden. Der Subnetz-IP-Bereich kann nicht den Wert des Subnetz-Gateways enthalten.

Wenn Sie eine externe IPAM-Integration für einen bestimmten IPAM-Anbieter verwenden, können Sie einen IPAM-IP-Bereich hinzufügen. Dieser Vorgang wird im Kontext eines allgemeinen Workflows für die externe IPAM-Integration unter [Konfigurieren eines Netzwerks und eines Netzwerkprofils in vRealize Automation Cloud Assembly zur Verwendung von IPAM-Anbieterwerten](#) beschrieben.

IP-Adressen

Zeigen Sie den Status (z. B. verfügbar oder zugeteilt) der definierten IP-Adressen an, die von Ihrer Organisation verwendet werden.

Lastausgleichsdienste

Zeigen Sie Informationen zu den verfügbaren Lastausgleichsdiensten für die Cloud-Konten des Kontos/der Region in Ihrer Organisation an. Sie können die konfigurierten Einstellungen für jeden verfügbaren Lastausgleichsdienst öffnen und anzeigen. Sie können auch Tags für einen Lastausgleichsdienst hinzufügen und entfernen.

Sicherheitsressourcen

Nachdem Sie ein Cloud-Konto hinzugefügt haben, erkennt die Datenerfassung die Netzwerk- und Sicherheitsinformationen des Cloud-Kontos und stellt diese Informationen zur Verwendung in Netzwerkprofilen und für andere Optionen bereit.

Sicherheitsgruppen und Firewallregeln unterstützen die Netzwerkisolierung.

Sicherheitsgruppen

Sie können in vRealize Automation Cloud Assembly erstellte bedarfsgesteuerte Sicherheitsgruppen sowie vorhandene Sicherheitsgruppen anzeigen, die in Quellenanwendungen (wie z. B. NSX-T und Amazon Web Services) erstellt wurden, die über die Datenerfassung bereitgestellt werden.

Sie können die verfügbaren Sicherheitsgruppen anzeigen und Tags für ausgewählte Sicherheitsgruppen hinzufügen oder entfernen. Sie können auch bedarfsgesteuerte Sicherheitsgruppen bearbeiten, die Sie erstellt haben. Cloud-Administratoren können Sicherheitsgruppen öffnen und bearbeiten. Ein Blueprint-Autor kann einer Maschinen-Netzwerkkarte (NIC) eine oder mehrere Sicherheitsgruppen zuweisen, um Netzwerkregeln und andere Sicherheitsaspekte für die Blueprint-Bereitstellung zu steuern.

Wenn Firewallregeln im Blueprint-Code einer Sicherheitsgruppenkomponente verwendet werden,

Bei vorhandenen Sicherheitsgruppen aus dem zugrundeliegenden Cloud-Konto-Endpoint, z. B. in den Anwendungen NSX-V, NSX-T oder Amazon Web Services, werden die Daten von vRealize Automation Cloud Assembly erfasst. Vorhandene Sicherheitsgruppen werden in der Spalte **Ursprung** als *Discovered* angezeigt und klassifiziert. Bedarfsgesteuerte Sicherheitsgruppen, die Sie in vRealize Automation Cloud Assembly entweder in einem Blueprint oder in einem Netzwerkprofil erstellen, werden in der Spalte **Ursprung** als *Managed by Cloud Assembly* angezeigt und klassifiziert. Nur Sicherheitsgruppen, die für die Verwendung in Ihrer Organisation zur Verfügung stehen, werden angezeigt.

Wenn Sie eine vorhandene Sicherheitsgruppe direkt in der Quellenanwendung (beispielsweise in der NSX-Quellenanwendung) bearbeiten anstatt in vRealize Automation Cloud Assembly, werden die Updates in vRealize Automation Cloud Assembly erst dann angezeigt, wenn Sie die Datenerfassung im zugehörigen Cloudkonto oder Integrationspunkt innerhalb von vRealize Automation Cloud Assembly ausführen.

Ein Cloud-Administrator kann einer vorhandenen Sicherheitsgruppe ein oder mehrere Tags zuweisen, damit diese in einem Blueprint verwendet werden kann. Eine Sicherheitsgruppe muss über mindestens ein Tag verfügen, ansonsten kann sie nicht in einem Blueprint verwendet werden. Ein Blueprint-Autor kann eine `Cloud.SecurityGroup`-Komponente in einem Blueprint verwenden, um eine vorhandene Sicherheitsgruppe mithilfe von Tag-Einschränkungen zuzuteilen. Im Blueprint muss eine `Cloud.SecurityGroup`-Sicherheitsgruppenkomponente auf eine Maschinen-Netzwerkkarte (NIC) angewendet werden, um Netzwerkregeln und andere Sicherheitsaspekte in der Blueprint-Bereitstellung zu steuern.

Informationen dazu, wie ein Administrator Tags zu einer vorhandenen Sicherheitsgruppe hinzufügen kann oder wie die Daten von Tags von der Sicherheitsgruppe in der Quellanwendung erfasst werden, finden Sie unter [Weitere Informationen zu Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

Beispiele dazu, wie ein Blueprint-Autor Tags in einer Blueprint-Sicherheitskomponente verwenden kann, finden Sie unter [Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in vRealize Automation Cloud Assembly-Blueprints](#).

Ein Blueprint-Codebeispiel finden Sie im Beispiel *Vorhandene Sicherheitsgruppe mit einem Einschränkungs-Tag, das auf eine Maschinen-Netzwerkkarte (NIC) angewendet wurde* unter [Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in vRealize Automation Cloud Assembly-Blueprints](#).

Weitere Informationen zur Verwendung von Sicherheitsgruppen in Netzwerkprofilen finden Sie unter [Weitere Informationen zu Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

Möglichkeiten zum Definieren von Sicherheitsgruppen in Netzwerkprofilen und Blueprints

Sie können die Funktionen der Sicherheitsgruppe auf eine der folgenden Arten nutzen:

- Vorhandene Sicherheitsgruppe, die in einem Netzwerkprofil angegeben ist

Sie können eine vorhandene Sicherheitsgruppe zu einem Netzwerkprofil hinzufügen. Wenn ein Blueprint dieses Netzwerkprofil verwendet, werden die zugehörigen Maschinen als Mitglieder der Sicherheitsgruppe zusammengefasst. Bei dieser Methode muss einem Blueprint keine Sicherheitsgruppenkomponente hinzugefügt werden.

- Sicherheitsgruppenkomponente, die der Maschinenkomponente in einem Blueprint zugeordnet ist

Sie können eine Sicherheitsgruppenkomponente per Drag & Drop auf einen Blueprint verschieben und die Sicherheitsgruppenkomponente an eine Maschinen-Netzwerkkarte binden, indem Sie Einschränkungs-Tags in der vorhandenen Sicherheitsgruppenkomponente im Blueprint und in der vorhandenen Sicherheitsgruppe in der Ressource verwenden, für die Daten erfasst wurden.

- NSX-T-Tags, die in Ihrer NSX-T-Anwendung angegeben werden
- Sie können ein NSX-T-Tag verwenden, das als Einschränkung für eine Netzwerkkomponente in einem Blueprint angegeben ist, wobei die Netzwerkkomponente mit einer Maschinen-Netzwerkkarte im Blueprint verbunden ist. Mit NSX-T-Tags können Sie Maschinen unter Verwendung eines vordefinierten NSX-T-Tags dynamisch gruppieren, dessen Daten aus dem NSX-T-Quell-Endpoint erfasst wurden. Verwenden Sie einen logischen Port, wenn Sie das NSX-T-Tag in NSX-T erstellen.

Speicherressourcen

Ein Cloud-Administrator kann mit Speicherressourcen und deren Funktionen arbeiten, die über die vRealize Automation Cloud Assembly-Datenerfassung aus verknüpften Cloud-Konten ermittelt werden.

Speicherressourcen-Funktionen werden über Tags offengelegt, die in der Regel vom Quell-Cloud-Konto stammen. Ein Cloud-Administrator kann jedoch mithilfe von vRealize Automation Cloud Assembly zusätzliche Tags direkt auf Speicherressourcen anwenden. Die zusätzlichen Tags können eine bestimmte Funktion für den Abgleich zum Zeitpunkt der Bereitstellung kennzeichnen.

Die Funktionen für Speicherressourcen werden im Rahmen der Definition eines vRealize Automation Cloud Assembly-Speicherprofils angezeigt. Weitere Informationen hierzu finden Sie unter [Weitere Informationen zu vRealize Automation Cloud Assembly-Speicherprofilen](#).

Maschinenressourcen

In vRealize Automation können alle Benutzer Maschinenressourcen überprüfen, die über die Datenerfassung offen gelegt werden.

Alle Maschinen in Ihren Projekten werden in der Liste „Maschinen“ angezeigt.

Nicht verwaltete Maschinen, die Cloud-Konten in Ihren Projekten zugeordnet sind, werden in dieser Liste wie verwaltete Maschinen angezeigt. Die Spalte „Ursprung“ gibt den Maschinenstatus an.

- **Erkannt:** Maschinen, die noch nicht integriert sind.
- **Bereitgestellt:** Maschinen, die noch nicht integriert sind oder noch nicht von vRealize Automation bereitgestellt wurden.

Sie können einen Arbeitslast-Onboarding-Plan verwenden, um nicht verwaltete Maschinen in die vRealize Automation-Verwaltung zu integrieren.

Informationen zur Verwendung von Onboarding-Plänen zwecks Integration von nicht verwalteten Maschinen in die vRealize Automation-Verwaltung finden Sie unter [Definition von Onboarding-Plänen in vRealize Automation Cloud Assembly](#).

Volume-Ressourcen

In vRealize Automation Cloud Assembly können alle Benutzer die Volume-Ressourcen überprüfen.

vRealize Automation Cloud Assembly zeigt Volumes oder logische Laufwerke an, die aus zwei Quellen stammen:

- Durch die Datenerfassung von Quell-Cloud-Konten erkannte Volumes
- Volumes, die mit von vRealize Automation Cloud Assembly bereitgestellten Arbeitslasten verknüpft sind

Sie können die Kapazität und Funktionen nach Volume oder logischem Laufwerk überprüfen. In der Liste sind auch Funktions-Tags verfügbar, die aus dem Quell-Cloud-Konto stammen oder in vRealize Automation Cloud Assembly selbst hinzugefügt wurden.

Weitere Informationen zu Ressourcen in vRealize Automation Cloud Assembly

vRealize Automation Cloud Assembly kann zusätzliche Informationen zu Ressourcen mit erfassten Daten, wie z. B. Kosten, bereitstellen.

Funktionsweise der Datenerfassung in vRealize Automation Cloud Assembly

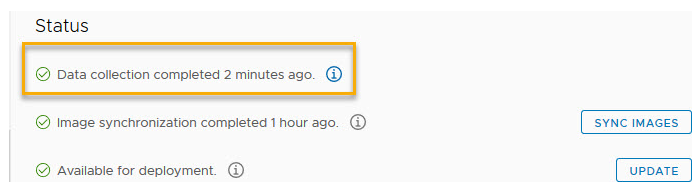
Nach der anfänglichen Datenerfassung erfolgt die Erfassung von Ressourcendaten automatisch alle 10 Minuten. Das Datenerfassungsintervall ist nicht konfigurierbar, und Sie können die Datenerfassung nicht manuell einleiten.

Sie können Informationen zur Erfassung von Ressourcendaten und zur Image-Synchronisierung für ein vorhandenes Cloud-Konto auf dessen Seite im Abschnitt „Status“ ermitteln. Wählen Sie hierzu **Infrastruktur > Verbindungen > Cloud-Konten** und klicken Sie dann auf **Öffnen** in einem beliebigen vorhandenen Cloud-Konto.

Sie können ein bestehendes Cloud-Konto öffnen und dessen verknüpfte Endpoint-Version im Abschnitt **Status** der zugehörigen Seite anzeigen. Wenn der zugeordnete Endpoint aktualisiert wurde, wird die neue Endpoint-Version während der Datenerfassung erkannt und im Abschnitt **Status** auf der Seite des Cloud-Kontos angezeigt.

Erfassen von Ressourcendaten

Die Datenerfassung erfolgt alle 10 Minuten. Jedes Cloud-Konto zeigt an, wann die Datenerfassung zuletzt abgeschlossen wurde.

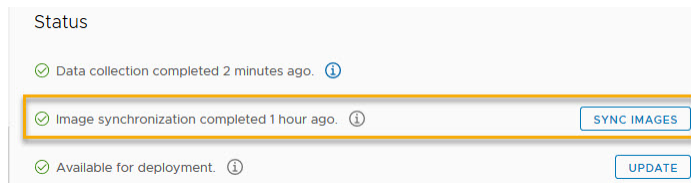


Erfassen von Image-Daten

Die Image-Synchronisierung erfolgt alle 24 Stunden. Sie können die Image-Synchronisierung für einige Cloud-Kontotypen einleiten. Um die Image-Synchronisierung einzuleiten, öffnen Sie das Cloud-Konto (**Infrastruktur > Cloud-Konten**, wählen Sie dann das gewünschte Cloud-Konto aus und öffnen Sie es) und klicken Sie auf die Schaltfläche **Images synchronisieren**. Es gibt keine Image-Synchronisierungsoption für NSX-Cloud-Konten.

Hinweis Images werden intern als „öffentlich“ oder „privat“ eingestuft. Öffentliche Images werden gemeinsam genutzt und sind nicht spezifisch für ein bestimmtes Cloud-Abonnement oder eine bestimmte Organisation. Private Images werden nicht gemeinsam genutzt und sind spezifisch für ein bestimmtes Abonnement. Öffentliche und private Images werden alle 24 Stunden automatisch synchronisiert. Eine Option auf der Seite „Cloud-Konto“ ermöglicht es Ihnen, die Synchronisierung für private Images auszulösen.

Auf der Seite „Cloud-Konto“ wird der Zeitpunkt angezeigt, an dem die Image-Synchronisierung abgeschlossen wurde.



Um Fault Tolerance und Hochverfügbarkeit in Bereitstellungen zu vereinfachen, stellt jeder NSX-T-Datencenter-Endpoint ein Cluster aus drei NSX Managern dar. Informationen hierzu finden Sie unter [Erstellen eines NSX-T-Cloud-Kontos in vRealize Automation Cloud Assembly](#).

Cloud-Konten und Onboarding-Pläne

Wenn Sie ein Cloud-Konto erstellen, werden die Daten aller Maschinen, die diesem Konto zugeordnet sind, erfasst und anschließend auf der Seite **Infrastruktur > Ressourcen > Maschinen** angezeigt. Wenn das Cloud-Konto über Maschinen verfügt, die außerhalb von vRealize Automation Cloud Assembly bereitgestellt wurden, können Sie einen Onboarding-Plan verwenden, der zulässt, dass die Maschinen von vRealize Automation Cloud Assembly verwaltet werden.

Informationen zum Hinzufügen von Cloud-Konten finden Sie unter [Hinzufügen von Cloud-Konten zu vRealize Automation Cloud Assembly](#).

Informationen zum Onboarding von nicht verwalteten Maschinen finden Sie unter [Definition von Onboarding-Plänen in vRealize Automation Cloud Assembly](#).

Definition der Kosten meiner vRealize Automation Cloud Assembly-Bereitstellung

Projektadministratoren und Cloud-Administratoren sind häufig mit der Verwaltung von Kosten beauftragt. Mit der Kostenverwaltung von vRealize Automation Cloud Assembly können Sie die monetären Auswirkungen Ihrer einzelnen Bereitstellungen nachvollziehen, sodass Sie Ressourcen besser verwalten können.

Bevor Sie die Kosten anzeigen können, müssen Sie „Kosten“ in vRealize Operations so konfigurieren, dass die Funktion mit vRealize Automation funktioniert, und aktivieren. Stellen Sie beim Konfigurieren von vRealize Operations mit vRealize Automation sicher, dass beide Anwendungen auf dieselbe Zeitzone festgelegt sind. Um die Zeitzone in vRealize Operations zu konfigurieren, aktivieren Sie SSH und melden sich bei jedem vRealize Operations-Knoten an, bearbeiten die Datei `$ALIVE_Base/user/conf/analytics/advanced.properties` und fügen `timeZoneUseInMeteringCalculation = <time zone>` hinzu.

Die Kosten einer Bereitstellung im Zeitverlauf werden auf der Bereitstellungskarte als die bisherigen monatlichen Kosten angezeigt, die zu Beginn eines jeden Monats auf null zurückgesetzt werden. Die Aufschlüsselungen der Komponentenkosten sind in den Bereitstellungsdetails verfügbar. Wenn diese Informationen auf der Bereitstellungsebene bereitgestellt werden, wird der Cloud-Administrator darüber informiert. Diese Informationen sind aber auch für Mitglieder hilfreich, um die Auswirkungen ihrer Arbeit auf Budgets und langfristige Entwicklung zu verstehen.

Aus ähnlichen geschäftlichen Gründen müssen Sie möglicherweise die Gesamtkosten für ein gesamtes Projekt verstehen. Die Kosten auf Projektebene spiegeln die vollständigen Kosten aller Bereitstellungen für das Projektteam wider. Um die vollständigen Projektkosten anzuzeigen, wählen Sie **Infrastruktur > Projekte > Projekt auswählen > Kosten**. Die Ergebnisse zeigen die Gesamtkosten und die Kostenaufschlüsselung nach Bereitstellung.

Hinweis Projektkosten beinhalten nur die Kosten für Private Cloud-Arbeitslasten. Wenn ein Projekt Bereitstellungen enthält, die zu Public Clouds gehören, sind die Kosten für diese Bereitstellungen nicht in den Projektkosten enthalten.

Kostenberechnung

Die anfänglichen Kosten, die Sie auf der Bereitstellungsebene für Ihre Computing- und Speicherressourcen sehen, basieren auf branchenüblichen Benchmark-Sätzen und werden dann im Laufe der Zeit berechnet. Der Kostensatz wird auf Hosts angewendet und der Dienst berechnet die CPU- und Arbeitsspeicherraten. Der Server berechnet die Kosten alle 24 Stunden neu.

Sie können den Kostenserver auch jederzeit manuell auf der Seite „vROPs Endpoint“ unter **Infrastruktur > Integrationen > vROPs Endpoint** > aktualisieren. Klicken Sie im Abschnitt „vCenter Server“ auf **Synchronisieren**. Wenn Sie den Kostenserver manuell mit der Option **Synchronisieren** aktualisieren, werden die Kosten für alle Projekte in der Organisation neu berechnet. Je nach Anzahl der Projekte in Ihrer Organisation ist dieser Vorgang möglicherweise intensiv und nimmt Zeit in Anspruch.

Eine Liste der unterstützten Ressourcen finden Sie unter [Liste der kalkulierten Komponententypen in vRealize Automation Cloud Assembly](#).

Anpassung der berechneten Sätze

Nach der Arbeit mit den Benchmark-Werten für einen bestimmten Zeitraum kann ein Cloud-Administrator feststellen, dass die tatsächlichen Kosten reduziert werden. Sie können die Rate anpassen, die zur Berechnung der Kosten verwendet wird, damit Sie Ihre Geschäftspraktiken besser widerspiegelt.

Wenn Sie Anpassungen vornehmen, werden die Berechnungsänderungen wiedergegeben, wenn der Dienst die Berechnungen das nächste Mal ausführt. Der Server berechnet die Werte alle 24 Stunden neu.

Liste der kalkulierten Komponententypen in vRealize Automation Cloud Assembly





vRealize Automation Cloud Assembly bietet standardmäßige Kosteninformationen für die folgenden Blueprint-Komponententypen.

Tabelle 4-3. Kalkulierte Komponententypen

Blueprint-Komponententyp	Dienstname/Objekttyp	Blueprint-Ressourcentyp	Anmerkungen
Cloud-unabhängig	Maschine	Cloud.Machine	Wenn eine unabhängige Maschine mit vSphere konfiguriert ist, können Sie die Bereitstellungskosten anzeigen.
	Festplatte	Cloud.Volume	Wenn eine unabhängige Festplatte an eine virtuelle Maschine angehängt ist, die mit vSphere konfiguriert ist, können Sie die Bereitstellungskosten anzeigen.
vSphere	vSphere-Maschine	Cloud.vSphere.Machine	Wird mithilfe eines Cloud-spezifischen Blueprints bereitgestellt.
	vSphere-Festplatte	Cloud.vSphere.Disk	Wird mithilfe eines Cloud-spezifischen Blueprints bereitgestellt, der an eine virtuelle Maschine angehängt ist.

Vorgehensweise zum Schätzen der Kosten einer Bereitstellung

Vor der Bereitstellung eines Katalogelements können Sie die Vorlaufkosten als Kostenschätzung für Ihre Bereitstellung verwenden.

Daily Cost Estimate		×
	vSphere machine with disk	\$2.14
	Cloud_vSphere_Machine_1	\$2.11
	Compute	\$1.97
	Storage	\$0.03
	Additional charges	\$0.11
	Cloud_vSphere_Disk_1	\$0.03
	Storage	\$0.03
	Cloud_vSphere_Network_1	
	<i>Cost for this resource is not supported.</i>	
		CLOSE

Bei den Vorlaufkosten einer Bereitstellung handelt es sich um eine Schätzung der täglichen Kosten, basierend auf der Zuteilung einer Ressource für ein bestimmtes Katalogelement, bevor es bereitgestellt wird. Nachdem ein Katalogelement bereitgestellt wurde, können Sie die Kosten vom Monatsanfang bis zum aktuellen Datum als Summe der Vorlaufkosten auf den Registerkarten **Bereitstellung** und **Infrastruktur > Projekte** anzeigen. Die Vorlaufkosten werden für Private Cloud-Ressourcen wie vSphere Machine und vSphere Disk, Cloud Assembly-Katalogelemente und Cloud-unabhängige Elemente mit vCenter, das für Private Cloud konfiguriert ist, unterstützt.

Hinweis Die Vorlaufkosten werden für Public Cloud-Ressourcen oder andere Private Cloud-Ressourcen als vSphere Machine oder Disk nicht unterstützt.

Voraussetzungen

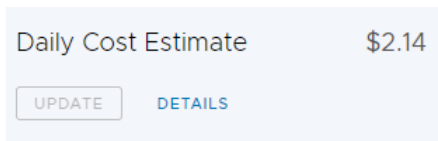
Um die Vorlaufkosten in vRealize Automation Cloud Assembly anzuzeigen, müssen Sie über einen vRealize Operations-Integrations-Endpoint verfügen, für den die Kostenberechnung aktiviert und die Währung voreingestellt sind.

Verfahren

- 1 Wählen Sie im Katalog ein Katalogelement aus und klicken Sie auf **Anfordern**.

Daily Cost Estimate	0.00
CALCULATE	DETAILS

- 2 Geben Sie die Details für Ihre Katalogelementanforderung ein und klicken Sie auf **Berechnen**.



Daily Cost Estimate \$2.14

UPDATE DETAILS

- 3 (Optional) Klicken Sie auf **Details**, um die Kostenaufschlüsselung im Fenster mit der Schätzung der täglichen Kosten anzuzeigen.

Nächste Schritte

Wenn die Schätzung der täglichen Kosten akzeptabel ist, klicken Sie auf **Senden**, um die Bereitstellungsanforderung fortzusetzen.

Vorgehensweise zum Feststellen der Gründe für hohe Kosten einer Bereitstellung

Bei der Kostenverwaltung stellen Projektadministratoren und Cloud-Administratoren unter Umständen fest, dass eine Bereitstellung teuer ist.

Die Kosten für eine Bereitstellung werden anhand ihrer Ressourcen berechnet. Gelegentlich können sich die Kosten für eine einzelne Bereitstellung durch die Kombination der Ressourcen erhöhen, wodurch die Bereitstellung gemessen an Ihrem Geschäftsbedarf teuer wird. Wenn Ihr Projekt also kostspielige Bereitstellungen enthält, steigen die Gesamtkosten für das Projekt entsprechend.

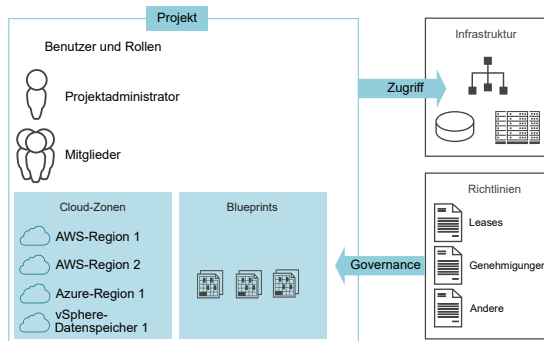
Ob Ihre Bereitstellung zu teuer ist, hängt von Ihrem Geschäftsbedarf ab. Sie können die Kostenaufschlüsselung Ihrer Bereitstellung anzeigen, indem Sie die Details einer Bereitstellung öffnen und auf **Kosten** klicken. Um die Kosten Ihrer Bereitstellung zu ändern, müssen Sie die Zusammensetzung der Ressourcen im Blueprint der Bereitstellung bearbeiten und das Ganze dann erneut bereitstellen.

Hinzufügen und Verwalten von vRealize Automation Cloud Assembly-Projekten

5

Mit Projekten wird gesteuert, wer Zugriff auf vRealize Automation Cloud Assembly-Blueprints hat und wo die Blueprints bereitgestellt werden. Mithilfe von Projekten verwalten und steuern Sie die Aufgaben, die von den Benutzern durchgeführt werden können, sowie die Cloud-Zonen, in denen Blueprints in Ihrer Cloud-Infrastruktur bereitgestellt werden können.

Cloud-Administratoren richten die Projekte ein, denen Benutzer und Cloud-Zonen hinzugefügt werden können. Jeder, der Blueprints erstellt und bereitstellt, muss Mitglied in mindestens einem Projekt sein.



Dieses Kapitel enthält die folgenden Themen:

- [Vorgehensweise zum Hinzufügen eines Projekts für mein vRealize Automation Cloud Assembly-Entwicklungsteam](#)
- [Weitere Informationen zu vRealize Automation Cloud Assembly-Projekten](#)

Vorgehensweise zum Hinzufügen eines Projekts für mein vRealize Automation Cloud Assembly-Entwicklungsteam

Sie erstellen ein Projekt, dem Sie Mitglieder und Cloud-Zonen hinzufügen, damit die Projektmitglieder ihre Blueprints in den zugeordneten Zonen bereitstellen können. Als vRealize Automation Cloud Assembly-Administrator erstellen Sie ein Projekt für ein Entwicklungsteam. Sie können dann einen Projektadministrator zuweisen oder selbst als Projektadministrator fungieren.

Wenn Sie einen Blueprint erstellen, wählen Sie zuerst das Projekt aus, mit dem er verknüpft werden soll. Das Projekt muss vorhanden sein. Erst dann können Sie den Blueprint erstellen.

Stellen Sie sicher, dass Ihre Projekte die geschäftlichen Anforderungen des Entwicklungsteams unterstützen.

- Stellt das Projekt die Ressourcen bereit, die die Ziele des Teams unterstützen? Ein Beispiel für die Unterstützung eines Blueprints durch die Infrastrukturressourcen und ein Projekt finden Sie unter [Anwendungsbeispiel: WordPress](#).

Dieses Verfahren basiert auf der Erstellung eines Anfangsprojekts, das nur die grundlegenden Konfigurationen enthält. Wenn das Entwicklungsteam seine Blueprints erstellt und bereitstellt, können Sie das Projekt unter Umständen ändern. Sie können Einschränkungen, benutzerdefinierte Eigenschaften und andere Optionen hinzufügen, um die Effizienz der Bereitstellung zu steigern. Weitere Informationen finden Sie in den Artikeln unter [Weitere Informationen zu vRealize Automation Cloud Assembly-Projekten](#).

Voraussetzungen

- Stellen Sie sicher, dass Sie die Cloud-Zonen konfiguriert haben. Weitere Informationen hierzu finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#).
- Stellen Sie sicher, dass Sie die Zuordnungen und Profile für die Regionen konfiguriert haben, die die Cloud-Zonen für dieses Projekt enthalten. Weitere Informationen hierzu finden Sie unter [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#).
- Stellen Sie sicher, dass Sie über die notwendigen Berechtigungen zum Durchführen dieser Aufgabe verfügen. Weitere Informationen hierzu finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Bestimmen Sie den Projektadministrator. Weitere Informationen zu den Funktionen eines Projektmanagers in vRealize Automation Cloud Assembly finden Sie unter [Definition der vRealize Automation Cloud Assembly-Benutzerrollen](#).
- Wenn Sie Projekten Active Directory-Gruppen hinzufügen, müssen Sie Active Directory-Gruppen für Ihre Organisation konfiguriert haben. Weitere Informationen finden Sie unter [Rollenzuweisungen für Gruppen in vRealize Automation bearbeiten](#) in *Verwalten von vRealize Automation*. Wenn Sie versuchen, nicht synchronisierte Gruppen zu einem Projekt hinzuzufügen, sind sie nicht verfügbar.

Verfahren

- 1 Wählen Sie **Infrastruktur > Konfigurieren > Projekte** aus und klicken Sie auf **Neues Projekt**.
- 2 Geben Sie den Projektnamen ein.
- 3 Klicken Sie auf die Registerkarte **Benutzer**.
 - a Damit nur der Besitzer auf Bereitstellungen von Projektmitgliedern zugreifen kann, deaktivieren Sie **Bereitstellungsfreigabe**.
 - b Fügen Sie Benutzer mit zugewiesenen Rollen hinzu.

- 4 Klicken Sie auf die Registerkarte **Bereitstellung** und fügen Sie eine oder mehrere Cloud-Zonen hinzu.

Die Cloud-Zonen müssen die Ressourcen enthalten, die die von den Benutzern bereitgestellten Blueprints unterstützen.

- 5 Klicken Sie auf **Erstellen**.

- 6 Klicken Sie zum Testen Ihres Projekts mit den Cloud-Zonen des Projekts auf der Seite „Projekte“ auf **Testkonfiguration**.

In der Simulation wird ein standardisierter hypothetischer Bereitstellungstest für die Ressourcen der Projekt-Cloud-Zone ausgeführt. Wenn der Test fehlschlägt, können Sie die Details überprüfen und die Ressourcenkonfiguration korrigieren.

Nächste Schritte

Erste Schritte mit Blueprints. Weitere Informationen hierzu finden Sie unter [Kapitel 6 Entwerfen Ihrer vRealize Automation Cloud Assembly-Bereitstellungen](#).

Weitere Informationen zu vRealize Automation Cloud Assembly-Projekten

Projekte stellen die Verbindung zwischen Blueprints und Ressourcen dar. Je besser Sie sich mit der Funktionsweise von Projekten und den Einsatzmöglichkeiten für Ihre Zwecke auskennen, desto effektiver können Sie den Entwicklungs- und Bereitstellungsprozess in vRealize Automation Cloud Assembly gestalten.

Verwenden von vRealize Automation Cloud Assembly-Projekt-Tags und benutzerdefinierten Eigenschaften

Als Administrator können Sie Kontrolleinschränkungen auf Projektebene oder benutzerdefinierte Eigenschaften hinzufügen, wenn sich die Anforderungen des Projekts von denen der vRealize Automation Cloud Assembly-Blueprints unterscheiden. Neben Einschränkungs-Tags können Sie während des Bereitstellungsvorgangs Ressourcen-Tags hinzufügen, die den bereitgestellten Ressourcen hinzugefügt werden, damit Sie die Ressourcen verwalten können.

Definition von Projekt-Ressourcen-Tags

Ein Projekt-Ressourcen-Tag fungiert als standardisiertes Bezeichnungs-Tag, mit dem Sie die bereitgestellten Ressourcen verwalten und die Konformität sicherstellen können.

Die in einem Projekt definierten Ressourcen-Tags werden allen Komponentenressourcen hinzugefügt, die als Teil dieses Projekts bereitgestellt wurden. Daraufhin können Sie die Ressourcen unter Verwendung des Standard-Taggings mit anderen Anwendungen verwalten.

Als Cloud-Administrator möchten Sie z. B. mithilfe einer Anwendung wie CloudHealth die Kosten verwalten. Sie fügen das `costCenter:eu-cc-1234`-Tag einem Projekt hinzu, das zur Entwicklung eines Personalwesen-Tools in der Europäischen Union dient. Wenn das Projektteam von diesem Projekt bereitgestellt wird, wird das Tag zu den bereitgestellten Ressourcen hinzugefügt. Anschließend konfigurieren Sie das Kosten-Tool, um die Ressourcen zu bezeichnen und zu verwalten, die dieses Tag enthalten. Andere Projekte mit anderen Kostenstellen hätten alternative Werte für den Schlüssel.

Definition von Projekteinschränkungs-Tags

Eine Projekteinschränkung fungiert als Governance-Definition. Es handelt sich um ein `key:value`-Tag, das die Ressourcen definiert, die von der Bereitstellungsanforderung in den Cloud-Zonen des Projekts verwendet oder vermieden werden.

Der Bereitstellungsprozess sucht nach Tags für die Netzwerke und den Speicher, die den Projekteinschränkungen entsprechen, und führt die Bereitstellung basierend auf passenden Tags aus.

Die Erweiterbarkeitseinschränkung wird zur Angabe der integrierten vRealize Orchestrator-Instanz genutzt, die für Erweiterbarkeits-Workflows verwendet werden soll.

Berücksichtigen Sie die folgenden Formate, wenn Sie Projekteinschränkungen konfigurieren.

- **key:value** und **key:value:hard**. Verwenden Sie dieses Tag in einem der beiden Formate, wenn der Blueprint auf Ressourcen mit dem entsprechenden Funktions-Tag bereitgestellt werden muss. Der Bereitstellungsvorgang schlägt fehl, wenn kein passendes Tag gefunden wird. Ein von den Mitgliedern eines Projekts bereitgestellter Blueprint muss beispielsweise in einem PCI-kompatiblen Netzwerk bereitgestellt werden. Sie verwenden `security:pci`. Wenn keine Netzwerke in den Cloud-Zonen des Projekts gefunden werden, schlägt die Bereitstellung fehl. Hiermit wird sichergestellt, dass keine unsicheren Bereitstellungen vorhanden sind.
- **key:value:soft**. Verwenden Sie dieses Tag, wenn Sie eine passende Ressource zwar bevorzugen, der Bereitstellungsvorgang aber ohne Fehler fortgesetzt werden soll, indem Ressourcen akzeptiert werden, bei denen das Tag nicht passt. Sie möchten beispielsweise, dass die Projektmitglieder ihre Blueprints in einem kostengünstigeren Speicher bereitstellen, lehnen es aber ab, dass sich die Speicherverfügbarkeit auf deren Bereitstellungsmöglichkeiten auswirkt. Sie verwenden `tier:silver:soft`. Befindet sich in den Cloud-Zonen des Projekts kein Speicher mit dem Tag `tier:silver`, wird der Blueprint weiterhin in anderen Speicherressourcen bereitgestellt.
- **!key:value**. Verwenden Sie dieses Tag mit „hard“ oder „soft“, wenn Sie die Bereitstellung für Ressourcen mit einem passenden Tag vermeiden möchten.

Aufgrund der höheren Priorität der Einschränkungs-Tags des Projekts überschreiben diese die Einschränkungs-Tags des Blueprints zur Bereitstellungszeit. Bei einem Blueprint, der nicht überschrieben werden darf, können Sie `failOnConstraintMergeConflict:true` im Blueprint verwenden. Beispiel: Ihr Projekt weist eine Netzwerkeinschränkung vom Typ `loc:london` auf, bei dem Blueprint handelt es sich jedoch um `loc:mumbai`. Fügen Sie eine Eigenschaft ähnlich dem folgenden Beispiel hinzu, wenn die Bereitstellung mit einem Einschränkungskonflikt fehlschlagen und der Projektstandort keinen Vorrang haben soll.

```
constraints:
  - tag: 'loc:mumbai'
failOnConstraintMergeConflict:true
```

Vorgehensweise zum Verwenden benutzerdefinierter Projekteigenschaften

Sie können eine benutzerdefinierte Projekteigenschaft für die Berichterstellung, zum Auslösen und Befüllen von Erweiterbarkeitsaktionen und Workflows sowie zum Überschreiben von Eigenschaften auf Blueprint-Ebene verwenden.

Durch Hinzufügen einer benutzerdefinierten Eigenschaft zu einer Bereitstellung können Sie den Wert auf der Benutzeroberfläche verwenden oder ihn mithilfe der API abrufen, um Berichte zu erzeugen.

Die Erweiterbarkeit kann auch eine benutzerdefinierte Eigenschaft für ein Erweiterbarkeitsabonnement verwenden.

Ein Blueprint verfügt unter Umständen über einen bestimmten Eigenschaftswert, den Sie für ein Projekt ändern möchten. Sie können einen alternativen Namen und einen anderen Wert als benutzerdefinierte Eigenschaft angeben.

Funktionsweise von vRealize Automation Cloud Assembly-Projekten zur Bereitstellungszeit

Projekte steuern den Benutzerzugriff auf die Cloud-Zonen und das Eigentum der Benutzer an den bereitgestellten Ressourcen. Ob Sie ein Cloud-Administrator oder ein Blueprint-Entwickler sind, Sie müssen verstehen, wie die Projekte zum Zeitpunkt der Bereitstellung funktionieren, damit Sie Ihre Bereitstellungen verwalten und Probleme beheben können.

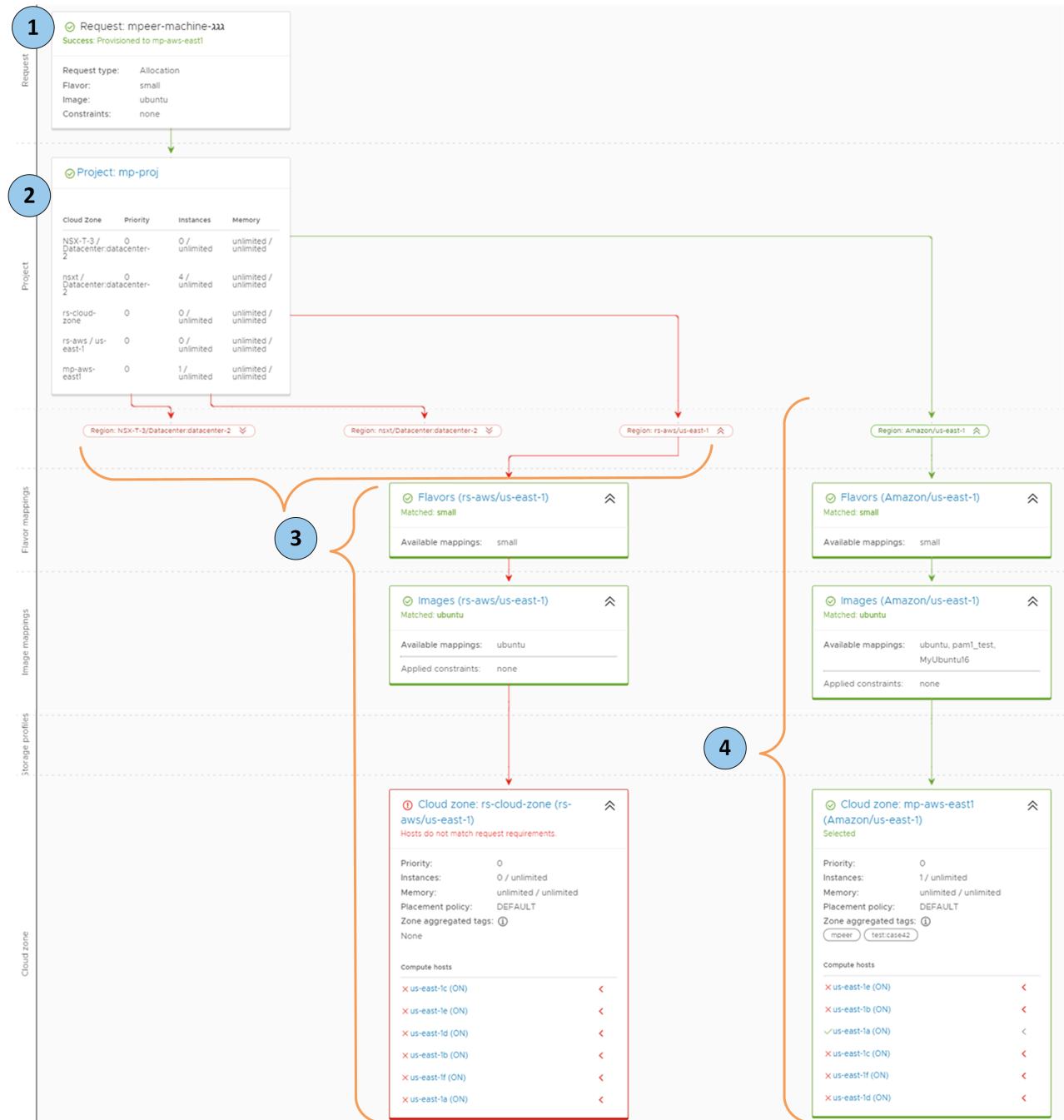
Als Cloud-Administrator, der Projekte für verschiedene Teams einrichtet, müssen Sie verstehen, wie Projekte festlegen, wo Blueprint-Komponenten bereitgestellt werden. Mit diesem Verständnis können Sie Projekte erstellen, die Blueprint-Entwickler unterstützen und fehlerhafte Bereitstellungen beheben.

Wenn Sie einen Blueprint erstellen, verknüpfen Sie ihn zunächst mit einem Projekt. Zum Zeitpunkt der Bereitstellung werden die Blueprint-Anforderungen anhand der Cloud-Zonen des Projekts ausgewertet, um den optimalen Bereitstellungsspeicherort zu finden.

Der folgende Workflow stellt den Vorgang dar.

- 1 Sie senden eine Blueprint-Bereitstellungsanforderung.

- Das Projekt wertet die Blueprint- und Projektanforderungen aus, wie z. B. Zuordnungs-, Image- und Einschränkungs-Tags. Die Anforderungen werden mit den Cloud-Zonen des Projekts verglichen, um eine Zone zu finden, die die Anforderungen unterstützt.
- Diese Zonen verfügen nicht über die Ressourcen zur Unterstützung der Anforderung.
- Diese Cloud-Zone unterstützt den Anforderungsbedarf und der Blueprint wird in dieser Cloud-Zonen-Kontobereich bereitgestellt.



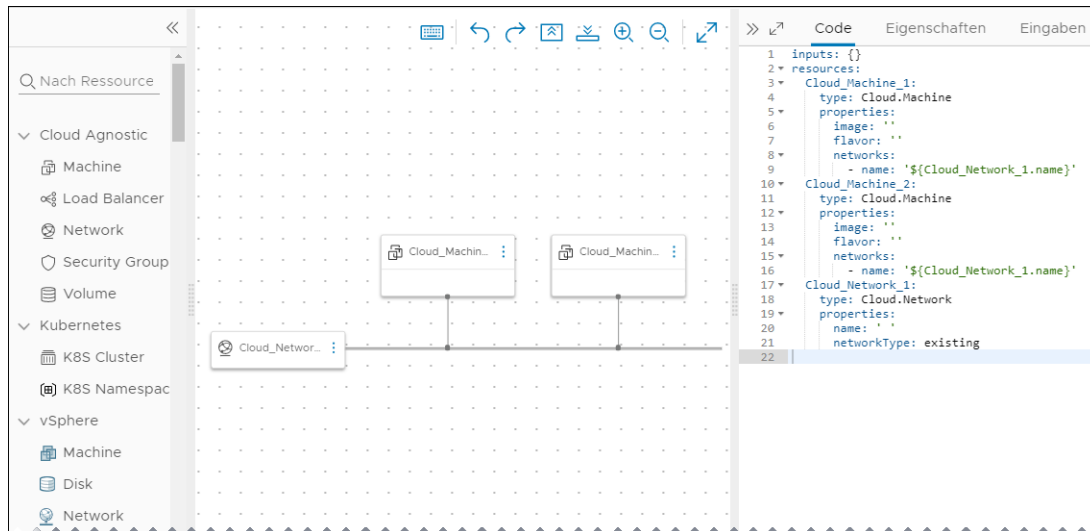
Entwerfen Ihrer vRealize Automation Cloud Assembly-Bereitstellungen

6

Bereitstellungen beginnen mit Blueprints, den Spezifikationen, die die Maschinen, Anwendungen und Dienste definieren, die Sie für Cloud-Ressourcen mithilfe von vRealize Automation Cloud Assembly bereitstellen.

Als Blueprint-Entwickler können Sie Blueprints entwerfen, die auf bestimmte Cloud-Anbieter abgestimmt sind. Alternativ dazu können Sie auch Cloud-unabhängige Blueprints erstellen. Die Cloud Zonen, die Ihrem Projekt zugeordnet sind, bestimmen den zur Auswahl stehenden Ansatz. Wenden Sie sich an Ihren Cloud-Administrator, um sicherzustellen, dass Sie wissen, aus welchen Ressourcen sich Ihre Cloud Zonen zusammensetzen.

Beachten Sie, dass die vRealize Automation Cloud Assembly-Blueprint-Erstellung ein „Infrastruktur-als-Code“-Prozess ist. Sie können Komponenten in der Design-Arbeitsfläche hinzufügen und verbinden, um den Vorgang zu starten. Anschließend vervollständigen Sie die Details mit dem Code-Editor rechts neben der Arbeitsfläche. Mit dem Code-Editor können Sie Code direkt eingeben oder Eigenschaftswerte in ein Formular eingeben.



Dieses Kapitel enthält die folgenden Themen:

- Vor dem Erstellen eines Blueprints

- [Möglichkeiten zum Erstellen von Blueprints](#)
- [Vorgehensweise zum Erstellen eines einfachen vRealize Automation Cloud Assembly-Blueprints von Grund auf](#)
- [Vorgehensweise zum Verbessern eines einfachen vRealize Automation Cloud Assembly-Blueprints](#)
- [Vorgehensweise zum Speichern verschiedener Versionen eines vRealize Automation Cloud Assembly-Blueprints](#)
- [Anpassen der Namen von bereitgestellten Ressourcen mithilfe von vRealize Automation Cloud Assembly](#)
- [Eigenschaften der vRealize Automation-Ressource](#)
- [Beispiele für Blueprint-Code](#)
- [Vorgehensweise zum Verwenden des vRealize Automation Cloud Assembly-Marketplace](#)
- [Vorgehensweise zum Erweitern und Automatisieren der Lebenszyklen von Anwendungen mit Erweiterbarkeit](#)

Vor dem Erstellen eines Blueprints

Sie können jederzeit einen vRealize Automation Cloud Assembly-Blueprint erstellen. Für die Bereitstellung müssen Sie jedoch zunächst die Infrastruktur Ihrer Cloud-Ressourcen definieren.

- [Kapitel 4 Erstellen der vRealize Automation Cloud Assembly-Ressourceninfrastruktur](#)

Darüber hinaus müssen Sie ein vRealize Automation Cloud Assembly-Projekt erstellen, das diese Infrastrukturressourcen als Cloud-Zonen enthält.

- [Verwenden von vRealize Automation Cloud Assembly-Projekt-Tags und benutzerdefinierten Eigenschaften](#)

Möglichkeiten zum Erstellen von Blueprints

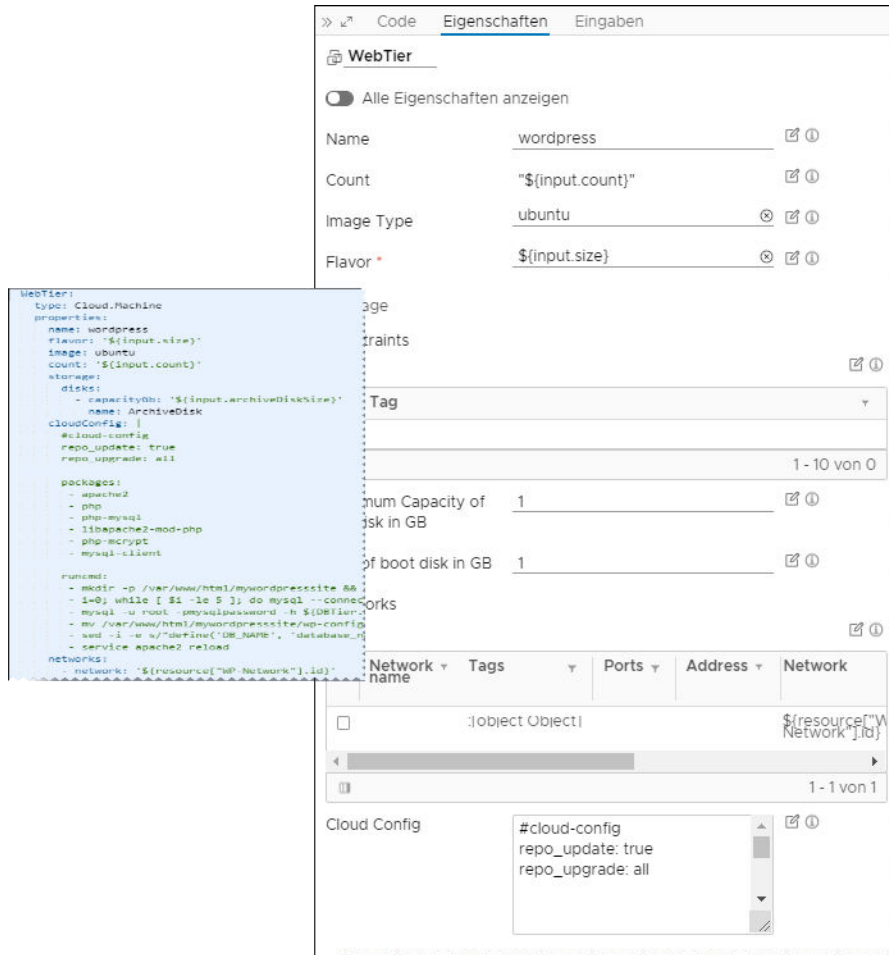
vRealize Automation Cloud Assembly erstellt und speichert Blueprints als Code, mit dem Sie Blueprints problemlos entwerfen und wieder verwenden können.

Sie können einen Blueprint aus einer leeren Arbeitsfläche erstellen oder den vorhandenen Code nutzen.

Die vRealize Automation Cloud Assembly-Blueprint-Design-Seite

Um einen Blueprint von Grund auf neu zu erstellen, wechseln Sie zu **Blueprints** und klicken Sie auf **Neu**. Ziehen Sie Komponenten auf die Arbeitsfläche, verbinden Sie sie und schließen Sie die Konfiguration im Code-Editor ab.

Mit dem Code-Editor können Sie Code direkt eingeben, ausschneiden, kopieren und einfügen. Wenn Sie nicht gern Code bearbeiten, können Sie eine Ressource in der Design-Arbeitsfläche auswählen, auf die Registerkarte **Eigenschaften** im Code-Editor klicken und die Werte dort eingeben. Die von Ihnen eingegebenen Eigenschaftswerte werden im Code so angezeigt, als hätten Sie sie direkt eingegeben.



Beachten Sie, dass Sie Code von einem Blueprint in einen anderen kopieren und einfügen können.

Klonen von Blueprints

Um einen Blueprint zu klonen, wechseln Sie zu **Blueprints**, wählen Sie eine Quelle aus und klicken Sie auf **Klonen**. Sie klonen einen Blueprint, um eine Kopie basierend auf der Quelle zu erstellen. Anschließend weisen Sie den Klon einem neuen Projekt zu oder verwenden ihn als Startcode für eine neue Anwendung.

Hochladen und Herunterladen

Der vRealize Automation Cloud Assembly-Marketplace enthält fertige Blueprints, mit denen Sie Ihre Prozesse beschleunigen können. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwenden des vRealize Automation Cloud Assembly-Marketplace](#).

Darüber hinaus können Sie den YAML-Code Ihrer Blueprints in einer Weise hochladen, herunterladen und freigeben, die für Ihren Standort sinnvoll ist. Sie können den Blueprint-Code auch mithilfe externer Editoren und Entwicklungsumgebungen ändern.

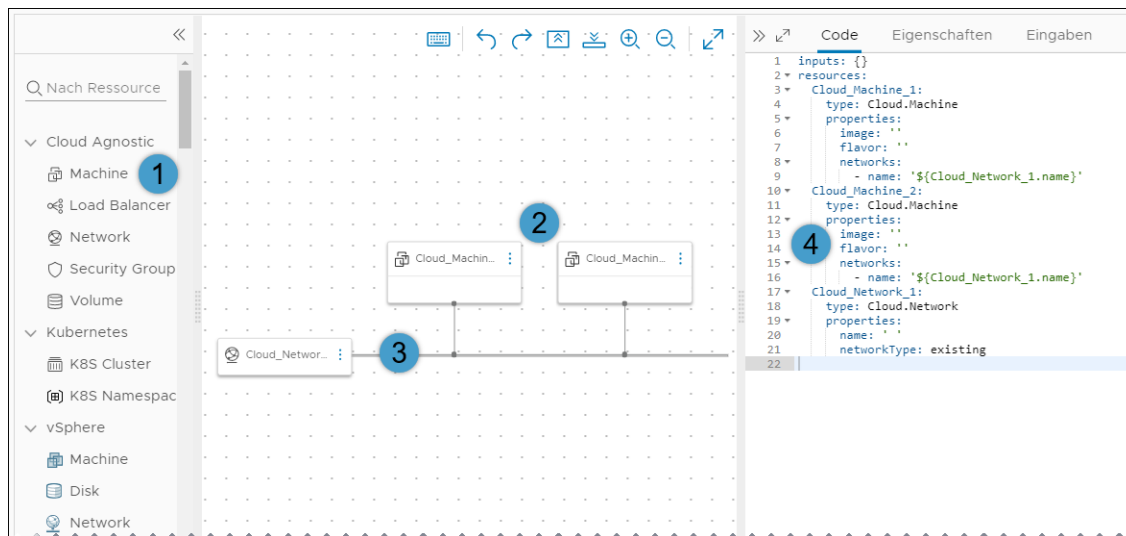
Hinweis Sie können freigegebenen Blueprint-Code validieren, indem Sie ihn im vRealize Automation Cloud Assembly-Code-Editor auf der Seite „Blueprint-Design“ überprüfen.

	Name	Quellcodeverwaltung	Projekt	Zuletzt aktualisiert	Aktualisiert von	Freigegebene Versionen
<input checked="" type="checkbox"/>	vSphere-With-Disk-Attached		test-AD-project	21. Jan. 2020, 12:48:57	sestervil@vmware.com	0 von 0
<input type="checkbox"/>	DB		0709-AWS-w2...	21. Jan. 2020, 11:34:36	pmartini@vmware.com	0 von 0
<input type="checkbox"/>	WordPress-BP		0709-AWS-w2...	20. Jan. 2020, 15:25:36	cani@vmware.com	0 von 0
<input type="checkbox"/>	git BPP		Azure Project1...	20. Jan. 2020, 15:18:40	cani@vmware.com	0 von 0
<input type="checkbox"/>	mp-nxsv		0717VSPHERE_PROJECTVSPHERE...	20. Jan. 2020, 10:03:51	pmartini@vmware.com	0 von 0
<input type="checkbox"/>	WP - PORT		wordpress project	19. Jan. 2020, 19:12:37	pmartini@vmware.com	2 von 2

Vorgehensweise zum Erstellen eines einfachen vRealize Automation Cloud Assembly-Blueprints von Grund auf

Auf der Seite „Entwerfen“ können Sie vRealize Automation Cloud Assembly-Blueprint-Spezifikationen für die Maschinen oder Anwendungen erstellen, die bereitgestellt werden sollen.

- 1 Suchen Sie nach den Komponenten.
- 2 Ziehen Sie die Komponenten auf die Arbeitsfläche.
- 3 Verbinden Sie die Komponenten.
- 4 Konfigurieren Sie die Komponenten, indem Sie den Blueprint-Code bearbeiten.



Auf der Seite „Entwerfen“ können Sie auch den Blueprint-Namen und die Version ändern, Versionen wiederherstellen oder einen Blueprint klonen bzw. bereitstellen.

Vorgehensweise zum Auswählen und Hinzufügen von vRealize Automation Cloud Assembly-Komponenten zu einem Blueprint

Bei den vRealize Automation Cloud Assembly-Komponenten handelt es sich um Ihre Blueprint-Bausteine. Auf der Seite „Entwerfen“ können Sie Cloud-unabhängige Komponenten oder Komponenten verwenden, die für einen Cloud-Anbieter spezifisch sind.

Komponenten können links auf der Seite „Entwerfen“ ausgewählt werden.

Cloud-unabhängige Komponenten

Sie können Cloud-unabhängige Komponenten für jeden Cloud-Anbieter bereitstellen. Zur Bereitstellungszeit verwendet die Bereitstellung passende Cloud-spezifische Ressourcen. Wenn ein Blueprint beispielsweise in AWS- und vSphere-Cloud-Zonen bereitgestellt werden soll, verwenden Sie Cloud-unabhängige Komponenten.

Komponenten der Cloud-Anbieter

Amazon Web Services-, Microsoft Azure- und VMware vSphere-Komponenten können ausschließlich in passenden AWS-, Azure- oder vSphere-Cloud-Zonen bereitgestellt werden.

Sie können Cloud-unabhängige Komponenten zu einem Blueprint hinzufügen, der Cloud-spezifische Komponenten für einen bestimmten Anbieter enthält. Achten Sie darauf, was von den Cloud-Zonen des Projekts in Bezug auf den Anbieter unterstützt wird.

Komponenten der Konfigurationsverwaltung

Die Komponenten der Konfigurationsverwaltung richten sich nach den integrierten Anwendungen. Eine Puppet-Komponente kann beispielsweise die Konfiguration der anderen Komponenten überwachen und erzwingen.

Vorgehensweise zum Verbinden von Blueprint-Ressourcen in vRealize Automation Cloud Assembly

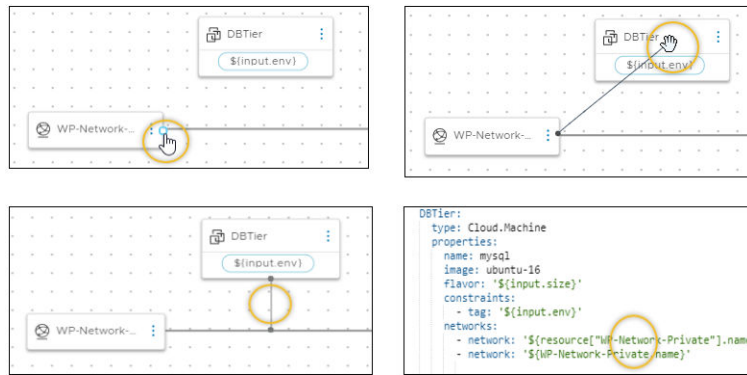
Verwenden Sie die grafische Design-Arbeitsfläche, um vRealize Automation Cloud Assembly-Blueprint-Ressourcen zu verbinden.

Sie können Ressourcen verbinden, wenn diese für eine Verbindung kompatibel sind. Beispiel:

- Verbinden eines Lastausgleichsdiensts mit einem Maschinen-Cluster.
- Verbinden einer Maschine mit einem Netzwerk.
- Verbinden des externen Speichers mit einer Maschine.

Um eine Verbindung herzustellen, bewegen Sie den Mauszeiger über den Rand einer Ressource, um die Verbindungsblase anzuzeigen. Klicken Sie dann auf die Blase, ziehen Sie sie auf die Zielressource und lassen Sie sie los.

Im Code-Editor wird zusätzlicher Code für die Quellressource im Zielressourcencode angezeigt.



Eine durchgehende Linie zwischen den Ressourcen zeigt an, dass die Ressourcen zum gleichen Ort führen müssen. Auch wenn Sie eine Verbindung auf der Arbeitsfläche hinzufügen können, schlägt die Bereitstellung fehl, wenn in Konflikt stehende Platzierungseinschränkungs-Tags vorhanden sind. Die Bereitstellung schlägt z. B. fehl, wenn Sie Ressourcen verbinden, von denen eine die harte Einschränkung auf eine Test-Cloud-Zone für us-west-1 und die andere auf eine Produktions-Cloud-Zone für us-east-1 aufweist.

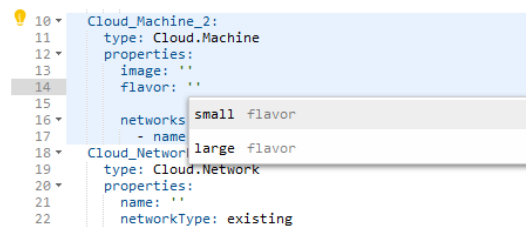
Vorgehensweise zum Erstellen eines gültigen vRealize Automation Cloud Assembly-Blueprint-Codes

Durch das Hinzufügen von vRealize Automation Cloud Assembly-Komponenten und das Verbinden dieser Komponenten auf der Arbeitsfläche wird nur Startercode erstellt. Um die Komponenten vollständig zu konfigurieren, bearbeiten Sie den Code.

Mit dem Code-Editor können Sie Code direkt eingeben oder Eigenschaftswerte in ein Formular eingeben. Um die direkte Code-Erstellung zu vereinfachen, enthält der vRealize Automation Cloud Assembly-Editor Funktionen zur Syntaxvervollständigung und Fehlerüberprüfung.

Editor-**Hinweise****Beispiel**

Verfügbare Werte

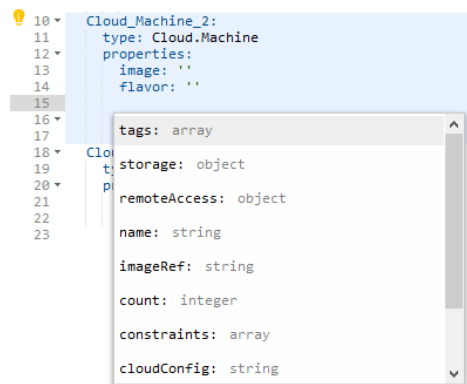


```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15   networks:
16     - name: 'small'
17     - name: 'large'
18 Cloud_Network_1:
19   type: Cloud.Network
20   properties:
21     name: ''
22     networkType: existing

```

Zulässige Eigenschaft

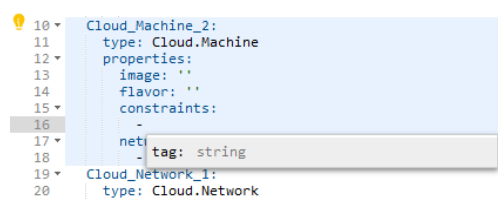


```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15   tags: array
16   storage: object
17   remoteAccess: object
18   name: string
19   imageRef: string
20   count: integer
21   constraints: array
22   cloudConfig: string

```

Untergeordnete Eigenschaft

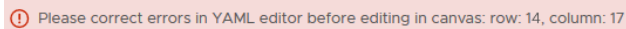
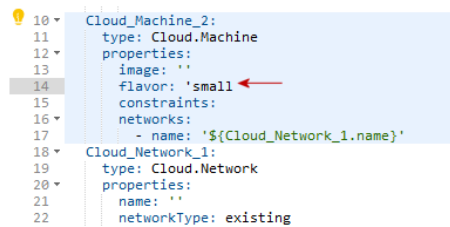


```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15     constraints:
16       - tag: string
17   networks:
18     - name: 'small'
19     - name: 'large'
20 Cloud_Network_1:
21   type: Cloud.Network

```

Syntaxfehler

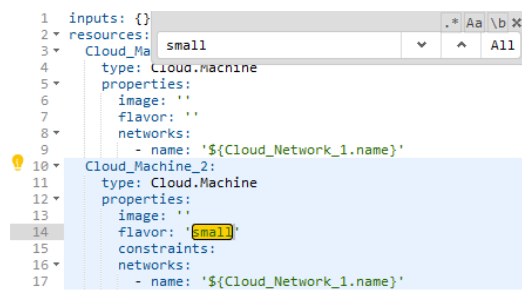



```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: 'small'
15     constraints:
16       - name: '${Cloud_Network_1.name}'
17   networks:
18     - name: 'small'
19     - name: 'large'
20 Cloud_Network_1:
21   type: Cloud.Network
22   properties:
23     name: ''
24     networkType: existing

```

Strg+F für die Suche



```

1 inputs: {}
2 resources:
3   Cloud_Machine_2:
4     type: Cloud.Machine
5     properties:
6       image: ''
7       flavor: ''
8       networks:
9         - name: '${Cloud_Network_1.name}'
10  Cloud_Machine_2:
11    type: Cloud.Machine
12    properties:
13      image: ''
14      flavor: 'small'
15      constraints:
16        - name: '${Cloud_Network_1.name}'

```

Editor-Hinweise	Beispiel
<p>Optionale Parameter</p> <p>Optionale Parameter einfügen</p> <ul style="list-style-type: none"> + attachedDisks + autoScaleConfiguration + cloudConfig + cloudConfigSettings 	<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>
<p>Schema-Hilfe</p> <p>cloudConfig</p> <p>Typ string</p> <p>When provisioning an instance, machine cloud-init startup instructions from user data fields. Sample cloud config instructions:</p> <pre> #cloud-config repo_update: true repo_upgrade: all packages: - httpd - mariadb-server runcmd: - [sh, -c, "amazon-linux-extras install -y - systemctl start httpd - sudo systemctl enable httpd </pre>	<pre> Tier: 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 </pre>

Vorgehensweise zum Verbessern eines einfachen vRealize Automation Cloud Assembly-Blueprints

Es gibt verbesserte Möglichkeiten für vRealize Automation Cloud Assembly-Blueprint-Code, mit denen ein einfacher Blueprint auf die nächste Ebene verschoben werden kann.

Die hier beschriebenen Techniken erfordern entsprechende Kenntnisse im Umgang mit Infrastrukturcode. Glücklicherweise ist der vRealize Automation Cloud Assembly-Code lesbar und einfach zu verstehen.

Anpassen eines vRealize Automation Cloud Assembly-Blueprints mithilfe von Benutzereingaben

Als Blueprint-Entwickler verwenden Sie Eingabeparameter, damit Benutzer zur Anforderungszeit benutzerdefinierte Auswahlen vornehmen können.

Wenn Benutzer Eingaben bereitstellen, müssen nicht mehr mehrere Kopien von Blueprints gespeichert werden, die sich nur geringfügig unterscheiden. Darüber hinaus können Eingaben einen Blueprint auf Tag-2-Vorgänge vorbereiten, da die Katalogbenutzer bei jeder Aktualisierung einer Bereitstellung zu neuen Eingaben aufgefordert werden.

Vorsicht Bestimmte Eigenschaftsänderungen führen dazu, dass eine Ressource neu erstellt wird. Wenn Sie beispielsweise `connection_string.name` unter `Cloud.Service.Azure.App.Service` ändern, wird die vorhandene Ressource gelöscht und eine neue Ressource erstellt.

Legen Sie beim Entwerfen von Eingaben zur Unterstützung von Tag-2-Änderungen fest, ob Eingaben zugelassen werden sollen, die Ressourcen löschen und neu erstellen. Informationen zu den Eigenschaften, die eine Ressource neu erstellen, finden Sie im konsolidierten Ressourcenschema unter [vRealize Automation-Blueprint-Schema in VMware {code}](#).

Die folgenden Eingaben zeigen, wie Sie einen Blueprint für einen MySQL-Datenbankserver erstellen, wobei Benutzer diesen Blueprint in verschiedenen Cloud-Ressourcenumgebungen bereitstellen und jedes Mal andere Funktionen und Anmeldedaten anwenden können.

Environment	env:dev	▼ ⓘ
Database Tier Size *	small	▼ ⓘ
Database Username *	ouradmin	
Database Password *	•••••	
MySQL Data Disk Size	4	⬆️ ⬆️ ⓘ

Vorgehensweise zum Definieren von Eingabeparametern für Blueprints

Fügen Sie dem Blueprint-Code einen `inputs`-Abschnitt hinzu, in dem Sie die auswählbaren Werte festlegen.

Im folgenden Beispiel können Maschinengröße, Betriebssystem und Anzahl der geclusterten Server ausgewählt werden.

```
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)

```

Wenn Sie mit der Bearbeitung von Code nicht vertraut sind, können Sie auf die Registerkarte **Eingaben** des Code-Editors klicken und dort entsprechende Einstellungen vornehmen. Das folgende Beispiel zeigt bestimmte Eingaben für die bereits erwähnte MySQL-Datenbank.

The screenshot shows the 'Inputs' tab of the vRealize Automation Cloud Assembly interface. It displays a table of blueprint inputs with columns for Name, Title, Type, and Default Value. Below the table, an 'Edit Blueprint Input' dialog is open for the 'size' input, showing fields for Name, Title, Description, Type, and Encrypted status.

Name	Title	Type	Default Value
size	Tier Machine Size	string	
username	Database Username	string	
userpassword	Database Password	string	****
databaseDiskSize	MySQL Data Disk Size	number	4

Edit Blueprint Input: size

Name *

Title

Description

Type

Encrypted ☐

Vorgehensweise zum Verweisen auf Eingabeparameter für Blueprints

Als Nächstes verweisen Sie im Abschnitt `resources` unter Verwendung der Syntax `${input.property-name}` auf einen Eingabeparameter.

Wenn ein Eigenschaftsname ein Leerzeichen enthält, trennen Sie ihn mit eckigen Klammern und doppelten Anführungszeichen ab, anstatt die Punktnotation zu verwenden: `${input["property name"]}`

Wichtig Im Blueprint-Code können Sie das Wort `input` nur verwenden, um einen Eingabeparameter anzugeben.

```
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      flavor: '${input.wp-size}'
      image: '${input.wp-image}'
      count: '${input.wp-count}'
```

Liste der Eingabeeigenschaften

Eigenschaft	Beschreibung
const	Wird mit <code>oneOf</code> verwendet. Der dem angezeigten Titel zugeordnete tatsächliche Wert.
default	Vorab befüllter Wert für die Eingabe. Der Standardwert muss den korrekten Typ aufweisen. Geben Sie kein Wort als Standardwert für eine Ganzzahl ein.
description	Benutzerhilfetext für die Eingabe.
encrypted	Gibt an, ob die vom Benutzer eingegebene Eingabe verschlüsselt werden soll, <code>True</code> oder <code>False</code> . Kennwörter werden in der Regel verschlüsselt.
enum	Ein Dropdown-Menü mit den zulässigen Werten. Verwenden Sie das folgende Beispiel als Formatierungshilfe. <div data-bbox="815 1417 1021 1493" data-label="Text"> <pre>enum: - value 1 - value 2</pre> </div>
format	Legt das erwartete Format für die Eingabe fest. Beispiel: <code>(25/04/19)</code> unterstützt Datum/Uhrzeit (<code>data-time</code>). Ermöglicht die Verwendung der Datumsauswahl in benutzerdefinierten vRealize Automation Service Broker-Formularen.
items	Deklariert Elemente innerhalb eines Arrays. Unterstützt „Zahl“, „Ganzzahl“, „Zeichenfolge“, „Boolesch“ oder „Objekt“.
maxItems	Maximale Anzahl der auswählbaren Elemente innerhalb eines Arrays.

Eigenschaft	Beschreibung
maxLength	Größte zulässige Anzahl an Zeichen für eine Zeichenfolge. Wenn Sie ein Feld beispielsweise auf 25 Zeichen begrenzen möchten, geben Sie <code>maxLength: 25</code> ein.
maximum	Größter zulässiger Wert für eine Zahl oder Ganzzahl.
minItems	Mindestanzahl der auswählbaren Elemente innerhalb eines Arrays.
minLength	Kleinste zulässige Anzahl an Zeichen für eine Zeichenfolge.
minimum	Kleinster zulässiger Wert für eine Zahl oder Ganzzahl.
oneOf	Ermöglicht, dass das Benutzereingabeformular einen Anzeigenamen (title) für einen weniger benutzerfreundlichen Wert (const) anzeigt. Wenn Sie einen Standardwert festlegen, legen Sie „const“ fest, nicht „title“. Gültig für die Verwendung mit den Typen „Zeichenfolge“, „Ganzzahl“ und „Zahl“.
pattern	Zulässige Zeichen für Zeichenfolgeneingaben in der Syntax für reguläre Ausdrücke. Beispiel: <code>'[a-z]+'</code> oder <code>'[a-z0-9A-Z@#&\$]+'</code>
Eigenschaften	Deklariert den key:value-Eigenschaftenblock für Objekte.
readOnly	Dient nur zur Bereitstellung einer Formularbezeichnung.
title	Wird mit oneOf verwendet. Der Anzeigename für einen const-Wert. Der Titel wird zum Zeitpunkt der Bereitstellung im Benutzereingabeformular angezeigt.
type	Datentyp „Zahl“, „Ganzzahl“, „Zeichenfolge“, „Boolesch“ oder „Objekt“.
writeOnly	Blendet im Formular die Tastenanschläge hinter Sternchen aus. Kann nicht mit „enum“ verwendet werden. Wird in benutzerdefinierten vRealize Automation Service Broker-Formularen als Kennwertfeld angezeigt.

Weitere Beispiele

Zeichenfolge mit Enumeration

```

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

```

Ganzzahl mit Mindest- und Höchstwert

```

count:
  type: integer
  title: Machine Count
  description: The number of machines that you want to deploy.
  maximum: 5
  minimum: 1
  default: 1

```

Array von Objekten

```

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

```

Zeichenfolge mit Anzeigenamen

```

platform:
  type: string
  oneOf:
    - title: AWS
      const: platform:aws
    - title: Azure
      const: platform:azure
    - title: vSphere
      const: platform:vsphere
  default: platform:aws

```

Zeichenfolge mit Mustervalidierung

```
username:
  type: string
  title: Username
  description: The name for the user that will be created when the machine is provisioned.
  pattern: ^[a-zA-Z]+$
```

Zeichenfolge als Kennwort

```
password:
  type: string
  title: Password
  description: The initial password that will be required to logon to the machine.
  Configured to reset on first login.
  writeOnly: true
```

Zeichenfolge als Textbereich

```
ssh_public_key:
  type: string
  title: SSH public key
  maxLength: 256
```

Boolesch

```
public_ip:
  type: boolean
  title: Assign public IP address
  description: Choose whether your machine should be internet facing.
  default: false
```

Vorgehensweise zum Festlegen der Reihenfolge der Komponentenbereitstellung in vRealize Automation Cloud Assembly

Wenn Sie einen vRealize Automation Cloud Assembly-Blueprint bereitstellen, benötigt eine Komponente möglicherweise eine andere Komponente, die zuerst verfügbar sein muss.

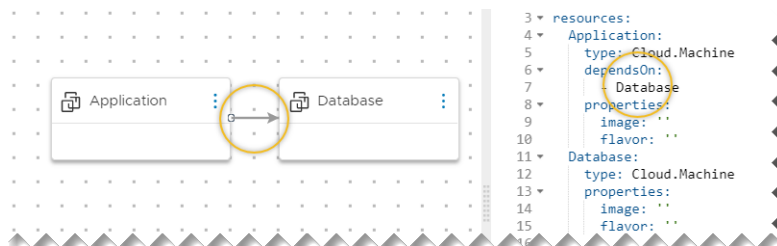
Vorgehensweise zum Erstellen einer expliziten Abhängigkeit

In bestimmten Fällen benötigt eine Komponente eine andere Komponente, die zuerst bereitgestellt werden muss. So muss beispielsweise zuerst ein Datenbankserver vorhanden sein, bevor ein Anwendungsserver erstellt und für den Zugriff darauf konfiguriert werden kann.

Eine explizite Abhängigkeit richtet die Build-Reihenfolge zum Zeitpunkt der Bereitstellung oder für vertikale oder horizontale Skalierungsaktionen ein. Sie können eine explizite Abhängigkeit mithilfe der grafischen Design-Arbeitsfläche oder des Code-Editors hinzufügen.

- Design-Arbeitsfläche – Zeichnen Sie eine Verbindung beginnend bei der abhängigen Komponente und endend bei der Komponente, die zuerst bereitgestellt werden muss.
- Code-Editor – Fügen Sie der abhängigen Komponente eine `dependsOn`-Eigenschaft hinzu und geben Sie die Komponente an, die zuerst bereitgestellt werden muss.

Eine explizite Abhängigkeit wird in Form eines durchgehenden Pfeils auf der Arbeitsfläche angegeben.



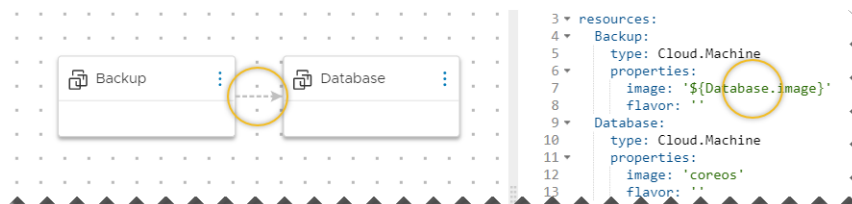
Erstellen einer impliziten Abhängigkeit oder Eigenschaftsbindung

In bestimmten Fällen benötigt eine Komponenteneigenschaft einen Wert, der sich in einer Eigenschaft einer anderen Komponente befindet. Beispiel: Ein Sicherungsserver benötigt möglicherweise das Betriebssystem-Image des zu sichernden Datenbankservers, d. h., der Datenbankserver muss zuerst vorhanden sein.

Eine implizite Abhängigkeit, die auch als Eigenschaftsbindung bezeichnet wird, steuert die Build-Reihenfolge, indem gewartet wird, bis die erforderliche Eigenschaft verfügbar ist. Erst danach wird die abhängige Komponente bereitgestellt. Sie fügen eine implizite Abhängigkeit mithilfe des Code-Editors hinzu.

- Bearbeiten Sie die abhängige Komponente, indem Sie eine Eigenschaft hinzufügen, die die Komponente und die Eigenschaft angibt, die zuerst vorhanden sein müssen.

Eine implizite Abhängigkeit oder Eigenschaftsbindung wird in Form eines gestrichelten Pfeils auf der Arbeitsfläche angegeben.



Vorgehensweise zum Verwenden von Ausdrücken zur flexibleren Gestaltung von vRealize Automation Cloud Assembly-Blueprint-Code

Zur Steigerung der Flexibilität können Sie dem vRealize Automation Cloud Assembly-Blueprint-Code Ausdrücke hinzufügen.

In Ausdrücken wird das Konstrukt `${expression}` verwendet, wie in den folgenden Beispielen gezeigt.

Die Beispiele sind so beschnitten, dass nur die wichtigen Zeilen angezeigt werden. Der gesamte, unbearbeitete Blueprint wird am Ende angezeigt.

Beispiele

Lassen Sie zu, dass der Benutzer zum Zeitpunkt der Bereitstellung den verschlüsselten Schlüssel einfügt, der für den Remote-Zugriff erforderlich ist:

```
inputs:
  sshKey:
    type: string
    maxLength: 500
resources:
  frontend:
    type: Cloud.Machine
    properties:
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'
```

Für die Bereitstellung von in VMware Cloud on AWS legen Sie den Ordernamen auf den erforderlichen Namen von *Workload* fest:

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
```

Versehen Sie zum Zeitpunkt der Bereitstellung die Maschine mit einem nur aus Kleinbuchstaben bestehenden *env*-Tag, das der ausgewählten Umgebung entspricht:

```
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)}'
```

Legen Sie die Anzahl der Maschinen im Front-End-Cluster auf eine (small) oder zwei (large) fest. Beachten Sie, dass der umfangreiche Cluster durch Löschung festgelegt wird:

```
inputs:
  envsize:
    type: string
    enum:
      - Small
      - Large
resources:
  frontend:
    type: Cloud.Machine
    properties:
      count: '${input.envsize == "Small" ? 1 : 2}'
```

Hängen Sie Maschinen an dasselbe *Standard*netzwerk an, indem Sie sie an die in der Netzwerkressource gefundene Eigenschaft binden:

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'
Cloud_Network_1:
```



```

type: Cloud.Network
properties:
  name: Default
  networkType: existing

```

Verschlüsseln Sie die für die API bereitgestellten Zugriffsanmeldedaten:

```

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

```

Ermitteln Sie die Adresse der API-Maschine:

```

resources:
  frontend:
    type: Cloud.Machine
    properties:
      cloudConfig: |
        runcmd:
          - echo ${resource.apitier.networks[0].address}
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - name: '${resource.Cloud_Network_1.name}'

```

Vollständiger Blueprint

```

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:
      - name: '${resource.Cloud_Network_1.name}'
apitier:
  type: Cloud.Machine
  properties:
    folderName: '${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:
      - name: '${resource.Cloud_Network_1.name}'
Cloud_Network_1:
  type: Cloud.Network
  properties:
    name: Default
    networkType: existing
  constraints:
    - tag: '${"env:" + to_lower(input.environment)}'

```

Ausdruckssyntax für vRealize Automation Cloud Assembly-Blueprint

Die vRealize Automation Cloud Assembly-Blueprint-Ausdruckssyntax macht alle Funktionen von Ausdrücken verfügbar.

Die Syntax wird in den Beispielen unter [Vorgehensweise zum Verwenden von Ausdrücken zur flexibleren Gestaltung von vRealize Automation Cloud Assembly-Blueprint-Code](#) nur teilweise dargestellt.

Literale

Folgende Literale werden unterstützt:

- Boolesch („true“ oder „false“)
- Ganzzahl
- Gleitkomma
- Zeichenfolge

Der umgekehrte Schrägstrich ist das Escapezeichen für doppeltes Anführungszeichen, einfaches Anführungszeichen und umgekehrten Schrägstrich selbst:

" wird wie folgt mit Escapezeichen geschützt: \"

' wird wie folgt mit Escapezeichen geschützt: \'

\ wird als \\ geschützt

Anführungszeichen müssen in einer Zeichenfolge nur dann mit Escapezeichen versehen werden, wenn die Zeichenfolge in Anführungszeichen desselben Typs eingeschlossen ist, wie im folgenden Beispiel gezeigt.

```
"I am a \"double quoted\" string inside \"double quotes\"."
```

- Null

Umgebungsvariablen

Umgebungsnamen:

- orgId
- projectId
- projectName
- deploymentId
- deploymentName
- blueprintId
- blueprintVersion
- blueprintName
- requestedBy (user)
- requestedAt (time)

Syntax:

```
env.ENV_NAME
```

Beispiel:

```
${env.blueprintId}
```

Ressourcenvariablen

Mithilfe von Ressourcenvariablen können Sie Bindungen mit Ressourceneigenschaften anderer Ressourcen erstellen.

Syntax:

```
resource.RESOURCE_NAME.PROPERTY_NAME
```

Beispiele:

- `${resource.db.id}`
- `${resource.db.networks[0].address}`
- `${resource.app.id}` (Gibt die Zeichenfolge für nicht geclusterte Ressourcen zurück, wobei „count“ nicht angegeben wird. Gibt das Array für geclusterte Ressourcen zurück.)
- `${resource.app[0].id}` (Gibt den ersten Eintrag für geclusterte Ressourcen zurück.)

Ressourcenvariablen vom Typ „Self“

Ressourcenvariablen vom Typ „Self“ sind nur für Ressourcen zulässig, die die Zuteilungsphase unterstützen. Ressourcenvariablen vom Typ „Self“ sind nur verfügbar (oder es wurde nur ein Wert für sie festgelegt), nachdem die Zuteilungsphase abgeschlossen ist.

Syntax:

```
self.property_name
```

Beispiel:

```
${self.address} (Gibt die Adresse zurück, die während der Zuteilungsphase zugewiesen wurde.)
```

Beachten Sie, dass für eine Ressource mit dem Namen `resource_x` `self.property_name` und `resource.resource_x.property_name` gleich sind und beide als Eigenreferenzen betrachtet werden.

Index der Clusteranzahl

Syntax:

```
count.index
```

Beispiel:

```
${count.index == 0 ? "primary" : "secondary"} (Gibt den Knotentyp für geclusterte Ressourcen zurück.)
```

Beschränkungen:

Die Verwendung von `count.index` für die Ressourcenzuteilung wird nicht unterstützt. Beispiel: Der folgende Kapazitätsausdruck schlägt fehl, wenn er auf die Position innerhalb eines zum Zeitpunkt der Eingabe erstellten Festplatten-Arrays verweist.

```
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
    properties:
      capacityGb: '${input.disks[count.index].size}'
      count: '${length(input.disks)}'
```

Bedingungen

Syntax:

- Gleichheitsoperatoren sind `==` und `!=`.
- Relationale Operatoren sind `<`, `>`, `<=` und `>=`.
- Logische Operatoren sind `&&`, `||` und `!`.
- In Bedingungsausdrücken wird das folgende Muster verwendet:

condition-expression ? true-expression : false-expression

Beispiele:

```
${input.count < 5 && input.size == 'small'}
```

```
${input.count < 2 ? "small" : "large"}
```

Arithmetische Operatoren

Syntax:

Operatoren sind `+`, `-`, `/`, `*` und `%`.

Beispiel:

```
${(input.count + 5) * 2}
```

Zeichenfolgenverkettung

Syntax:

`${'ABC' + 'DEF'}` ergibt ABCDEF.

Operatoren [] und .

Der Ausdruck folgt bei der Vereinheitlichung der Behandlung der Operatoren [] und . dem ECMAScript.

Also ist `expr.identifizier` äquivalent mit `expr["identifizier"]`. Der Bezeichner wird verwendet, um ein Literal zu konstruieren, dessen Wert der Bezeichner ist, und dann wird der []-Operator mit diesem Wert verwendet.

Beispiel:

```
${resource.app.networks[0].address}
```

Wenn darüber hinaus eine Eigenschaft ein Leerzeichen enthält, trennen Sie sie mit eckigen Klammern und doppelten Anführungszeichen ab, anstatt die Punktnotation zu verwenden.

Falsch:

```
input.operating system
```

Richtig:

```
input["operating system"]
```

Aufbau einer Zuordnung

Syntax:

```
${{'key1':'value1', 'key2':input.key2}}
```

Aufbau eines Arrays

Syntax:

```
${['key1','key2']}
```

Beispiel:

```
${[1,2,3]}
```

Funktionen

Syntax:

```
${function(arguments...)}
```

Beispiel:

```
${to_lower(resource.app.name)}
```

Tabelle 6-1. Funktionen

Funktion	Beschreibung
abs(number)	Absoluter Zahlenwert
floor(number)	Gibt den größten (am nächsten an positiv unendlich liegenden) Wert zurück, der kleiner oder gleich dem Argument und gleich einer mathematischen Ganzzahl ist
ceil(number)	Gibt den kleinsten (am nächsten an negativ unendlich liegenden) Wert zurück, der größer oder gleich dem Argument und gleich einer mathematischen Ganzzahl ist
to_lower(str)	Konvertiert eine Zeichenfolge in Kleinbuchstaben
to_upper(str)	Konvertiert eine Zeichenfolge in Großbuchstaben
contains(array, value)	Überprüft, ob ein Array einen bestimmten Wert enthält
contains(string, value)	Überprüft, ob eine Zeichenfolge einen bestimmten Wert enthält
join(array, delim)	Verknüpft ein Array von Zeichenfolgen mit einem Trennzeichen und gibt eine Zeichenfolge zurück
split(string, delim)	Teilt eine Zeichenfolge mit Trennzeichen und gibt ein Array von Zeichenfolgen zurück
slice(array, begin, end)	Gibt ein Segment eines Arrays vom Anfangsindex bis zum Endindex zurück
reverse(array)	Keht die Reihenfolge der Einträge eines Arrays um
starts_with(subject, prefix)	Überprüft, ob die Zeichenfolge mit einer bestimmten Präfixzeichenfolge beginnt
ends_with(subject, suffix)	Überprüft, ob die Zeichenfolge mit einer bestimmten Suffixzeichenfolge endet
replace(string, target, replacement)	Ersetzt die Zeichenfolge, die die Zielzeichenfolge enthält, mit der Zielzeichenfolge
substring(string, begin, end)	Gibt eine Teilzeichenfolge der Zeichenfolge vom Anfangsindex bis zum Endindex zurück
format(format, values...)	Gibt eine formatierte Zeichenfolge unter Verwendung des Class Formatter -Formats und der zugehörigen Werte von Java zurück.
keys(map)	Gibt die Schlüssel der Zuordnung zurück
values(map)	Gibt die Werte der Zuordnung zurück
merge(map, map)	Gibt eine zusammengeführte Zuordnung zurück
length(string)	Gibt die Zeichenfolgenlänge zurück
length(array)	Gibt die Array-Länge zurück
max(array)	Gibt den maximalen Wert aus einem Array von Zahlen zurück

Tabelle 6-1. Funktionen (Fortsetzung)

Funktion	Beschreibung
min(array)	Gibt den Mindestwert aus einem Array von Zahlen zurück
sum(array)	Gibt die Summe aller Werte aus einem Array von Zahlen zurück
avg(array)	Gibt den Durchschnittswert aller Werte aus einem Array von Zahlen zurück
digest(value, type)	Gibt einen Digest des Werts unter Verwendung des unterstützten Typs (md5, sha1, sha256, sha384, sha512) zurück
to_string(value)	Gibt eine Zeichenfolgendarstellung des Werts zurück
to_number(string)	Analysiert eine Zeichenfolge als Zahl
not_null(array)	Gibt den ersten Eintrag zurück, der nicht null ist
base64_encode(string)	Gibt einen base64-codierten Wert zurück
base64_decode(string)	Gibt einen decodierten base64-Wert zurück
now()	Gibt die aktuelle Uhrzeit im Format ISO-8601 zurück
uuid()	Gibt eine zufallsgenerierte UUID zurück
from_json(string)	Analysiert eine JSON-Zeichenfolge
to_json(value)	Serialisiert einen Wert als JSON-Zeichenfolge
json_path(value, path)	Wertet den Pfad anhand des Werts unter Verwendung von XPath for JSON aus.
matches(string, regex)	Überprüft, ob die Zeichenfolge mit einem Regex-Ausdruck übereinstimmt
url_encode(string)	Codiert eine Zeichenfolge unter Verwendung der URL-Codierungsspezifikation
trim(string)	Entfernt vorangestellte und nachgestellte Leerzeichen

Vorgehensweise zum automatischen Initialisieren einer Maschine in einem vRealize Automation Cloud Assembly-Blueprint

Sie können die Maschineninitialisierung in vRealize Automation Cloud Assembly mithilfe von vSphere-Anpassungsspezifikationen oder durch direkte Ausführung von Befehlen anwenden.

Eine Eigenschaft in Ihrem-Blueprint-Code kann anhand des Namens auf eine vSphere-Anpassungsspezifikation verweisen. Alternativ können Sie einen cloudConfig-Abschnitt zum Blueprint hinzufügen, in den bestimmte Befehle eingebettet werden.

Gehen Sie beim Versuch, eingebettete Befehle mit der Initialisierung der Anpassungsspezifikation zu kombinieren, umsichtig vor. Sie sind in formaler Hinsicht nicht kompatibel und verursachen möglicherweise inkonsistente oder unerwünschte Ergebnisse, wenn sie gemeinsam verwendet werden.

Ein Beispiel dazu, wie eine Anpassungsspezifikation die Befehle in einem cloudConfig-Abschnitt stört, finden Sie unter [Vorgehensweise zum Bereitstellen einer Linux-Maschine mit einer statischen IP-Adresse](#).

vSphere-Anpassungsspezifikationen in vRealize Automation Cloud Assembly-Blueprints

Bei der Bereitstellung auf vSphere-basierten Cloud-Zonen können Sie Gastbetriebssystemeinstellungen mithilfe von Anpassungsspezifikationen zur Bereitstellungszeit anwenden.

Die Anpassungsspezifikation muss in vSphere auf dem Ziel vorhanden sein, auf dem die Bereitstellung erfolgt.

Bearbeiten Sie den Blueprint-Code direkt. Das folgende Beispiel verweist auf eine `cloud-assembly-linux`-Anpassungsspezifikation für einen WordPress-Host auf vSphere.

```
resources:
  WebTier:
    type: Cloud.vSphere.Machine
    properties:
      name: wordpress
      cpuCount: 2
      totalMemoryMB: 1024
      imageRef: 'Template: ubuntu-18.04'
      customizationSpec: 'cloud-assembly-linux'
      resourceGroupName: '/Datacenters/Datacenter/vm/deployments'
```

Anpassungsspezifikationen und Initialisierungsbefehle

Wenn die Bereitstellungserfahrung mit den derzeit von Ihnen in vSphere durchgeführten Aktionen übereinstimmen soll, ist die Verwendung von Anpassungsspezifikationen möglicherweise der beste Ansatz. Eine neutralere Herangehensweise zur Erweiterung auf eine Hybrid-Bereitstellung oder eine Bereitstellung mit mehreren Clouds hingegen besteht in der Einbettung von Initialisierungsbefehlen in einen cloudConfig-Abschnitt eines Blueprints.

Weitere Informationen zu cloudConfig-Abschnitten in Blueprints finden Sie unter [Konfigurationsbefehle in vRealize Automation Cloud Assembly-Blueprints](#).

Konfigurationsbefehle in vRealize Automation Cloud Assembly-Blueprints

Sie können einen cloudConfig-Abschnitt im vRealize Automation Cloud Assembly-Blueprint-Code einfügen, dem Sie zur Bereitstellungszeit auszuführende Initialisierungsbefehle für Maschinen hinzufügen.

- Linux – Initialisierungsbefehle folgen dem offenen [cloud-init](#)-Standard.
- Windows – Initialisierungsbefehle verwenden [Cloudbase-init](#).

Hinweis Linux [cloud-init](#) und Windows [Cloudbase-init](#) verwenden nicht dieselbe Syntax. Ein cloudConfig-Abschnitt für ein Betriebssystem funktioniert nicht in einem Maschinen-Image des anderen Betriebssystems.

Sie verwenden Initialisierungsbefehle, um die Anwendung von Daten oder Einstellungen zum Zeitpunkt der Instanzerstellung zu automatisieren, wodurch Benutzer, Berechtigungen, Installationen oder andere befehlsbasierte Vorgänge angepasst werden können. Zu den Beispielen gehören:

- Festlegen eines Hostnamens
- Erstellen und Einrichten von privaten SSH-Schlüsseln
- Installieren von Paketen

In vRealize Automation Cloud Assembly können Sie bei der Konfiguration der Infrastruktur Initialisierungsbefehle auch im Vorfeld zu einem Maschinen-Image hinzufügen. Alle Blueprints, die auf das Quell-Image verweisen, erhalten dieselbe Initialisierung.

Wichtig Möglicherweise verfügen Sie über eine Image-Zuordnung und einen Blueprint, die beide Initialisierungsbefehle enthalten. Zum Zeitpunkt der Bereitstellung werden die Befehle zusammengeführt. Die konsolidierten Befehle werden dann von vRealize Automation Cloud Assembly ausgeführt.

Wenn derselbe Befehl an beiden Positionen angezeigt wird, aber unterschiedliche Parameter enthält, wird nur der Image-Zuordnungsbefehl ausgeführt.

Zusätzliche Informationen hierzu finden Sie unter [Weitere Informationen zu Image-Zuordnungen in vRealize Automation Cloud Assembly](#).

Der folgende cloudConfig-Beispielabschnitt stammt aus dem Blueprint-Code des [WordPress-Anwendungsbeispiel: Erstellen eines einfachen Blueprints](#) für den Linux-basierten MySQL-Server.

Um die korrekte Interpretation von Befehlen zu gewährleisten, fügen Sie immer einen senkrechten Strich (`cloudConfig: |`) hinzu, wie in der Abbildung gezeigt.

Wenn ein cloud-init-Skript unerwartetes Verhalten aufweist, überprüfen Sie die erfasste Konsolenausgabe in `/var/log/cloud-init-output.log` zur Fehlerbehebung. Weitere Informationen zu cloud-init finden Sie in der [cloud-init-Dokumentation](#).

```
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
```

Vorgehensweise zum Bereitstellen einer Linux-Maschine mit einer statischen IP-Adresse

Bei der Bereitstellung in vSphere muss für eine statische IP-Adresse von vRealize Automation Cloud Assembly eine vSphere-Anpassungsspezifikation erzeugt werden, die cloud-init-Befehle stören kann.

Problem

- Ein vRealize Automation Cloud Assembly-Blueprint enthält `assignment: static`, um eine statische IP-Adresse auf eine virtuelle vSphere-Maschine anzuwenden.
- Der Blueprint enthält darüber hinaus einen cloudConfig-Abschnitt mit Initialisierungsbefehlen, die mithilfe von „cloud-init“ ausgeführt werden.

- Um der virtuellen Maschine eine statische IP-Adresse zuzuweisen, erzeugt vRealize Automation Cloud Assembly dynamisch eine anzuwendende vSphere-Anpassungsspezifikation.
- Bei jeder Anwendung einer Anpassungsspezifikation startet der letzte Vorgang die virtuelle Maschine neu.
- Da die Ausführung der cloud-init-Befehle in der Anpassungsspezifikation nicht angegeben wird, kommt es beim Neustart zu einer Unterbrechung der Befehlsausführung.
- Die cloud-init-Befehle werden ausschließlich beim ersten Start ausgeführt und im Fall einer Unterbrechung nicht automatisch wiederhergestellt.
- Die resultierende virtuelle Maschine wird nur teilweise konfiguriert.

Problemumgehung

Erstellen Sie eine Maschinenvorlage mit einer zeitgesteuerten Deaktivierung von cloud-init. Stellen Sie dann basierend auf der Vorlage Maschinen bereit, damit die Anpassungsspezifikation und der Neustart vor cloud-init erfolgen können.

Beispielverfahren – Ubuntu 18.04

Die folgenden Schritte gelten für Ubuntu 18.04. Sie müssen unter Umständen Änderungen vornehmen und die hier dargestellten Konzepte für andere Linux-Versionen oder -Angebote anpassen.

- 1 Erstellen Sie die virtuelle Maschine und aktualisieren Sie sie auf die gewünschten Versionen und Pakete.

Denken Sie daran, dass cloud-init im Gegensatz zu Ubuntu 18.04 in anderen Linux-Angeboten unter Umständen nicht vorinstalliert ist.

- 2 Konfigurieren Sie cloud-init neu, indem Sie die Datenquelle auf OVF setzen.

```
sudo dpkg-reconfigure cloud-init
```

- 3 Bearbeiten Sie die folgende Datei.

```
/etc/cloud/cloud.cfg
```

- a Aktivieren Sie die herkömmliche Gastbetriebssystemanpassung (GOSC), indem Sie folgende Zeile hinzufügen.

```
disable_vmware_customization: true
```

- b Stellen Sie sicher, dass die Netzwerkkonfiguration aktiviert ist. Löschen oder kommentieren Sie die Einstellung zum Deaktivieren gegebenenfalls aus.

```
network:
  # config: disabled
```

Alternativ können Sie auch alle Konfigurationsdateien in folgendem Verzeichnis überprüfen.

```
/etc/cloud/cloud.cfg.d/*
```

Löschen Sie alle Dateien, die eine Einstellung vom Typ `network: {config: disabled}` enthalten.

- 4 Bearbeiten Sie die folgende Datei.

```
/usr/lib/tmpfiles.d/tmp.conf
```

- Verhindern Sie, dass das temporäre Verzeichnis entfernt wird, indem Sie die Einstellung auskommentieren.

```
# D /tmp 1777 root root -
```

- 5 Bearbeiten Sie die folgende Datei.

```
/lib/systemd/system/open-vmtools.service
```

- Konfigurieren Sie open-vmtools für den Start nach dbus.service, indem Sie die folgende Zeile unter dem Abschnitt `[Unit]` hinzufügen.

```
After=dbus.service
```

- 6 Erstellen Sie die neue leere Datei, die cloud-init deaktiviert.

```
sudo touch /etc/cloud/cloud-init.disabled
```

- 7 Erstellen Sie ein Skript vom Typ „re_init.sh“. Nachdem ein Cron-Auftrag für die Anpassungsspezifikation aufgeschoben wurde, wird cloud-init vom Skript erneut aktiviert und initialisiert.

```
sudo rm -rf /etc/cloud/cloud-init.disabled
sudo cloud-init init
sleep 20
sudo cloud-init modules --mode config
sleep 20
sudo cloud-init modules --mode final
```

- 8 Fügen Sie die Ausführungsberechtigung für das Skript hinzu.

```
sudo chmod +x re_init.sh
```

- 9 Erstellen Sie den Cron-Auftrag, der mit einer Verzögerung von 90 Sekunden beim Start ausgeführt wird. Geben Sie `crontab -e` und Folgendes ein:

```
@reboot ( sleep 90 ; sh /script_path/delay_init.sh )
```

Sie können mehr als 90 Sekunden angeben, wenn Anpassungsspezifikationen und Neustarts mehr Zeit in Anspruch nehmen.

- 10 Erstellen Sie ein Skript vom Typ „cleaner.sh“, das die Vorlage bereinigt. Ersetzen Sie `cloudadmin` durch Ihren eigenen Benutzer, den Sie während der Installation des Betriebssystems eingerichtet haben.

Das Beispielskript ist Ubuntu-spezifisch. Zum Erstellen eines Skripts für andere Linux-Angebote schließen Sie die hervorgehobenen, obligatorischen Abschnitte unbedingt ein.

```
#!/bin/bash

# Add usernames to add to /etc/sudoers for passwordless sudo
users=("ubuntu" "cloudadmin")

for user in "${users[@]}"
do
cat /etc/sudoers | grep ^$user
RC=$?
if [ $RC != 0 ]; then
bash -c "echo \"$user ALL=(ALL) NOPASSWD:ALL\" >> /etc/sudoers"
fi
done

#grab Ubuntu Codename
codename="$(lsb_release -c | awk {'print $2'})"

#Stop services for cleanup
service rsyslog stop

#clear audit logs
if [ -f /var/log/audit/audit.log ]; then
cat /dev/null > /var/log/audit/audit.log
fi
if [ -f /var/log/wtmp ]; then
cat /dev/null > /var/log/wtmp
fi
if [ -f /var/log/lastlog ]; then
cat /dev/null > /var/log/lastlog
fi

#cleanup persistent udev rules
if [ -f /etc/udev/rules.d/70-persistent-net.rules ]; then
rm /etc/udev/rules.d/70-persistent-net.rules
fi

#cleanup /tmp directories
rm -rf /tmp/*
rm -rf /var/tmp/*

#cleanup current ssh keys
#rm -f /etc/ssh/ssh_host_*

#cat /dev/null > /etc/hostname

#cleanup apt
apt-get clean

#Clean Machine ID

truncate -s 0 /etc/machine-id
```

```
rm /var/lib/dbus/machine-id
ln -s /etc/machine-id /var/lib/dbus/machine-id

#Clean Cloud-init
cloud-init clean --logs --seed

#cleanup shell history
history -w
history -c
```

- 11 Fügen Sie die Ausführungsberechtigung für das Vorlagenbereinigungsskript hinzu.

```
sudo chmod +x cleaner.sh
```

- 12 In Ubuntu 18.04 benötigt das Bereinigungsskript Root-Berechtigungen. Bearbeiten Sie die folgende Datei.

```
/etc/ssh/sshd_config
```

- a Stellen Sie sicher, dass Sie zum Root-Benutzer wechseln können.

```
PermitRootLogin yes
```

- b Legen Sie ein Kennwort für Root fest.

```
sudo passwd root
```

- 13 Führen Sie das Bereinigungsskript aus.

```
sudo ./script_path/cleaner.sh
```

- 14 (Optional) Stellen Sie aus Sicherheitsgründen Schritt 12 wieder her, um weitere Root-Anmeldungen zu verhindern.

- 15 Fahren Sie die virtuelle Maschine herunter und verwenden Sie vSphere, um sie in eine Vorlage umzuwandeln.

Vorlagenaktualisierungen

Der Cron-Auftrag wird bei jeder Aktualisierung der Vorlage ausgeführt. Wenn das Update länger als die Verzögerung (z. B. 90 Sekunden) dauert, müssen Sie vor dem Herunterfahren der Vorlage die `/etc/cloud/cloud-init.disabled`-Datei erneut hinzufügen und das Bereinigungsskript erneut ausführen. Andernfalls wird cloud-init beim ersten Start nicht deaktiviert und beim Neustart der Anpassungsspezifikation kommt es erneut zu einer Störung der cloud-init-Befehle.

Fehlerbehebung

Wenn Sie den Verdacht haben, dass die vSphere-Anpassungsspezifikation den Abschluss von cloud-init verhindert, deaktivieren Sie vorübergehend die Anpassungsspezifikation und ermitteln Sie, ob cloud-init wie erwartet abgeschlossen werden kann. Verwenden Sie zum temporären Deaktivieren der Anpassungsspezifikation die Eigenschaft `customizeGuestOS: false`.

```
properties:
  image: ubuntu
  cpuCount: 1
  totalMemoryMB: 8192
  customizeGuestOS: false
```

Vorgehensweise zum Warten auf die Initialisierung einer vRealize Automation Cloud Assembly-Bereitstellung

In bestimmten Fällen muss eine virtuelle Maschine vollständig gestartet werden, bevor mit der vRealize Automation Cloud Assembly-Bereitstellung fortgefahren wird.

Das Bereitstellen einer Maschine, auf der noch Pakete installiert und ein Webserver gestartet werden, kann beispielsweise dazu führen, dass ein vorschneller Benutzer versucht, eine Anwendung zu öffnen, bevor sie verfügbar ist.

Stellen Sie bei Verwendung dieser Funktion folgende Überlegungen an.

- Die Funktion verwendet das Modul `cloud-init phone_home` und ist verfügbar, wenn Linux-Maschinen bereitgestellt werden.
- Phone home steht für Windows aufgrund von `Cloudbase-init`-Einschränkungen nicht zur Verfügung.
- Phone home kann sich wie eine explizite Abhängigkeit auf die Bereitstellungsreihenfolge auswirken, ist aber flexibler im Hinblick auf Zeitplanungs- und Verarbeitungsoptionen.

Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Festlegen der Reihenfolge der Komponentenbereitstellung in vRealize Automation Cloud Assembly](#).

- Phone home benötigt cloudConfig-Befehle im Blueprint.
- Ihre Kreativität spielt eine Rolle. Ein cloudConfig-Abschnitt kann eine eingebettete Wartezeit zwischen Vorgängen enthalten, die zusammen mit phone home verwendet werden können.
- Blueprint-basiertes phone home funktioniert nicht, wenn die Maschinenvorlage bereits Einstellungen für das phone_home-Modul enthält
- Die Maschine muss über ausgehenden Kommunikationszugriff auf vRealize Automation Cloud Assembly verfügen.

Um unter Verwendung von phone home in vRealize Automation Cloud Assembly auf die Initialisierung der Maschine zu warten, fügen Sie dem Blueprint einen `cloudConfigSettings`-Abschnitt hinzu:

```
cloudConfigSettings:
  phoneHomeShouldWait: true
```



```
phoneHomeTimeoutSeconds: 600
phoneHomeFailOnTimeout: true
```

Eigenschaft	Beschreibung
phoneHomeShouldWait	Warten auf Initialisierung, True oder False.
phoneHomeTimeoutSeconds	Zeitpunkt, an dem entschieden werden muss, ob die Bereitstellung trotz laufender Initialisierung fortgesetzt wird. Standardwert ist 10 Minuten.
phoneHomeFailOnTimeout	Fortsetzen der Bereitstellung nach einer Zeitüberschreitung, True oder False. Wenn Sie die Bereitstellung dennoch fortsetzen, beachten Sie, dass sie aus anderen Gründen fehlschlagen kann.

Vorgehensweise zum Aktivieren des Remotezugriffs in vRealize Automation Cloud Assembly-Blueprints

Für den Remotezugriff auf eine von vRealize Automation Cloud Assembly bereitgestellte Maschine fügen Sie vor der Bereitstellung Eigenschaften zum Blueprint für diese Maschine hinzu.

Für den Remotezugriff können Sie eine der folgenden Authentifizierungsoptionen konfigurieren.

Hinweis Wenn Schlüssel kopiert werden müssen, können Sie auch einen cloudConfig-Abschnitt im Blueprint erstellen, um die Schlüssel bei der Bereitstellung automatisch zu kopieren. Die Spezifikationen sind hier nicht dokumentiert. Unter [Vorgehensweise zum automatischen Initialisieren einer Maschine in einem vRealize Automation Cloud Assembly-Blueprint](#) finden Sie jedoch allgemeine Informationen zu cloudConfig.

Erzeugen eines Schlüsselpaars zur vRealize Automation Cloud Assembly-Bereitstellungszeit

Wenn Sie nicht über ein eigenes öffentliches/privates Schlüsselpaar für die RAS-Authentifizierung verfügen, können Sie vRealize Automation Cloud Assembly veranlassen, ein Schlüsselpaar zu erzeugen.

Verwenden Sie den folgenden Code als Richtlinie.

- 1 Fügen Sie vor der Bereitstellung in vRealize Automation Cloud Assembly `remoteAccess-` Eigenschaften zum Blueprint hinzu, wie im Beispiel gezeigt.

Der Benutzername ist optional. Wenn Sie ihn weglassen, erzeugt das System eine zufällige ID als Benutzernamen.

Beispiel:

```
type: Cloud.Machine
properties:
  name: our-vm2
  image: Linux18
  flavor: small
  remoteAccess: authentication: generatedPublicPrivatekey username: testuser
```

- 2 Stellen Sie in vRealize Automation Cloud Assembly die Maschine aus dem zugehörigen Blueprint bereit und weisen Sie ihr den Status „Gestartet“ zu.

Der Bereitstellungsvorgang erzeugt die Schlüssel.

- 3 Suchen Sie in den Eigenschaften **Bereitstellungen > Topologie** nach dem Schlüsselnamen.
- 4 Verwenden Sie die Cloud-Anbieter-Oberfläche, wie beispielsweise den vSphere Client, um auf die Befehlszeile der bereitgestellten Maschine zuzugreifen.
- 5 Erteilen Sie dem privaten Schlüssel Leseberechtigungen.

```
chmod 600 key-name
```

- 6 Navigieren Sie zur vRealize Automation Cloud Assembly-Bereitstellung, wählen Sie die Maschine aus und klicken Sie auf **Aktionen > Privaten Schlüssel abrufen**.

- 7 Kopieren Sie die Datei des privaten Schlüssels auf Ihre lokale Maschine.

Ein typischer lokaler Dateipfad lautet `/home/username/.ssh/key-name`.

- 8 Öffnen Sie eine SSH-Remote-Sitzung und stellen Sie eine Verbindung zur bereitgestellten Maschine her.

```
ssh -i key-name user-name@machine-ip
```

Bereitstellen des eigenen öffentlichen/privaten Schlüsselpaars in vRealize Automation Cloud Assembly

Viele Unternehmen erstellen und verteilen eigene öffentliche/private Schlüsselpaare für die Authentifizierung.

Verwenden Sie den folgenden Code als Richtlinie.

- 1 Rufen Sie in der lokalen Umgebung Ihr öffentliches/privates Schlüsselpaar ab oder erzeugen Sie es.

Im Moment müssen Sie die Schlüssel nur lokal erzeugen und speichern.

- 2 Fügen Sie vor der Bereitstellung in vRealize Automation Cloud Assembly `remoteAccess-`Eigenschaften zum Blueprint hinzu, wie im Beispiel gezeigt.

`sshKey` enthält den langen alphanumerischen Wert, der innerhalb der öffentlichen Schlüsseldatei `key-name.pub` gefunden wurde.

Der Benutzername ist optional und wird erstellt, damit Sie sich anmelden können. Wenn Sie ihn weglassen, erzeugt das System eine zufällige ID als Benutzernamen.

Beispiel:

```
type: Cloud.Machine
properties:
  name: our-vm1
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: ssh-rsa Iq+5aQgBP3ZNT4o1baP5Ii+dstIcowRRkyobbfpA1mj9ts1f
qGxvU66PX9IeZax5hZvNWFgjw6ag+Z1zndOLhVdVoW49f274/mIRi1d7Uuw...
    username: testuser
```

- 3 Stellen Sie in vRealize Automation Cloud Assembly die Maschine aus dem zugehörigen Blueprint bereit und weisen Sie ihr den Status „Gestartet“ zu.
- 4 Greifen Sie mit dem Client des Cloud-Anbieters auf die bereitgestellte Maschine zu.
- 5 Fügen Sie die Datei des öffentlichen Schlüssels zum Stammordner auf der Maschine hinzu. Verwenden Sie den Schlüssel, den Sie in `remoteAccess.sshKey` angegeben haben.
- 6 Stellen Sie sicher, dass das Gegenstück zur privaten Schlüsseldatei auf Ihrem lokalen Computer vorhanden ist.

Der Schlüssel lautet in der Regel `/home/username/.ssh/key-name` ohne die PUB-Erweiterung.

- 7 Öffnen Sie eine SSH-Remote-Sitzung und stellen Sie eine Verbindung zur bereitgestellten Maschine her.

```
ssh -i key-name user-name@machine-ip
```

Bereitstellen eines AWS-Schlüsselpaars in vRealize Automation Cloud Assembly

Durch Hinzufügen eines AWS-Schlüsselpaarnamens zum Blueprint können Sie remote auf eine Maschine zugreifen, die von vRealize Automation Cloud Assembly für AWS bereitgestellt wird.

Beachten Sie, dass AWS-Schlüsselpaare regionsspezifisch sind. Wenn Sie Arbeitslasten in `us-east-1` bereitstellen, muss das Schlüsselpaar in `us-east-1` vorhanden sein.

Verwenden Sie den folgenden Code als Richtlinie. Diese Option steht nur für AWS-Cloud-Zonen zur Verfügung.

```
type: Cloud.Machine
properties:
  image: Ubuntu
  flavor: small
  remoteAccess: authentication: keyPairName keyPair: cas-test
constraints:
  - tag: 'cloud:aws'
```

Angeben eines Benutzernamens und Kennworts in vRealize Automation Cloud Assembly

Durch Hinzufügen eines Benutzernamens und Kennworts zum Blueprint erhalten Sie einfachen Remote-Zugriff auf eine Maschine, die von vRealize Automation Cloud Assembly für Azure bereitgestellt wird.

Obwohl die Remoteanmeldung mit Benutzername und Kennwort weniger sicher ist, kann sie in Ihrer Situation unter Umständen notwendig sein. Denken Sie daran, dass bestimmte Cloud-Anbieter oder Konfigurationen diese weniger sichere Option unter Umständen nicht unterstützen.

- 1 Fügen Sie vor der Bereitstellung in vRealize Automation Cloud Assembly `remoteAccess`-Eigenschaften zum Blueprint hinzu, wie im Beispiel gezeigt.

Legen Sie den Benutzernamen und das Kennwort für das Konto fest, mit dem Sie sich wahrscheinlich anmelden werden.

Beispiel:

```
type: Cloud.Machine
properties:
  name: our-vm3
  image: Linux18
  flavor: small
  remoteAccess: authentication: usernamePassword username: testuser password: admin123
```

- 2 Stellen Sie in vRealize Automation Cloud Assembly die Maschine aus dem zugehörigen Blueprint bereit und weisen Sie ihr den Status „Gestartet“ zu.
- 3 Navigieren Sie zur Schnittstelle Ihres Cloud-Anbieters und greifen Sie auf die bereitgestellte Maschine zu.
- 4 Erstellen oder aktivieren Sie das Konto auf der bereitgestellten Maschine.
- 5 Öffnen Sie über Ihren lokalen Computer eine Remotesitzung anhand der IP-Adresse oder des FQDN der bereitgestellten Maschine und melden Sie sich wie gewohnt mit dem Benutzernamen und Kennwort an.

Vorgehensweise zum Speichern verschiedener Versionen eines vRealize Automation Cloud Assembly-Blueprints

Als Blueprint-Entwickler können Sie einen Snapshot eines funktionierenden Blueprints sicher erfassen, bevor Sie weitere Änderungen vornehmen.

Zur Bereitstellungszeit können Sie eine der bereitzustellenden Versionen auswählen.

Vorgehensweise zum Erfassen einer Blueprint-Version

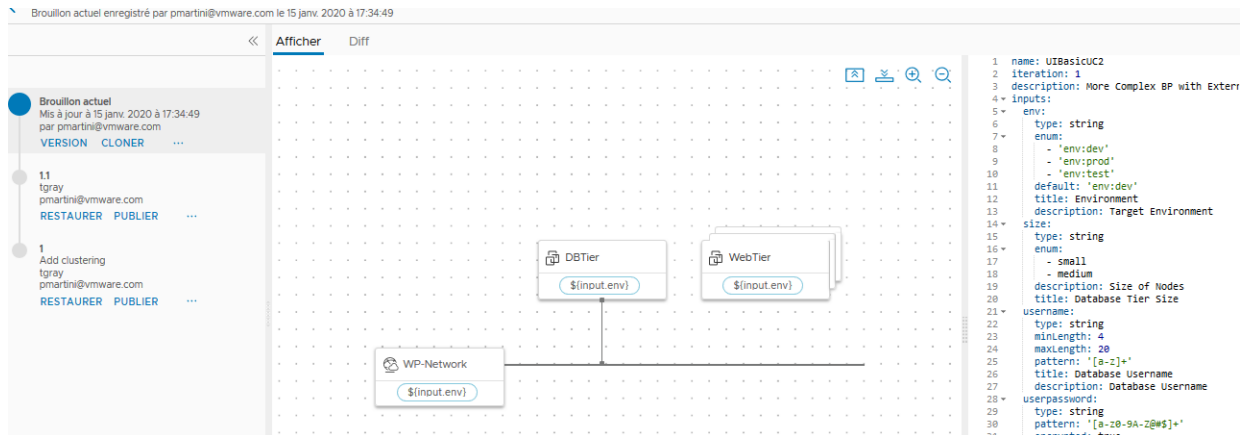
Klicken Sie auf der Entwurfsseite auf **Version** und geben Sie einen Namen an.

Der Name muss alphanumerisch sein und darf keine Leerzeichen enthalten. Nur Punkte, Bindestriche und Unterstriche sind als Sonderzeichen zulässig.

Vorgehensweise zum Wiederherstellen einer älteren Version

Klicken Sie auf der Entwurfsseite auf **Versionsverlauf**.

Wählen Sie auf der linken Seite eine ältere Version aus, um sie auf der Arbeitsfläche und im Code-Editor zu überprüfen. Wenn Sie die gewünschte Version gefunden haben, klicken Sie auf **Wiederherstellen**. Beim Wiederherstellen wird der aktuelle Entwurf überschrieben, ohne dass benannte Versionen entfernt werden.



Vorgehensweise zum Freigeben einer Version für Benutzer von vRealize Automation Service Broker

Klicken Sie auf der Entwurfsseite auf **Versionsverlauf**.

Wählen Sie auf der linken Seite eine Version aus und klicken Sie auf **Freigeben**. Sie können den aktuellen Entwurf erst nach der Versionierung freigeben.

Wenn mehr als eine Version eines Blueprints freigegeben wird, verwendet vRealize Automation Service Broker den neuesten Blueprint.

Vorgehensweise zum Vergleichen von Blueprint-Versionen

Wenn sich Änderungen und Versionen anhäufen, möchten Sie unter Umständen die Unterschiede zwischen ihnen ermitteln.

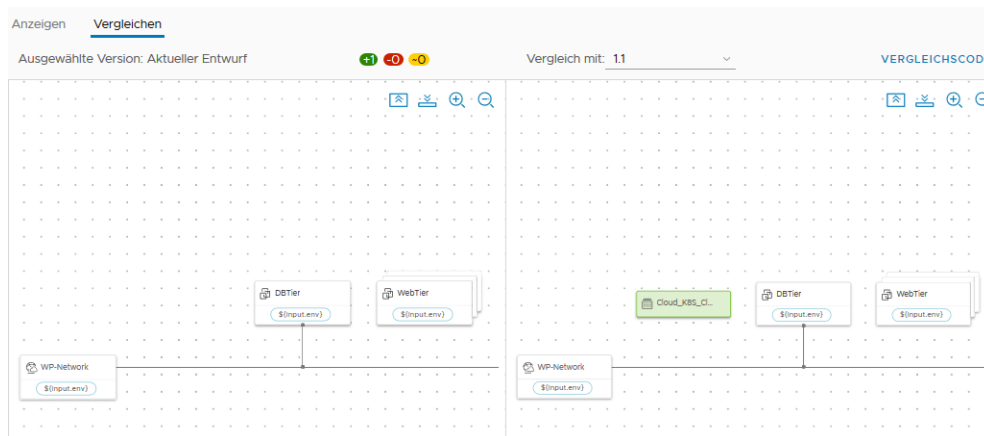
Wählen Sie in der Ansicht „Versionsverlauf“ eine Version aus und klicken Sie auf **Vergleichen**. Wählen Sie dann im Dropdown-Menü **Vergleichen mit** eine andere Version für den Vergleich aus.

Beachten Sie, dass Sie zwischen der Überprüfung von Code-Unterschieden oder Unterschieden in der visuellen Topologie umschalten können.

Abbildung 6-1. Code-Unterschiede

Anzeigen	Vergleichen
Ausgewählte Version: Aktueller Entwurf	Vergleich mit: 1.1 OPTISCH VERGLEICHE
<pre> 50 @@ -50,8 +50,16 @@ 51 maximum: 10 52 title: Wordpress Archive Disk Size 53 description: Size of Wordpress archive disk 54 resources: </pre>	<pre> 50 maximum: 10 51 title: Wordpress Archive Disk Size 52 description: Size of Wordpress archive disk 53 resources: 54 + Cloud_K8S_Cluster_1: 55 + type: Cloud.K8S.Cluster 56 + metadata: 57 + layoutPosition: 58 + - 0 59 + - 0 60 + properties: 61 + hostname: '' </pre>
<pre> 54 DBTier: 55 type: Cloud.Machine 56 metadata: 57 layoutPosition: </pre>	<pre> 62 DBTier: 63 type: Cloud.Machine 64 metadata: 65 layoutPosition: </pre>

Abbildung 6-2. Unterschiede in der visuellen Topologie



Vorgehensweise zum Klonen eines Blueprints

Sie können auf der Entwurfsseite mithilfe von **Aktionen > Klonen** eine Kopie des aktuellen Blueprints zur alternativen Entwicklung erstellen. Dieser Vorgang ist jedoch nicht mit dem Speichern einer Version gleichzusetzen.

Anpassen der Namen von bereitgestellten Ressourcen mithilfe von vRealize Automation Cloud Assembly

Als Cloud- oder Projektadministrator verfügen Sie über eine vorgeschriebene Benennungskonvention für Ressourcen in Ihrer Umgebung, und Sie möchten, dass die bereitgestellte Ressource diese Konvention ohne Benutzereingriff befolgt. Sie können eine Benennungsvorlage für alle Bereitstellungen aus einem vRealize Automation Cloud Assembly-Projekt erstellen.

Beispielsweise besteht Ihre Host-Benennungskonvention darin, eine Ressource mit einem Präfix wie *projectname-sitecode-costcenter-whereDeployed-identifier* zu versehen. Sie konfigurieren die benutzerdefinierte Benennungsvorlage für die Maschinen für jedes Projekt. Einige der Vorlagenvariablen werden während der Bereitstellung aus dem System abgerufen, andere basieren auf benutzerdefinierten Eigenschaften des Projekts.

Alle Ressourcennamen müssen eindeutig sein. Mit der Eigenschaft „inkrementelle Zahl“ können Sie die Eindeutigkeit gewährleisten. Die Zahlen werden für alle Bereitstellungen inkrementiert, einschließlich der von vRealize Automation Cloud Assembly benannten Bereitstellungen. Wenn Ihr System umfangreicher wird, wird die Nummerierung möglicherweise zufällig angezeigt, aber sie gewährleistet dennoch die Eindeutigkeit.

Neben den hier angegebenen Beispielen können Sie auch den Benutzernamen, das verwendete Image, andere integrierte Optionen und einfache Zeichenfolgen hinzufügen. Beim Erstellen der Vorlage werden Hinweise zu möglichen Optionen angegeben.

Beachten Sie, dass es sich bei einigen der angezeigten Werte nur um Anwendungsfallbeispiele handelt. Sie können diese nicht eins zu eins auf Ihre Umgebung übertragen. Überlegen Sie sich, wo Sie Ihre eigenen Ersetzungen vornehmen würden, oder übernehmen Sie eine Hochrechnung aus den Beispielwerten, um die Anforderungen an die Verwaltung Ihrer eigenen Cloud-Infrastruktur und -Bereitstellung zu erfüllen.

Voraussetzungen

- Sie müssen die Benennungskonvention kennen, die Sie für Bereitstellungen aus einem Projekt verwenden möchten.
- Bei diesem Verfahren wird davon ausgegangen, dass Sie einen einfachen Blueprint erstellt haben oder erstellen können, mit dem Sie Ihre benutzerdefinierte Host-Präfixbenennung testen.

Verfahren

- 1 Klicken Sie auf **Infrastruktur > Projekte**.
- 2 Wählen Sie ein vorhandenes Projekt aus oder erstellen Sie ein neues Projekt.
- 3 Suchen Sie auf der Registerkarte **Bereitstellung** den Abschnitt „Benutzerdefinierte Eigenschaften“ und erstellen Sie die Eigenschaften für den Site-Code und die Kostenstellenwerte.

Hier können Sie die angezeigten Werte durch die passenden Werte für Ihre Umgebung ersetzen.

Benutzerdefinierte Eigenschaften
Geben Sie die benutzerdefinierten Eigenschaften an, die allen Anforderungen in diesem Projekt hinzugefügt werden sollen. ①

Benutzerdefinierte Eigenschaften festlegen	Name	Wert
	siteCode	BGL
	costCenter	IT-research

Benutzerdefinierte Benennung
Geben Sie die Benennungsvorlage an, die für in diesem Projekt bereitgestellte Maschinen, Netzwerke, Sicherheitsgruppen und Festplatten verwendet werden soll.

Vorlage: `${project.name}-${resource.siteCode}-${resource` ①
Hint: Avoid conflicting names by generating digits in names. `${#####}`

- a Erstellen Sie eine benutzerdefinierte Eigenschaft mit dem Namen **siteCode** und dem Wert **BGL**.
 - b Fügen Sie eine weitere benutzerdefinierte Eigenschaft mit dem Namen **costCenter** und dem Wert **IT-Research** hinzu.
- 4 Suchen Sie den Abschnitt „Benutzerdefinierte Benennung“ und fügen Sie die folgende Vorlage hinzu.

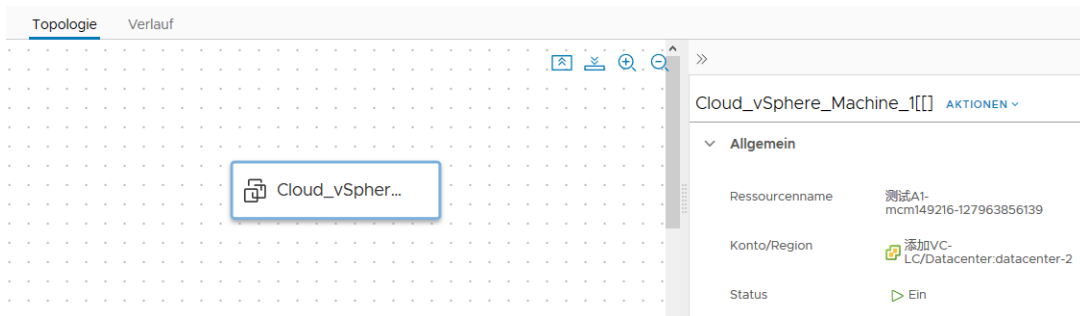
```
${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}
```

Sie können zwar die Zeichenfolge hineinkopieren, aber wenn es sich um Ihre erste Benennungsvorlage handelt, empfiehlt es sich unter Umständen, beim Erstellen der Vorlage den Hinweistext und die Schnellauswahl zu verwenden.

- 5 Stellen Sie einen Blueprint bereit, der dem Projekt zugeordnet ist, um zu überprüfen, ob der benutzerdefinierte Name auf die Ressource angewendet wird.
 - a Klicken Sie auf die Registerkarte **Blueprints** und dann auf einen Blueprint, der dem Projekt zugeordnet ist.
 - b Stellen Sie den Blueprint bereit.

Die Registerkarte **Bereitstellungen** wird geöffnet. Darauf wird Ihre in Bearbeitung befindliche Bereitstellung angezeigt.

- c Wenn die Bereitstellung abgeschlossen ist, klicken Sie auf den Namen der Bereitstellung.
- d Beachten Sie auf der Registerkarte **Topologie** im rechten Fensterbereich, dass Ihr benutzerdefinierter Name der Ressourcenname ist.



- 6 Wenn Sie einen Test-Blueprint zur Überprüfung der Benennungskonvention bereitgestellt haben, können Sie die Bereitstellung löschen.

Nächste Schritte

Erstellen Sie benutzerdefinierte Benennungsvorlagen für Ihre anderen Projekte.

Eigenschaften der vRealize Automation-Ressource

Mit dem vRealize Automation-Editor „Infrastruktur-als-Code“ können Sie auf die Hilfe zur Syntax und Codeergänzung klicken bzw. den Mauszeiger darüber halten. Den vollständigen Satz von Blueprint-Ressourceneigenschaften, die manchmal als benutzerdefinierte Eigenschaften bezeichnet werden, finden Sie im konsolidierten Ressourcenschema.

Das Schema ist auf der VMware {code}-Website verfügbar. Folgen Sie dem Link und klicken Sie auf **Modelle**, um die Ressourcenobjekte aufzuführen, die für Blueprints verfügbar sind.

- [vRealize Automation-Ressourcentypschema in VMware {code}](#)

Beispiele für Blueprint-Code

Die Kombinations- und Anwendungsmöglichkeiten von Blueprint-Code in vRealize Automation Cloud Assembly sind nahezu unbegrenzt.

Erfolgreicher Code stellt häufig den besten Ausgangspunkt für die weitere Entwicklung dar. Wenn Sie sich an einem Beispiel orientieren, ersetzen Sie Ressourcennamen, Werte usw., um Ihre Site-Einstellungen anzuwenden.

Beispiele für vSphere-Maschinen in vRealize Automation Cloud Assembly-Blueprints

In diesen allgemeinen Beispielen werden vSphere-Ressourcen innerhalb von vRealize Automation Cloud Assembly-Blueprints definiert.

Ressource	Blueprint – Beispiel
Virtuelle vSphere-Maschine mit CPU, Arbeitsspeicher und Betriebssystem	<pre>resources: demo-machine: type: Cloud.vSphere.Machine properties: name: demo-machine cpuCount: 1 totalMemoryMB: 1024 image: ubuntu</pre>
vSphere-Maschine mit einer Datenspeicherressource	<pre>resources: demo-vsphere-disk-001: type: Cloud.vSphere.Disk properties: name: DISK_001 type: 'HDD' capacityGb: 10 dataStore: 'datastore-01' provisioningType: thick</pre>
vSphere-Maschine mit angehängter Festplatte	<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: >- https://bintray.com/vmware/photon/ download_file?file_path=2.0%2FRC%2Fova%2Fphoton- custom-hw11-2.0-315b961.ova attachedDisks: - source: '\${demo-vsphere-disk-001.id}'</pre>
vSphere-Maschine von einem Linked Clone-Image (Schrägstrich und Name des Snapshots anhängen)	<pre>resources: demo-machine: type: Cloud.vSphere.Machine properties: imageRef: 'demo-machine/snapshot-01' cpuCount: 1 totalMemoryMB: 1024</pre>

Ressource	Blueprint – Beispiel
vSphere-Maschine in einem bestimmten Ordner in vCenter	<pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: name: demo-machine cpuCount: 2 totalMemoryMB: 1024 imageRef: ubuntu resourceGroupName: 'myFolder' </pre>
vSphere-Maschine mit mehreren Netzwerkkarten	<pre> resources: demo-machine: type: Cloud.Machine properties: image: ubuntu flavor: small networks: - name: '\${network-01.name}' deviceIndex: 0 - name: '\${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>
vSphere-Maschine aus einem Snapshot-Image	<pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: imageRef: 'demo-machine/snapshot-01' cpuCount: 1 totalMemoryMB: 1024 </pre>
vSphere-Maschine mit einem angehängten Tag in vCenter	<pre> resources: demo-machine: type: Cloud.Machine properties: flavor: small image: ubuntu tags: - key: env value: demo </pre>
vSphere-Maschine mit einer Anpassungsspezifikation	<pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: name: demo-machine image: ubuntu flavor: small customizationSpec: Linux </pre>

Ressource	Blueprint – Beispiel
vSphere-Maschine mit einer vSphere-Netzwerkkomponente und statischer IP-Adresse	<pre> resources: demo-network: type: Cloud.vSphere.Network properties: name: demo-network demo-machine: type: Cloud.vSphere.Machine properties: image: ubuntu flavor: small networks: - name: demo-network assignment: static </pre>
vSphere-Maschine mit Remotezugriff	<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: >- 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" >> /etc/sudoers.d/\${input.username} </pre>

Dokumentiertes Beispiel für einen vRealize Automation Cloud Assembly-Blueprint

Mithilfe der umfassenden in diesem Beispiel enthaltenen Kommentare können Sie die Struktur und den Zweck der Abschnitte in einem vRealize Automation Cloud Assembly-Blueprint überprüfen.

```
# *****
#
# This WordPress blueprint is enhanced with comments to explain its
# parameters.
#
# Try cloning it and experimenting with its YAML code. If you're new to
# YAML, visit yaml.org for general information.
#
# The blueprint deploys a minimum of 3 virtual machines and runs scripts
# to install packages.
#
# *****
#
# -----
# Blueprints need a descriptive name and version if
# source controlled in git.
# -----
name: WordPress Blueprint 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 blueprint 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:
            - name: '${resource.WP_Network.name}'
              network: '${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 extensibility 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 extensiblity 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: '${"env:" + to_lower(input.platform)}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
#     networks:
#       - name: '${resource.WP_Network.name}'
#         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 blueprint useful. Note that
# you can also access an API to create blueprints, 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  
#  
# *****
```

Beispiele für Netzwerke, Sicherheit und Lastausgleichsdienste in vRealize Automation Cloud Assembly-Blueprints

Diese Blueprint-Codebeispiele veranschaulichen einige grundlegende Netzwerk-, Sicherheits- und Lastausgleichsdienst-Konfigurationen.

Beispiele detaillierter Implementierungsszenarien für Netzwerk und Sicherheit finden Sie in den folgenden VMware-Blogs:

- [vRealize Automation Cloud Assembly-Lastausgleichsdienst mit NSX-T Deep Dive](#)
- [Netzwerkautomatisierung mit Cloud Assembly und NSX – Teil 1](#) (beinhaltet die Verwendung von NSX-T- und vCenter-Cloud-Konten und Netzwerk-CIDR)
- [Netzwerkautomatisierung mit Cloud Assembly und NSX – Teil 2](#) (beinhaltet die Verwendung vorhandener und ausgehender Netzwerktypen)
- [Netzwerkautomatisierung mit Cloud Assembly und NSX – Teil 3](#) (beinhaltet die Verwendung vorhandener und bedarfsgesteuerter Sicherheitsgruppen)
- [Netzwerkautomatisierung mit Cloud Assembly und NSX – Teil 4](#) (beinhaltet die Verwendung vorhandener und bedarfsgesteuerter Lastausgleichsdienste)

Eine vollständige Zusammenfassung aller Blueprint-Schema-Optionen finden Sie unter [vRealize Automation-Ressourcentypschemata](#).

Weitere Informationen zu Netzwerktypen finden Sie unter [Verwenden von Netzwerken und Netzwerkprofilen in vRealize Automation Cloud Assembly](#).

Ressourcenszenario	Code eines Beispiel-Blueprints
vSphere-Maschine mit mehreren Netzwerkkarten	<pre> resources: demo-machine: type: Cloud.vSphere.Machine properties: image: ubuntu flavor: small networks: - name: '\${network-01.id}' deviceIndex: 0 - name: '\${network-02.id}' deviceIndex: 1 Cloud_vSphere_Network_1: type: Cloud.vSphere.Network properties: networkType: existing name: network-01 Cloud_vSphere_Network_2: type: Cloud.NSX.Network properties: networkType: existing name: network-02 </pre>
Public Cloud-Maschine zur Verwendung interner IP-Adressen anstelle öffentlicher IP-Adressen	<pre> resources: wf_proxy: type: Cloud.Machine properties: image: ubuntu 16.04 flavor: small constraints: - tag: 'platform:vsphere' networks: - name: '\${resource.wf_net.id}' assignPublicIpAddress: false </pre>
Geroutetes Netzwerk für NSX-V oder NSX-T unter Verwendung des NSX-Netzwerkkomponententyps	<pre> Cloud_NSX_Network_1: type: Cloud.NSX.Network properties: networkType: routed </pre>
<p>Taggen von logischen NSX-Switches für ein ausgehendes Netzwerk</p> <p>Weitere Informationen zu diesem Szenario finden Sie im Community-Blogbeitrag Erstellen von Tags in NSX mit Cloud Assembly.</p>	<pre> Cloud_NSX_Network_1: type: Cloud.NSX.Network properties: networkType: outbound tags: - key: app value: opencart </pre>

Ressourcenszenario	Code eines Beispiel-Blueprints
<p>Vorhandene Sicherheitsgruppe mit einem Einschränkungstags, das auf eine Maschinen-Netzwerkkarte (NIC) angewendet wurde</p> <p>Um eine vorhandene Sicherheitsgruppe zu verwenden, geben Sie <i>Vorhanden</i> für die Eigenschaft <code>securityGroupType</code> in der Komponente <code>Cloud.SecurityGroup</code> ein.</p> <p>Um eine bedarfsgesteuerte Sicherheitsgruppe zu erstellen, geben Sie <i>Neu</i> für die Eigenschaft <code>securityGroupType</code> in der Komponente <code>Cloud.SecurityGroup</code> ein.</p> <p>Sie können einer <code>Cloud.SecurityGroup</code>-Komponente Tags zuweisen, um vorhandene Sicherheitsgruppen mithilfe von Tag-Einschränkungen zuzuteilen. Sicherheitsgruppen, die keine Tags enthalten, können im Blueprint nicht verwendet werden.</p>	<pre>formatVersion: 1 inputs: {} resources: allowSsh_sg: type: Cloud.SecurityGroup properties: securityGroupType: Vorhanden 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>

Ressourcenszenario	Code eines Beispiel-Blueprints
<p>Bedarfsgesteuertes Netzwerk mit einem 1-armigen Lastausgleichsdienst</p>	<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>
<p>Vorhandenes Netzwerk mit einem Lastausgleichsdienst</p>	<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>

Ressourcenszenario	Code eines Beispiel-Blueprints
	<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>

Puppet-fähiger Blueprint mit Zugriff auf Benutzername und Kennwort

In diesem Beispiel fügen Sie Puppet-Konfigurationsverwaltung zu einem Blueprint hinzu, der auf einer vCenter-Computing-Ressource mit Zugriff auf Benutzernamen und Kennwort bereitgestellt wird.

Dieses Verfahren zeigt ein Beispiel dafür, wie Sie eine Puppet-fähige einsetzbare Ressource erstellen können, die eine Authentifizierung von Benutzername und Kennwort erfordert. Der Zugriff auf Benutzername und Kennwort bedeutet, dass sich der Benutzer manuell von der Computing-Ressource bei der primären Puppet-Maschine anmelden muss, um die Verwaltung der Puppet-Konfiguration aufzurufen.

Optional können Sie RAS-Authentifizierung konfigurieren, die die Konfigurationsverwaltung in einem Blueprint einrichtet, sodass die Computing-Ressource die Authentifizierung mit der primären Puppet-Maschine verarbeitet. Wenn der Remotezugriff aktiviert ist, generiert die Computing-Ressource automatisch einen Schlüssel für die Kennwortauthentifizierung. Ein gültiger Benutzername ist weiterhin erforderlich.

Weitere Beispiele dafür, wie Sie verschiedene Puppet-Szenarien in vRealize Automation Cloud Assembly-Blueprints konfigurieren können, finden Sie unter [Blueprint-Beispiele für die Puppet-Konfigurationsverwaltung in AWS](#) und [vCenter – Beispiele für Puppet-Konfigurations-Blueprints](#).

Voraussetzungen

- Richten Sie eine Puppet Enterprise-Instanz in einem gültigen Netzwerk ein.
- Fügen Sie mithilfe der Integrations-Funktion Ihre Puppet Enterprise-Instanz vRealize Automation Cloud Assembly hinzu. Weitere Informationen finden Sie unter [Konfigurieren der Puppet Enterprise-Integration in vRealize Automation Cloud Assembly](#).
- Richten Sie ein vSphere-Konto und eine vCenter-Computing-Ressource ein.

Verfahren

- 1 Fügen Sie eine Verwaltungskomponente für die Puppet-Konfiguration zu einer vSphere-Computing-Ressource auf der Arbeitsfläche für den gewünschten Blueprint hinzu.
 - a Wählen Sie **Infrastruktur > Verwalten > Integrationen** aus.
 - b Klicken Sie auf **Integration hinzufügen** und wählen Sie „Puppet“ aus.
 - c Geben Sie auf der Seite „Puppet-Konfiguration“ die entsprechenden Informationen ein.

Konfiguration	Beschreibung	Beispielwert
Hostname	Der Hostname oder die IP-Adresse der primären Puppet-Maschine.	Puppet-Ubuntu
SSH-Port	SSH-Port für die Kommunikation zwischen vRealize Automation Cloud Assembly und der primären Puppet-Maschine. (Optional)	–
Geheimer Schlüssel für automatische Signierung	Der auf der primären Puppet-Maschine konfigurierte gemeinsame geheime Schlüssel, der von Knoten zur Unterstützung von Zertifikatsanforderungen für automatische Signierung bereitgestellt werden soll.	Benutzerspezifisch
Speicherort	Geben Sie an, ob sich die primäre Puppet-Maschine in einer Private Cloud oder Public Cloud befindet. Hinweis Die cloud-übergreifende Bereitstellung wird nur unterstützt, wenn eine Verbindung zwischen der Computing-Ressource der Bereitstellung und der primären Puppet-Maschine besteht.	
Cloud proxy	Nicht erforderlich für Public Cloud-Konten, wie zum Beispiel Microsoft Azure oder Amazon Web Services. Wenn Sie ein vCenter-basiertes Cloud-Konto verwenden, wählen Sie den entsprechenden cloud proxy für Ihr Konto aus.	–
Benutzername	SSH- und RBAC-Benutzername für primäre Puppet-Maschine.	Benutzerspezifisch. YAML-Wert ist '\$ {input.username}'
Kennwort	SSH- und RBAC-Kennwort für primäre Puppet-Maschine.	Der benutzerspezifische YAML-Wert ist '\$ {input.password}'
Sudo-Befehle für diesen Benutzer verwenden	Wählen Sie diese Option aus, um sudo-Befehle für procidd zu verwenden.	true

Konfiguration	Beschreibung	Beispielwert
Name	Name der primären Puppet-Maschine	PEMasterOnPrem
Beschreibung		

- 2 Fügen Sie die Eigenschaften für Benutzername und Kennwort zur Puppet-YAML hinzu, wie im folgenden Beispiel gezeigt.
- 3 Stellen Sie sicher, dass der Wert für die RAS-Eigenschaft der Puppet-YAML auf `authentication: username and password` festgelegt ist, wie im folgenden Beispiel gezeigt.

Beispiel: YAML-Code für vCenter-Benutzername und -Kennwort

Das folgende Beispiel zeigt den repräsentativen YAML-Code für das Hinzufügen der Benutzernamen- und Kennwortauthentifizierung für eine vCenter-Computing-Ressource.

```
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}'
```

Blueprint-Beispiele für die Puppet-Konfigurationsverwaltung in AWS

Für die Konfiguration von Blueprints zur Unterstützung der Puppet-basierten Konfigurationsverwaltung in AWS-Computing-Ressourcen stehen mehrere Möglichkeiten zur Verfügung.

Puppet-Verwaltung in AWS mit Benutzername und Kennwort

Beispiel für...	Beispiel einer Blueprint-YAML
<p>Authentifizierung der Cloud-Konfiguration auf einem beliebigen unterstützten Amazon-Maschinen-Image.</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" >> /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>Authentifizierung der Cloud-Konfiguration auf einem benutzerdefinierten Amazon-Maschinen-Image mit einem vorhandenen Benutzer.</p>	<pre> inputs: username: type: string title: Username default: puppet password: type: string title: Password encrypted: true default: VMware@123 </pre>

Beispiel für...	Beispiel einer Blueprint-YAML
	<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>

Puppet-Verwaltung in AWS mit erzeugtem PublicPrivateKey

Beispiel für...	Beispiel einer Blueprint-YAML
remoteAccess.authentication- Authentifizierung in AWS mit generatedPublicPrivateKey- Zugriff	<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>

vCenter – Beispiele für Puppet-Konfigurations-Blueprints

Es gibt mehrere Optionen zum Konfigurieren von Blueprints zur Unterstützung der Puppet-basierten Konfigurationsverwaltung auf vCenter-Computing-Ressourcen.

Puppet auf vSphere mit Benutzernamen- und Kennwortauthentifizierung.

Das folgende Beispiel zeigt den YAML-Code für Puppet auf einer vSphere-OVA mit Benutzernamen- und Kennwortauthentifizierung.

Tabelle 6-2.

Beispiel für...	Beispiel einer Blueprint-YAML
<p>YAML-Code für Puppet auf einer vSphere-OVA mit Benutzernamen- und Kennwortauthentifizierung.</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: >- 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 AAAAB3NzaClyc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX dmettem@dmettem-m01.vmware.com runcmd: - echo "Defaults:\${input.username} </pre>
<p>YAML-Code für Puppet auf einer vSphere-OVA mit Benutzernamen- und Kennwortauthentifizierung auf der Computing-Ressource.</p>	<pre> inputs: username: type: string title: Username default: puppet </pre>

Tabelle 6-2. (Fortsetzung)

Beispiel für...	Beispiel einer Blueprint-YAML
YAML-Code für Puppet auf einem vCenter mit aktiviertem RAS-Authentifizierungskennwort für die Computing-Ressource.	<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: >- 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>
	<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>

Tabelle 6-2. (Fortsetzung)

Beispiel für...	Beispiel einer Blueprint-YAML
	<pre> 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}' </pre>

Puppet auf vSphere mit generierter PublicPrivateKey-Authentifizierung

Tabelle 6-3.

Beispiel für...	Beispiel einer Blueprint-YAML
YAML-Code für Puppet auf einer vSphere-OVA mit generierter PublicPrivateKey-Authentifizierung auf der Computing-Ressource.	<pre> inputs: {} resources: Machine: 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: 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>

Vorgehensweise zum Verwenden des vRealize Automation Cloud Assembly-Marketplace

Zum Schnellstarten Ihrer Ressourcenbibliothek laden Sie Dateien aus dem vRealize Automation Cloud Assembly-Marketplace herunter.

Der Marketplace stellt fertige Blueprints und Open Virtualization-Images bereit, die in [VMware Solution Exchange](#) verwaltet werden. Solution Exchange-Dateien werden mit `cloud assembly` gekennzeichnet und auf der vRealize Automation Cloud Assembly-Registerkarte „Marketplace“ angezeigt.

Zugriff auf den Marketplace

In vRealize Automation Cloud Assembly wählen Sie **Infrastruktur > Verbindungen > Integrationen** aus. Klicken Sie auf **Integration hinzufügen**, dann auf **My VMware** und stellen Sie die Anmeldedaten für Ihr My VMware-Konto bereit.

Vorgehensweise zum Herunterladen und Verwenden von Marketplace-Blueprint-Dateien

Klicken Sie auf der Registerkarte **Marketplace** auf **Abrufen** und akzeptieren Sie die Blueprint-Lizenzbedingungen. Anschließend können Sie den Blueprint zu einem vRealize Automation Cloud Assembly-Projekt hinzufügen oder einfach herunterladen. Sie können einen Blueprint auf der Registerkarte **Blueprints** hochladen.

Stellen Sie sich für ein projektbasiertes Beispiel vor, dass Sie Projektadministrator für ein Big-Data-Projekt sind. Zur Unterstützung Ihres Teams suchen Sie nach einem Marketplace-Hadoop-Blueprint, den Sie dem Teamprojekt hinzufügen. Anschließend passen Sie den Blueprint für Ihre Ressourcenumgebung an und geben ihn frei. Dann importieren Sie den Blueprint in den vRealize Automation Service Broker-Katalog, damit er vom Team bereitgestellt werden kann.

Vorgehensweise zum Herunterladen und Verwenden von Marketplace-Image-Dateien

Klicken Sie auf der Registerkarte **Marketplace** auf **Abrufen** und akzeptieren Sie die Lizenzbedingungen der OVF- oder OVA-Datei. Anschließend können Sie das OVF- oder das OVA-Image herunterladen und im Blueprint-Code darauf verweisen.

In Fortführung des vorherigen Beispiels benötigt Ihr Team möglicherweise Zugriff auf eine Hadoop-Version. Sie können eine Hadoop-OVF-Datei herunterladen und zu Cloud-Kontoressourcen wie einer vCenter Server-Inhaltsbibliothek hinzufügen. Anschließend aktualisieren Sie den gesamten Blueprint-Code, der auf das OVF-Image verweisen muss.

Vorgehensweise zum Erweitern und Automatisieren der Lebenszyklen von Anwendungen mit Erweiterbarkeit

Sie können die Lebenszyklen Ihrer Anwendungen unter Verwendung von Erweiterbarkeitsaktionen oder vRealize Orchestrator-Workflows mit Erweiterbarkeitsabonnements verlängern.

Mithilfe der vRealize Automation Cloud Assembly-Erweiterbarkeit können Sie einem Ereignis unter Verwendung von Abonnements eine Erweiterbarkeitsaktion oder einen vRealize Orchestrator-Workflow zuweisen. Wenn das angegebene Ereignis auftritt, initiiert das Abonnement die auszuführende Aktion oder den auszuführenden Workflow, und alle Abonnenten werden benachrichtigt.

Erweiterbarkeitsaktionen

Bei Erweiterbarkeitsaktionen handelt es sich um kleine, einfache Code-Skripts zur Angabe und Durchführung einer Aktion. Sie können Erweiterbarkeitsaktionen aus vordefinierten vRealize Automation Cloud Assembly-Aktionsvorlagen oder aus einer ZIP-Datei importieren. Zum Erstellen von benutzerdefinierten Skripten für Ihre Erweiterbarkeitsaktionen können Sie auch den Aktions-Editor verwenden. Durch die Verknüpfung mehrerer Aktionssskripts in einem Skript erstellen Sie einen Aktionsablauf. Mithilfe von Aktionsabläufen können Sie eine Abfolge von Aktionen erstellen. Informationen zur Verwendung von Aktionsabläufen finden Sie unter [Definition eines Aktionsablaufs](#).

vRealize Orchestrator-Workflows

Durch die Integration von vRealize Automation Cloud Assembly in Ihre vorhandene vRealize Orchestrator-Umgebung können Sie Workflows in Ihren Erweiterbarkeitsabonnements verwenden.

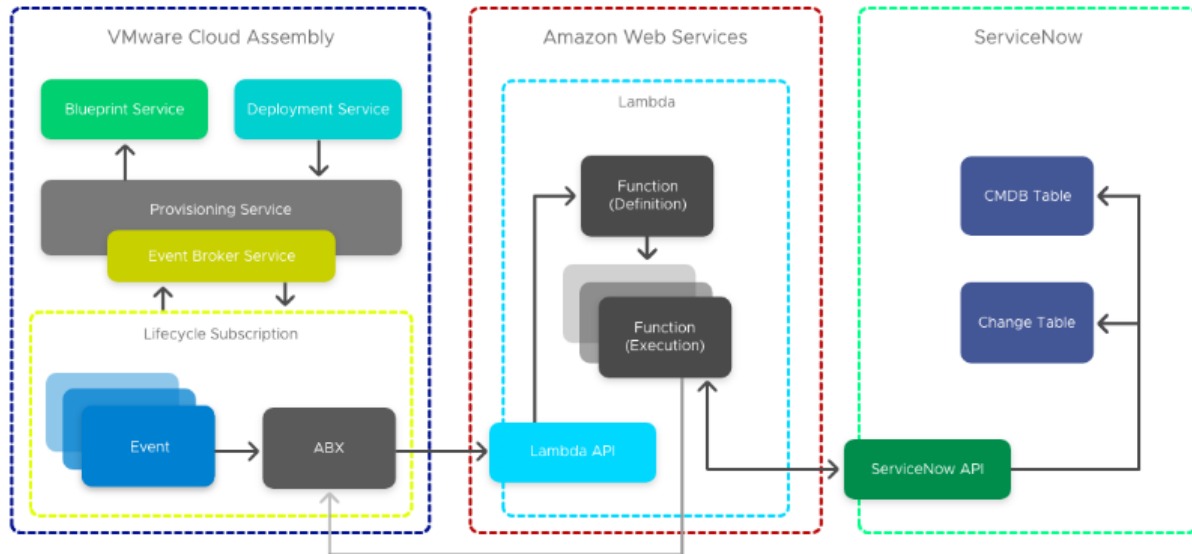
Abonnements für Erweiterbarkeitsaktionen

Sie können einem vRealize Automation Cloud Assembly-Abonnement eine Erweiterbarkeitsaktion zuweisen, um den Lebenszyklus Ihrer Anwendung zu verlängern.

Hinweis Bei den folgenden Abonnements handelt es sich um Anwendungsbeispiele. Sie umfassen nicht alle Funktionen von Erweiterbarkeitsaktionen.

Vorgehensweise zum Integrieren von Cloud Assembly in ServiceNow unter Verwendung von Erweiterbarkeitsaktionen

Mithilfe von Erweiterbarkeitsaktionen können Sie vRealize Automation Cloud Assembly mit einer ITSM-Unternehmenslösung wie ServiceNow integrieren.

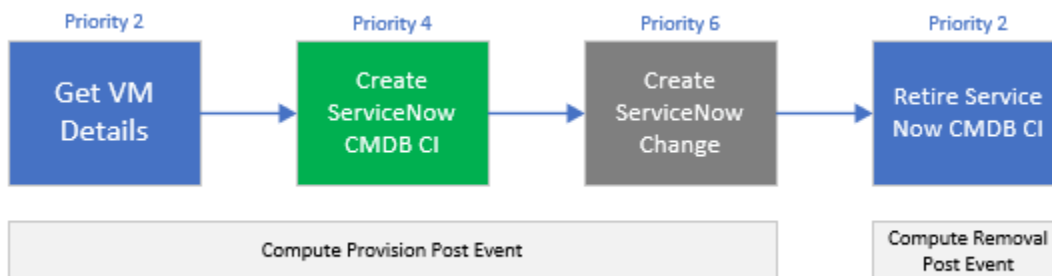


Unternehmensbenutzer integrieren ihre Cloud Management Plattform zu Übereinstimmungszwecken in der Regel mit einer ITSM- (IT Service Management) und einer CMDB-Plattform (Configuration Management Database). Gemäß diesem Beispiel können Sie vRealize Automation Cloud Assembly unter Verwendung von Erweiterbarkeitsaktionsskripts in ServiceNow für CMDB und ITSM integrieren.

Hinweis Sie können ServiceNow auch unter Verwendung von vRealize Orchestrator-Workflows in vRealize Automation Cloud Assembly integrieren. Informationen zum Integrieren von ServiceNow mithilfe von Workflows finden Sie unter [Vorgehensweise zum Integrieren von Cloud Assembly für ITSM mit ServiceNow unter Verwendung von vRealize Orchestrator-Workflows](#).

Zum Erstellen dieser Integration verwenden Sie vier Erweiterbarkeitsaktionsskripts. Die ersten drei Skripts werden nacheinander während der Bereitstellung beim Compute-Bereitstellungs-Post-Ereignis initiiert. Das vierte Skript löst das Compute-Entfernungs-Post-Ereignis aus.

Weitere Informationen zu Ereignisthemen finden Sie unter [Mit vRealize Automation Cloud Assembly bereitgestellte Ereignisthemen](#).



VM-Details abrufen

Das Skript zum Abrufen von VM-Details erfasst zusätzliche Nutzlastdetails, die für die CI-Erstellung erforderlich sind, sowie ein Identitätstoken, das in Amazon Web Services Systems Manager Parameter Store (SSM) gespeichert ist. Außerdem aktualisiert dieses Skript `customProperties` mit zusätzlichen Eigenschaften für die spätere Verwendung.

ServiceNow-CMDB-Konfigurationselement erstellen

Das Skript zum Erstellen des ServiceNow-CMDB-Konfigurationselements übergibt die Instanz-URL von ServiceNow als Eingabe und speichert die Instanz in SSM, um die Sicherheitsanforderungen zu erfüllen. Dieses Skript liest auch die Antwort der eindeutigen Datensatz-ID (`sys_id`) der ServiceNow-CMDB. Es übergibt die Antwort als Ausgabe und schreibt die benutzerdefinierte Eigenschaft `serviceNowSysId` während der Erstellung. Dieser Wert wird verwendet, um das Konfigurationselement als veraltet zu markieren, wenn die Instanz gelöscht wird.

Hinweis Unter Umständen müssen Ihrer Amazon Web Services-Rolle für vRealize Automation services zusätzliche Berechtigungen zugewiesen werden, damit Lambda auf den SSM-Parameterspeicher zugreifen kann.

ServiceNow-Änderung erstellen

Dieses Skript beendet die ITSM-Integration, indem die Instanz-URL von ServiceNow als Eingabe übergeben und die ServiceNow-Anmeldedaten als SSM gespeichert werden, um die Sicherheitsanforderungen zu erfüllen.

ServiceNow-Änderung erstellen

Das Skript zum Zurückziehen des ServiceNow-CMDB-Konfigurationselements fordert ServiceNow zum Anhalten auf und markiert das Konfigurationselement basierend auf der benutzerdefinierten Eigenschaft `serviceNowSysId`, die im Erstellungsskript erstellt wurde, als veraltet.

Voraussetzungen

- Bevor Sie diese Integration konfigurieren, filtern Sie alle Ereignisabonnements mit der bedingten Blueprint-Eigenschaft: `event.data["customProperties"]`
`["enable_servicenow"] == "true"`

Hinweis Diese Eigenschaft ist in Blueprints vorhanden, die eine ServiceNow-Integration benötigen.

- Installierte Python-Anwendung.

Weitere Informationen zum Filtern von Abonnements finden Sie unter [Erstellen eines Erweiterbarkeitsabonnements](#).

Verfahren

- 1 Öffnen Sie eine Eingabeaufforderung auf Ihrer virtuellen Maschine.

2 Führen Sie das Skript zum Abrufen von VM-Details aus.

```
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 Führen Sie die Aktion zur Erstellung des CMDB-Konfigurationselements aus.

```
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 Führen Sie das Erstellungsaktionsskript aus.

```

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)

```

Ergebnisse

vRealize Automation Cloud Assembly wurde erfolgreich in ITSM-Lösung ServiceNow integriert.

Nächste Schritte

Wenn Sie möchten, können Sie das Konfigurationselement mithilfe der Aktion zum Zurückziehen des CMDB-Konfigurationselements als veraltet markieren:

```

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)

```

Weitere Informationen zur Verwendung von Erweiterbarkeitsaktionen für die Integration von ServiceNow in vRealize Automation Cloud Assembly finden Sie im Blogbeitrag [Extending Cloud Assembly with Action Based Extensibility for ServiceNow Integration](#), der die Erweiterung von Cloud Assembly mit aktionsbasierter Erweiterbarkeit für die ServiceNow-Integration behandelt.

Vorgehensweise zum Taggen virtueller Maschinen während der Bereitstellung mithilfe von Erweiterbarkeitsaktionen

Sie können Erweiterbarkeitsaktionen in Verbindung mit Abonnements verwenden, um das Tagging von VMs zu automatisieren und zu vereinfachen.

Als Cloud-Administrator können Sie Bereitstellungen, die automatisch mit bestimmten Eingaben und Ausgaben gekennzeichnet werden, unter Verwendung von Erweiterbarkeitsaktionen und -abonnements erstellen. Wenn eine neue Bereitstellung für das Projekt erstellt wird, das das Abonnement zum Kennzeichnen der VM enthält, löst das Bereitstellungsereignis die Ausführung des Skripts „VM kennzeichnen“ aus und die Tags werden automatisch angewendet. Dies spart Zeit und fördert die Effizienz bei gleichzeitiger Vereinfachung der Bereitstellungsverwaltung.

Voraussetzungen

- Zugriff auf Anmeldedaten für Cloud-Administrator.
- Amazon Web Services-Rolle für Lambda-Funktionen.

Verfahren

- 1 Navigieren Sie zu **Erweiterbarkeit > Bibliothek > Aktionen > Neue Aktion** und erstellen Sie eine Aktion mit den folgenden Parametern.

Parameter	Beschreibung
Aktionsname	Name der Erweiterbarkeitsaktion, vorzugsweise mit „TagVM“ als Präfix oder Suffix.
Projekt	Projekt zum Testen der Erweiterbarkeitsaktion.
Aktionsvorlage	VM kennzeichnen
Laufzeit	Python
Skriptquelle	Skript schreiben

- 2 Geben Sie **Handler** als **Hauptfunktion** ein.
- 3 Fügen Sie Tagging-Eingaben zum Testen der Erweiterbarkeitsaktion hinzu.
Beispiel: resourceNames = ["DB_VM"] und Ziel = world.
- 4 Klicken Sie auf **Speichern**, um Ihre Änderungen zu speichern.
- 5 Klicken Sie auf **Test**, um die Aktion zu testen.
- 6 Klicken Sie auf **Schließen**, um den Aktions-Editor zu beenden.
- 7 Navigieren Sie zu **Erweiterbarkeit > Abonnements**.
- 8 Klicken Sie auf **Neues Abonnement**.

9 Geben Sie die folgenden Abonnementdetails ein.

Details	Einstellung
Ereignisthema	Wählen Sie ein Ereignisthema aus, das sich auf die Kennzeichnungsphase der VM bezieht. Beispiel: Computing-Zuteilung. Hinweis Tags müssen Teil des Abonnementschemas sein.
Wird blockiert	Legen Sie als Zeitüberschreitung für das Abonnement 1 Minute fest.
Ausführbarer Elementtyp	Wählen Sie einen ausführbaren Typ der Erweiterbarkeitsaktion aus.
Ausführbare ID	Wählen Sie Ihre benutzerdefinierte Erweiterbarkeitsaktion aus.

- Klicken Sie auf **Erstellen**, um Ihr benutzerdefiniertes Abonnement für Erweiterbarkeitsaktionen zu speichern.
- Erstellen Sie unter **Neuer Blueprint** einen neuen Blueprint mit zwei virtuellen Maschinen: Anwendungs-VM und Datenbank-VM.

The screenshot displays the vRealize Automation Cloud Assembly interface. The top navigation bar includes 'Bereitstellungen', 'Blueprints', 'Infrastruktur', 'Erweiterbarkeit', and 'Marketplace'. The 'Blueprints' tab is active, showing the 'ABX' blueprint. The left sidebar lists various resource categories: Cloud Agnostic (Machine, Load Balancer, Network, Security Group, Volume), Kubernetes (K8S Cluster, K8S Namespace), vSphere (Machine, Disk, Network), NSX (Load Balancer, Network), and AWS. The main canvas shows two VMs, 'Application_VM' and 'DB_VM', being added to the blueprint. The right pane shows the 'Code' tab with the following JSON configuration:

```

1 inputs: {}
2 resources:
3   Application_VM:
4     type: Cloud.Machine
5     properties:
6       image: 'ubuntu'
7       flavor: 'small'
8   DB_VM:
9     type: Cloud.Machine
10    properties:
11      image: 'ubuntu'
12      flavor: 'small'
13

```

At the bottom, there are buttons for 'BEREITSTELLEN', 'TESTEN', 'VERSION', and 'SCHLIESSEN'. A status message indicates 'Zuletzt gespeichert vor 3 Minuten'.

- Klicken Sie auf **Bereitstellen**, um die VMs bereitzustellen.
- Stellen Sie während der Bereitstellung sicher, dass das Ereignis initiiert und die Erweiterbarkeitsaktion ausgeführt wird.

- 14 Um zu überprüfen, ob die Tags korrekt angewendet wurden, navigieren Sie zu **Infrastruktur > Ressourcen > Maschinen**.

Weitere Informationen zu Erweiterbarkeitsaktionen

Bei der aktionsbasierten Erweiterbarkeit (ABX) werden optimierte Codeskripts in vRealize Automation Cloud Assembly verwendet, um Erweiterbarkeitsaktionen zu automatisieren.

Die aktionsbasierte Erweiterbarkeit stellt eine einfache und flexible Laufzeit-Engine-Schnittstelle bereit, auf der Sie kleine skriptfähige Aktionen definieren und so konfigurieren können, dass sie bei bestimmten, vom EBS (Event Broker Service) bereitgestellten Ereignissen initiiert werden.

Sie können diese Erweiterbarkeitsaktionsskripts in vRealize Automation Cloud Assembly erstellen und zu Abonnements zuweisen. Ähnlich wie Workflows wird das Erweiterbarkeitsaktionsskript ausgelöst, wenn ein in ein Erweiterbarkeitsabonnement aufgenommenes Ereignis eintritt. Erweiterbarkeitsaktionsskripts werden verwendet, um Aufgaben und Schritte schnell und einfach zu automatisieren. Im Gegensatz zu Workflows, die lokal mithilfe eines vRealize Orchestrator-Servers gehostet werden, werden sie ebenfalls in der Cloud gehostet. Weitere Informationen zur Integration von vRealize Automation Cloud Assembly in einen vRealize Orchestrator-Server finden Sie unter [Konfigurieren der vRealize Orchestrator-Integration in Cloud Assembly](#).

Die aktionsbasierte Erweiterbarkeit bietet Folgendes:

- Eine Alternative zu vRealize Orchestrator-Workflows, die kleine und wiederverwendbare skriptfähige Aktionen für einfache Integrationen und Anpassungen verwendet.
- Eine Möglichkeit zur Wiederverwendung von Aktionsvorlagen, die wiederverwendbare parametrisierte Aktionen enthalten.

Sie können Erweiterbarkeitsaktionen erstellen, indem Sie entweder einen benutzerdefinierten Aktionsskriptcode schreiben oder einen vordefinierten Skriptcode als ZIP-Paket importieren. Die aktionsbasierte Erweiterbarkeit unterstützt die Laufzeitumgebungen „Node.js“ und „Python“ und basiert auf Amazon Web Services Lambda. Aus diesem Grund benötigen Sie ein aktives Abonnement bei Amazon Web Services Identity and Access Management (IAM) und müssen Amazon Web Services als Endpoint in vRealize Automation Cloud Assembly konfigurieren. Informationen zu den ersten Schritten mit Amazon Web Services Lambda finden Sie unter [ABX: Serverlose Erweiterbarkeit von Cloud Assembly-Diensten](#).

Hinweis Erweiterbarkeitsaktionen sind projektspezifisch.

Vorgehensweise zum Erstellen von Erweiterbarkeitsaktionen

Mit vRealize Automation Cloud Assembly können Sie Erweiterbarkeitsaktionen zur Verwendung in Erweiterbarkeitsabonnements erstellen.

Bei Erweiterbarkeitsaktionen handelt es sich um äußerst anpassbare, einfache und flexible Methoden zum Verlängern des Lebenszyklus einer Anwendung mithilfe von benutzerdefiniertem Skriptcode und Aktionsvorlagen. Aktionsvorlagen enthalten vordefinierte Parameter, die beim Einrichten der Grundlage Ihrer Erweiterbarkeitsaktion hilfreich sind.

Es gibt zwei Methoden zum Erstellen einer Erweiterbarkeitsaktion:

- Schreiben von benutzerdefiniertem Code für ein Skript für Erweiterbarkeitsaktionen.

Hinweis Das Schreiben von benutzerdefiniertem Code im Erweiterbarkeitsaktionseditor erfordert möglicherweise eine aktive Internetverbindung.

- Importieren eines Bereitstellungspakets als ZIP-Paket für eine Erweiterbarkeitsaktion. Informationen zum Erstellen eines ZIP-Pakets für Erweiterbarkeitsaktionen finden Sie unter [Erstellen eines ZIP-Pakets für Erweiterbarkeitsaktionen der Python-Laufzeit](#) oder [Erstellen eines ZIP-Pakets für Erweiterbarkeitsaktionen der Node.js-Laufzeit](#).

In den folgenden Schritten wird die Vorgehensweise zum Erstellen einer Erweiterbarkeitsaktion beschrieben, die Amazon Web Services als FaaS-Anbieter verwendet.

Voraussetzungen

- Mitgliedschaft in einem aktiven und gültigen Projekt.
- Konfigurierte Amazon Web Services-Rolle für Lambda-Funktionen. Beispiel: `AWSLambdaBasicExecutionRole`.
- Aktivierte Cloud-Administratorrolle oder `iam:PassRole`-Berechtigungen.

Verfahren

- 1 Klicken Sie auf **Erweiterbarkeit > Bibliothek > Aktionen**.
- 2 Klicken Sie auf **Neue Aktion**.
- 3 Geben Sie einen Namen für die Aktion ein und wählen Sie ein Projekt aus.
- 4 Klicken Sie auf **Weiter**.
- 5 Suchen Sie nach einer Aktionsvorlage und wählen Sie sie aus.

Hinweis Um eine benutzerdefinierte Aktion ohne Verwendung einer Aktionsvorlage zu erstellen, wählen Sie **Benutzerdefiniertes Skript** aus.

Neue konfigurierbare Parameter werden angezeigt.

- 6 Wählen Sie **Skript schreiben** oder **Paket importieren** aus.
- 7 Wählen Sie eine Aktion vom Typ **Laufzeit** aus.
- 8 Geben Sie einen Namen für die **Hauptfunktion** für den Einstiegspunkt der Aktion ein.

Hinweis Für Aktionen, die aus einem ZIP-Paket importiert werden, muss die Hauptfunktion auch den Namen der Skriptdatei mit dem Einstiegspunkt enthalten. Wenn die Hauptskriptdatei beispielsweise den Namen `main.py` aufweist und `handler (context, inputs)` als Einstiegspunkt verwendet wird, muss der Name der Hauptfunktion `main.handler` lauten.

- 9 Legen Sie die **Eingabe**- und **Ausgabe**-Parameter der Aktion fest.

- 10 (Optional) Fügen Sie Anwendungsabhängigkeiten zur Aktion hinzu.

Hinweis Für Aktionen, die aus einem ZIP-Paket importiert werden, werden Anwendungsabhängigkeiten automatisch hinzugefügt.

- 11 Um Zeitüberschreitungs- und Arbeitsspeichergrenzwerte festzulegen, aktivieren Sie die Option **Benutzerdefinierte Zeitüberschreitung und Grenzwerte festlegen**.
- 12 Um Ihre Aktion zu testen, klicken Sie auf **Speichern** und dann auf **Testen**.

Nächste Schritte

Nachdem Ihre Erweiterbarkeitsaktion erstellt und überprüft wurde, können Sie sie einem Abonnement zuweisen.

Hinweis Erweiterbarkeitsabonnements verwenden die neueste freigegebene Version einer Erweiterbarkeitsaktion. Nachdem Sie eine neue Version einer Aktion erstellt haben, klicken Sie oben rechts im Editor-Fenster auf **Versionen**. Um die Version der Aktion, die Sie in Ihrem Abonnement verwenden möchten, freizugeben, klicken Sie auf **Freigeben**.

Exportieren und Importieren von Erweiterbarkeitsaktionen

Mit vRealize Automation Cloud Assembly können Sie Erweiterbarkeitsaktionen für die Verwendung in unterschiedlichen Projekten exportieren und importieren.

Voraussetzungen

Eine vorhandene Erweiterbarkeitsaktion.

Verfahren

- 1 Exportieren Sie eine Erweiterbarkeitsaktion.
 - a Navigieren Sie zu **Erweiterbarkeit > Bibliothek > Aktionen**.
 - b Wählen Sie eine Erweiterbarkeitsaktion aus und klicken Sie auf **Exportieren**.
Das Aktionsskript und seine Abhängigkeiten werden als ZIP-Datei in Ihrer lokalen Umgebung gespeichert.
- 2 Importieren Sie eine Erweiterbarkeitsaktion.
 - a Navigieren Sie zu **Erweiterbarkeit > Bibliothek > Aktionen**.
 - b Klicken Sie auf **Importieren**.
 - c Wählen Sie die exportierte Erweiterbarkeitsaktion aus und weisen Sie sie einem Projekt zu.
 - d Klicken Sie auf **Importieren**.

Hinweis Wenn die importierte Erweiterbarkeitsaktion dem angegebenen Projekt bereits zugewiesen ist, werden Sie aufgefordert, eine Konfliktlösungsrichtlinie auszuwählen.

Alternative Sie können auch Aktionsskripte importieren, indem Sie die Option **Paket importieren** direkt im Aktions-Editor auswählen.

Erstellen eines ZIP-Pakets für Erweiterbarkeitsaktionen der Python-Laufzeit

Sie können ein ZIP-Paket mit dem Python-Skript und Abhängigkeiten erstellen, die von Ihren vRealize Automation Cloud Assembly-Erweiterbarkeitsaktionen verwendet werden.

Zwei Methoden stehen zur Erstellung des Skripts für Ihre Erweiterbarkeitsaktionen zur Verfügung:

- Direktes Schreiben des Skripts im Erweiterbarkeitsaktionseditor in vRealize Automation Cloud Assembly.
- Erstellen des Skripts in Ihrer lokalen Umgebung und Hinzufügen des Skripts zu einem ZIP-Paket mit allen relevanten Abhängigkeiten.

Mithilfe eines ZIP-Pakets können Sie eine benutzerdefinierte vorkonfigurierte Vorlage mit Aktionsskripts und Abhängigkeiten erstellen, die Sie zur Verwendung in Erweiterbarkeitsaktionen in vRealize Automation Cloud Assembly importieren können.

Darüber hinaus können Sie ein ZIP-Paket in Szenarien verwenden, in denen Module, die mit Abhängigkeiten in Ihrem Aktionsskript verknüpft sind, nicht vom vRealize Automation Cloud Assembly-Dienst aufgelöst werden können. Dies ist beispielsweise der Fall, wenn kein Internetzugriff in Ihrer Umgebung möglich ist.

Sie können auch ein ZIP-Paket verwenden, um Erweiterbarkeitsaktionen mit mehreren Python-Skriptdateien zu erstellen. Die Verwendung mehrerer Skriptdateien kann nützlich sein, um die Struktur des Codes der Erweiterbarkeitsaktionen zu verwalten.

Voraussetzungen

Laden Sie bei Verwendung von Python 3.3 oder früher das Installationsprogramm für das PIP-Paket herunter und konfigurieren Sie es. Weitere Informationen finden Sie im [Python-Paketindex](#).

Verfahren

- 1 Erstellen Sie auf Ihrem lokalen Computer einen Ordner für das Aktionsskript und die Abhängigkeiten.

Beispiel: `/home/user1/zip-action`.

- 2 Fügen Sie dem Ordner das Python-Hauptaktionsskript oder Skripts hinzu.

Beispiel: `/home/user1/zip-action/main.py`.

- 3 (Optional) Fügen Sie dem Ordner alle Abhängigkeiten für das Python-Skript hinzu.
 - a Erstellen Sie eine Datei vom Typ `requirements.txt`, die die Abhängigkeiten enthält. Weitere Informationen finden Sie unter [Anforderungsdateien](#).
 - b Öffnen Sie eine Linux-Shell.

Hinweis Die Laufzeit der aktionsbasierten Erweiterbarkeit (ABX) in vRealize Automation Cloud Assembly ist Linux-basiert. Aus diesem Grund machen in einer Windows-Umgebung kompilierte Python-Abhängigkeiten das erzeugte ZIP-Paket für die Erstellung von Erweiterbarkeitsaktionen möglicherweise unbrauchbar. Deshalb müssen Sie eine Linux-Shell verwenden.

- c Installieren Sie die Datei `requirements.txt` im Skriptordner, indem Sie den folgenden Befehl ausführen:

```
pip install -r requirements.txt --target=home/user1/zip-action
```

- 4 Wählen Sie im zugewiesenen Ordner die Skriptelemente und gegebenenfalls die Datei `requirements.txt` aus und komprimieren Sie die Elemente in einem ZIP-Paket.

Hinweis Sowohl die Skript- als auch die Abhängigkeitselemente müssen auf der Root-Ebene des ZIP-Pakets gespeichert werden. Beim Erstellen des ZIP-Pakets in einer Linux-Umgebung tritt möglicherweise ein Problem auf, wenn der Paketinhalt nicht auf der Root-Ebene gespeichert wird. Wenn dieses Problem auftritt, erstellen Sie das Paket, indem Sie den Befehl `zip -r` in der Befehlszeilen-Shell ausführen.

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

Nächste Schritte

Verwenden Sie das ZIP-Paket, um ein Erweiterbarkeitsaktionsskript zu erstellen. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Erstellen von Erweiterbarkeitsaktionen](#).

Erstellen eines ZIP-Pakets für Erweiterbarkeitsaktionen der Node.js-Laufzeit

Sie können ein ZIP-Paket mit dem Skript „Node.js“ und Abhängigkeiten erstellen, die von Ihren vRealize Automation Cloud Assembly-Erweiterbarkeitsaktionen verwendet werden.

Zwei Methoden stehen zur Erstellung des Skripts für Ihre Erweiterbarkeitsaktionen zur Verfügung:

- Direktes Schreiben des Skripts im Erweiterbarkeitsaktionseditor in vRealize Automation Cloud Assembly.
- Erstellen des Skripts in Ihrer lokalen Umgebung und Hinzufügen des Skripts zu einem ZIP-Paket mit allen relevanten Abhängigkeiten.

Mithilfe eines ZIP-Pakets können Sie eine benutzerdefinierte vorkonfigurierte Vorlage mit Aktionsskripts und Abhängigkeiten erstellen, die Sie zur Verwendung in Erweiterbarkeitsaktionen in vRealize Automation Cloud Assembly importieren können.

Darüber hinaus können Sie ein ZIP-Paket in Szenarien verwenden, in denen Module, die mit Abhängigkeiten in Ihrem Aktionsskript verknüpft sind, nicht vom vRealize Automation Cloud Assembly-Dienst aufgelöst werden können. Dies ist beispielsweise der Fall, wenn kein Internetzugriff in Ihrer Umgebung möglich ist.

Weiterhin können Sie Pakete verwenden, um Erweiterbarkeitsaktionen mit mehreren Node.js-Skriptdateien zu erstellen. Die Verwendung mehrerer Skriptdateien kann nützlich sein, um die Struktur des Codes der Erweiterbarkeitsaktionen zu verwalten.

Verfahren

- 1 Erstellen Sie auf Ihrem lokalen Computer einen Ordner für das Aktionsskript und die Abhängigkeiten.

Beispiel: `/home/user1/zip-action`.

- 2 Fügen Sie dem Ordner das Hauptaktionsskript „Node.js“ oder Skripts hinzu.

Beispiel: `/home/user1/zip-action/main.js`.

- 3 (Optional) Fügen Sie dem Ordner alle Abhängigkeiten für das Skript „Node.js“ hinzu.

- a Erstellen Sie eine Datei vom Typ `package.json` mit Abhängigkeiten im Skriptordner. Weitere Informationen finden Sie unter [Erstellen einer package.json-Datei](#) und [Angaben von „dependencies“ und „devDependencies“ in einer package.json-Datei](#).

- b Öffnen Sie eine Befehlszeilen-Shell.

- c Navigieren Sie zu dem Ordner, den Sie für das Aktionsskript und die Abhängigkeiten erstellt haben.

```
cd /home/user1/zip-action
```

- d Installieren Sie die Datei `package.json` im Skriptordner, indem Sie den folgenden Befehl ausführen:

```
npm install --production
```

Hinweis Mit diesem Befehl wird ein Verzeichnis vom Typ `node_modules` in Ihrem Ordner erstellt.

- 4 Wählen Sie im zugewiesenen Ordner die Skriptelemente und gegebenenfalls das Verzeichnis `node_modules` aus und komprimieren Sie die Elemente in einem ZIP-Paket.

Hinweis Sowohl die Skript- als auch die Abhängigkeitselemente müssen auf der Root-Ebene des ZIP-Pakets gespeichert werden. Beim Erstellen des ZIP-Pakets in einer Linux-Umgebung tritt möglicherweise ein Problem auf, wenn der Paketinhalt nicht auf der Root-Ebene gespeichert wird. Wenn dieses Problem auftritt, erstellen Sie das Paket, indem Sie den Befehl `zip -r` in der Befehlszeilen-Shell ausführen.

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

Nächste Schritte

Verwenden Sie das ZIP-Paket, um ein Erweiterbarkeitsaktionsskript zu erstellen. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Erstellen von Erweiterbarkeitsaktionen](#).

Konfigurieren von Cloud-spezifischen Erweiterbarkeitsaktionen

Sie können Erweiterbarkeitsaktionen so konfigurieren, dass sie mit Ihren Cloud-Konten verwendet werden können.

Beim Erstellen einer Erweiterbarkeitsaktion können Sie sie mit verschiedenen Cloud-basierten Konten konfigurieren und verknüpfen:

- Microsoft Azure
- Amazon Web Services

Voraussetzungen

Ein gültiges Cloud-Konto ist erforderlich.

Verfahren

- 1 Wählen Sie **Erweiterbarkeit > Bibliothek > Aktion**.
- 2 Klicken Sie auf **Neue Aktion**.
- 3 Geben Sie die Aktionsparameter nach Bedarf an.
- 4 Wählen Sie im Dropdown-Menü **FaaS-Anbieter** Ihren Cloud-Kontoanbieter aus oder wählen Sie **Auto** aus.

Hinweis Wenn Sie **Automatisch** auswählen, wird der FaaS-Anbieter automatisch von der Aktion festgelegt.

- 5 Klicken Sie auf **Speichern**.

Ergebnisse

Ihre Erweiterbarkeitsaktion ist für die Verwendung mit dem konfigurierten Cloud-Konto verknüpft.

Konfigurieren lokaler Erweiterbarkeitsaktionen

Sie können Ihre Erweiterbarkeitsaktionen für die Verwendung eines lokalen FaaS-Anbieters anstelle eines Amazon Web Services- oder Microsoft Azure-Cloud-Kontos konfigurieren.

Indem Sie einen lokalen FaaS-Anbieter für Ihre Erweiterbarkeitsaktionen verwenden, können Sie lokale Dienste wie LDAP, CMDB oder vCenter-Datencenter in Ihren vRealize Automation Cloud Assembly-Erweiterbarkeitsabonnements verwenden.

Verfahren

- 1 Klicken Sie auf **Erweiterbarkeit > Bibliothek > Aktionen**.
- 2 Klicken Sie auf **Neue Aktion**.
- 3 Geben Sie einen Namen und ein Projekt für die Erweiterbarkeitsaktion ein.
- 4 (Optional) Geben Sie eine Beschreibung für die Erweiterbarkeitsaktion ein.
- 5 Klicken Sie auf **Weiter**.
- 6 Erstellen oder importieren Sie das Skript für die Erweiterbarkeitsaktion.
- 7 Klicken Sie auf das Dropdown-Menü **FaaS-Anbieter** und wählen Sie **Lokal** aus.
- 8 Um die neue Erweiterbarkeitsaktion zu speichern, klicken Sie auf **Speichern**.

Nächste Schritte

Verwenden Sie die zuvor erstellte Erweiterbarkeitsaktion in Ihren vRealize Automation Cloud Assembly-Erweiterbarkeitsabonnements.

Definition eines Aktionsablaufs

Bei Aktionsabläufen handelt es sich um eine Gruppe von Erweiterbarkeitsaktionsskripts, mit deren Hilfe Lebenszyklen und Automatisierung weiter verlängert werden können.

Alle Aktionsabläufe beginnen mit `flow_start` und enden mit `flow_end`. Mithilfe der folgenden Aktionsablaufelemente können Sie mehrere Erweiterbarkeitsaktionsskripts miteinander verknüpfen:

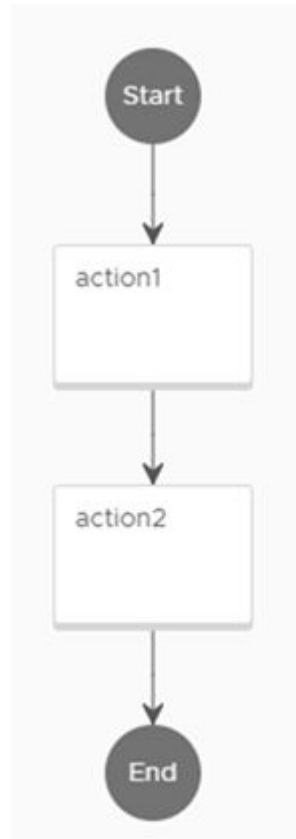
- [Sequenzielle Aktionsabläufe](#) – Mehrere Erweiterbarkeitsaktionsskripts werden nacheinander ausgeführt.
- [Fork-Aktionsabläufe](#) – Mehrere Erweiterbarkeitsaktionsskripts oder -abläufe, die über aufgeteilte Pfade zur gleichen Ausgabe beitragen.
- [Join-Aktionsabläufe](#) – Mehrere Erweiterbarkeitsaktionsskripts oder -abläufe, die miteinander verbunden werden und zur gleichen Ausgabe beitragen.
- [Bedingte Aktionsabläufe](#) – Mehrere Erweiterbarkeitsaktionsskripts oder -abläufe, die ausgeführt werden, nachdem eine Bedingung erfüllt worden ist.

Sequenzielle Aktionsabläufe

Mehrere Erweiterbarkeitsaktionsskripts, die nacheinander ausgeführt werden.

```
version: "1"
flow:
  flow_start:
    next: action1
  action1:
    action: <action_name>
    next: action2
  action2:
    action: <action_name>
    next: flow_end
```

Hinweis Sie können zu einer vorherigen Aktion zurückkehren, indem Sie sie als `next:`-Aktion zuweisen. In diesem Beispiel können Sie beispielsweise anstelle von `next: flow_end` `next: action1` eingeben, um Aktion1 erneut auszuführen und die Abfolge der Aktionen neu zu starten.



Fork-Aktionsabläufe

Mehrere Erweiterbarkeitsaktionsskripts oder -abläufe, die Pfade aufteilen, um zur gleichen Ausgabe beizutragen.

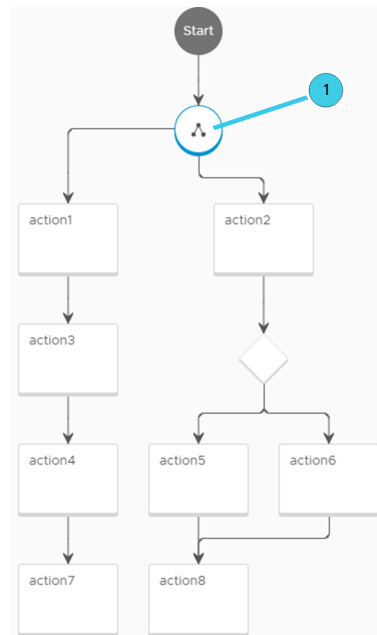
```

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>

```

Hinweis Sie können zu einer vorherigen Aktion zurückkehren, indem Sie sie als `next:`-Aktion zuweisen.

Beispiel: Statt `next: flow_end` zum Beenden des Aktionsablaufs zu verwenden, können Sie `next: action1` eingeben, um Aktion1 erneut auszuführen und die Abfolge der Aktionen neu zu starten.



1 Fork-Element

Join-Aktionsabläufe

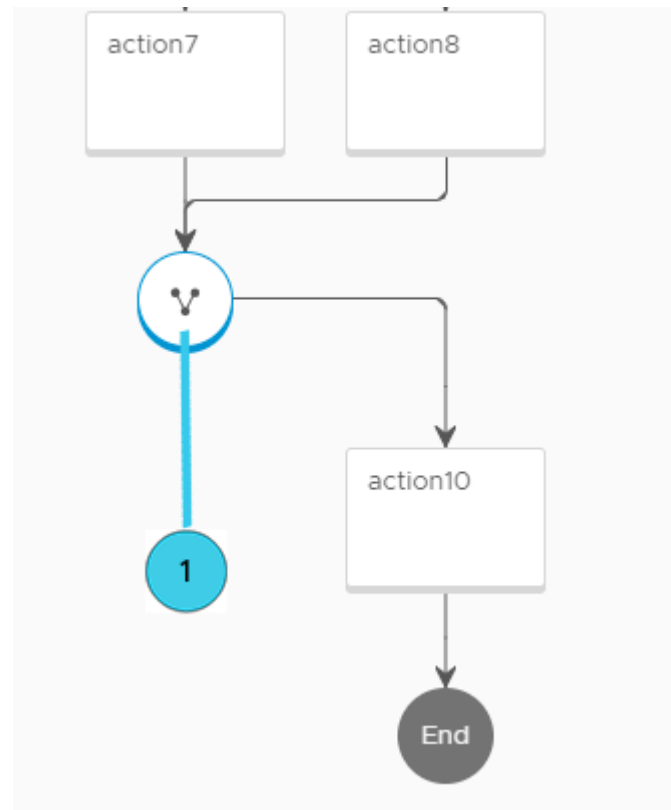
Mehrere Erweiterbarkeitsaktionsskripts oder -abläufe, die mehrere Pfade bündeln und zur gleichen Ausgabe beitragen.

```

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

```

Hinweis Sie können zu einer vorherigen Aktion zurückkehren, indem Sie sie als `next:`-Aktion zuweisen. In diesem Beispiel können Sie beispielsweise anstelle von `next: flow_end` `next: action1` eingeben, um Aktion1 erneut auszuführen und die Abfolge der Aktionen neu zu starten.



1 Join-Element

Bedingte Aktionsabläufe

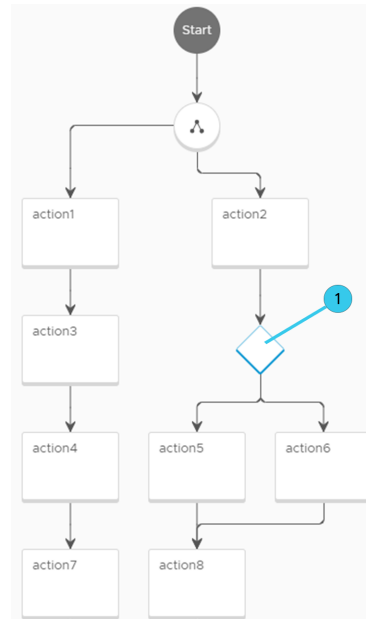
Mehrere Erweiterbarkeitsaktionsskripts oder -abläufe, die ausgeführt werden, wenn eine Bedingung mithilfe eines Switch-Elements erfüllt ist.

In bestimmten Fällen muss die Bedingung gleich `true` sein, damit die Aktion ausgeführt werden kann. In anderen Fällen (wie in diesem Beispiel) werden Parameterwerte benötigt, die erfüllt sein müssen, bevor eine Aktion ausgeführt werden kann. Wenn keine der Bedingungen erfüllt ist, schlägt der Aktionsablauf fehl.

```

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 Switch-Element

Hinweis Sie können zu einer vorherigen Aktion zurückkehren, indem Sie sie als `next`:-Aktion zuweisen.

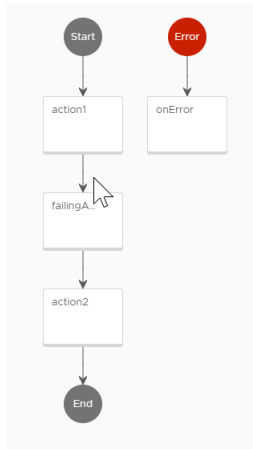
Beispiel: Statt `next: flow_end` zum Beenden des Aktionsablaufs zu verwenden, können Sie `next: action1` eingeben, um Aktion1 erneut auszuführen und die Abfolge der Aktionen neu zu starten.

Vorgehensweise zum Verwenden eines Fehlerhandlers in Aktionsabläufen

Indem Sie ein Fehlerbehandlungselement verwenden, können Sie Ihren Aktionsablauf so konfigurieren, dass er in bestimmten Phasen des Ablaufs einen Fehler anzeigt.

Ein Fehlerbehandlungselement erfordert zwei Eingaben:

- Angegebene Fehlermeldung der fehlgeschlagenen Aktion.
- Eingaben des Aktionsablaufs.



Wenn eine Aktion in Ihrem Ablauf fehlschlägt und der Aktionsablauf ein Fehlerbehandlungselement enthält, wird eine Fehlermeldung ausgegeben, die Sie über den Aktionsablauf informiert. Der Fehlerhandler ist eine eigene Aktion. Das folgende Skript ist ein Beispiel für einen Fehlerhandler, der in einem Aktionsablauf verwendet werden kann.

```
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
```

Sie können die erfolgreichen und fehlgeschlagenen Ausführungen im Fenster „Aktionsausführungen“ anzeigen.

Status	Aktion	Aktions-ID
Abgeschlossen	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
Fehlgeschlagen	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
Abgeschlossen	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
Abgeschlossen	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6

In diesem Beispiel wurde der Ablauf „flow-with-handler“, der das Fehlerbehandlungselement enthält, erfolgreich ausgeführt. Eine der Aktionen im Ablauf ist jedoch fehlgeschlagen, was den Fehlerhandler dazu veranlasst hat, einen Fehler zu melden.

Vorgehensweise zum Verfolgen von Aktionsausführungen

Auf der Registerkarte „Aktionsausführungen“ wird ein Protokoll mit den von Abonnements ausgelösten Erweiterbarkeitsaktionen und deren Status angezeigt.

Sie können das Protokoll der Aktionsausführungen mithilfe von **Erweiterbarkeit > Aktivität > Aktionsausführungen** anzeigen. Darüber hinaus können Sie die Liste der Aktionsausführungen gleichzeitig nach einer oder mehreren Eigenschaften filtern. Zur Anzeige weiterer Details zu einer einzelnen Aktionsausführung klicken Sie auf die Ausführungs-ID.

Fehlerbehebung bei fehlgeschlagenen Ausführungen von Erweiterbarkeitsaktionen

Wenn die Ausführung der Erweiterbarkeitsaktion fehlschlägt, können Sie zu Korrekturzwecken Fehlerbehebungsschritte durchführen.

Wenn eine Aktionsausführung fehlschlägt, erhalten Sie möglicherweise eine Fehlermeldung, einen Fehlerstatus und ein Fehlerprotokoll. Wenn Ihre Aktionsausführung fehlschlägt, ist dies entweder auf einen Bereitstellungs- oder einen Codefehler zurückzuführen.

Problem	Lösung
Bereitstellungsfehler	Diese Fehler ergeben sich aus Problemen im Zusammenhang mit der Konfiguration des Cloud-Kontos, der Aktionsbereitstellung oder anderen Abhängigkeiten, die die Bereitstellung der Aktion verhindern können. Stellen Sie sicher, dass das von Ihnen verwendete Projekt innerhalb des konfigurierten Cloud-Kontos definiert ist und Berechtigungen zum Ausführen von Funktionen erteilt wurde. Bevor Sie die Aktion erneut initiieren, können Sie die Aktion anhand eines bestimmten Projekts auf der Seite „Details“ der Aktion testen.
Codefehler	Diese Fehler sind auf ungültige Skripts oder ungültigen Code zurückzuführen. Verwenden Sie die Aktionsprotokolle zur Fehlerbehebung und Korrektur der ungültigen Skripts.

Abonnements für Erweiterbarkeits-Workflows

Sie können Ihre von vRealize Orchestrator gehosteten Workflows mit vRealize Automation Cloud Assembly verwenden, um den Lebenszyklus Ihrer Anwendung zu verlängern.

Vorgehensweise zum Ändern der Eigenschaften virtueller Maschinen mithilfe eines vRealize Orchestrator-Workflow-Abonnements

Sie können einen vorhandenen vRealize Orchestrator-Workflow verwenden, um Eigenschaften virtueller Maschinen zu ändern und virtuelle Maschinen zu Active Directory hinzuzufügen.

Im Abonnementschema wird das Format der Nutzlast für EBS-Nachrichten (Event Broker Service, Ereignisbrokerdienst) definiert. Um die EBS-Nachrichtennutzlast innerhalb eines Workflows zu empfangen und zu verwenden, müssen Sie die Workflow-Eingabeparameter „inputProperties“ definieren.

Voraussetzungen

- Cloud-Administratorbenutzerrolle
- Vorhandene lokale vRealize Orchestrator-Workflows.
- Erfolgreiche Integration und Verbindung zum vRealize Orchestrator-Clientserver.

Verfahren

- 1 Klicken Sie auf **Erweiterbarkeit > Abonnements**.
- 2 Klicken Sie auf **Neues Abonnement**.
- 3 Erstellen Sie ein Abonnement mit den folgenden Parametern:

Parameter	Wert
Name	RenameVM
Ereignisthema	Wählen Sie ein Ereignisthema aus, das für die gewünschte vRealize Orchestrator-Integration geeignet ist. Beispiel: Computing-Zuteilung.
Blockierend/Nicht blockierend	Nicht blockierend
Ausführbares Element	Wählen Sie einen ausführbaren vRealize Orchestrator-Typ aus.
Ausführbare ID	Wählen Sie den gewünschten Workflow aus. Beispiel: VM-Namen festlegen.

- 4 Zum Speichern des Abonnements klicken Sie auf **Speichern**.
- 5 Weisen Sie Ihr Abonnement zu und aktivieren Sie es, indem Sie einen Blueprint erstellen oder einen vorhandenen Blueprint bereitstellen.

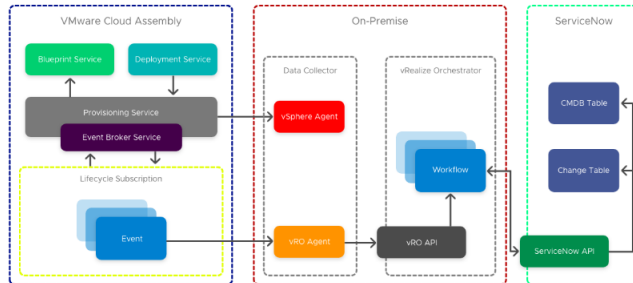
Nächste Schritte

Stellen Sie mithilfe einer der folgenden Methoden sicher, dass der Workflow erfolgreich initiiert wurde:

- Vergewissern Sie sich, dass der Workflow das Protokoll ausführt. Klicken Sie dazu auf **Erweiterbarkeit > Aktivität > Workflow-Ausführungen**.
- Öffnen Sie den vRealize Orchestrator-Client und überprüfen Sie den Workflow-Status, indem Sie zum Workflow navigieren und den Status überprüfen oder indem Sie die Registerkarte der entsprechenden Protokolle öffnen.

Vorgehensweise zum Integrieren von Cloud Assembly für ITSM mit ServiceNow unter Verwendung von vRealize Orchestrator-Workflows

Mithilfe von gehosteten vRealize Orchestrator-Workflows können Sie vRealize Automation Cloud Assembly mit ServiceNow für ITSM-Übereinstimmung integrieren.



Unternehmensbenutzer integrieren ihre Cloud Management Plattform zu Übereinstimmungszwecken in der Regel mit einer ITSM- (IT Service Management) und einer CMDB-Plattform (Configuration Management Database). Im Anschluss an dieses Beispiel können Sie vRealize Automation Cloud Assembly mit ServiceNow für CMDB und ITSM unter Verwendung von gehosteten vRealize Orchestrator-Workflows integrieren. Bei Verwendung von vRealize Orchestrator-Integrationen und -Workflows sind Funktions-Tags besonders nützlich, wenn Sie über mehrere Instanzen für verschiedene Umgebungen verfügen. Weitere Informationen zu Funktions-Tags finden Sie unter [Verwenden von Funktions-Tags in vRealize Automation Cloud Assembly](#).

Hinweis Sie können ServiceNow unter Verwendung von Erweiterbarkeitsaktionsskripts auch mit vRealize Automation Cloud Assembly integrieren. Informationen zum Integrieren von ServiceNow mithilfe von Erweiterbarkeitsaktionsskripten finden Sie unter [Vorgehensweise zum Integrieren von Cloud Assembly in ServiceNow unter Verwendung von Erweiterbarkeitsaktionen](#).

In diesem Beispiel besteht die ServiceNow-Integration aus drei Workflows auf oberster Ebene. Jeder Workflow verfügt über eigene Abonnements, sodass Sie jede Komponente einzeln aktualisieren und durchlaufen lassen können.

- Einstiegspunkt des Ereignisabonnements – Einfache Protokollierung, gibt gegebenenfalls den anfordernden Benutzer und die vCenter-VM an.
- Integrations-Workflow – trennt Objekte und speist Eingaben in den technischen Workflow ein, verarbeitet Protokollierungs-, Eigenschaften- und Ausgabeaktualisierungen.
- Technischer Workflow – nachgelagerte Systemintegration für die ServiceNow-API zum Erstellen der CMDB-CI-, CR- und CAS-IaaS-API mit zusätzlichen VM-Eigenschaften außerhalb der Nutzlast.

Voraussetzungen

- Eine eigenständige oder geclusterte vRealize Orchestrator-Umgebung.

- Eine vRealize Orchestrator-Integration in vRealize Automation Cloud Assembly. Informationen zur Integration einer eigenständigen vRealize Orchestrator-Instanz in vRealize Automation Cloud Assembly finden Sie unter [Konfigurieren der vRealize Orchestrator-Integration in Cloud Assembly](#).

Verfahren

- 1 Erstellen und speichern Sie in vRealize Orchestrator eine Konfigurationsdatei, die eine allgemeine, in mehreren Workflows verwendete Konfiguration enthält.
- 2 Speichern Sie Ihr CAS-API-Token am selben Speicherort wie die Konfigurationsdatei aus Schritt 1.

Hinweis Das CAS-API-Token weist ein Ablaufdatum auf.

- 3 Erstellen Sie mit dem bereitgestellten Skriptelement einen Workflow in vRealize Orchestrator. Dieses Skript verweist auf einen REST-Host und sucht nach diesem. Es standardisiert auch REST-Aktionen, die einen optionalen Parameter eines Tokens verwenden, der als zusätzlicher Autorisierungs-Header hinzugefügt wird.

```
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) ;

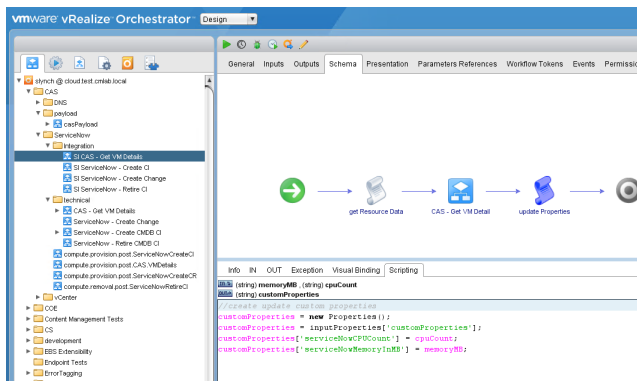
try{
    var vm = JSON.parse(vmResponse);
} catch (ex) {
    throw ex + " failed to parse vm details"
}

System.log("cpuCount: " + vm["customProperties"]["cpuCount"]);
System.log("memoryInMB: " + vm["customProperties"]["memoryInMB"]);

cpuCount = vm["customProperties"]["cpuCount"];
memoryMB = vm["customProperties"]["memoryInMB"];

```

Dieses Skript sendet die Ausgabe `cpuCount` und `memoryMB` an den übergeordneten Workflow und aktualisiert die vorhandenen `customProperties`-Eigenschaften. Diese Werte können bei der Erstellung der CMDB in nachfolgenden Workflows verwendet werden.



- 4 Fügen Sie das ServiceNow-CMDB-Skriptelement zum Erstellen des Konfigurationselements zu Ihrem Workflow hinzu. Dieses Element sucht mithilfe des Konfigurationselements nach dem ServiceNow-REST-Host, erstellt einen REST-Vorgang für die `cmdb_ci_vmware_instance`-Tabelle sowie basierend auf Workflow-Eingaben für Post-Daten eine Zeichenfolge aus Inhaltsobjekten und gibt die zurückgegebene `sys_id` aus.

```
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;

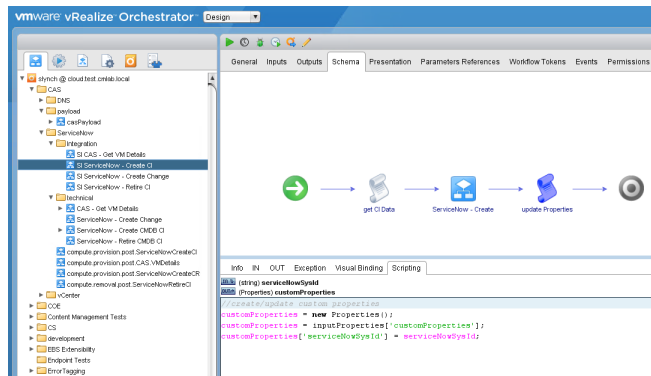
postContent = JSON.stringify(contentObject);
System.log("JSON: " + postContent);

// (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'];
```

- Erstellen Sie mithilfe der Ausgabe aus dem untergeordneten Workflow unter Verwendung der vorhandenen `customProperties` ein Eigenschaftenobjekt und überschreiben Sie die Eigenschaft `serviceNowSysId` mit dem Wert aus ServiceNow. Diese eindeutige ID wird in der CMDB verwendet, um eine Instanz beim Löschen als veraltet zu kennzeichnen.



Ergebnisse

vRealize Automation Cloud Assembly wurde erfolgreich mit ITSM-Lösung ServiceNow integriert. Weitere Informationen zur Verwendung von Workflows für die Integration von ServiceNow in vRealize Automation Cloud Assembly finden Sie im Blogbeitrag [Extending Cloud Assembly with vRealize Orchestrator for ServiceNow Integration](#), der die Erweiterung von Cloud Assembly mit vRealize Orchestrator für die ServiceNow-Integration behandelt.

Weitere Informationen zu Workflow-Abonnements

Wenn Sie die vRealize Orchestrator-Integration in vRealize Automation Cloud Assembly verwenden, können Sie die Lebenszyklen von Anwendungen mit Workflows verlängern.

vRealize Automation Cloud Assembly enthält die vRealize Orchestrator-Integration, die Sie zum Importieren und Verknüpfen vorhandener lokaler Workflows mit einem Abonnement verwenden können. Diese Workflows werden auf dem vRealize Orchestrator-Server verwaltet. Sie können Workflows nur mithilfe des vRealize Orchestrator-Clients erstellen, ändern und löschen.

Empfehlungen zum Erstellen von vRealize Orchestrator-Workflows

Ein Workflow-Abonnement basiert auf einem bestimmten Themenschema. Die Abonnements müssen mit den korrekten Eingabeparametern konfiguriert werden, damit sie die vRealize Orchestrator-Workflows initiieren und mit den Ereignisdaten verwendet werden können.

Workflow-Eingabeparameter

Der benutzerdefinierte Workflow kann alle Parameter oder einen einzelnen Parameter enthalten, der alle Daten in der Nutzlast verbraucht.

Zur Verwendung eines einzelnen Parameters konfigurieren Sie einen Parameter mit dem Typ `Properties` und dem Namen `inputProperties`.

Workflow-Ausgabeparameter

Der benutzerdefinierte Workflow kann Ausgabeparameter mit Relevanz für nachfolgende Ereignisse enthalten, die für ein Antwortereignisthema erforderlich sind.

Wenn ein Ereignisthema eine Antwort erwartet, müssen die Workflow-Ausgabeparameter mit dem Antwortschema übereinstimmen.

Vorgehensweise zum Verfolgen von Workflow-Ausführungen

Auf der Registerkarte **Workflow-Ausführungen** wird ein Protokoll der von Abonnements ausgelösten Workflows und deren Status angezeigt.

Sie können das Protokoll der Workflow-Ausführungen mit einem Klick auf **Erweiterbarkeit > Aktivität > Workflow-Ausführungen** anzeigen.

Fehlerbehebung bei fehlgeschlagenen Workflow-Abonnements

Wenn Ihr Workflow-Abonnement fehlschlägt, können Sie Fehlerbehebungsschritte durchführen, um es zu korrigieren.

Fehlgeschlagene Workflow-Ausführungen können dazu führen, dass Ihr Workflow-Abonnement nicht erfolgreich gestartet oder abgeschlossen wird. Fehler bei der Workflow-Ausführung können das Ergebnis mehrerer allgemeiner Probleme sein.

Problem	Ursache	Lösung
Ihr vRealize Orchestrator-Workflow-Abonnement wurde nicht erfolgreich gestartet oder abgeschlossen.	Sie haben ein Workflow-Abonnement konfiguriert, um beim Eingang der Ereignismeldung einen benutzerdefinierten Workflow auszuführen, aber der Workflow wird nicht erfolgreich gestartet oder abgeschlossen.	<ol style="list-style-type: none"> 1 Stellen Sie sicher, dass das Workflow-Abonnement ordnungsgemäß gespeichert wurde. 2 Stellen Sie sicher, dass die Bedingungen des Workflow-Abonnements ordnungsgemäß konfiguriert sind. 3 Stellen Sie sicher, dass vRealize Orchestrator den angegebenen Workflow enthält. 4 Stellen Sie sicher, dass der Workflow in vRealize Orchestrator ordnungsgemäß konfiguriert ist.
Die Genehmigungsanforderung für Ihr vRealize Orchestrator-Workflow-Abonnement wurde nicht ausgeführt.	Sie haben ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“ konfiguriert, um einen vRealize Orchestrator-Workflow auszuführen. Der Workflow wird nicht ausgeführt, wenn eine Maschine, die den definierten Kriterien entspricht, im Service Catalog angefordert wird.	<p>Zur erfolgreichen Ausführung einer Genehmigung für ein Workflow-Abonnement müssen Sie sicherstellen, dass alle Komponenten ordnungsgemäß konfiguriert sind.</p> <ol style="list-style-type: none"> 1 Stellen Sie sicher, dass die Genehmigungsrichtlinie aktiv ist und ordnungsgemäß angewendet wurde. 2 Stellen Sie sicher, dass Ihr Workflow-Abonnement ordnungsgemäß konfiguriert und gespeichert wurde. 3 Überprüfen Sie die Ereignisprotokolle auf Meldungen im Zusammenhang mit Genehmigungen.
Die Genehmigungsanforderung für Ihr vRealize Orchestrator-Workflow-Abonnement wurde abgelehnt.	Sie haben ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“ konfiguriert, das einen angegebenen vRealize Orchestrator-Workflow ausführt. Die Anforderung wird jedoch auf der externen Genehmigungsebene abgelehnt. Eine mögliche Ursache ist ein interner Fehler bei der Workflow-Ausführung in vRealize Orchestrator. Beispielsweise fehlt der Workflow oder der vRealize Orchestrator-Server wird nicht ausgeführt.	<ol style="list-style-type: none"> 1 Überprüfen Sie die Protokolle auf Meldungen im Zusammenhang mit Genehmigungen. 2 Stellen Sie sicher, dass der vRealize Orchestrator-Server ausgeführt wird. 3 Stellen Sie sicher, dass vRealize Orchestrator den angegebenen Workflow enthält.

Weitere Informationen zu Erweiterbarkeitsabonnements

Sie können die Lebenszyklen Ihrer Anwendungen unter Verwendung von Erweiterbarkeitsaktionen oder gehosteten vRealize Orchestrator-Workflows mit Erweiterbarkeitsabonnements verlängern.

Wenn ein auslösendes Ereignis in Ihrer Umgebung auftritt, wird das Abonnement initiiert und der angegebene Workflow oder die angegebene Erweiterbarkeitsaktion wird ausgeführt. Sie können Systemereignisse im Ereignisprotokoll, Workflow-Ausführungen im Fenster „Workflow-Ausführungen“ und Aktionsausführungen im Fenster „Aktion ausführen“ anzeigen. Abonnements sind projektspezifisch, d. h., sie sind über das angegebene Projekt mit Blueprints und Bereitstellungen verknüpft.

Terminologie der Erweiterbarkeit

Beim Arbeiten mit Erweiterbarkeit und Abonnements in vRealize Automation Cloud Assembly stoßen Sie möglicherweise auf spezielle Terminologie für die Abonnements und den Ereignisbrokerdienst.

Tabelle 6-4. Terminologie der Erweiterbarkeit

Begriff	Beschreibung
Ereignisthema	Beschreibt mehrere Ereignisse mit derselben logischen Absicht und derselben Struktur. Jedes Ereignis stellt eine Instanz eines Ereignisthemas dar. Sie können bestimmten Ereignisthemen blockierende Parameter zuweisen. Weitere Informationen finden Sie unter Blockieren von Ereignisthemen .
Ereignis	Bezeichnet eine Statusänderung beim Producer oder den Elementen, die vom Producer verwaltet werden. Beim Ereignis handelt es sich um das Element, das Informationen zum Auftreten des Ereignisses aufzeichnet.
Ereignisbrokerdienst	Mit diesem Dienst werden Nachrichten gesendet, die von einem Producer für die abonnierten Verbraucher veröffentlicht werden.
Nutzlast	Die Ereignisdaten, die alle relevanten Eigenschaften im Zusammenhang mit dem entsprechenden Ereignisthema enthalten.
Abonnement	Gibt an, dass ein Abonnent über ein Ereignis informiert werden möchte, indem ein Ereignisthema abonniert und die Kriterien definiert werden, die die Benachrichtigung auslösen. Abonnements verknüpfen Erweiterbarkeitsaktionen oder Workflows mit auslösenden Ereignissen, mit denen Teile des Anwendungslebenszyklus automatisiert werden.
Abonnent	Die Benutzer, die über die Ereignisse informiert werden, die basierend auf der Abonnementdefinition für den Ereignisbrokerdienst veröffentlicht werden. Der Abonnent kann auch als Verbraucher bezeichnet werden.

Tabelle 6-4. Terminologie der Erweiterbarkeit (Fortsetzung)

Begriff	Beschreibung
Systemadministrator	Ein Benutzer mit Berechtigungen zum Erstellen, Lesen, Aktualisieren und Löschen der Mandanten- und System-Workflow-Abonnements mithilfe von vRealize Automation Cloud Assembly.
Workflow-Abonnement	Legt das Ereignisthema und die Bedingungen fest, die einen vRealize Orchestrator-Workflow auslösen.
Aktionsabonnement	Gibt das Ereignisthema und die Bedingungen an, die die Ausführung einer Erweiterbarkeitsaktion auslösen.
Workflow	Ein vRealize Orchestrator-Workflow, der in vRealize Automation Cloud Assembly integriert ist. Sie können diese Workflows mit Ereignissen in Abonnements verknüpfen.
Erweiterbarkeitsaktion	Ein optimiertes Codeskript, das nach einem auslösenden Ereignis innerhalb eines Abonnements ausgeführt werden kann. Erweiterbarkeitsaktionen ähneln Workflows, sind aber einfacher. Erweiterbarkeitsaktionen können innerhalb von vRealize Automation Cloud Assembly angepasst werden.
Aktionsausführungen	Zugreifbar über die Registerkarte Aktionsausführungen . Bei einer Aktionsausführung handelt es sich um ein detailliertes Protokoll der Erweiterbarkeitsaktionen, die als Reaktion auf auslösende Ereignisse ausgeführt wurden.

Blockieren von Ereignisthemen

Bestimmte Ereignisthemen unterstützen die Blockierung von Ereignissen. Das Verhalten eines Erweiterbarkeitsabonnements hängt davon ab, ob diese Ereignistypen vom Thema unterstützt werden und wie das Abonnement konfiguriert ist.

vRealize Automation Cloud Assembly-Erweiterbarkeitsabonnements können zwei grundlegende Typen von Ereignisthemen verwenden: nicht blockierende und blockierende Ereignisthemen. Der Typ des Ereignisthemas definiert das Verhalten des Erweiterbarkeitsabonnements.

Nicht blockierende Ereignisthemen

Mit nicht blockierenden Ereignisthemen können nur nicht blockierende Abonnements erstellt werden. Nicht blockierende Abonnements werden asynchron ausgelöst, und Sie können sich nicht auf die Reihenfolge verlassen, in der die Abonnements ausgelöst werden.

Blockieren von Ereignisthemen

Bestimmte Ereignisthemen unterstützen die Blockierung. Wenn ein Abonnement als „Blockierend“ gekennzeichnet ist, werden alle Nachrichten, die die festgelegten Bedingungen erfüllen, von keinem anderen Abonnement mit übereinstimmenden Bedingungen empfangen, bis das ausführbare Element des blockierenden Abonnements ausgeführt wird.

Blockierende Abonnements werden in der Reihenfolge der Priorität ausgeführt. Der höchste Prioritätswert ist 0 (null). Wenn Sie über mehr als ein blockierendes Abonnement für das gleiche Ereignisthema mit der gleichen Prioritätsstufe verfügen, werden die Abonnements in umgekehrter alphabetischer Reihenfolge basierend auf dem Namen des Abonnements ausgeführt. Nachdem alle blockierenden Abonnements verarbeitet wurden, wird die Nachricht gleichzeitig an alle nicht blockierenden Abonnements gesendet. Da die blockierenden Abonnements synchron ausgeführt werden, umfasst die geänderte Ereignisnutzlast das aktualisierte Ereignis, wenn die nachfolgenden Abonnements benachrichtigt werden.

Sie können blockierende Ereignisthemen verwenden, um mehrere Abonnements zu verwalten, die voneinander abhängig sind.

Sie verfügen beispielsweise über zwei Bereitstellungs-Workflow-Abonnements, bei denen das zweite Abonnement von den Ergebnissen des ersten abhängt. Beim ersten Abonnement wird während der Bereitstellung eine Eigenschaft geändert und das zweite zeichnet die neue Eigenschaft, z. B. einen Maschinennamen, in einem Dateisystem auf. Das Abonnement „ChangeProperty“ hat die Priorität 0 und „RecordProperty“ hat die Priorität 1, da das zweite Abonnement die Ergebnisse des ersten Abonnements verwendet. Bei der Bereitstellung einer Maschine wird die Ausführung des Abonnements „ChangeProperty“ gestartet. Da die Bedingungen des Abonnements „RecordProperty“ auf einer Bedingung nach der Bereitstellung basieren, löst ein Ereignis das Abonnement „RecordProperty“ aus. Da der Workflow „ChangeProperty“ aber ein blockierender Workflow ist, wird das Ereignis erst dann empfangen, wenn der Workflow abgeschlossen ist. Nachdem der Maschinename geändert und das erste Workflow-Abonnement abgeschlossen wurde, wird das zweite Workflow-Abonnement ausgeführt und der Maschinename im Dateisystem aufgezeichnet.

Ausführbares Wiederherstellungselement

Zum Blockieren von Ereignisthemen können Sie dem Abonnement ein ausführbares Wiederherstellungselement hinzufügen. Das ausführbare Wiederherstellungselement in einem Abonnement wird ausgeführt, wenn das primäre ausführbare Element ausfällt. Sie können beispielsweise ein Workflow-Abonnement erstellen, bei dem das primäre ausführbare Element ein Workflow ist, der Datensätze in einem CMDB-System, wie z. B. ServiceNow, erstellt. Selbst wenn das Workflow-Abonnement fehlschlägt, werden möglicherweise einige Datensätze im CMDB-System erstellt. In diesem Szenario kann ein ausführbares Wiederherstellungselement verwendet werden, um die vom fehlgeschlagenen ausführbaren Element im CMDB-System hinterlassenen Datensätze zu bereinigen.

Für Anwendungsfälle, die mehrere Abonnements enthalten, die voneinander abhängig sind, können Sie dem ausführbaren Wiederherstellungselement eine `ebs.recover.continuation`-Eigenschaft hinzufügen. Mit dieser Eigenschaft können Sie steuern, ob der Erweiterbarkeitsdienst mit dem nächsten Abonnement in Ihrer Kette fortgesetzt werden muss, falls das aktuelle Abonnement fehlschlägt.

Mit vRealize Automation Cloud Assembly bereitgestellte Ereignisthemen

vRealize Automation Cloud Assembly enthält vordefinierte Ereignisthemen.

Ereignisthemen

Ereignisthemen sind die Kategorien, in denen ähnliche Ereignisse zusammengefasst werden. Einem Abonnement zugewiesene Ereignisthemen definieren das Ereignis, das das Abonnement auslöst. Die folgenden Ereignisthemen werden standardmäßig mit vRealize Automation Cloud Assembly bereitgestellt. Alle Themen können zum Hinzufügen oder Aktualisieren benutzerdefinierter Eigenschaften oder Tags der Ressource verwendet werden. Wenn ein vRealize Orchestrator-Workflow oder eine Erweiterbarkeitsaktion fehlschlägt, schlägt die entsprechende Aufgabe ebenfalls fehl.

Tabelle 6-5. Cloud Assembly-Ereignisthemen

Ereignisthema	Blockierbar	Beschreibung
<code>Blueprint.configuration</code>	Nein	Wird ausgegeben, wenn ein Blueprint-Konfigurationsereignis wie das Erstellen oder Löschen eines Blueprints auftritt, und kann für die Benachrichtigung externer Systeme über solche Ereignisse nützlich sein.
<code>Blueprint.version.configuration</code>	Nein	Wird ausgegeben, wenn ein neues Blueprint-Versionereignis auftritt, wie beispielsweise das Erstellen, die Freigabe, das Aufheben der Freigabe oder die Wiederherstellung einer Version. Dieses Ereignisthema kann bei der Integration von Drittanbieter-Versionskontrollsystemen nützlich sein.
<code>Compute allocation</code>	Ja	Ereignisse, die vor der Zuteilung von <code>resourcenames</code> und <code>hostselections</code> ausgegeben wurden. Beide Eigenschaften können in dieser Phase geändert werden.
<code>Compute post provision</code>	Ja	Ereignisse, die nach der erfolgreichen Bereitstellung einer Ressource ausgegeben werden.
<code>Compute post removal</code>	Ja	Ereignisse, die nach dem Entfernen einer Computing-Ressource ausgegeben wurden.
<code>Compute provision</code>	Ja	Ereignisse, die vor dem Bereitstellen der Ressource auf dem Hypervisor auftreten. Hinweis Sie können die zugewiesene IP-Adresse ändern.
<code>Compute removal</code>	Ja	Ereignisse, die vor dem Entfernen der Ressource ausgegeben werden.

Tabelle 6-5. Cloud Assembly-Ereignisthemen (Fortsetzung)

Ereignisthema	Blockierbar	Beschreibung
Compute reservation	Ja	Ereignisse, die zum Zeitpunkt der Reservierung ausgegeben werden. Hinweis Sie können die Reihenfolge der Platzierungen ändern.
Deployment action completed	Ja	Wird nach Abschluss einer Bereitstellungsaktion ausgegeben.
Deployment action requested	Ja	Wird vor Abschluss einer Bereitstellungsaktion ausgegeben.
Deployment completed	Ja	Wird nach der Bereitstellung einer Blueprint- oder Kataloganforderung ausgegeben.
Deployment onboarded	Nein	Wird bei der Integration einer neuen Bereitstellung ausgegeben.
Deployment requested	Ja	Wird vor der Bereitstellung einer Blueprint- oder Kataloganforderung ausgegeben.
Deployment resource action completed	Ja	Wird nach der Bereitstellung einer Ressourcenaktion ausgegeben.
Deployment resource action requested	Ja	Wird vor der Bereitstellung einer Ressourcenaktion ausgegeben.
Deployment resource completed	Ja	Wird nach der Bereitstellung einer Bereitstellungsressource ausgegeben.
Deployment resource requested	Ja	Wird vor der Bereitstellung einer Bereitstellungsressource ausgegeben.
Disk allocation	Ja	Wird für die Vorabzuteilung von Datenträgerressourcen ausgegeben.
Disk post removal	Ja	Wird nach dem Löschen einer Datenträgerressource ausgegeben.
Disk post resize	Ja	Wird nach einer Größenänderung der Datenträgerressource ausgegeben.
EventLog	Ja	Verwandte Ereignisse werden protokolliert.
Kubernetes cluster allocation	Ja	Wird für die Vorabzuteilung von Ressourcen für einen Kubernetes-Cluster ausgegeben.
Kubernetes cluster post provision	Ja	Wird nach der Bereitstellung eines Kubernetes-Clusters ausgegeben.
Kubernetes cluster post removal	Ja	Wird nach dem Löschen eines Kubernetes-Clusters ausgegeben.
Kubernetes cluster provision	Ja	Wird vor der Bereitstellung eines Kubernetes-Clusters ausgegeben.

Tabelle 6-5. Cloud Assembly-Ereignisthemen (Fortsetzung)

Ereignisthema	Blockierbar	Beschreibung
Kubernetes cluster removal	Ja	Wird vor dem Initiieren des Prozesses zum Löschen eines Kubernetes-Clusters ausgegeben.
Load balancer post provision	Ja	Wird nach der Bereitstellung eines Lastausgleichsdiensts ausgegeben.
Network Configure	Ja	Ereignisse, die bei der Konfiguration des Netzwerks während der Computing-Zuteilung ausgegeben werden. Hinweis Das Thema „Netzwerkkonfiguration“ unterstützt mehrere IP-Adressen/Netzwerkkarten.
Project Lifecycle	Nein	Ereignisse, die beim Erstellen, Aktualisieren oder Löschen eines Projekts ausgegeben werden.

Ereignisschema

Nach dem Hinzufügen eines Ereignisthemas können Sie das Ereignisschema anzeigen. Dieses Schema definiert die Struktur der Ereignisnutzlast oder `inputProperties`.

Protokoll der Erweiterbarkeitereignisse

Auf der Registerkarte „Erweiterbarkeitereignisse“ wird eine Liste aller Ereignisse angezeigt, die in Ihrer Umgebung aufgetreten sind.

Sie können das Protokoll der Erweiterbarkeitereignisse mithilfe von **Erweiterbarkeit > Ereignisse** anzeigen. Darüber hinaus können Sie die Liste der Ereignisse gleichzeitig anhand einer oder mehrerer Eigenschaften filtern. Zur Anzeige weiterer Details zu einem einzelnen Ereignis klicken Sie auf die Ereignis-ID.

Bereitstellungen Blueprints Infrastruktur **Erweiterbarkeit** Marketplace GEFÜHRTES SETUP

Ergebnisse 3444 Elemente

Filtern...

ID	Zeitstempel	Ereignisthema	Benutzername	Ziel-ID	Beschreibung
eabc02cb-9dfc-4bf4-897a-93526799a c06	15.01.20, 10:55	Deployment completed	mwiemer@vmware.com		
dd89b1be-1c76-4937-a991-f58c559a5 4b4	15.01.20, 10:55	Deployment requested	mwiemer@vmware.com		
d4b40f20-9792-4c48-aec0-0a16ce7e aab6	14.01.20, 18:14	Blueprint configuration	pmartini@vmware.com		
1d86cdf8-9555-d312-1836-00d8b2a26 2e6	14.01.20, 16:46	Project Lifecycle Event Topic	project-service	1d0b482d-8396-46a1-a493-54ed2d29 9ec8	update
dfcf544c-d263-c30c-3b58-23b31be25f 6a	14.01.20, 16:31	EventLog	vro-gateway	77ef8b8b-3180-44d0-a69a-503f27ed c42c	Workflow run [77ef8b8b-3180-44d0-a69a-503f27edc42c] status changed to [FAILED]. workflowid: [f246b7b5-fe89-4da5-a640-36ffc6874069] eventid:[33c01131-c9f9-3255-a686-a67671f44bcb]

Abonnements
Bibliothek
Ereignisthemen
Aktionen
Workflows
Aktivität
Aktionsausführungen
Workflow-Ausführungen

SUPPORT

Erstellen eines Erweiterbarkeitsabonnements

Bei Verwendung von vRealize Orchestrator-Integrations- oder -Erweiterbarkeitsaktionen mit vRealize Automation Cloud Assembly können Sie Abonnements erstellen, um Ihre Anwendungen zu erweitern.

Mithilfe von Erweiterbarkeitsabonnements können Sie Ihre Anwendungen erweitern, indem Sie Workflows oder Aktionen bei bestimmten Lebenszyklusereignissen auslösen. Sie können auch Filter auf Ihre Abonnements anwenden, um boolesche Bedingungen für das angegebene Ereignis festzulegen. Beispiel: Das Ereignis und der Workflow/die Aktion werden nur ausgelöst, wenn der boolesche Ausdruck „true“ ist. Dies ist hilfreich, wenn Sie das Auslösen von Ereignissen und Aktionen steuern möchten.

Tipp Verwenden Sie im Textfeld zum Filtern von Ereignissen in Themen unter Windows die Tasten „Alt + Leertaste“ oder unter Mac die Tasten „Option + Leertaste“, um Filteroptionen anzuzeigen.

Voraussetzungen

- Cloud-Administratorbenutzerrolle
- Bei Verwendung von vRealize Orchestrator-Workflows:
 - Die Bibliothek des eingebetteten vRealize Orchestrator-Clients oder die Bibliothek einer integrierten externen vRealize Orchestrator-Instanz.
- Bei Verwendung von Erweiterbarkeitsaktionen:
 - Vorhandene Skripts für Erweiterbarkeitsaktionen. Weitere Informationen finden Sie unter [Vorgehensweise zum Erstellen von Erweiterbarkeitsaktionen](#).

Verfahren

- 1 Klicken Sie auf **Erweiterbarkeit > Abonnements**.
- 2 Klicken Sie auf **Neue Integration**.
- 3 Geben Sie die Details Ihres Abonnements ein.
- 4 Wählen Sie ein **Ereignisthema** aus.
- 5 (Optional) Legen Sie die Bedingungen für das Ereignisthema fest.
- 6 (Optional) Konfigurieren Sie gegebenenfalls das Blockierungsverhalten für das Ereignisthema.
- 7 Klicken Sie auf **Ausführbares Element** und dann im Dropdown-Menü auf **vRO-Workflow** oder **ABX-Aktion**.
- 8 Wählen Sie einen Workflow oder eine Erweiterbarkeitsaktion aus, die Sie in Ihrem Abonnement ausführen möchten.
- 9 (Optional) Um den Projektumfang des Erweiterungsabonnements zu definieren, deaktivieren Sie **Beliebiges Projekt** und klicken Sie auf **Projekte hinzufügen**.
- 10 Klicken Sie auf **Erstellen** und speichern Sie Ihr Abonnement.

Ergebnisse

Ihr Abonnement wird erstellt. Wenn ein durch das ausgewählte Ereignisthema kategorisiertes Ereignis eintritt, wird der verknüpfte vRealize Orchestrator-Workflow oder die Erweiterbarkeitsaktion initiiert und alle Abonnenten werden benachrichtigt.

Nächste Schritte

Nach dem Erstellen Ihres Abonnements können Sie einen Blueprint erstellen oder bereitstellen, um das Abonnement zu verknüpfen und zu verwenden. Darüber hinaus können Sie den Status der Workflow-Ausführung auf der Registerkarte **Erweiterbarkeit** innerhalb von vRealize Automation Cloud Assembly überprüfen. Bei Abonnements mit vRealize Orchestrator-Workflows können Sie auch die Ausführungen und den Workflowstatus über den vRealize Orchestrator-Client überwachen.

Fehlerbehebung für ein Erweiterbarkeitsabonnement

Beheben Sie Fehler bei Erweiterungsabonnements.

Wenn Ihr Abonnement fehlschlägt, ist dies häufig auf Fehler bei Ihrem Workflow oder Erweiterbarkeitsaktionsskript zurückzuführen.

Anzeigen von Themenparametern und Nutzlast

Sie können ein Skript vom Typ „Sichern der Themenparameter eines Abonnements“ verwenden, um bestimmte Parameter und die Nutzlast Ihrer virtuellen Maschine in einer beliebigen Ereignisphase anzuzeigen.

Dieses Skript eignet sich vor allem zum Debuggen und Überprüfen verfügbarer Eingaben für Ihren vRealize Orchestrator-Workflow. Um alle Parameter Ihrer virtuellen Maschine anzuzeigen, verwenden Sie folgendes Skript mit Ihrem Workflow:

```
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")
```

Versionsverlauf eines Abonnements

Wenn Ihr Abonnement fehlschlägt, können Sie den Versionsverlauf anzeigen.

Anzeigen des Versionsverlaufs des Abonnements

Auf der Registerkarte „Versionsverlauf“ können Sie den Änderungsverlauf Ihres Abonnements mit dem Benutzer und dem Änderungsdatum anzeigen. Wenn Ihr Abonnement fehlschlägt oder nicht ordnungsgemäß ausgeführt wird, kann der Versionsverlauf helfen, die Ursache zu ermitteln.

The screenshot displays the vRealize Automation Cloud Assembly interface. The top navigation bar includes tabs for 'Bereitstellungen', 'Blueprints', 'Infrastruktur', 'Erweiterbarkeit' (which is the active tab), and 'Marketplace'. The left sidebar contains a search icon and several categories: 'Ereignisse', 'Abonnements' (highlighted with a blue circle '1'), 'Bibliothek', and 'Aktivität'. The main content area is titled 'Test subscription - Versionsverlauf' (highlighted with a blue circle '2'). It features a list of subscription events and a detailed view of the selected event. The event list shows three entries, with the first one (highlighted with a blue circle '3') being the most recent. The detailed view on the right shows the JSON payload for the selected event.

Time	Owner	Subscription ID
13.01.20, 15:09 – aktuell	rishi@vmware.com	sub_1578899317099
13.01.20, 15:09	rishi@vmware.com	sub_1578899317099
13.01.20, 15:08	rishi@vmware.com	sub_1578899317099

```

{
  "id": "sub_1578899317099",
  "type": "RUNNABLE",
  "eventTopicId": "kubernetes.cluster.allocation.pre",
  "name": "Test subscription",
  "orgId": "9d9648a7-115a-4a06-a613-a0c4077469f7",
  "ownerId": "rishi@vmware.com",
  "subscriberId": "abx-svc",
  "blocking": false,
  "description": "",
  "criteria": "",
  "constraints": {
    "projectId": null
  },
  "timeout": 0,
  "broadcast": false,
}

```

1

Öffnen Sie Ihr Abonnement auf der Registerkarte **Abonnements**.

2

Um den Versionsverlauf anzuzeigen, klicken Sie auf **Versionsverlauf**.

3

Sie können auf jeden Änderungseintrag klicken, um den entsprechenden Abonnementcode anzuzeigen, der mit der Änderung verknüpft ist.

Verwalten von vRealize Automation Cloud Assembly-Bereitstellungen

7

Als vRealize Automation Cloud Assembly-Blueprint-Entwickler verwenden Sie die Registerkarte „Bereitstellung“, um Ihre Bereitstellungen zu verwalten. Sie können Fehler in Bereitstellungsprozessen beheben, Änderungen vornehmen und nicht verwendete Bereitstellungen löschen.

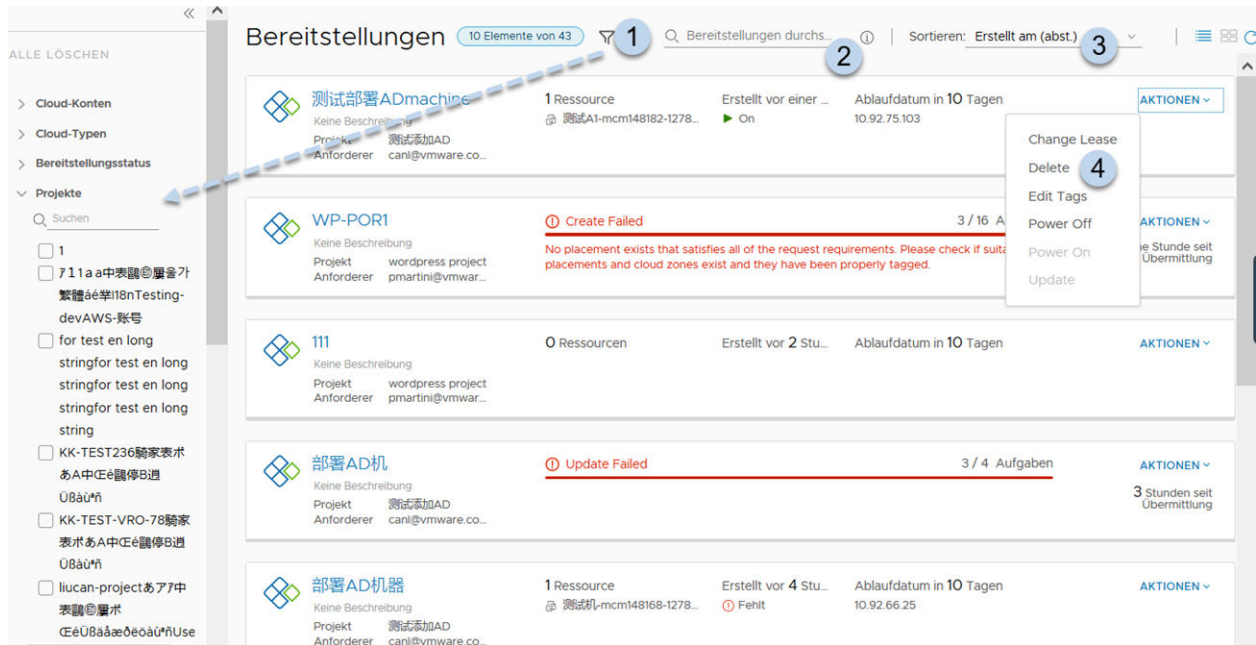
Bei Bereitstellungen handelt es sich um die bereitgestellten Instanzen von Blueprints. Auf der Registerkarte „Bereitstellungen“ werden erfolgreiche und fehlgeschlagene Bereitstellungen aufgeführt. Sie verwenden die Seite, um Ihre erfolgreichen Bereitstellungen zu verwalten oder um mit der Fehlerbehebung bei fehlgeschlagenen Anforderungen zu beginnen.

Arbeiten mit Bereitstellungskarten

Sie können mithilfe der Kartenliste nach Ihren Bereitstellungen suchen und diese verwalten. Sie können nach bestimmten Bereitstellungen suchen und diese filtern und anschließend Aktionen für diese Bereitstellungen ausführen.

- 1 Filtern Sie Ihre Anforderungen auf der Basis von Attributen.
- 2 Suchen Sie auf der Basis von Schlüsselwörtern oder Anforderer nach Bereitstellungen.
- 3 Sortieren Sie die Liste nach Uhrzeit oder Name.
- 4 Führen Sie Aktionen auf Bereitstellungsebene in der Bereitstellung aus, einschließlich Löschen nicht verwendeter Bereitstellungen zur Rückforderung von Ressourcen.

Sie können auch die Bereitstellungskosten, Ablaufdaten und Statusangaben anzeigen.



Dieses Kapitel enthält die folgenden Themen:

- Vorgehensweise zum Überwachen aktiver Bereitstellungen in vRealize Automation Cloud Assembly
- Vorgehensweise beim Fehlschlagen einer vRealize Automation Cloud Assembly-Bereitstellung
- Vorgehensweise zum Verwalten des Lebenszyklus einer abgeschlossenen vRealize Automation Cloud Assembly-Bereitstellung
- Welche Aktionen kann ich auf vRealize Automation Cloud Assembly-Bereitstellungen ausführen?

Vorgehensweise zum Überwachen aktiver Bereitstellungen in vRealize Automation Cloud Assembly

Nach dem Bereitstellen eines vRealize Automation Cloud Assembly-Blueprints können Sie Ihre Anforderung überwachen, um sicherzustellen, dass die Ressourcen bereitgestellt und aktiv sind. Beginnend mit der Bereitstellungskarte können Sie die Bereitstellung Ihrer Ressourcen überprüfen. Als Nächstes können Sie die Bereitstellungsdetails überprüfen.

Verfahren

- 1 Klicken Sie auf **Bereitstellungen** und suchen Sie gegebenenfalls mithilfe von Filtern und der Suchfunktion nach Ihrer aktuellen Bereitstellungskarte.

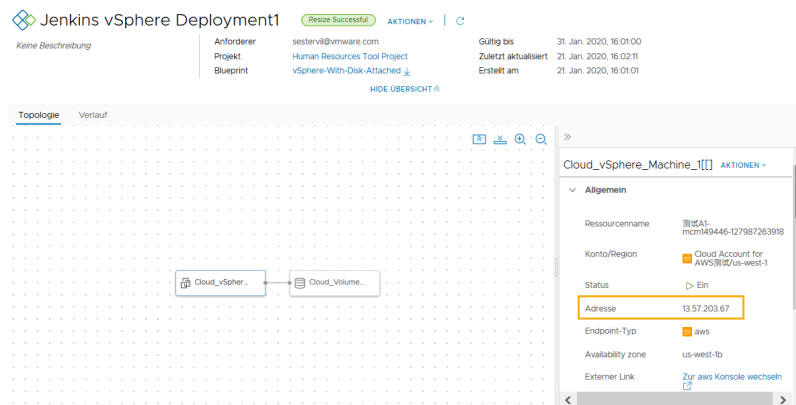
2 Überprüfen Sie den Kartenstatus.

Bei Ausführung der Bereitstellung wird auf der Prozessleiste die Anzahl der verbleibenden Aufgaben angezeigt. Bei erfolgreichem Abschluss der Bereitstellung werden auf der Karte die grundlegenden Details zur Bereitstellung angezeigt.



3 Um zu ermitteln, wo Ihre Ressourcen bereitgestellt wurden, klicken Sie auf den Namen der Bereitstellung und überprüfen Sie die Details auf der Seite „Topologie“.

Sie benötigen wahrscheinlich die IP-Adresse für die primäre Komponente. Beachten Sie beim Klicken auf jede Komponente, dass die bereitgestellten Informationen für die jeweilige Komponente spezifisch sind. In diesem Beispiel wird die IP-Adresse hervorgehoben.



Die Verfügbarkeit der externen Verknüpfung hängt vom Cloud-Anbieter ab. Wenn sie verfügbar ist, benötigen Sie die Anmeldedaten für diesen Anbieter, um auf die Komponente zuzugreifen.

Nächste Schritte

- Sie können Änderungen an Ihrer Bereitstellung vornehmen. Weitere Informationen hierzu finden Sie unter [Vorgehensweise zum Verwalten des Lebenszyklus einer abgeschlossenen vRealize Automation Cloud Assembly-Bereitstellung](#).
- Wenn die Bereitstellung fehlschlägt, finden Sie weitere Informationen unter [Vorgehensweise beim Fehlschlagen einer vRealize Automation Cloud Assembly-Bereitstellung](#).

Vorgehensweise beim Fehlschlagen einer vRealize Automation Cloud Assembly-Bereitstellung

Ihre Bereitstellungsanforderung kann aus verschiedenen Gründen fehlschlagen. Ursachen können der Netzwerkdatenverkehr, ein Mangel an Ressourcen beim Anbieter der Ziel-Cloud oder eine fehlerhafte Bereitstellungsspezifikation sein. Es ist auch möglich, dass die Bereitstellung erfolgreich verlaufen ist, aber offenbar nicht funktioniert. Sie können vRealize Automation Cloud Assembly verwenden, um Ihre Bereitstellung und alle Fehlermeldungen zu überprüfen und um

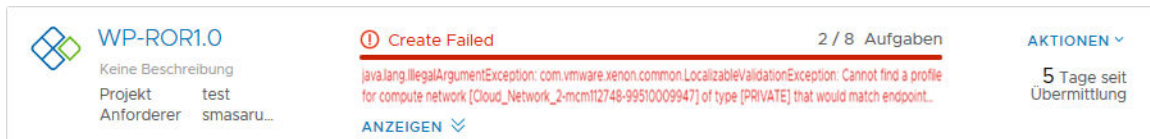
herauszufinden, ob es an der Umgebung oder der angeforderten Arbeitslastspezifikation liegt oder ob andere Gründe dafür verantwortlich sind.

Sie verwenden diesen Workflow, um mit der Recherche zu beginnen. Während dieses Prozesses finden Sie unter Umständen heraus, dass der Fehler auf ein vorübergehendes Umgebungsproblem zurückzuführen ist. Indem Sie sicherstellen, dass die Bedingungen verbessert wurden, kann dieses Problem durch eine erneute Bereitstellung der Anforderung behoben werden. In anderen Fällen müssen Sie unter Umständen andere Bereiche genau überprüfen.

Als Projektmitglied können Sie die Anforderungsdetails in vRealize Automation Cloud Assembly überprüfen.

Verfahren

- 1 Um festzustellen, ob eine Anforderung fehlgeschlagen ist, klicken Sie auf die Registerkarte **Bereitstellungen** und suchen Sie nach der Bereitstellungskarte.



Fehlgeschlagene Bereitstellungen werden auf der Karte angezeigt.

- a Schauen Sie sich die Fehlermeldung an.
- b Um weitere Informationen zu erhalten, klicken Sie auf den Bereitstellungsnamen für die Bereitstellungsdetails.

2 Klicken Sie auf der Seite mit den Bereitstellungsdetails auf die Registerkarte **Verlauf**.

WP-POR1 Create Failed AKTIONEN WP - POR1

Anforderer: pmartini@vmware.com
Projekt: wordpress project
Blueprint: WP - POR1

Gültig bis: 29. Jan. 2020, 17:09:00
Zuletzt aktualisiert: 19. Jan. 2020, 17:09:29
Erstellt am: 19. Jan. 2020, 17:09:19

HIDE ÜBERSICHT

Topologie **Verlauf**

ALLE ANFORDERUNGEN (1)

19. Jan. 2020, 17:09:19 CREATE pmartini@vmwa.

Create Failed Angefordert von: pmartini@vmware.com [Diagramm wird bereitgestellt](#)

ⓘ No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist and they have been properly tagged.

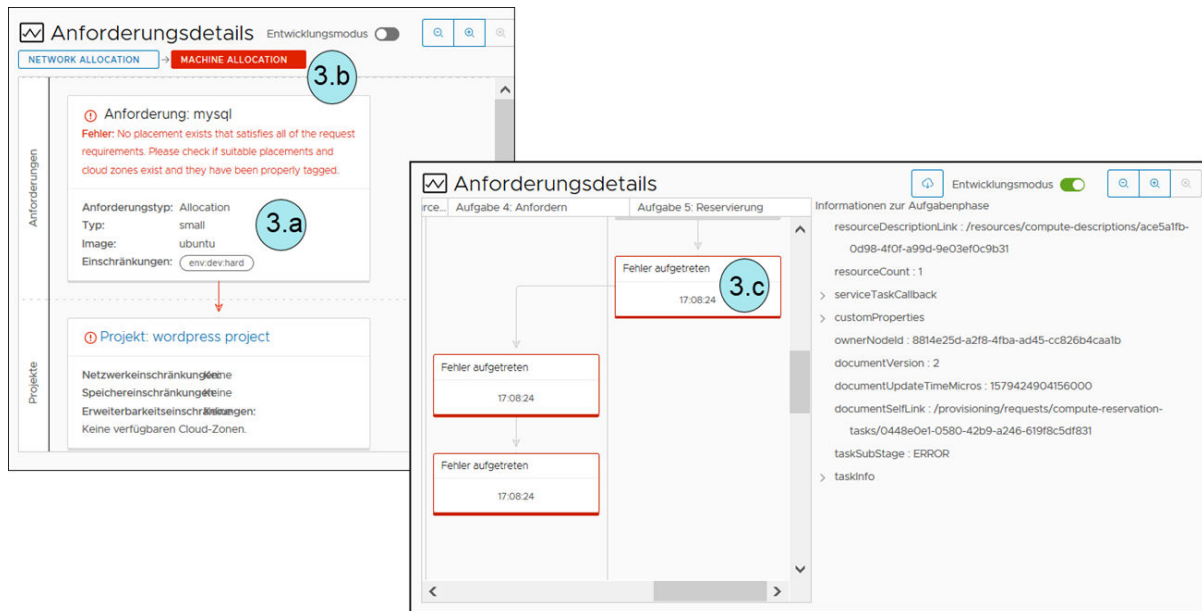
Ereignisse Anforderungsdetails

Zeitstempel	Status	Ressourcentyp	Ressourcenname	Details
19. Jan. 2020, 17:09:29	REQUEST_FAILED			No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist and they have been properly tagged.
19. Jan. 2020, 17:09:29	COMPLETION_FINISHED			
19. Jan. 2020, 17:09:21	COMPLETION_IN_PROGRESS			
19. Jan. 2020, 17:09:21	ALLOCATE_FAILED	Cloud.Machine	DBTier	No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist and they have been properly tagged.
19. Jan. 2020, 17:09:21	ALLOCATE_IN_PROGRESS	Cloud.Machine	DBTier	

- Überprüfen Sie die Ereignisstruktur, um herauszufinden, an welcher Stelle der Bereitstellungsvorgang fehlgeschlagen ist. Diese Struktur ist nützlich, wenn Sie eine Bereitstellung ändern, die Änderung aber fehlschlägt.

In der Struktur wird auch angezeigt, wann Sie Bereitstellungsaktionen ausführen. Sie können die Struktur auch zur Behebung der Fehler bei fehlgeschlagenen Änderungen verwenden.
- Unter **Details** finden Sie eine ausführlichere Version der Fehlermeldung.
- Wurde als Element ein vRealize Automation Cloud Assembly-Blueprint angefordert, können Sie mithilfe der Verknüpfung rechts neben der Meldung die vRealize Automation Cloud Assembly-Anwendung öffnen und die **Anforderungsdetails** anzeigen.

- 3 Unter **Anforderungsdetails** finden Sie den Bereitstellungs-Workflow für fehlgeschlagene Komponenten, anhand dessen Sie das Problem untersuchen können.

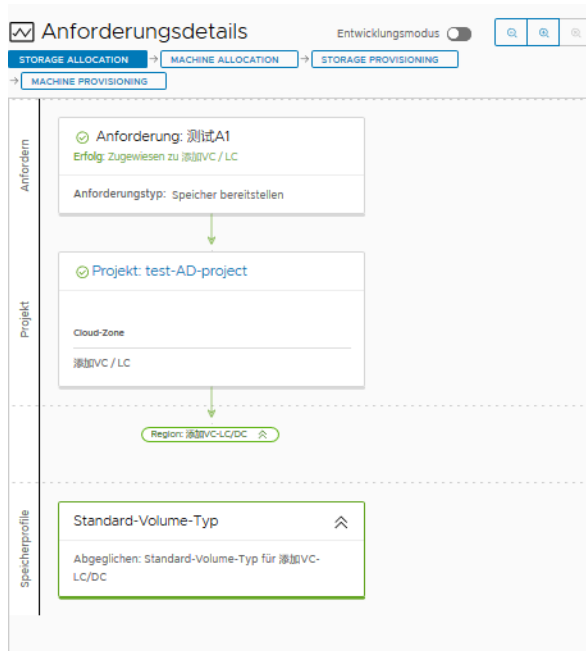


- Schauen Sie sich die Fehlermeldung an.
 - Sie können den **Entwicklungsmodus** aktivieren, um zwischen dem einfachen Bereitstellungs-Workflow und einem ausführlichen Flussdiagramm zu wechseln.
 - Klicken Sie auf die Karte, um das Bereitstellungsskript zu überprüfen.
- 4 Beheben Sie die Fehler und stellen Sie den Blueprint erneut bereit.

Die Fehler liegen möglicherweise in der Konstruktion des Blueprints oder sind auf die Konfiguration Ihrer Infrastruktur zurückzuführen.

Nächste Schritte

Nach der Behebung der Fehler und der Bereitstellung des Blueprints werden Informationen ähnlich dem folgenden Beispiel in den Anforderungsdetails angezeigt. Zur Anzeige der Anforderungsdetails wählen Sie **Infrastruktur > Aktivität > Anforderungen** aus.



Vorgehensweise zum Verwalten des Lebenszyklus einer abgeschlossenen vRealize Automation Cloud Assembly-Bereitstellung

Nach dem Bereitstellen und Ausführen einer Bereitstellung stehen Ihnen mehrere Aktionen zur Verfügung, die Sie zum Verwalten der Bereitstellung ausführen können. Die Lebenszyklusverwaltung kann das Ein- und Ausschalten, das Ändern der Größe und das Löschen einer Bereitstellung umfassen. Sie können auch verschiedene Aktionen für einzelne Komponenten ausführen, um diese zu verwalten.

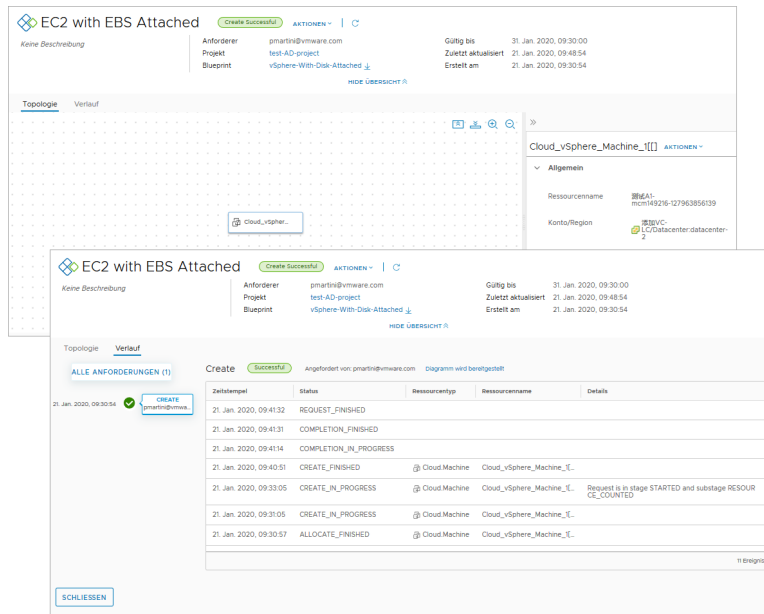
Verfahren

- 1 Klicken Sie auf **Bereitstellungen** und suchen Sie nach der Bereitstellung.
- 2 Klicken Sie für den Zugriff auf die Bereitstellungsdetails auf den Namen der Bereitstellung.

Sie können die Registerkarte „Topologie“ verwenden, um die Bereitstellungsstruktur und die Bereitstellungsressourcen anzuzeigen.

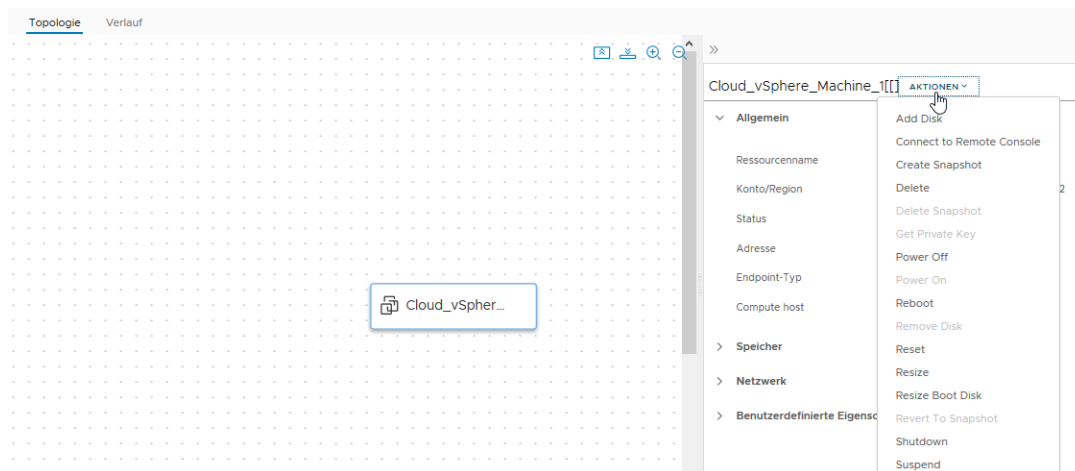
Die Registerkarte „Verlauf“ enthält alle Bereitstellungsereignisse sowie sämtliche Ereignisse, die sich auf Aktionen beziehen, die nach der Bereitstellung des angeforderten Elements ausgeführt wurden. Treten während des Bereitstellungsvorgangs Probleme auf, können Sie die Ereignisse auf der Registerkarte „Verlauf“ zur Fehlerbehebung nutzen.

Auf der Registerkarte „Kosten“ werden die aktuellen Kosten bestimmter Komponenten seit ihrer Bereitstellung angezeigt.



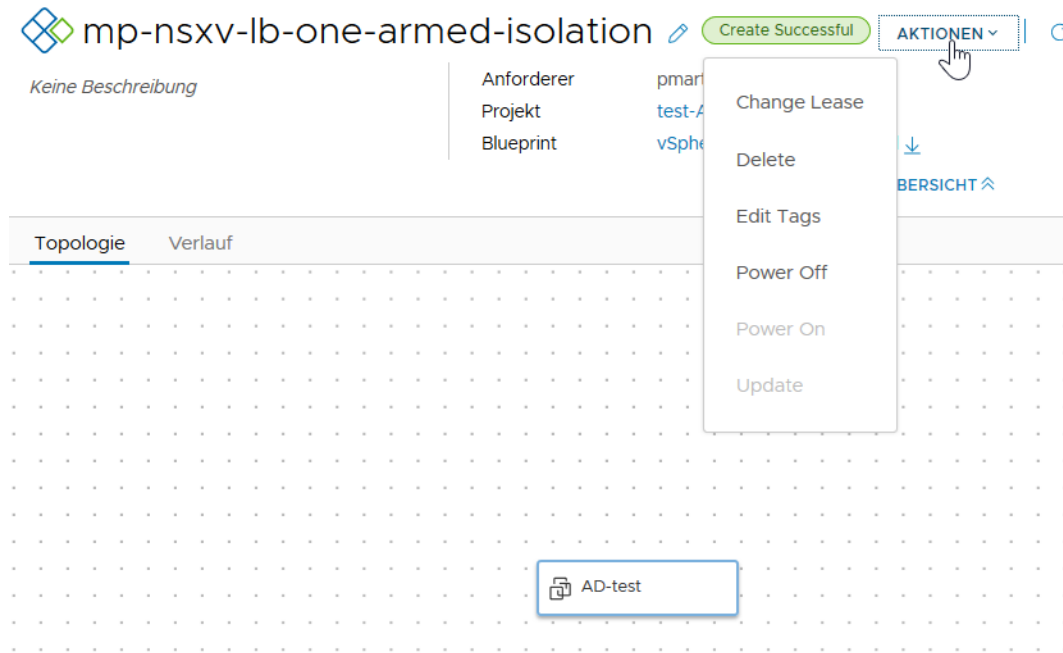
- 3 Wenn Sie feststellen, dass die aktuelle Konfiguration einer Bereitstellung zu kostspielig ist, und Sie die Größe einer Komponente ändern möchten, wählen Sie die Komponente auf der Seite „Topologie“ aus und klicken Sie dann auf der Seite zu der Komponente auf **Aktionen > Größe ändern**.

Die verfügbaren Aktionen richten sich nach der Komponente, dem Cloud-Konto und Ihren Berechtigungen.



- 4 Im Verlauf des Entwicklungslebenszyklus wird eine Ihrer Bereitstellungen nicht mehr benötigt. Um die Bereitstellung zu entfernen und Ressourcen zurückzufordern, wählen Sie **Aktionen > Löschen** aus.

Die verfügbaren Aktionen richten sich nach dem Status der Bereitstellung.



Nächste Schritte

Weitere Informationen über mögliche Aktionen finden Sie unter [Welche Aktionen kann ich auf vRealize Automation Cloud Assembly-Bereitstellungen ausführen?](#).

Welche Aktionen kann ich auf vRealize Automation Cloud Assembly-Bereitstellungen ausführen?

Nachdem Sie Blueprints bereitgestellt haben, können Sie Aktionen in vRealize Automation Cloud Assembly ausführen, um die Ressourcen zu verwalten. Die verfügbaren Aktionen hängen vom Ressourcentyp und davon ab, ob die Aktionen auf einem bestimmten Cloud-Konto oder einer Integrationsplattform unterstützt werden.

Welche Aktionen verfügbar sind, hängt auch davon ab, welche Ausführungsberechtigungen Ihr Administrator Ihnen erteilt hat.

Als Administrator oder Projektadministrator können Sie Richtlinien für Tag-2-Aktionen in vRealize Automation Service Broker einrichten. Weitere Informationen finden Sie im Abschnitt zur [Vorgehensweise für das Erteilen von Berechtigungen an Verbraucher für Service Broker Tag-2-Aktionsrichtlinien](#).

Tabelle 7-1. Liste der möglichen Aktionen

Aktion	Gilt für die folgenden Ressourcentypen	Für die folgenden Cloud-Konten oder -Integrationen	Beschreibung
Festplatte hinzufügen	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Fügt vorhandenen virtuellen Maschinen zusätzliche Festplatten hinzu.
Lease ändern	Bereitstellungen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere 	<p>Ändert das Datum und die Uhrzeit, zu denen die Lease abläuft.</p> <p>Wenn eine Lease abläuft, wird die Bereitstellung gelöscht und die Ressourcen werden zurückgefordert.</p> <p>Lease-Richtlinien werden in vRealize Automation Service Broker festgelegt.</p>
Mit Remote-Konsole verbinden	Maschinen	<ul style="list-style-type: none"> ■ VMware vSphere 	<p>Öffnen Sie eine Remotesitzung auf der ausgewählten Maschine.</p> <p>Überprüfen Sie die folgenden Anforderungen für eine erfolgreiche Verbindung.</p> <ul style="list-style-type: none"> ■ Stellen Sie als Verbraucher der Bereitstellung sicher, dass die bereitgestellte Maschine eingeschaltet ist.
Snapshot erstellen	Maschinen	<ul style="list-style-type: none"> ■ Google Cloud Platform ■ VMware vSphere 	<p>Erstellt einen Snapshot der virtuellen Maschine.</p> <p>Wenn Ihnen in vSphere nur zwei Snapshots zur Verfügung stehen und Sie diese bereits erstellt haben, ist dieser Befehl erst wieder verfügbar, nachdem Sie einen Snapshot gelöscht haben.</p>
Löschen	Bereitstellungen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	<p>Löscht eine Bereitstellung.</p> <p>Alle Ressourcen werden gelöscht und anschließend zurückgefordert.</p> <p>Wenn ein Löschvorgang fehlschlägt, können Sie die Löschaktion für eine Bereitstellung ein zweites Mal ausführen. Während des zweiten Versuchs können Sie Fehler beim Löschen ignorieren auswählen. Wenn Sie diese Option auswählen, wird die Bereitstellung gelöscht, aber die Ressourcen werden möglicherweise nicht zurückgefordert. Sie sollten die Systeme, auf denen die Bereitstellung erfolgt ist, daraufhin überprüfen, ob alle Ressourcen entfernt wurden. Ist dies nicht der Fall, müssen Sie die restlichen Ressourcen auf diesen Systemen manuell löschen.</p>
	Maschinen und Lastausgleichsdienste	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere 	Löscht eine Maschine oder einen Lastausgleichsdienst aus einer Bereitstellung. Diese Aktion könnte dazu führen, dass die Bereitstellung unbrauchbar wird.
Snapshot löschen	Maschinen	<ul style="list-style-type: none"> ■ VMware vSphere ■ Google Cloud Platform 	Löscht einen Snapshot der virtuellen Maschine.

Tabelle 7-1. Liste der möglichen Aktionen (Fortsetzung)

Aktion	Gilt für die folgenden Ressourcentypen	Für die folgenden Cloud-Konten oder -Integrationen	Beschreibung
Tags bearbeiten	Bereitstellungen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere 	Fügt Ressourcen-Tags hinzu oder ändert Ressourcen-Tags, die auf einzelne Bereitstellungsressourcen angewendet werden.
Ausschalten	Bereitstellungen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere 	Schaltet die Bereitstellung aus, ohne das Gastbetriebssystem herunterzufahren.
	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Schaltet die virtuelle Maschine aus, ohne die Gastbetriebssysteme herunterzufahren.
Einschalten	Bereitstellungen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere 	Schaltet die Bereitstellung ein. Wenn die Ressourcen angehalten wurden, wird der normale Betrieb an dem Punkt fortgesetzt, an dem die Ressourcen angehalten wurden.
	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Schaltet die Maschine ein. Wenn die Maschine angehalten wurde, wird der normale Betrieb an dem Punkt fortgesetzt, an dem die Maschine angehalten wurde.
Neu starten	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ VMware vSphere 	Startet das Gastbetriebssystem auf einer virtuellen Maschine neu. VMware Tools muss auf einer vSphere-Maschine installiert sein, damit diese Aktion verwendet werden kann.
Neu konfigurieren	Lastausgleichsdienste	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ Google Cloud Platform ■ VMware vSphere 	Ändert die Einstellungen für das Lastausgleichsprotokoll, den Port, die Systemzustandskonfiguration und die Mitgliederpools.
Festplatte entfernen	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Entfernt Festplatten von vorhandenen virtuellen Maschinen.
Zurücksetzen	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ VMware vSphere 	Erzwingt den Neustart einer virtuellen Maschine, ohne das Gastbetriebssystem herunterzufahren.
Größe anpassen	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ Google Cloud Platform ■ VMware vSphere 	Erhöht oder verringert die CPU und den Arbeitsspeicher einer virtuellen Maschine.

Tabelle 7-1. Liste der möglichen Aktionen (Fortsetzung)

Aktion	Gilt für die folgenden Ressourcentypen	Für die folgenden Cloud-Konten oder -Integrationen	Beschreibung
Größe von Startlaufwerk ändern	Maschinen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Vergrößert oder verkleinert das Startlaufwerk.
Festplattenengröße ändern	Speicherfestplatte	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Google Cloud Platform 	Erhöht die Kapazität einer Speicherfestplatte.
Neu starten	Maschinen	<ul style="list-style-type: none"> ■ Microsoft Azure 	Führt eine ausgeführte Maschine herunter und startet sie neu.
Snapshot wiederherstellen	Maschinen	<ul style="list-style-type: none"> ■ Google Cloud Platform ■ VMware vSphere 	Stellt einen vorherigen Snapshot der Maschine wieder her. Es muss ein Snapshot vorhanden sein, damit Sie diese Aktion ausführen können.
Puppet-Aufgabe durchführen	Verwaltete Ressourcen	<ul style="list-style-type: none"> ■ Puppet Enterprise 	Führt die ausgewählte Aufgabe auf Maschinen in Ihrer Bereitstellung aus. Die Aufgaben werden in Ihrer Puppet-Instanz definiert. Sie müssen in der Lage sein, die Aufgabe zu erkennen und die Eingabeparameter anzugeben.
Herunterfahren	Maschinen	<ul style="list-style-type: none"> ■ VMware vSphere 	Führt das Gastbetriebssystem herunter und schaltet die Maschine aus. VMware Tools muss auf der Maschine installiert sein, damit diese Aktion verwendet werden kann.
Anhalten	Maschinen	<ul style="list-style-type: none"> ■ Microsoft Azure ■ VMware vSphere 	Hält die Maschine an, damit sie nicht verwendet werden kann und keine Systemressourcen außer dem verwendeten Speicher verbraucht.
Aktualisieren	Bereitstellungen	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere 	Ändert die Bereitstellung basierend auf den Eingabeparametern.
Tags aktualisieren	Maschinen und Festplatten	<ul style="list-style-type: none"> ■ Amazon Web Services ■ Microsoft Azure ■ VMware vSphere 	Fügt ein Tag hinzu, das auf eine einzelne Ressource angewendet wird, bearbeitet es oder löscht es.