

# vRealize Automation Cloud Assembly 사용 및 관 리

2022년 10월

vRealize Automation 8.3

다음 VMware 웹 사이트에서 최신 기술 문서를 확인할 수 있습니다.

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

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

**VMware 코리아**  
서울시 강남구  
영동대로 517  
아셈타워 13층  
(우) 06164  
전화: +82 2 3016 6500  
팩스: +82 2 3016 6501  
[www.vmware.com/kr](http://www.vmware.com/kr)

Copyright © 2022 VMware, Inc. All rights reserved. 저작권 및 상표 정보

# 목차

## 1 vRealize Automation Cloud Assembly 개요 7

vRealize Automation Cloud Assembly 작동 방식 8

## 2 자습서 11

vSphere 인프라와 배포 설정 및 테스트 13

운영 워크로드 구성 및 프로비저닝 30

태그를 사용하여 vSphere 리소스 관리 38

다중 클라우드 인프라 및 배포 48

1부: 예제 인프라 구성 48

2부: 예제 프로젝트 생성 54

3부: 예제 클라우드 템플릿 설계 및 배포 55

VMware Cloud on AWS 구성 72

기본 VMware Cloud on AWS 워크플로 구성 72

VMware Cloud on AWS에서 격리된 네트워크 구성 85

Infoblox에 대한 외부 IPAM 통합 구성 89

다운로드 패키지를 배포하기 전에 Infoblox 애플리케이션에서 필요한 확장 가능 특성 추가 91

외부 IPAM 제공자 패키지 다운로드 및 배포 92

IPAM 통합 지점에 대한 실행 환경 생성 93

Infoblox에 대한 외부 IPAM 통합 추가 95

기존 네트워크에 대해 외부 IPAM을 사용하도록 네트워크 및 네트워크 프로파일 구성 98

외부 IPAM 제공자 범위 할당을 사용하는 클라우드 템플릿 정의 및 배포 101

클라우드 템플릿에서 IPAM 통합을 위해 Infoblox 특정 속성 사용 103

Infoblox 필터를 사용하여 네트워크 데이터 수집을 제어 107

## 3 조직에 vRealize Automation Cloud Assembly 설정 109

vRealize Automation 사용자 역할이란? 109

조직 및 서비스 사용자 역할 111

사용자 지정 사용자 역할 124

사용 사례: 사용자 역할을 통해 액세스를 제어하는 방법 127

클라우드 계정 추가 147

클라우드 계정을 사용한 작업에 필요한 자격 증명 148

Microsoft Azure 클라우드 계정 생성 166

Amazon Web Services 클라우드 계정 생성 167

Google Cloud Platform 클라우드 계정 생성 168

vCenter 클라우드 계정 생성 169

NSX-V 클라우드 계정 생성 171

NSX-T 클라우드 계정 생성	172
VMware Cloud on AWS 클라우드 계정 생성	175
VMware Cloud Foundation 클라우드 계정 생성	176
다른 애플리케이션과 통합	178
GitLab 및 GitHub 통합을 사용하는 방법	178
외부 IPAM 통합을 구성하는 방법	184
최신 외부 IPAM 통합 패키지로 업그레이드하는 방법	185
vRealize Automation Cloud Assembly에서 My VMware 통합 구성	186
Cloud Assembly에서 vRealize Orchestrator 통합 구성	187
vRealize Automation Cloud Assembly에서 Kubernetes를 사용하는 방법	190
vRealize Automation Cloud Assembly의 구성 관리	207
SaltStack Config 통합 구성	216
vRealize Automation Cloud Assembly에서 Active Directory 통합을 생성하는 방법	229
VMware SDDC Manager 통합 구성	231
vRealize Operations Manager와 통합	232
온보딩 계획이란?	244
선택한 시스템을 단일 배포로 온보딩	245
개별 배포로서 온보딩 규칙으로 필터링된 시스템	247
고급 구성	253
인터넷 프록시 서버를 구성하는 방법	253
여러 vCenter에 대한 NSX-T 매핑으로 수행할 수 있는 작업	256
NSX 클라우드 계정 연결을 제거하면 어떻게 됩니까?	257
IPAM SDK를 사용하여 제공자별 외부 IPAM 통합 패키지를 생성하는 방법	258
Azure VMware Solution 또는 Google Cloud VMware Engine과 함께 vRealize Automation 사용	258
<b>4 리소스 인프라 구축</b>	<b>260</b>
클라우드 영역을 추가하는 방법	260
클라우드 영역에 대해 알아보기	261
버전 매핑을 추가하는 방법	263
버전 매핑에 대해 알아보기	264
이미지 매핑을 추가하는 방법	264
이미지 매핑에 대해 알아보기	264
네트워크 프로파일을 추가하는 방법	268
네트워크 프로파일에 대해 알아보기	269
네트워크 설정 사용	275
보안 그룹 설정 사용	278
로드 밸런서 설정 사용	280
외부 IPAM 통합에 대해 주문형 네트워크를 지원하도록 네트워크 프로파일을 구성하는 방법	281
외부 IPAM 통합에 대해 기존 네트워크를 지원하도록 네트워크 프로파일을 구성하는 방법	284

스토리지 프로파일을 추가하는 방법	284
스토리지 프로파일에 대해 알아보기	284
가격 책정 카드를 사용하는 방법	285
vSphere 및 VMC에 대한 가격 책정 카드를 생성하는 방법	287
태그를 사용하는 방법	291
태그 지정 전략 생성	293
vRealize Automation Cloud Assembly에서 기능 태그 사용	294
vRealize Automation Cloud Assembly에서 제약 조건 태그 사용	296
표준 태그	298
vRealize Automation Cloud Assembly가 태그를 처리하는 방법	299
간단한 태그 지정 구조를 설정하는 방법	299
리소스로 작업하는 방법	301
계산 리소스	301
네트워크 리소스	301
보안 리소스	303
스토리지 리소스	304
시스템 리소스	304
볼륨 리소스	305
리소스에 대해 알아보기	305
vRealize Automation으로 다중 제공자 테넌트 리소스 구성	324
vRealize Automation용 가상 개인 영역을 생성하는 방법	324
vRealize Automation 테넌트에 대한 가상 개인 영역 구성 관리	328
vRealize Automation 테넌트에 대한 글로벌 이미지 및 버전 매핑 생성	329
vRealize Automation에 대한 테넌트 특정 이미지 및 버전 매핑 구성	332
최신 버전의 vRealize Automation에서 기존 가상 개인 영역 사용	333
<b>5 프로젝트 추가 및 관리</b>	<b>335</b>
개발 팀용 프로젝트를 추가하는 방법	335
프로젝트에 대해 알아보기	337
프로젝트 태그 및 사용자 지정 속성 사용	338
프로젝트 비용이란?	339
배포 시간에 프로젝트가 작동하는 방식	340
<b>6 배포 설계</b>	<b>342</b>
클라우드 템플릿을 생성하는 방법	343
간단한 클라우드 템플릿을 처음부터 생성하는 방법	345
리소스를 선택하고 클라우드 템플릿에 추가하는 방법	346
클라우드 템플릿 리소스를 연결하는 방법	346
유요한 클라우드 템플릿 코드를 생성하는 방법	347
다른 버전을 저장하는 방법	349

간단한 클라우드 템플릿을 향상시키는 방법	351
사용자 입력이 클라우드 템플릿을 사용자 지정할 수 있는 방법	352
리소스 플래그가 요청을 사용자 지정하는 방법	357
리소스 배포 순서를 설정하는 방법	359
표현식을 사용하여 클라우드 템플릿 코드를 보다 다양하게 만드는 방법	360
비밀 속성을 생성하고 참조하는 방법	370
클라우드 템플릿에서 원격 액세스를 사용하도록 설정하는 방법	371
다른 설계에서 동일한 속성을 재사용하는 방법	374
클라우드 템플릿에서 소켓당 코어 및 CPU 수를 정의하는 방법	383
설계에 고급 기능을 추가하는 방법	384
배포된 리소스의 이름을 사용자 지정하는 방법	384
시스템 및 디스크 클러스터를 생성하는 방법	386
클라우드 템플릿에서 시스템을 자동으로 초기화하는 방법	388
클라우드 템플릿에서 사용할 사용자 지정 리소스 유형을 생성하는 방법	401
2일차 변경을 준비하는 방법	412
확장성을 사용하여 애플리케이션 수명 주기를 연장 및 자동화하는 방법	422
리소스 속성 소개	461
일부 코드 예	461
클라우드 템플릿의 vSphere 리소스 예시	462
검토 가능한 클라우드 템플릿	465
클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예	472
사용자 이름 및 암호 액세스를 사용하는 Puppet 지원 클라우드 템플릿	495
Terraform 구성을 포함하는 방법	505
Terraform 런타임 환경 준비	505
Terraform 구성 준비	512
Terraform 구성에 대한 설계	514
Terraform 구성에 대해 알아보기	519
마켓플레이스를 사용하는 방법	521

## 7 배포 관리 523

배포를 모니터링하는 방법	525
vRealize Automation Cloud Assembly 배포 실패 시 수행할 수 있는 작업	527
완료된 배포의 수명주기를 관리하는 방법	530
배포에서 실행할 수 있는 작업	533

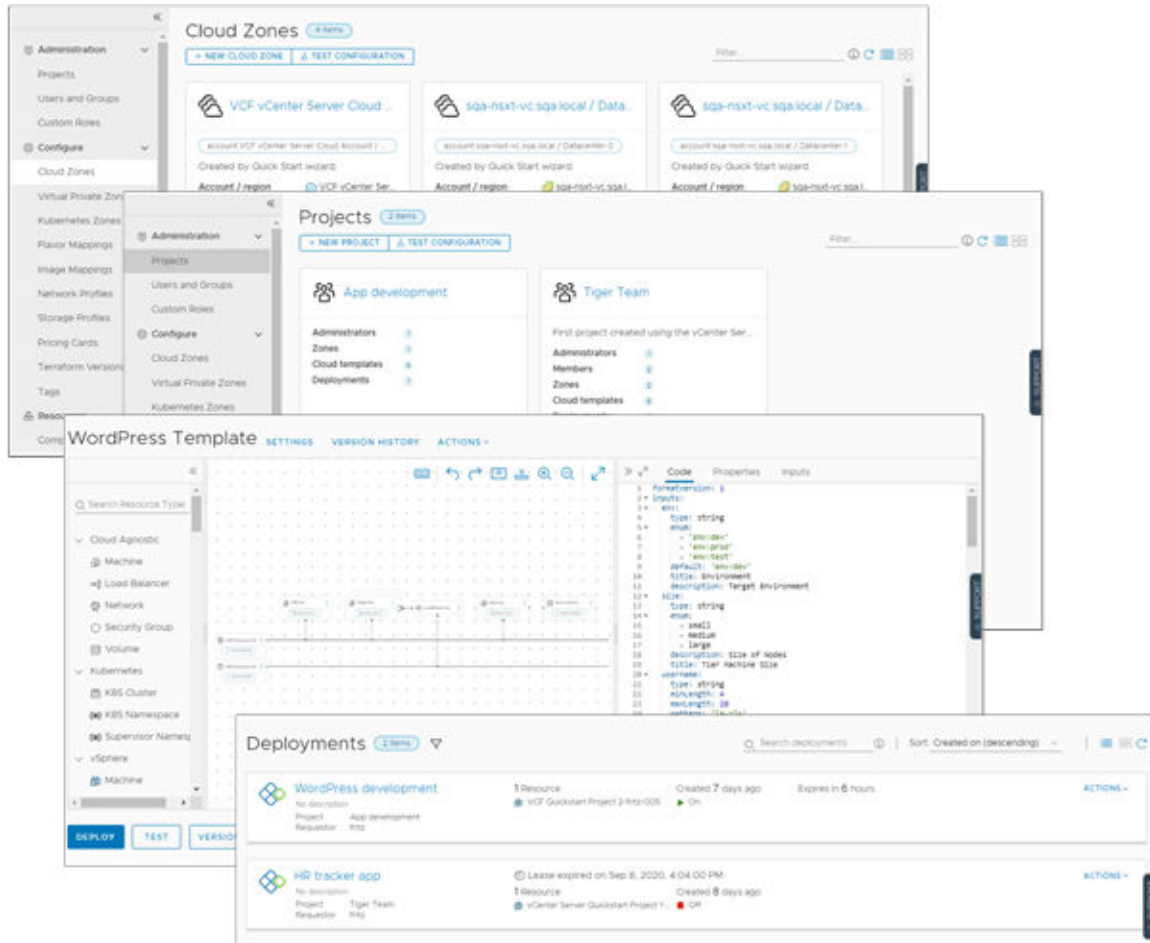
# vRealize Automation Cloud Assembly 개요

# 1

vRealize Automation Cloud Assembly는 생성한 시스템, 애플리케이션 및 서비스를 해당 리소스에 배포할 수 있도록 공용 및 사설 클라우드 공급업체에 연결하는 데 사용됩니다. 사용자와 팀 멤버는 개발에서 테스트 및 운영에 이르는 반복적인 워크플로를 지원하는 환경에서 코드로 구현되는 클라우드 템플릿을 개발합니다. 프로비저닝 시 다양한 클라우드 벤더에 걸쳐 배포할 수 있습니다. 이 서비스는 관리되는 VMware SaaS 및 NaaS 기반 프레임워크입니다.

vRealize Automation Cloud Assembly 개요에는 다음과 같은 기본 기능이 포함되어 있습니다.

- [인프라] 탭에서는 클라우드 벤더 리소스와 사용자를 추가하고 구성할 수 있습니다. 또한 이 탭에는 배포된 클라우드 템플릿에 대한 정보도 표시됩니다.
- [설계] 탭은 개발 홈 역할을 하며 캔버스 및 YAML 편집기를 사용하여 시스템과 애플리케이션을 개발하고 배포할 수 있습니다.
- [마켓플레이스] 탭에서는 OVA 또는 OVF를 지원하는 템플릿 라이브러리 및 액세스 기능을 구축하는 데 도움이 되는 VMware Solution Exchange 클라우드 템플릿 및 이미지를 제공합니다.
- [확장성] 탭에서는 애플리케이션 수명 주기를 연장하고 자동화할 수 있습니다. 확장성 작업 또는 vRealize Orchestrator 워크플로를 트리거하는 데 사용되는 이벤트를 구독할 수 있습니다.
- [배포] 탭에는 프로비저닝된 리소스의 현재 상태가 표시됩니다. 세부 정보 및 기록 정보를 활용하여 배포를 관리할 수 있습니다.
- [테넌트 관리] 탭은 서비스 제공자가 구성한 여러 테넌트를 표시하며 가상 개인 영역을 할당하거나 할당 취소할 수 있도록 합니다.



본 장은 다음 항목을 포함합니다.

- [vRealize Automation Cloud Assembly 작동 방식](#)

## vRealize Automation Cloud Assembly 작동 방식

vRealize Automation Cloud Assembly는 클라우드 템플릿 개발 및 배포 서비스입니다. 사용자와 팀 멤버는 이 서비스를 사용하여 클라우드 벤더 리소스에 시스템, 애플리케이션 및 서비스를 배포합니다.

일반적으로 클라우드 관리자라고 하는 Cloud Assembly 관리자는 프로비저닝 인프라를 설정하고 사용자와 리소스를 그룹화하는 프로젝트를 생성합니다.

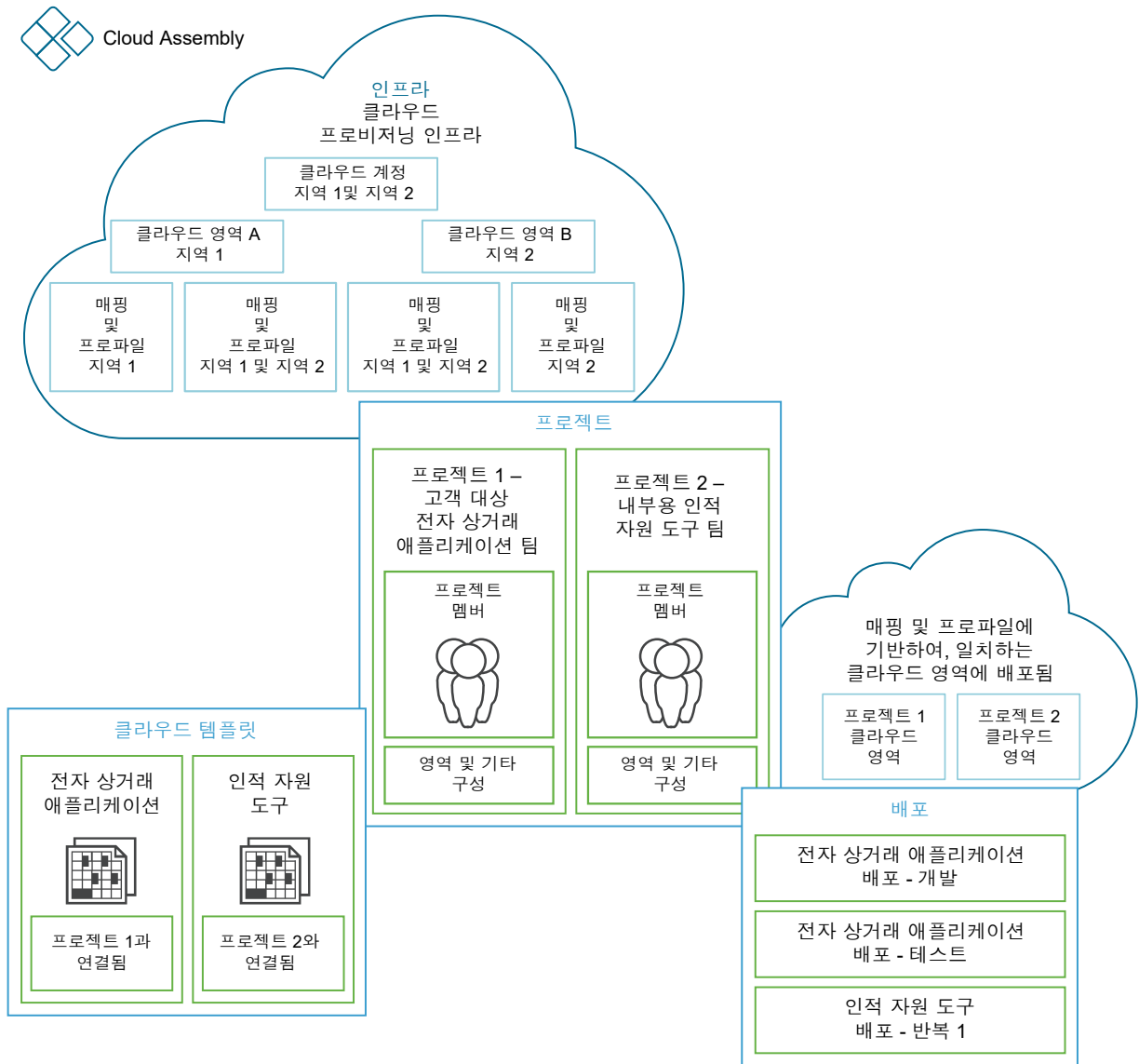
- 클라우드 벤더 계정을 추가합니다. [vRealize Automation Cloud Assembly에 클라우드 계정 추가 항목을 참조하십시오.](#)
- 개발자가 배포할 때 사용할 수 있는 지역 또는 데이터스토어, 즉 클라우드 영역을 결정합니다. [vRealize Automation Cloud Assembly 클라우드 영역에 대해 알아보기 항목을 참조하십시오.](#)
- 클라우드 영역을 정의하는 정책을 생성합니다. [장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축 항목을 참조하십시오.](#)

- 개발자를 클라우드 영역에 그룹화하는 프로젝트를 생성합니다. [vRealize Automation Cloud Assembly 프로젝트 태그 및 사용자 지정 속성 사용](#) 항목을 참조하십시오.

클라우드 템플릿 개발자는 프로젝트 하나 이상의 멤버입니다. 개발자는 클라우드 템플릿을 생성하여 프로젝트 중 하나와 연결된 클라우드 영역에 배포합니다.

- 캔버스를 사용하여 프로젝트를 위한 클라우드 템플릿을 개발합니다. 프로젝트 관리자는 마켓플레이스를 사용하여 VMware Solution Exchange에서 템플릿 및 지원 이미지를 다운로드할 수 있습니다. [장 6 vRealize Automation Cloud Assembly 배포 설계 및 vRealize Automation Cloud Assembly 마켓플레이스를 사용하는 방법](#) 항목을 참조하십시오.
- 정책과 제약 조건을 기반으로 클라우드 템플릿을 프로젝트 클라우드 영역에 배포합니다.
- 사용되지 않는 애플리케이션을 삭제하는 등 배포를 관리합니다. [장 7 vRealize Automation Cloud Assembly 배포 관리](#) 항목을 참조하십시오.

vRealize Automation Cloud Assembly를 시작합니다. 인프라를 정의한 후 클라우드 템플릿을 생성하고 배포하는 방법의 예시를 보려면 [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트](#) 항목을 참조하십시오.



# Cloud Assembly 자습서

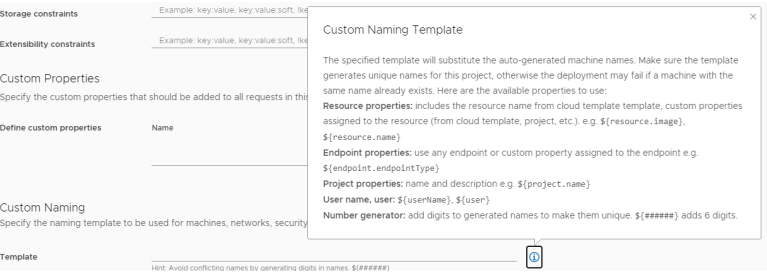
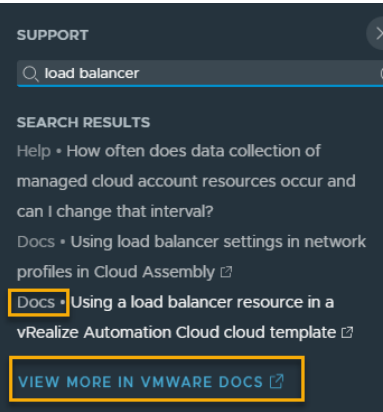
## 2

이 자습서에서는 vRealize Automation Cloud Assembly 전문가가 되는 데 도움이 되는 일반적인 작업을 수행하는 방법을 보여줍니다.

시작하면 자습서의 단계 외에도 이 가이드에 추가 정보가 제공됩니다. 관련 항목에 대한 링크가 제공됩니다.

### 사용자 지원에 액세스

사용자 지원은 애플리케이션 전체에 제공됩니다. 사용자 지원은 기능을 이해하는 데 유용하며, 텍스트 상자를 채우는 방법에 대한 결정을 내리는 데 유용합니다. 외부 설명서는 보다 심층적인, 코드 샘플 및 사용 사례를 제공합니다.

지원 유형	지원에 액세스하는 방법	예
필드 수준 표 지원 도움말	필드 옆에 있는 정보 아이콘 (i)을 클릭합니다.	
상황별 지원 패널 도움말	이름 및 조직 옆에 있는 도움말 아이콘 (?)을 클릭합니다.	
외부 설명서에 액세스	Docs 레이블이 있는 문서 제목을 클릭하거나 VMware Docs에서 더 보기를 클릭합니다.	

본 장은 다음 항목을 포함합니다.

- 자습서: vRealize Automation Cloud Assembly에서 vSphere 인프라와 배포 설정 및 테스트
- 자습서: 운영 워크로드를 프로비저닝하도록 vRealize Automation Cloud Assembly 구성
- 자습서: vRealize Automation Cloud Assembly에서 태그를 사용하여 vSphere 리소스 관리
- 자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트
- 자습서: vRealize Automation에 대해 VMware Cloud on AWS 구성
- Tutorial: vRealize Automation에 대한 제공자별 외부 IPAM 통합 구성

## 자습서: vRealize Automation Cloud Assembly에서 vSphere 인프라와 배포 설정 및 테스트

vRealize Automation를 처음 사용하거나 복습 과정이 필요한 경우, 이 자습서는 vRealize Automation Cloud Assembly 구성 프로세스를 안내합니다. 클라우드 vSphere 계정 끝점을 추가하고, 인프라를 정의하고, 프로젝트에 사용자를 추가한 다음, vSphere 리소스 유형을 기반으로 VMware Cloud Templates을 사용하여 워크로드를 설계하고 배포하는 과정을 통해 프로세스를 학습합니다.

이 자습서는 시작에 불과하지만, 다수의 공용 클라우드와 사설 클라우드에서 작동하는 셀프 서비스 자동화 및 반복적인 개발을 제공하는 것을 목표로 합니다. 이 자습서는 VMware vCenter Server 및 NSX-T에 중점을 두고 있습니다. 이 워크플로를 완료한 후에는 학습한 내용을 적용하여 더 많은 유형의 클라우드 계정을 추가하고 보다 정교한 클라우드 템플릿을 제공할 수 있습니다.

단계를 진행하는 동안 데이터 예제가 제공됩니다. 이러한 예제를 사용자의 환경에 맞는 값으로 바꾸십시오.

이 자습서의 모든 단계는 vRealize Automation Cloud Assembly에서 수행합니다.

이 자습서는 각 필수 구성 요소를 구성하는 과정을 안내합니다.

- **1단계: vCenter Server 및 NSX 클라우드 계정 추가.** 클라우드 계정은 vRealize Automation Cloud Assembly를 클라우드 벤더 끝점에 연결하는 자격 증명입니다.
- **2단계: 클라우드 영역 계산 리소스 정의.** 클라우드 영역은 계정/지역에서 선택한 다음, 프로젝트 요구 사항과 규정 준수 및 비용 관리 목표를 기반으로 서로 다른 프로젝트에 할당하는 계산 리소스입니다.
- **3단계: 계정/지역에 사용할 수 있는 가능한 리소스 구성.** 인프라 리소스는 클라우드 템플릿에 사용되는 계정/지역과 연결된 계산, 스토리지, 네트워크 및 기타 리소스의 정의입니다.
- **4단계: 프로젝트 생성.** 프로젝트는 프로젝트의 애플리케이션 개발 목표를 기반으로 사용자에게 클라우드 영역에 대한 액세스 권한을 부여하는 방법입니다.
- **5단계: 기본 클라우드 템플릿 설계 및 배포.** 클라우드 템플릿은 반복적으로 개발하고 배포하는 애플리케이션 워크로드에 대한 정의입니다.

이러한 구성 프로세스는 Cloud Assembly 개발 경험의 기초입니다. 인프라를 구축하고 클라우드 템플릿 개발 기술이 충분히 발달하면, 이 워크플로를 반복하고 확장하게 됩니다.

### 시작하기 전에

- Cloud Assembly 관리자 역할이 있는지 확인합니다. vRealize Automation의 조직 및 서비스 사용자 역할의 내용을 참조하십시오.
- vRealize Automation 콘솔에서 VMware vCenter Server 또는 VMware Cloud Foundation 빠른 시작 마법사를 아직 사용하지 않았다면 지금 사용할 수 있습니다.

마법사 기반 워크플로에는 이 자습서의 구성이 전부는 아니지만 대부분 포함됩니다.

이 자습서는 작업 인프라를 구성하고 워크로드를 배포하는 방법에 대한 이해를 높일 수 있는 실습 환경입니다.

"시작" 가이드에서 [Cloud Assembly](#)를 설정하는 방법을 참조하십시오.

- vRealize Automation Cloud Assembly에서 사용할 수 있는 단계별 설정을 아직 사용하지 않았다면 지금 사용할 수 있습니다. 단계별 설정은 이 자습서에서 수행하는 대부분의 절차를 안내합니다. 단계별 설정을 열려면 탭 표시줄의 오른쪽에서 **단계별 설정**을 클릭합니다.
- vCenter Server 및 NSX 자격 증명이 있는지 확인합니다. 자격 증명에 필요한 사용 권한에 대한 자세한 내용은 [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#)의 내용을 참조하십시오. 프로젝트에 사용자를 더 추가할 계획이면 해당 사용자가 vRealize Automation Cloud Assembly 서비스의 멤버인지 확인합니다.

## 1단계: vCenter Server 및 NSX 클라우드 계정 추가

클라우드 계정은 vRealize Automation가 vCenter Server 및 연결된 NSX 서버에 연결하는 데 사용하는 자격 증명을 제공합니다.

- 1 vCenter Server 클라우드 계정을 추가합니다.

vCenter Server 클라우드 계정은 vRealize Automation Cloud Assembly가 리소스를 검색하고 클라우드 템플릿을 배포하는 데 사용하는 vCenter 자격 증명을 제공합니다.

vCenter Server 클라우드 계정에 대한 자세한 내용은 [vRealize Automation에서 vCenter 클라우드 계정 생성](#)에서 참조하십시오.

- a **인프라 > 연결 > 클라우드 계정**을 선택합니다.
- b **클라우드 계정 추가**를 클릭하고 **vCenter**를 선택합니다.
- c 값을 입력합니다.

**New Cloud Account**

Name \* vCenter Server Account

Description

vCenter Server Credentials

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

Username \* mgmt@cmbu.local

Password \* .....

VALIDATE ✓ Credentials validated successfully. ✕

Configuration

Allow provisioning to these datacenters \* ☒ wld01-DC

☒ Create a cloud zone for the selected datacenters

NSX cloud account

Capabilities

Capability tags  ⓘ

**ADD** **CANCEL**

이러한 값은 예시일 뿐입니다. 실제 값은 사용자의 환경에 따라 다릅니다.

설정	샘플 값
이름	vCenter Server 계정
vCenter IP 주소/FQDN	your-dev-vcenter.company.com
사용자 이름 및 암호	vCenterCredentials@yourCompany.com

- d 자격 증명을 확인하려면 **검증**을 클릭합니다.
  - e 이 데이터 센터에 대한 **프로비저닝 허용**을 설정하려면 데이터 센터를 하나 이상 선택합니다.
  - f NSX 클라우드 계정을 건너뛰니다. 나중에 이를 구성하여 vCenter Server 계정을 NSX Cloud 계정에 연결할 것입니다.
  - g **추가**를 클릭합니다.
- 2 연결된 NSX 클라우드 계정을 추가합니다.

NSX-T 클라우드 계정은 vRealize Automation Cloud Assembly가 네트워크 리소스를 검색하고 클라우드 템플릿을 사용하여 네트워크를 배포하는 데 사용하는 NSX-T 자격 증명을 제공합니다.

NSX-T 클라우드 계정에 대한 자세한 내용은 [vRealize Automation](#)에서 [vCenter 클라우드 계정 생성](#)에서 참조하십시오.

- 인프라 > 연결 > 클라우드 계정을 선택합니다.
- 클라우드 계정 추가를 클릭하고 NSX-T 또는 NSX-V를 선택합니다. 이 자습서에서는 **NSX-T**를 사용합니다.
- 값을 입력합니다.

이러한 값은 예시일 뿐입니다. 실제 값은 사용자의 환경에 따라 다릅니다.

설정	샘플 값
이름	NSX-T 계정
vCenter IP 주소/FQDN	your-dev-NSX-vcenter.company.com
사용자 이름 및 암호	NSXCredentials@yourCompany.com
NSX 모드	무엇을 선택해야 할지 잘 모르겠다면 제품 내 도움말에 사용하여 도움을 받을 수 있습니다. 필드 오른쪽에서 정보 아이콘을 클릭합니다. 필드 수준 도움말은 옵션을 구성하는 데 도움이 될 수 있는 정보를 포함하고 있습니다. 이 예에서는 <b>정책</b> 을 선택합니다.

- 자격 증명을 확인하려면 **검증**을 클릭합니다.

- e 이전 단계에서 생성한 vCenter 클라우드 계정을 연결하려면 **추가**를 클릭한 다음 **vCenter 계정**을 선택합니다.

이 vCenter 클라우드 계정 연결은 네트워크 보안을 보장합니다.

- f NSX Cloud 계정 페이지에서 **추가**를 클릭합니다.

## 2단계: 클라우드 영역 계산 리소스 정의

클라우드 영역은 계정/지역의 계산 리소스 그룹으로, 프로젝트에서 사용할 수 있게 됩니다. 프로젝트 멤버는 할당된 클라우드 영역의 리소스를 사용하여 클라우드 템플릿을 배포합니다. 프로젝트 클라우드 템플릿이 배포되는 위치를 보다 세부적으로 제어하려는 경우에는 여러 계산 리소스를 사용하여 여러 클라우드 영역을 생성할 수 있습니다.

계정/지역은 클라우드 벤더가 리소스를 격리된 지역이나 데이터스토어에 연결하는 방법입니다. 계정은 클라우드 계정 유형을 나타내고 지역은 지역 또는 데이터스토어를 나타냅니다. vCenter Server는 데이터스토어를 사용하고 프로비저닝 리소스는 선택한 클러스터 및 리소스 풀입니다.

이 자습서에서는 프로젝트 개발 팀의 목표와 예산 및 관리 요구 사항을 지원하는 리소스가 클라우드 영역에 포함되어 있는지 확인해야 합니다.

클라우드 영역에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 클라우드 영역에 대해 알아보기](#) 항목을 참조하십시오.

- 1 **인프라 > 구성 > 클라우드 영역**을 선택합니다.

- 2 vCenter Server 인스턴스에 대해 추가된 클라우드 영역을 클릭하고 값을 입력합니다.

**vCenter Account Cloud Zone**
DELETE

Summary
Compute
Projects

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

Account / region \*

vCenter Account / wld01-DC

Name \*

vCenter Account Cloud Zone

Description

Placement policy \*

DEFAULT

Folder

Select folder

Capabilities

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

Capability tags

Enter capability tags

SAVE

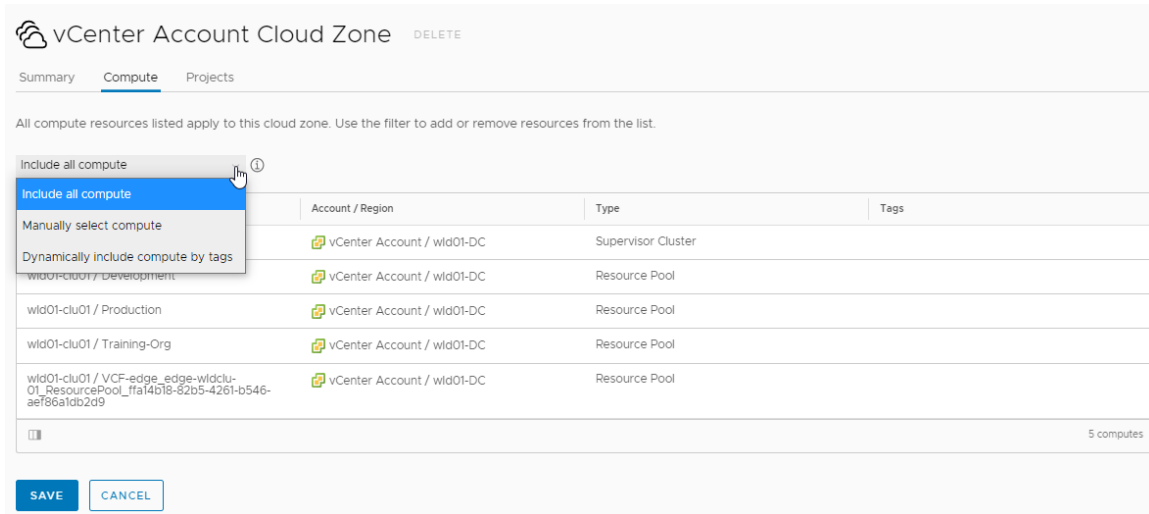
CANCEL

설정	샘플 값
계정/지역	vCenter 계정/데이터 센터 이름
이름	vCenter Server 클라우드 영역 이 값을 생성한 후에는 변경할 수 없습니다. 다른 vCenter Server에 대해 다른 데이터 센터를 구성하려면 계정/지역을 선택할 수 있는 새 클라우드 영역을 생성해야 합니다.
설명	개발을 위한 모든 vCenter Server 계산 리소스입니다.
정책	기본값 필드 값에 대한 질문이 있는 경우에는 도움말을 참조하는 것을 잊지 마십시오.

모든 값은 예시일 뿐입니다. 영역 세부 사항은 사용자 환경에 따라 다릅니다.

### 3 계산 탭을 클릭하고 계산 리소스가 모두 존재하는지 확인합니다.

하나를 제외해야 하는 경우 **수동으로 계산 선택**으로 전환하고 클라우드 영역에 포함하려는 항목만 추가합니다.



4 저장을 클릭합니다.

5 추가 클라우드 영역에 대해 이 프로세스를 반복합니다. 이때, 영역 이름이 고유한지 확인해야 합니다.

### 3단계: 계정/지역에 사용할 수 있는 가능한 리소스 구성

계정/지역을 클라우드 영역에 추가했습니다. 이제 클라우드 영역에 대해 가능한 시스템 크기(버전 매핑), 이미지 매핑, 네트워크 프로파일 및 스토리지 프로파일을 정의합니다. 클라우드 템플릿을 배포할 때 매핑 및 프로파일 정의가 일치하는지 평가되어 적절한 시스템 크기(버전), 이미지, 네트워크 및 스토리지가 워크로드에 포함되는지 확인합니다.

1 계정/지역에 대한 버전 매핑을 구성합니다.

버전 매핑을 티셔츠 크기 조정이라고도 합니다. 클라우드 템플릿이 구성된 방식에 따라, 적용된 버전 매핑이 CPU 및 메모리의 수를 결정합니다.

버전 매핑에 대한 자세한 내용은 [vRealize Automation의 버전 매핑에 대해 알아보기](#) 항목을 참조하십시오.

a 인프라 > 구성 > 버전 매핑을 선택합니다.

b 새 버전 매핑을 클릭하고 소규모, 중간 규모 및 대규모 시스템을 정의하는 값을 입력합니다.

사용된 값은 샘플 값입니다. 관련 계정/지역을 선택하고 크기를 정의해야 합니다.

The screenshot shows the 'small' flavor configuration page. It includes a 'Flavor name' field with the value 'small' and a 'Configuration' table. The table has two columns: 'Account / Region' and 'Value'. The first row shows 'vCenter Account / wld01-DC' for the account and '2' for the value. The second row shows '1' for the value. The unit is set to 'GB'.

설정	샘플 값
버전 이름	small
계정/지역	vCenter 계정/데이터 센터
CPU 값	2
메모리 값	1GB

- c **생성**을 클릭합니다.
- d 추가 크기를 생성하려면 계정/지역에 대해 중형 및 대형 버전 매핑을 구성합니다.

설정	샘플 값
버전 이름	medium
계정/지역	vCenter 계정/데이터 센터
CPU 값	4
메모리 값	2 GB
버전 이름	large
계정/지역	vCenter 계정/데이터 센터
CPU 값	8
메모리 값	4 GB

## 2 계정/지역에 대한 이미지 매핑을 구성합니다.

이미지는 클라우드 템플릿에 있는 시스템의 운영 체제입니다. vCenter Server 이미지를 사용하는 경우에는 vCenter 템플릿을 선택합니다.

이미지 매핑에 대한 자세한 내용은 [vRealize Automation의 이미지 매핑에 대해 알아보기](#) 항목을 참조하십시오.

- a **인프라 > 구성 > 이미지 매핑**을 선택합니다.
- b **새 이미지 매핑**을 클릭하고 계정/지역에 대한 이미지를 검색합니다.
- 사용된 값은 샘플 값입니다. 해당 계정/지역에서 검색된 관련 이미지를 선택해야 합니다.

설정	샘플 값
이미지 이름	centos
계정/지역	vCenter 계정
이미지	centos7

c **생성**을 클릭합니다.

d 이 프로세스를 반복하여 추가 이미지 매핑을 생성합니다. 예: 계정/지역에 대한 Ubuntu 매핑.

### 3 네트워크 프로파일을 구성합니다.

네트워크 프로파일은 계정/지역에 사용할 수 있는 네트워크 및 네트워크 설정을 정의합니다. 프로파일은 대상 배포 환경을 지원해야 합니다.

이 작업은 성공을 위한 최소 구성 정보를 제공합니다. 네트워크 프로파일에 대해 자세히 알아보려면 [vRealize Automation의 네트워크 프로파일에 대해 알아보기](#)부터 참조하십시오.

a **인프라 > 구성 > 네트워크 프로파일**을 선택합니다.

b **새 네트워크 프로파일**을 클릭하고 vCenter 계정/데이터 센터 계정/지역에 사용할 프로파일을 생성합니다.

Network Profile [DELETE](#)

Summary Networks Network Policies Load Balancers Security Groups

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

Account / region vCenter Account / wld01-DC

Name \* Network Profile

Description Networks for development teams.

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

Capability tags Enter capability tags

설정	샘플 값
계정/지역	vCenter 계정/데이터 센터
이름	네트워크 프로파일
설명	개발 팀의 네트워크입니다.

- c 네트워크 탭을 클릭하고 네트워크 추가를 클릭합니다.

Network Profile [DELETE](#)

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

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

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

<input type="checkbox"/>	Name	Account / Region	Zone	Network Domain	CIDR	Support Public IP	Default for Zone	Origin	Tags
<input type="checkbox"/>	DevProject-004	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.1.64/27	--	--	Deployed	
<input type="checkbox"/>	External-mcm1373520-150877845350	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	172.16.1.64/28	--	--	Discovered	
<input type="checkbox"/>	seg-domain-c8e2a5580e-2772-43f5-9ea-eddc05e35996-vmware-system-nsx-0	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	10.244.0.0/28	--	--	Discovered	<a href="#">external_id8...</a> <a href="#">ncp/project_u...</a> <a href="#">ncp/cluster d...</a> <a href="#">ncp/version1...</a> <a href="#">ncp/project.v...</a>

1 - 3 of 3 networks

- d 애플리케이션 개발 팀에서 사용하도록 설정할 NSX 네트워크를 선택합니다.

이 예에는 DevProject-004라는 NSX-T 네트워크가 있습니다.

- e 네트워크 정책 탭을 클릭하고 정책을 생성합니다.

설정	샘플 값
격리 정책	없음
Tier-0 논리적 라우터	Tier-0-router
Edge 클러스터	EdgeCluster

f **생성**을 클릭합니다.

#### 4 스토리지 프로파일을 구성합니다.

스토리지 프로파일은 계정/지역에 대한 디스크를 정의합니다. 프로파일은 대상 배포 환경을 지원해야 합니다.

스토리지 프로파일에 대한 자세한 내용은 [vRealize Automation의 스토리지 프로파일에 대해 알아보기](#)에서 참조하십시오.

a **인프라 > 구성 > 스토리지 프로파일**을 선택합니다.

b **새 스토리지 프로파일**을 클릭하고 vCenter Server/데이터 센터 계정/지역에 사용할 프로파일을 생성합니다.

테이블에 지정되어 있는 경우가 아니면 기본값을 유지합니다.

설정	샘플 값
계정/지역	vCenter 계정/데이터 센터
이름	스토리지 프로파일
데이터스토어/클러스터	용량이 충분하고 모든 호스트에서 액세스할 수 있는 데이터스토어를 선택했습니다.
이 지역의 기본 설정 스토리지	확인란을 선택합니다.

c 생성을 클릭합니다.

## 4단계: 프로젝트 생성

프로젝트 목표에 대해 실제로 생각하기 시작하는 단계입니다.

- 애플리케이션 클라우드 템플릿을 생성하고 배포하기 위해 계산 리소스에 액세스해야 하는 사용자는 누구입니까? 다른 프로젝트 역할을 통해 보고 수행할 수 있는 작업에 대한 자세한 내용은 [vRealize Automation의 조직 및 서비스 사용자 역할](#)의 내용을 참조하십시오.
- 프로젝트의 멤버가 개발에서 운영으로 이동하는 애플리케이션을 생성합니까? 필요한 리소스는 무엇입니까?
- 어떤 클라우드 영역이 필요합니까? 프로젝트의 각 영역에 어떤 우선 순위와 제한을 두어야 합니까?

이 자습서에서는 사내 소프트웨어 애플리케이션을 생성하고 확장하는 개발 팀을 지원합니다.

이 작업은 성공을 위한 최소 구성 정보를 제공합니다. 프로젝트에 대해 자세히 알아보려면 [vRealize Automation Cloud Assembly 프로젝트에 대해 알아보기](#)부터 참조하십시오.

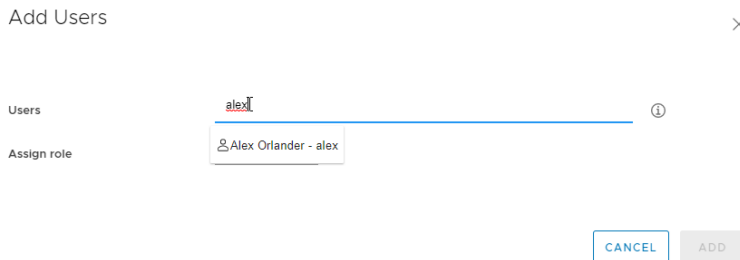
1 **인프라 > 관리 > 프로젝트**를 선택합니다.

2 **새 프로젝트**를 클릭하고 **Development Project**를 이름으로 입력합니다.

3 **사용자** 탭을 클릭한 후 **사용자 추가**를 클릭합니다.

이때 사용자를 추가할 필요는 없습니다. 하지만 다른 사용자가 클라우드 템플릿을 사용하여 작업하도록 하려면 해당 사용자는 프로젝트의 멤버여야 합니다.

4 이메일 주소를 입력하여 개별 사용 권한에 따라 사용자를 프로젝트 멤버 또는 관리자로 추가합니다.

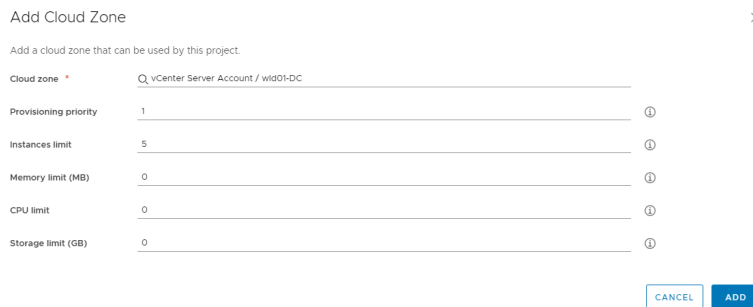


The 'Add Users' dialog box shows a search for 'alex' in the 'Users' field. Below, the 'Assign role' dropdown is set to 'Alex Orlander - alex'. At the bottom are 'CANCEL' and 'ADD' buttons.

5 **프로비저닝**을 클릭하고 **영역 추가 > 클라우드 영역**을 클릭합니다.

6 사용자가 배포할 수 있는 클라우드 영역을 추가합니다.

프로젝트의 클라우드 영역에 대한 리소스 제한을 설정할 수도 있습니다. 나중에 다른 프로젝트에 대해 다른 제한을 설정할 수 있습니다.



The 'Add Cloud Zone' dialog box shows a table for configuring cloud zone settings. The 'Cloud zone' field is set to 'Q vCenter Server Account / wid01-DC'. Other fields include 'Provisioning priority' (1), 'Instances limit' (5), 'Memory limit (MB)' (0), 'CPU limit' (0), and 'Storage limit (GB)' (0). At the bottom are 'CANCEL' and 'ADD' buttons.

프로젝트 클라우드 영역 설정	샘플 값
클라우드 영역	vCenter 계정 클라우드 영역
프로비저닝 우선 순위	1
인스턴스 제한	5

7 프로젝트에 클라우드 영역을 더 추가합니다.

8 **생성**을 클릭합니다.

- 9 프로젝트가 클라우드 영역에 추가되었는지 확인하려면 **인프라 > 구성 > 클라우드 영역**을 선택하고 vCenter 계정 영역 클라우드 영역 카드를 열어 **프로젝트** 탭을 검토합니다. Development Project가 표시되어야 합니다.

## 5단계: 기본 클라우드 템플릿 설계 및 배포

클라우드 템플릿을 설계 및 배포하여 인프라가 템플릿을 지원하도록 적절하게 구성되었는지 확인합니다. 나중에 프로젝트 요구 사항을 충족하는 애플리케이션을 생성할 때 템플릿에 구축할 수 있습니다.

클라우드 템플릿을 구축하는 가장 좋은 방법은 구성 요소별로 구축하는 것으로, 각 변경 사이에 배포되는지 확인합니다. 이 자습서에서는 간단한 시스템부터 시작하여 더 많은 리소스를 반복적으로 추가합니다.

이 절차의 예에서는 YAML 코드 편집기를 사용합니다. 이것은 코드 조각을 제공하는 더 쉬운 방법입니다. 그러나 대화 상자 기반 사용자 인터페이스를 사용하려는 경우에는 **입력**을 클릭합니다.

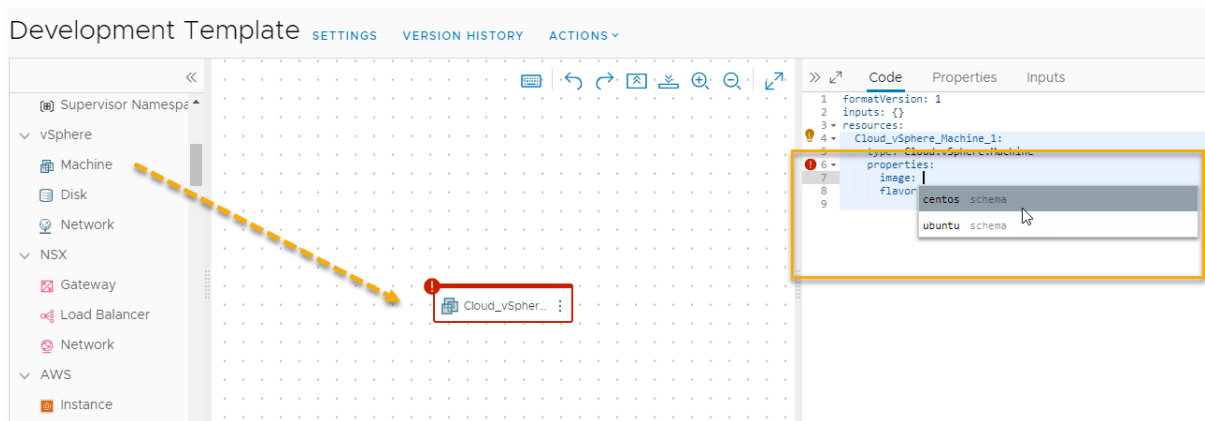
클라우드 템플릿으로 할 수 있는 일은 이 자습서에 제공된 것보다 훨씬 더 많습니다. 자세히 알아보려면 [장 6 vRealize Automation Cloud Assembly 배포 설계부터 참조하십시오](#).

이 자습서에서는 vSphere 및 NSX 리소스 유형을 사용합니다. 이러한 리소스 유형은 vCenter Server 클라우드 계정 끝점에만 배포할 수 있습니다. 클라우드 애그노스틱 리소스 유형을 사용하여 모든 끝점에 배포할 수 있는 클라우드 템플릿을 생성할 수도 있습니다. 인프라를 구성하고 끝점에 대한 템플릿을 설계하는 방법에 대한 예는 [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트](#)의 내용을 참조하십시오.



이 절차의 기본 단계를 설명하는 비디오는 기본 클라우드 템플릿을 설계하고 배포하는 방법을 참조하십시오.

- 1 **설계 > 클라우드 템플릿**을 선택합니다.
- 2 **새로 만들기 > 빈 캔버스**를 선택합니다.
- 3 **이름**으로 **Development Template**을 입력하고 **프로젝트**로 **Development Project**를 선택한 다음 **생성**을 클릭합니다.
- 4 설계 캔버스에 vSphere 시스템을 추가하고 테스트 및 배포합니다.



- a [리소스 유형] 창에서 **vSphere 시스템**을 캔버스로 끌어옵니다.

**코드** 창에 시스템에 대한 **YAML**이 표시되고 비어 있는 이미지 값, 미리 정의된 **CPU** 및 메모리 속성이 표시됩니다. 유연한 크기 조정을 지원하도록 이 템플릿을 구성할 수 있습니다.

- b 이미지 값을 선택하려면 **image**의 작은따옴표 사이에 포인터를 두고 구성된 이미지 목록에서 **centos**를 선택합니다.

사용된 값은 샘플 값입니다. **centos** 이미지를 구성하지 않은 경우에는 직접 구성된 이미지를 선택합니다.

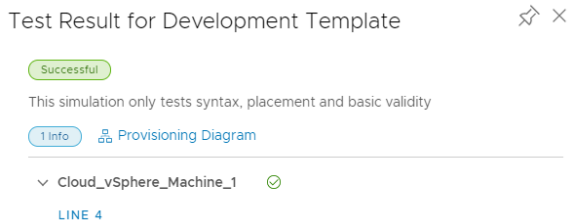
- c 이미지 속성 아래에 줄을 만들고 **flavor**를 입력하거나 선택한 다음 목록에서 **small**을 선택합니다.
- d **cpuCount** 및 **totalMemory**를 삭제합니다.

**YAML**은 이 예와 유사합니다.

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

- e **테스트**를 클릭합니다.

테스트를 통해 클라우드 템플릿의 구문 및 배치를 검증할 수 있습니다. 테스트에 성공한다고 해서 항상 템플릿을 오류 없이 배포할 수 있는 것은 아닙니다.



테스트에 실패하면 **프로비저닝 다이어그램**을 클릭하고 실패 지점을 찾습니다. 문제 해결을 위한 다이어그램 사용에 대한 자세한 내용은 **기본 클라우드 템플릿 테스트**의 내용을 참조하십시오.

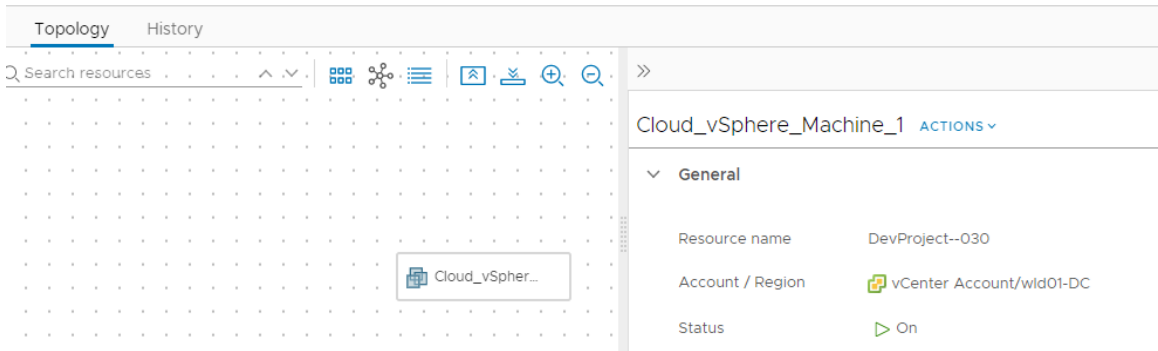
- f **배포**를 클릭합니다.

- g **배포 이름**으로 **DevTemplate - machine**을 입력하고 **배포**를 클릭합니다.

**DevTemplate** 배포 세부 정보 페이지 또는 **[배포]** 탭에서 배포의 진행률을 추적할 수 있습니다.

배포에 실패하는 경우 문제를 해결하고 템플릿을 수정할 수 있습니다. **vRealize Automation Cloud Assembly** 배포 실패 시 수행할 수 있는 작업의 내용을 참조하십시오.

배포에 성공하면 **[배포]** 탭의 이 예와 유사합니다.

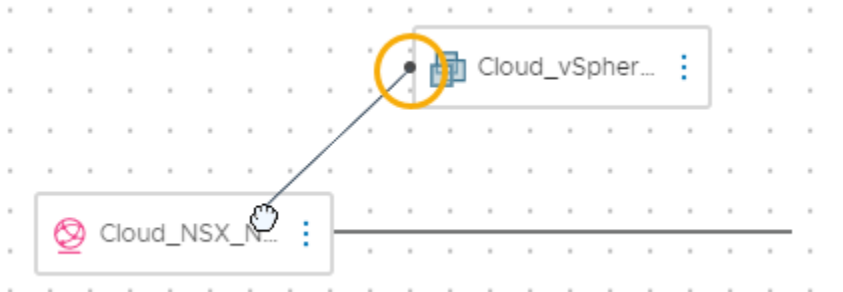


##### 5 템플릿에 버전을 지정하고 네트워크를 추가합니다.

클라우드 템플릿 버전 관리는 **Service Broker** 카탈로그에서 해당 템플릿을 사용할 수 있도록 하는 데 필요하지만 개발 중에 되돌릴 수 있는 좋은 버전이 있으면 유용합니다.

- a 설계 캔버스에서 템플릿을 엽니다.
- b 버전을 클릭하고 **Simple deployable machine**과 유사한 **설명**을 입력한 후 **생성**을 클릭합니다.
- c [리소스 유형] 창에서 **NSX 네트워크** 리소스 유형을 캔버스로 끌어옵니다.
- d 시스템을 네트워크에 연결합니다.

시스템 구성 요소에서 작은 원을 클릭하고 연결을 네트워크로 끌어옵니다.

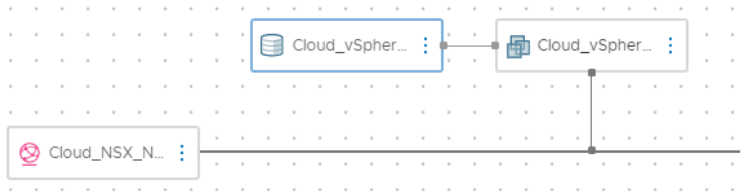


이제 **YAML**이 이 예와 유사하게 표시됩니다.

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

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

- e 템플릿을 검증하려면 **테스트**를 클릭합니다.
  - f **배포**를 클릭합니다.
  - g 이름으로 **DevTemplate - machine - network**를 입력하고 **배포**를 클릭합니다.
  - h 진행률을 추적하고 성공적인 배포를 검토합니다.
- 6 템플릿에 버전을 지정하고 데이터 디스크를 추가합니다.
- a 설계 캔버스에서 템플릿을 엽니다.
  - b 템플릿에 버전을 지정합니다.
- 설명으로 **Machine with existing network**를 입력합니다.
- c [리소스 유형] 창에서 **vSphere 디스크** 리소스 유형을 캔버스에 끌어옵니다.
  - d 디스크를 시스템에 연결합니다.



이제 **YAML**이 이 예와 유사하게 표시됩니다.

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

- e 템플릿을 테스트합니다.
- f **DevTemplate - machine - network - storage**라는 이름으로 템플릿을 배포합니다.
- g 진행률을 추적하고 성공적인 배포를 검토합니다.
- h 템플릿에 버전을 지정합니다.

설명으로 **Machine with existing network and storage disk**를 입력합니다.

이 최종 버전을 통해 작업 템플릿을 서비스 카탈로그에 추가할 수 있습니다.

## 자습서 결과

Cloud Assembly를 올바르게 작동하는 시스템으로 구성된 워크플로를 완료했습니다. 이제 다음과 같은 개념을 숙지하고 있습니다.

- 클라우드 계정은 vRealize Automation Cloud Assembly를 클라우드 벤더 끝점에 연결하는 자격 증명입니다.
- 클라우드 영역은 계정/지역에서 선택한 다음, 프로젝트 요구 사항과 비용 관리 목표를 기반으로 서로 다른 프로젝트에 할당하는 계산 리소스입니다.
- 인프라 리소스는 클라우드 템플릿에 사용되는 계정/지역과 연결된 리소스의 정의입니다.
- 프로젝트는 프로젝트의 애플리케이션 개발 목표를 기반으로 사용자에게 클라우드 영역에 대한 액세스 권한을 부여하는 방법입니다.
- 클라우드 템플릿은 반복적으로 개발하고 배포하는 애플리케이션 워크로드에 대한 정의입니다.

이 자습서는 vRealize Automation Cloud Assembly 개발 경험의 기초입니다. 이 프로세스를 사용하여 인프라를 구축하고 클라우드 템플릿 개발 기술을 발전시킬 수 있습니다.

## 자습서: 운영 워크로드를 프로비저닝하도록 vRealize Automation Cloud Assembly 구성

클라우드 관리자가 프로젝트에 대한 배포 프로세스를 자동화하면 클라우드 템플릿 디자이너가 템플릿을 생성하고 배포할 때 vRealize Automation Cloud Assembly에서 작업이 수행됩니다. 예를 들어 워크로드는 특정 사용자 지정 시스템 이름 패턴으로 배포되고, 시스템은 특정 **Active Directory** 조직 구성 단위에 추가되며, 특정 **DNS** 및 **IP** 범위가 사용됩니다.

프로젝트 배포에 대한 프로세스를 자동화하면 다양한 데이터 센터와 클라우드 환경에서 여러 프로젝트를 보다 쉽게 관리할 수 있습니다.

모든 작업을 완료할 필요는 없습니다. 관리 목표에 따라 이러한 작업을 혼합하고 일치시킬 수 있습니다. 가능한 작업의 목록은 다음과 같습니다.

- 시스템 이름 사용자 지정
- **Active Directory** 시스템 레코드 생성
- 네트워크 **DNS** 및 내부 **IP** 범위 설정

## 시작하기 전에

이 자습서를 사용하려면 인프라가 구성되어 있고 시스템 및 네트워크가 있는 클라우드 템플릿이 배포되어 있어야 합니다. 다음 사항이 시스템에 이미 구성되어 있는지 확인합니다.

- 인프라 자습서에 지정된 모든 단계를 성공적으로 수행했습니다. [자습서: vRealize Automation Cloud Assembly에서 vSphere 인프라와 배포 설정 및 테스트](#)의 내용을 참조하십시오.
- Cloud Assembly 관리자 역할이 있습니다. [vRealize Automation의 조직 및 서비스 사용자 역할의 내용](#)을 참조하십시오.

## 시스템 이름 사용자 지정

이 작업의 목표는 개발 프로젝트용으로 배포된 시스템의 이름이 프로젝트의 비용 센터, 배포 시 선택한 리소스 유형, 고유성을 보장하기 위한 증분 번호를 기반으로 지정되도록 하는 것입니다. 예: DevProject-centos-021.

이 예제를 명명 요구 사항에 맞게 조정할 수 있습니다.

프로젝트에 대한 자세한 내용은 [장 5 vRealize Automation Cloud Assembly 프로젝트 추가 및 관리](#)의 내용을 참조하십시오.



사용자 지정 명명 예제를 보여주는 비디오는 배포를 위한 사용자 지정 명명 템플릿을 생성하는 방법을 참조하십시오.

- 1 **인프라 > 프로젝트**를 선택합니다.
- 2 기존 프로젝트를 선택하거나 새로 생성합니다.  
이 자습서에서 프로젝트 이름은 Development Project입니다.
- 3 **생성**을 클릭합니다.
- 4 [프로젝트] 페이지에서, 프로젝트를 구성할 수 있도록 타일에서 프로젝트 이름을 클릭합니다.
- 5 **사용자** 탭을 클릭하고 이 프로젝트의 멤버인 사용자를 추가합니다.
- 6 **프로비저닝** 탭을 클릭합니다.
  - a [영역] 섹션에서 **영역 추가**를 클릭하고 이 프로젝트에 대해 워크로드가 배포된 가능한 클라우드 영역을 추가합니다.
  - b [사용자 지정 속성] 섹션에서 이름이 **costCenter**이고 값이 **DevProject**인 사용자 지정 속성을 추가합니다.

**Custom Properties**  
Specify the custom properties that should be added to all requests in this project. ⓘ

Define custom properties	Name	Value
	costCenter	DevProject

**Custom Naming**  
Specify the naming template to be used for machines, networks, security groups and disks provisioned in this project.

**Template** ⓘ  
`${resource.costCenter}-${resource.osType}-${###}`  
Hint: Avoid conflicting names by generating digits in names. \${#####}

- c [사용자 지정 명명] 섹션에서 다음 명명 템플릿을 추가합니다.

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

`${resource.osType}`은 클라우드 템플릿을 배포할 때 선택한 운영 체제를 기반으로 합니다.

- 7 **저장**을 클릭합니다.

- 8 운영 체제 유형에 대한 입력 값으로 클라우드 템플릿을 업데이트합니다.

입력 값은 사용자를 위한 배포 요청 양식을 사용자 지정하고 개발 프로세스를 간소화할 수 있는 직접적인 방법입니다. 입력 값을 생성하면 단일 클라우드 템플릿을 사용하여 다양한 구성의 워크로드를 배포할 수 있습니다. 예: 크기 또는 운영 체제.

이 예에서는 이전 자습서의 **Development Template**을 사용합니다. **5단계: 기본 클라우드 템플릿 설계 및 배포**의 내용을 참조하십시오.

- a **설계**를 선택하고 **Development Template**을 엽니다.
- b [코드] 창에서 **YAML**을 다음과 같이 변경하여 업데이트합니다.
- **Inputs** 섹션에서 **osType**을 추가합니다.

다음 단계에서는 `osType` 입력도 이미지 지정에 사용되는 것을 볼 수 있습니다. **enum** 섹션에 문자열을 추가하는 경우 값(이 예에서는 `centos` 및 `ubuntu`)이 **인프라 > 구성 > 이미지 매핑**에서 정의한 이미지 이름과 일치해야 합니다. 예를 들어 이미지 매핑 이름이 `centos`가 아닌 **CentOS** 경우 입력 섹션에서 **CentOS**를 사용해야 합니다.

```
inputs:
  osType:
    type: string
    title: OS Type
```

```
description: Select the operating system.
enum:
  - centos
  - ubuntu
```

- Cloud\_vSphere\_Machine\_1 섹션에서 image를 osType 입력 매개 변수(`${input.osType}`)로 업데이트하고 동일한 입력 매개 변수를 사용하여 osType 사용자 지정 속성을 추가합니다.

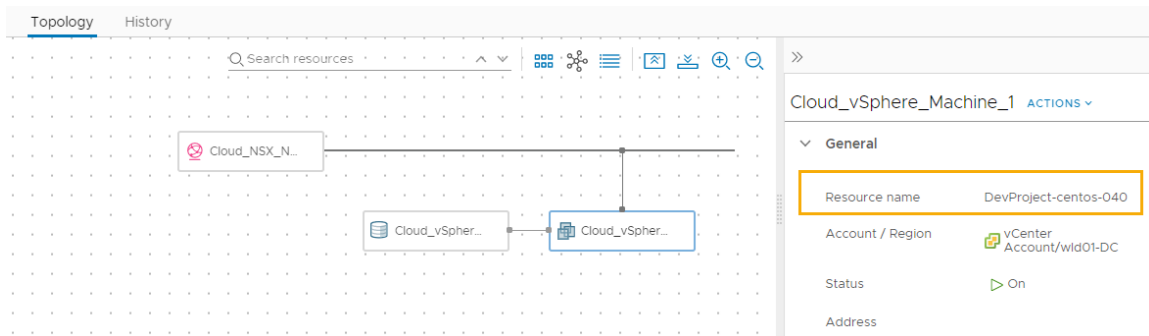
```
resources:
  Cloud_vSphere_Disk_1:
    type: Cloud.vSphere.Disk
    properties:
      capacityGb: 1
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: ${input.osType}
      osType: ${input.osType}
      flavor: small
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'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

- c 배포를 클릭하고 사용자 지정 이름 배포 테스트를 이름으로 입력합니다.
- d 다음을 클릭합니다.
- e 드롭다운 메뉴에서 **centos** 운영 체제를 선택합니다.

The screenshot shows the 'Deployment Inputs' configuration screen. The 'OS Type' dropdown menu is open, displaying 'centos' as the selected option, with 'centos' and 'ubuntu' as available choices. The left sidebar indicates the current step is '2 Deployment inputs'. At the bottom right, there are three buttons: 'CANCEL', 'PREVIOUS', and 'DEPLOY'.

- f 배포를 클릭합니다.
- 9 진행률을 추적하고 성공적인 배포를 검토합니다.

이 예에서 시스템 이름은 DevProject-centos-026입니다. 참고로, 이 예는 작업의 시작 부분에서 참조된 자습서를 기반으로 합니다.



## Active Directory 시스템 레코드 생성

워크로드를 프로비저닝할 때 Active Directory에서 시스템 레코드를 생성할 수 있습니다. 프로젝트 배포에 대해 이 작업을 자동으로 수행하도록 vRealize Automation Cloud Assembly를 구성하면 클라우드 관리자의 워크로드를 줄일 수 있습니다.

1 Active Directory 통합을 추가합니다.

a 인프라 > 연결 > 통합을 선택합니다.

이러한 단계는 이 AD 시스템 레코드 자습서와 관련된 기본 Active Directory 구성을 다룹니다. Active Directory 통합에 대한 자세한 내용은 [vRealize Automation Cloud Assembly에서 Active Directory 통합을 생성하는 방법](#)의 내용을 참조하십시오.

b 통합 추가를 클릭하고 Active Directory를 클릭합니다.

The screenshot shows the 'Active Directory Integration' configuration page. The left sidebar has a menu with 'Image Mappings', 'Network Profiles', 'Storage Profiles', 'Pricing Cards', 'Terraform Versions', 'Tags', 'Resources', 'Compute', 'Networks', 'Security', 'Storage', 'Machines', 'Volumes', 'Kubernetes', 'Activity', 'Requests', 'Events Log', and 'Connections'. The 'Summary' tab is selected. The 'Status' is 'OK'. The 'Activate integration' toggle is turned on. The 'Name' field is 'Active Directory Integration'. The 'Description' field is empty. The 'Active Directory Credentials' section includes: 'LDAP host / IP' (ldap://cmbu-sc2dc-01.cmbu.local:389), 'Running environment' (embedded-ABX-onprem), 'Username' (cmbu/administrator), and 'Password' (empty). The 'Base DN' field is highlighted with a yellow box and contains 'ou=AppDev,dc=cmbu,dc=local'. At the bottom, there is a 'VALIDATE' button and a warning message: 'Validate credentials before making changes'.

c 이 통합에 사용하는 이름을 입력합니다.

d LDAP 호스트/IP 및 연결된 자격 증명을 입력합니다.

e 기본 DN을 입력합니다.

이 자습서에서 예는 **ou=AppDev,dc=cmbu,dc=local**입니다. AppDev는 프로젝트에 대해 추가할 컴퓨터 OU의 상위 OU입니다.

f **추가**를 클릭합니다.

2 통합에 프로젝트를 추가합니다.

3 Active Directory 통합에서 **프로젝트** 탭을 클릭하고 **프로젝트 추가**를 클릭합니다.

Add Projects

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

a App Development 프로젝트를 선택합니다.

b 상대 DN을 입력합니다. 예를 들어 **OU=AppDev-Computers**를 입력합니다.

c 재정의의를 그대로 유지하고 꺼진 스위치를 무시합니다.

이 절차는 프로젝트의 프로세스 자동화에 중점을 둡니다. 템플릿에서 수행할 수 있는 사용자 지정에 대한 것은 아닙니다.

d **추가**를 클릭합니다.

4 통합에 대한 변경 내용을 저장하려면 **저장**을 클릭합니다.

5 프로젝트에 대한 클라우드 템플릿을 배포하고 시스템이 올바른 Active Directory OU에 추가되었는지 확인합니다.

## 네트워크 DNS 및 내부 IP 범위 설정

네트워크 프로파일을 추가하거나 업데이트하여 DNS 서버와 내부 IP 범위를 포함합니다.

vSphere, NSX-V 또는 NSX-T에 대한 클라우드 계정을 이미 생성한 상태여야 합니다. 자습서: [vRealize Automation Cloud Assembly에서 vSphere 인프라와 배포 설정 및 테스트](#) 또는 [vRealize Automation Cloud Assembly에 클라우드 계정 추가](#)의 내용을 참조하십시오.

1 **인프라 > 구성 > 네트워크 프로파일**을 선택합니다.

2 기존 프로파일을 선택하거나 새로 생성합니다.

### 3 요약 탭에서 계정/지역을 선택하고 이름을 입력합니다.

이 자습서에서 네트워크 프로파일 이름은 Network Profile입니다.

### 4 네트워크를 추가합니다.

- 네트워크 탭을 클릭합니다.
- 네트워크 추가를 클릭합니다.
- NSX 또는 vSphere 네트워크를 하나 이상 추가합니다.
- 추가를 클릭합니다.

### 5 DNS 서버를 구성합니다.

- 네트워크 탭의 네트워크 목록에서 네트워크 이름을 클릭합니다.

Summary Networks Network Policies Load Balancers Security					
Networks listed here are used when provisioning to existing, on-demand, or p					
<a href="#">+ ADD NETWORK</a> <a href="#">TAGS</a> <a href="#">MANAGE IP RANGES</a> <a href="#">X REMOVE</a>					
<input type="checkbox"/>	Name ↑	Account / Region	Zone	Network Domain	CIDR
<input type="checkbox"/>	DevProject--004	NSX-T Account		overlay-tz-sc2vc05-vip-nsx-mgmt.cmbu.local	192.168.1.64 /27

- 이 네트워크에서 사용하려는 DNS 서버 IP 주소를 입력합니다.

DevProject--004

DNS servers

192.168.1.22  
192.168.1.23

DNS search domains

company.local

DNS Servers

Use a comma separated list or new lines.

- 저장을 클릭합니다.

### 6 네트워크의 IP 범위를 지정합니다.

- 네트워크 목록에서 네트워크 이름 옆에 있는 확인란을 선택합니다.

Network Profile [DELETE](#)

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

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

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

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

1

- b IP 범위 관리를 클릭합니다.
- c [IP 범위 관리] 대화 상자에서 새 IP 범위를 클릭합니다.

## New IP Range

**Network \*** NSX-mcm1376447-151082888186

**Source** ☒ Internal ☐ External

**Name \*** DevProject Range

**Description**

**CIDR** 192.168.100.32/28

**Start IP address \*** 192.168.100.34

**End IP address \*** 192.168.100.46

- d 이름을 입력합니다.

예: **DevProject Range**.

e 범위를 정의하려면 **시작 IP 주소** 및 **끝 IP 주소**를 입력합니다.

f **추가**를 클릭합니다.

g 범위를 더 추가하거나 **닫기**를 클릭합니다.

7 개발 프로젝트에 구성해 놓은 연결된 네트워크 계정/지역이 포함된 클라우드 영역을 추가합니다.

8 프로젝트에 대한 클라우드 템플릿을 배포하고 지정된 IP 범위 내에 시스템이 프로비저닝되었는지 확인합니다.

## 자습서: vRealize Automation Cloud Assembly에서 태그를 사용하여 vSphere 리소스 관리

태그는 리소스와 연결하고 템플릿에 포함할 수 있는 강력한 메타데이터입니다. 워크로드 배치 및 리소스 레이블 지정을 비롯한 다양한 관리 시나리오에서 태그를 사용할 수 있습니다.

### 태그에 대한 간단한 소개

이 섹션은 제공된 단계에 적용되는 태그에 대한 간단한 소개입니다. 태그에 대한 자세한 내용은 [태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#)에서 참조하십시오.

#### ■ 기능 및 제약 조건 태그

태그를 사용하면 리소스 기능을 기반으로 배포를 제어할 수 있습니다. 예를 들어 클라우드 관리자는 반복적으로 개발된 클라우드 템플릿을 개발 관련 리소스 풀에 배포하고 운영에 적합한 템플릿은 다른 리소스 풀에 배포하려고 합니다.

- 기능 태그는 리소스에 추가되어 리소스의 기능을 정의합니다.

- 제약 조건 태그는 클라우드 템플릿에서 사용되며, 배포된 리소스에서 사용할 리소스를 정의합니다.

#### ■ 태그에 레이블 지정

리소스를 관리하기 위해 태그를 개체 레이블 또는 설명으로 추가할 수 있습니다. 관리 가능성에는 리소스 검색 결과 개선, 유사한 개체 간의 차별화, 사용자 지정 정보로 개체에 주석 달기, 타사 시스템에 정보 제공, 보안 그룹화 멤버 자격 조건 생성, 연결된 SDDC 도메인 간의 일관성 보장 등이 있습니다.

### 시작하기 전에

- 자습서: [vRealize Automation Cloud Assembly에서 vSphere 인프라와 배포 설정 및 테스트](#)에서 정의된 리소스 및 클라우드 템플릿을 검토합니다. 이 자습서에 사용된 샘플 값이 여기에서 사용됩니다.

## 태그를 사용하여 워크로드 배치 관리

여기 간단한 예제에서는 개발 및 운영 환경 태그를 사용하여 기능 및 제약 조건 태그를 사용하는 방법을 보여줍니다. 먼저 vCenter Server 리소스 풀 계산 리소스에 대한 기능 태그를 추가한 다음 클라우드 템플릿에 이 태그를 포함합니다. 클라우드 템플릿 예제에서는 입력을 사용하여 배포하는 사용자가 개발과 운영 리소스 풀 중 어디에 배포할지를 선택할 수 있도록 하는 방법을 보여줍니다.

동일한 태그를 사용하여 다중 클라우드 환경에서 배치를 정의하는 방법에 대한 예는 [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트](#)에서 참조하십시오.

1 리소스 풀에 기능 태그를 추가합니다.

- a 인프라 > 리소스 > 계산을 선택합니다.
- b 클라우드 영역을 열고 **계산**을 클릭합니다.



c 개발 워크로드를 배포할 리소스 풀을 찾아서 클릭합니다.

이 자습서에서는 다음 샘플 값을 사용합니다. 이러한 값은 예시일 뿐입니다. 실제 값은 사용자의 환경에 따라 다릅니다.

샘플 리소스 풀	샘플 태그
wid01-clu01 / Development	env:dev
wid01-clu01 / Production	env:prod

d **env.dev** 태그를 추가하고 **저장**을 클릭합니다.

**wld01-clu01 / Development**

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

**Name** wld01-clu01 / Development

**Type** VM\_HOST

**Tags** env:dev X Enter a new tag

**SAVE** **CANCEL**

- e 운영 워크로드를 배포할 리소스 풀에 대해 이 프로세스를 반복하고 **env:prod** 태그를 추가합니다.
- 2 클라우드 영역의 리소스 풀에 기능 태그가 추가되어 있는지 확인합니다.
  - a **인프라 > 구성 > 클라우드 영역**을 선택합니다.
  - b 프로젝트와 연결된 클라우드 영역을 열고 **계산**을 클릭합니다.

이 예에서 클라우드 영역은 vCenter 계정 클라우드 영역이고 태그는 리소스 풀 2개(wld01-clu01 / Development 및 wld01-clu01 / Production)에 추가되었습니다.

**vCenter Account Cloud Zone** DELETE

Summary **Compute** Projects

All compute resources listed apply to this cloud zone. Use the filter to add or remove resources from the list.

Include all compute

Name	Account / Region	Type	Tags
10.176.152.27	vCenter Account / wld01-DC	Host	
wld01-clu01	vCenter Account / wld01-DC	Supervisor Cluster	
wld01-clu01 / Development	vCenter Account / wld01-DC	Resource Pool	env:dev
wld01-clu01 / Production	vCenter Account / wld01-DC	Resource Pool	env:prod
wld01-clu01 / Training-Org	vCenter Account / wld01-DC	Resource Pool	
wld01-clu01 / VCF-edge_edge-wldclu-01_ResourcePool_fta14b18-82b5-4261-b546-aef86a1db2d9	vCenter Account / wld01-DC	Resource Pool	

- 3 클라우드 템플릿에 제약 조건 태그를 추가합니다.
 

제약 조건 태그는 템플릿이 배포되는 위치를 제한하는 데 사용됩니다.

  - a **설계 > 클라우드 템플릿**을 선택한 다음 템플릿을 엽니다.
 

이 자습서에서 템플릿 이름은 **Development Template**입니다.
  - b 코드 창에서 템플릿에 대한 **YAML**을 검토합니다.

이 YAML이 자습서의 시작점입니다.

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

- c `${input.placement}`를 변수로 사용하여 제약 조건 태그를 `Cloud_vSphere_Machine_1` 리소스에 추가합니다.

```
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos
      flavor: medium
      constraints:
        - tag: '${input.placement}'
      networks:
        - network: '${resource.Cloud_NSX_Network_1.id}'
      attachedDisks:
        - source: '${resource.Cloud_vSphere_Disk_1.id}'
```

- d 입력 섹션에서 배치 변수를 정의합니다.

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

- e 최종 YAML이 다음 예와 유사한지 확인합니다.

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

- f 사용 가능한 리소스에 대해 태그 변수를 사용해 보려면 **테스트**를 클릭한 다음 **env:dev**를 선택합니다.

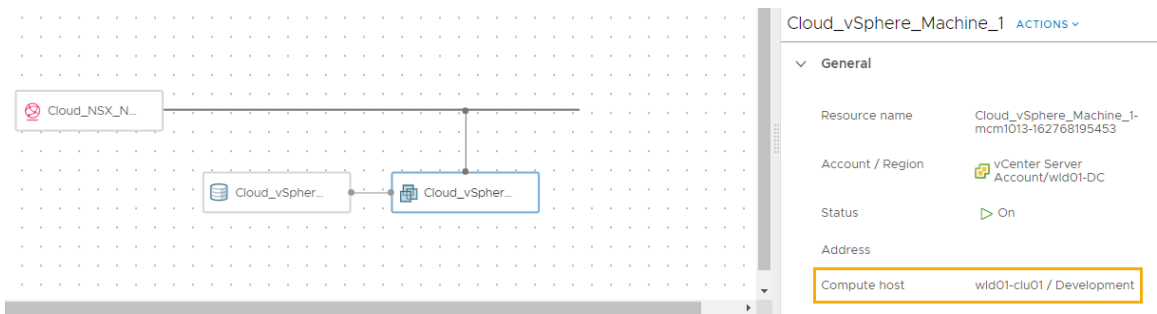


**env:prod**를 사용하여 테스트를 반복합니다. 두 테스트가 모두 성공하면 템플릿을 배포하여 작동하는지 확인합니다.

- 4 템플릿을 배포하여 워크로드 배치를 테스트합니다.
  - a 클라우드 템플릿 디자이너에서 **배포**를 클릭합니다.
  - b **Deployment Tag Dev**를 **배포 이름**으로 입력하고 **다음**을 클릭합니다.

- c **배포용 배치 선택** 드롭다운 메뉴에서 **env:dev**를 선택하고 **배포**를 클릭합니다.
- 5 템플릿이 선택한 리소스 풀에 리소스를 배포했는지 확인합니다.
  - a **배포** 탭에서 **Deployment Tag Dev** 배포를 찾습니다.
  - b 배포 세부 정보를 열고 **토폴로지**를 클릭합니다.
  - c vSphere 시스템을 클릭하고 오른쪽 창에서 시스템 정보를 확장합니다.
  - d **일반** 섹션에서 **계산 호스트**를 찾아서 **env:dev** 태그와 일치하는 리소스 풀과 값이 일치하는지 확인합니다.

이 예에서는 값이 wid01-clu01 / Development이며 선택한 제약 조건 태그를 기반으로 올바른 리소스 풀에 워크로드가 배포된 것을 보여줍니다.



- e 배포 프로세스를 반복하고, 이번에는 **env:prod**를 선택합니다.

## vCenter Server 및 NSX-T에서 사용할 수 있는 레이블로 태그 추가

배포에 태그를 추가하여 리소스를 관리하는 데 사용할 수 있습니다.

이 예에서는 MySQL 시스템 및 네트워크를 식별하는 태그를 추가합니다. 또한 웹 네트워크를 식별하는 태그를 추가합니다. 주문형 네트워크 대비 기존 네트워크에서 태그가 작동하는 방식 때문에 두 가지 중 선택할 수 있습니다.

- 이전 섹션에서 사용한 기존 네트워크 프로파일을 사용하는 경우 **NGINX:web** 태그가 **NSX-T**의 기존 개체에 추가되지 않습니다. 따라서 **NSX-T**에서는 이 태그와 관련된 확인 단계를 무시할 수 있습니다.
- 주문형 네트워크 프로파일을 생성하는 경우 라우팅된/주문형 네트워크를 사용하도록 **YAML**에서 네트워크를 업데이트할 수 있습니다. 이 예제에서는 **NSX-T**의 새 개체에서 **NGINX:web** 태그를 선언할 수 있도록 주문형 네트워크가 사용됩니다.

다음 **YAML**은 이전 예제에서 가져왔으며, 라우팅된 주문형 **networkType**을 사용하는 것만 다릅니다. 여기에는 제약 조건 태그가 포함됩니다.

이 자습서에서는 다음 샘플 값을 사용합니다. 이러한 값은 예시일 뿐입니다. 실제 값은 사용자의 환경에 따라 다릅니다.

```
formatVersion: 1
inputs:
  placement:
    type: string
```

```

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

```

- 1 **설계 > 클라우드 템플릿**을 선택한 다음 템플릿을 엽니다.
- 2 **Cloud\_vSphere\_Machine\_1** 속성에서 다음 태그를 추가합니다.

```

tags:
  - key: db
    value: mysql

```

- 3 **VM NIC** 태그를 추가합니다.

```

tags:
  - key: db
    value: mysql

```

- 4 **NSX 논리적 스위치/세그먼트** 태그를 추가합니다.

```

tags:
  - key: NGINX
    value: web

```

## 5 YAML이 다음 예와 유사한지 확인합니다.

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

## 6 템플릿 배포

이 예에서는 **Development template w tags**라는 이름을 사용합니다.

## 7 배포의 태그를 확인하려면 배포를 열고 **토폴로지**를 클릭합니다.

- a 토폴로지에서 시스템을 클릭합니다.
- b 시스템의 **일반** 섹션을 확장하고 태그 레이블을 찾습니다.

태그 값은 db:mysql입니다.

- c 네트워크 섹션을 확장하고 네트워크 태그 열을 찾습니다.

태그 값은 db:mysql입니다.

Development template w tags Create Successful ACTIONS ▾ | ↻

No description

Owner fritz  
Requestor fritz  
Project Development Project  
Cloud Template Development Template ↕

Expires on Never  
Last updated Mar 8, 2021, 4:31:01 PM  
Created on Mar 8, 2021, 4:09:14 PM

HIDE SUMMARY ⌵

Topology History

Search resources

Cloud\_NSX\_N...

Cloud\_vSphere...

Cloud\_vSphere...

Cloud\_vSphere\_Machine\_1 ACTIONS ▾

General

Resource name Cloud\_vSphere\_Machine\_1-mcm1019-163638575175  
Account / Region vCenter Server Account/wld01-DC  
Status On  
Address  
Compute host wld01-clu01 / Development

Tags db:mysql

Storage

Network

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

Custom properties

- d 토폴로지에서 네트워크를 클릭하고 일반 섹션을 확장하여 태그 레이블을 찾습니다.

태그 값은 NGINX:web입니다.

Topology History

Search resources

Cloud\_NSX\_N...

Cloud\_vSphere...

Cloud\_vSphere...

Cloud\_NSX\_Network\_1 ACTIONS ▾

General

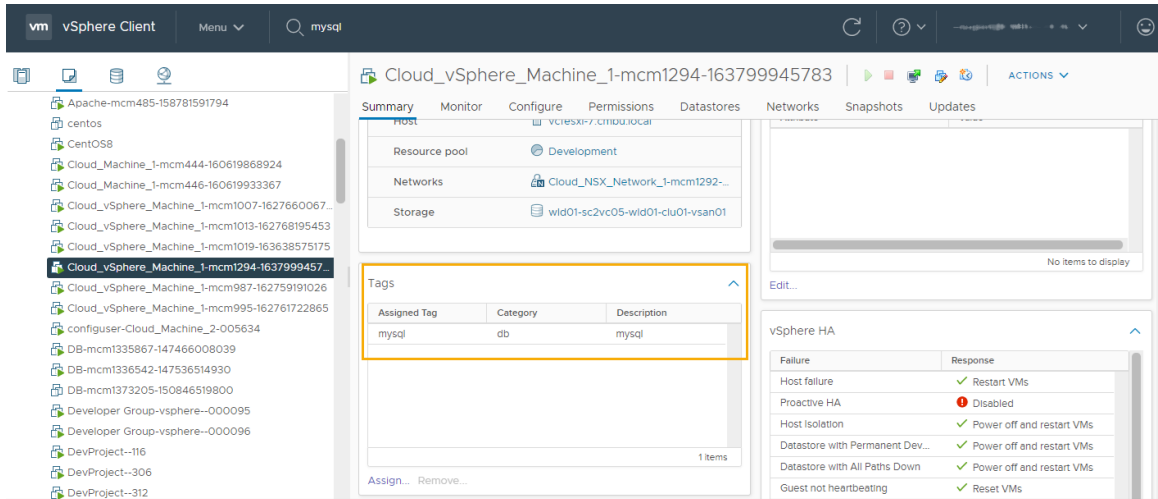
Resource name Cloud\_NSX\_Network\_1-mcm1292-163799928607  
Account NSX-T Account  
Network type routed  
CIDR 192.168.150.0/28

Tags NGINX:web

Custom properties

- 8 vCenter Server에서 태그를 확인하려면 이 워크로드가 배포된 vCenter Server 인스턴스에 로그인합니다.

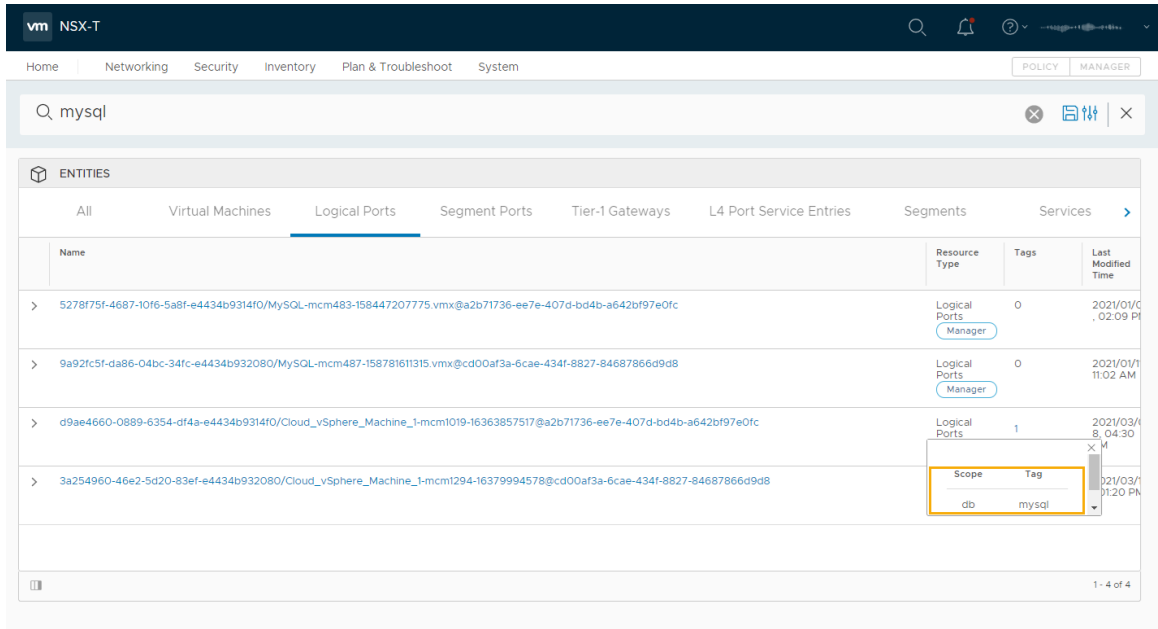
- a 가상 시스템을 찾아서 태그 창을 찾습니다.



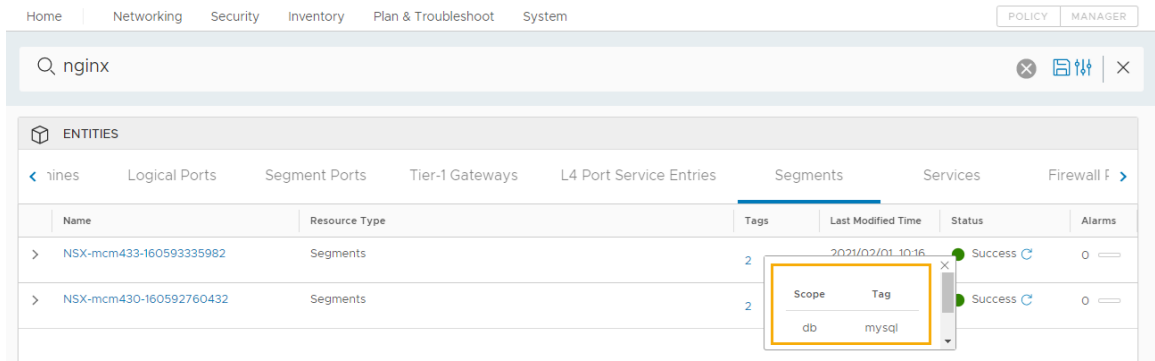
9 NSX-T의 태그를 확인하려면 이 네트워크가 구성된 NSX-T 인스턴스에 로그인합니다.

- a 오른쪽 상단에서 **정책**을 클릭합니다.
- b NIC와 연결된 db:mysql 태그를 찾으려면 **mysql**을 검색합니다.
- c **논리적 포트**를 클릭하고 배포된 vSphere 시스템을 찾습니다.
- d [태그] 열에서 숫자를 클릭합니다.

범위 및 태그는 각각 db 및 mysql입니다.



- e 세그먼트와 연결된 NGINX:web 태그를 찾으려면 네트워크를 검색합니다.  
이 예시에서 네트워크 이름은 **Cloud\_NSX\_Network\_1-mcm1292-163799928607**입니다.
- f 세그먼트 행을 찾고 태그 열에서 숫자를 클릭합니다.  
범위 및 태그는 각각 NGINX 및 web입니다.



## 자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트

이 전체적인 vRealize Automation Cloud Assembly 자습서는 다중 클라우드 설정에서 배포하는 방법을 보여줍니다. 동일한 클라우드 템플릿을 둘 이상의 제공자(여기에서는 AWS 및 Microsoft Azure)에 배포합니다.

이 예제에서 애플리케이션은 WordPress 사이트입니다. 단계별 설정을 살펴보면 전체 설계를 완성하는 과정을 이해할 수 있습니다.

여기에 보이는 이름과 값은 예시일 뿐입니다. 자체 환경에서 이러한 값을 그대로 사용해서는 안 됩니다.

자체 클라우드 인프라 및 배포 요구 사항에 맞게, 예제 값을 대체할 부분을 고려하십시오.

### 1부: 예제 vRealize Automation Cloud Assembly 인프라 구성

먼저 vRealize Automation Cloud Assembly 엔지니어링 사용자가 나중에 애플리케이션을 개발하고 테스트하고 운영 환경에 배치할 수 있는 리소스를 구성합니다.

인프라에는 클라우드 대상 및 WordPress 사이트에 필요한 가용 시스템, 네트워크 및 스토리지에 대한 정의가 포함됩니다.

#### 1. 클라우드 계정 추가

이 단계에서는 클라우드 관리자가 클라우드 계정 2개를 추가합니다. 예제 프로젝트는 AWS에서 개발 및 테스트 작업을 수행한 후 Azure에서 운영 환경으로 전환합니다.

##### 절차

- 1 **인프라 > 연결 > 클라우드 계정**으로 이동합니다.
- 2 **클라우드 계정 추가**를 클릭하고 Amazon Web Services를 선택한 후 값을 입력합니다.

설정	샘플 값
액세스 키 ID	R5SDR3PXVV2ZW8B7YNSM
비밀 액세스 키	SZXAINXU4UHNAQ1E156S

설정	샘플 값
이름	OurCo-AWS
설명	WordPress

모든 값은 예시일 뿐입니다. 계정 세부 사항은 다를 수 있습니다.

- 3 자격 증명을 확인하려면 **검증**을 클릭합니다.
- 4 **추가**를 클릭합니다.
- 5 새로 추가된 계정 **구성**을 편집하여 **us-east-1** 및 **us-west-2** 지역으로의 프로비저닝을 허용합니다.
- 6 **클라우드 계정 추가**를 클릭하고 Microsoft Azure를 선택한 후 값을 입력합니다.

설정	샘플 값
구독 ID	ef2avpf-dfdv-zxluguii-g4h0-i8ep2jwp4c9arbfe
테넌트 ID	dso9wv3-4zgc-5nrcy5h3m-4skf-nnovp40wfxsro22r
클라이언트 애플리케이션 ID	bg224oq-3ptp-mbhi6aa05-q511-uflyjr2sttyik6bs
클라이언트 애플리케이션 비밀 키	7uqxi57-0wtn-kymgf9wcj-t2l7-e52e4nu5fig4pmd
이름	OurCo-Azure
설명	WordPress

- 7 자격 증명을 확인하려면 **검증**을 클릭합니다.
- 8 **추가**를 클릭합니다.
- 9 새로 추가된 계정 **구성**을 편집하여 **East US** 지역으로의 프로비저닝을 허용합니다.

## 2. 클라우드 영역 추가

이 예제 단계에서는 클라우드 관리자가 개발, 테스트 및 운영 환경에 대해 하나씩 총 3개의 클라우드 영역을 추가합니다.

클라우드 영역은 프로젝트가 WordPress 사이트를 지원하기 위해 시스템, 네트워크 및 스토리지를 배포하는 리소스입니다.

### 절차

- 1 **인프라 > 구성 > 클라우드 영역**으로 이동합니다.
- 2 **새 클라우드 영역**을 클릭하고 개발 환경에 대한 값을 입력합니다.

클라우드 영역 설정	샘플 값
계정/지역	OurCo-AWS/us-east-1
이름	OurCo-AWS-US-East

클라우드 영역 설정	샘플 값
설명	WordPress
배치 정책	기본값
기능 태그	env:dev

모든 값은 예시일 뿐입니다. 실제 영역 세부 정보는 다를 수 있습니다.

- 3 계산을 클릭하고, 필요한 영역이 표시되는지 확인합니다.
- 4 생성을 클릭합니다.
- 5 테스트 및 운영 환경에 대한 값을 사용하여 이 프로세스를 두 번 반복합니다.

클라우드 영역 설정	샘플 값
계정/지역	OurCo-AWS/us-west-2
이름	OurCo-AWS-US-West
설명	WordPress
배치 정책	기본값
기능 태그	env:test

클라우드 영역 설정	샘플 값
계정/지역	OurCo-Azure/East US
이름	OurCo-Azure-East-US
설명	WordPress
배치 정책	기본값
기능 태그	env:prod

### 3. 버전 매핑 추가

이 예제 단계에서는 클라우드 관리자가 배포 환경마다 다를 수 있는 용량 요구 사항을 고려하기 위해 버전 매핑을 추가합니다.

버전 매핑은 다양한 크기의 시스템 배포를 설명하며, 비공식적으로 티셔츠 크기 조정이라고 합니다.

#### 절차

- 1 **인프라 > 구성 > 버전 매핑**으로 이동합니다. 각 클라우드 영역에 대해 소규모, 중간 규모 및 대규모 버전을 허용해야 합니다.

## 2 새 버전 매핑을 클릭하고 개발 클라우드 영역에 대한 값을 입력합니다.

설정	샘플 값
버전 이름	small
계정/지역 값	OurCo-AWS/us-east-1 t2.micro
계정/지역 값	OurCo-AWS/us-west-2 t2.micro
계정/지역 값	OurCo-Azure/East US Standard_A0

모든 값은 예시일 뿐입니다. 실제 버전은 다를 수 있습니다.

## 3 생성을 클릭합니다.

## 4 중형 및 대형 버전에 대한 값을 사용하여 이 프로세스를 두 번 반복합니다.

설정	샘플 값
버전 이름	medium
계정/지역 값	OurCo-AWS/us-east-1 t2.medium
계정/지역 값	OurCo-AWS/us-west-2 t2.medium
계정/지역 값	OurCo-Azure/East US Standard_A3

설정	샘플 값
버전 이름	large
계정/지역 값	OurCo-AWS/us-east-1 t2.large
계정/지역 값	OurCo-AWS/us-west-2 t2.large
계정/지역 값	OurCo-Azure/East US Standard_A7

## 4. 이미지 매핑 추가

이 예제 단계에서, 클라우드 관리자는 Ubuntu를 위한 이미지 매핑, WordPress 서버를 위한 호스트 및 해당 MySQL 데이터베이스 서버를 추가합니다.

이미지 매핑을 추가하여 운영 체제를 계획합니다. 각 클라우드 영역에는 Ubuntu 이미지 매핑이 필요합니다.

**절차**

- 1 **인프라 > 구성 > 이미지 매핑**으로 이동합니다.
- 2 **새 이미지 매핑**을 클릭하고 Ubuntu 서버에 대한 값을 입력합니다.

설정	샘플 값
이미지 이름	ubuntu
계정/지역 값	OurCo-AWS/us-east-1 ubuntu-16.04-server-cloudimg-amd64
계정/지역 값	OurCo-AWS/us-west-2 ubuntu-16.04-server-cloudimg-amd64
계정/지역 값	OurCo-Azure/East US azul-zulu-ubuntu-1604-923eng

모든 값은 예시일 뿐입니다. 이미지는 달라집니다.

- 3 **생성**을 클릭합니다.

**5. 네트워크 프로파일 추가**

이 예제 단계에서는 클라우드 관리자가 각 클라우드 영역에 네트워크 프로파일을 추가합니다.

관리자는 WordPress 시스템이 사용할 네트워크 및 로드 밸런서의 다른 측면에서 사용할 두 번째 네트워크를 각 프로파일에 추가합니다. 두 번째 네트워크는 결과적으로 사용자가 연결하는 네트워크입니다.

**절차**

- 1 **인프라 > 구성 > 네트워크 프로파일**로 이동합니다.
- 2 **새 네트워크 프로파일**을 클릭하고 개발 클라우드 영역에 사용할 프로파일을 생성합니다.

네트워크 프로파일 설정	샘플 값
계정/지역	OurCo-AWS/us-east-1
이름	devnets
설명	WordPress

- 3 **네트워크**를 클릭하고 **네트워크 추가**를 클릭합니다.
- 4 wpnet 및 appnet-public을 선택하고 **추가**를 클릭합니다.  
모든 값은 예시일 뿐입니다. 네트워크 이름은 다를 수 있습니다.
- 5 **생성**을 클릭합니다.

이 Wordpress 예제에서는 네트워크 정책 또는 네트워크 보안 설정을 지정할 필요가 없습니다.

- 6 프로세스를 두 번 반복하여 Wordpress 예제 테스트 클라우드 영역과 운영 클라우드 영역에 사용할 네트워크 프로파일을 생성합니다. 각 프로세스에서 wpnet 네트워크와 appnet-public 네트워크를 추가합니다.

네트워크 프로파일 설정	샘플 값
계정/지역	OurCo-AWS/us-west-2
이름	testnets
설명	WordPress

네트워크 프로파일 설정	값
계정/지역	OurCo-Azure/East US
이름	prodnets
설명	WordPress

## 6. 스토리지 프로파일 추가

이 예제 단계에서는 클라우드 관리자가 각 클라우드 영역에 스토리지 프로파일을 추가합니다.

관리자는 운영 영역에 고속 스토리지를 배치하고 개발 및 테스트 영역에 일반 스토리지를 배치합니다.

### 절차

- 1 **인프라 > 구성 > 스토리지 프로파일**로 이동합니다.
- 2 **새 스토리지 프로파일**을 클릭하고 개발 클라우드 영역에 사용할 프로파일을 생성합니다.

계정/지역을 선택하면 추가적인 필드가 나타납니다.

스토리지 프로파일 설정	샘플 값
계정/지역	OurCo-AWS/us-east-1
이름	OurCo-AWS-US-East-Disk
설명	WordPress
디바이스 유형	EBS
볼륨 유형	범용 SSD
기능 태그	storage:general

모든 값은 예시일 뿐입니다.

- 3 **생성**을 클릭합니다.

4 이 프로세스를 반복하여 테스트 클라우드 영역에 사용할 프로파일을 생성합니다.

스토리지 프로파일 설정	샘플 값
계정/지역	OurCo-AWS/us-west-2
이름	OurCo-AWS-US-West-Disk
설명	WordPress
디바이스 유형	EBS
볼륨 유형	범용 SSD
기능 태그	storage:general

5 이 프로세스를 반복하여 운영 클라우드 영역에 사용할 프로파일을 생성합니다. 운영 클라우드 영역은 Azure 영역이므로 설정이 다릅니다.

스토리지 프로파일 설정	샘플 값
계정/지역	OurCo-Azure/East US
이름	OurCo-Azure-East-US-Disk
설명	WordPress
스토리지 유형	관리 디스크
디스크 유형	프리미엄 LRS
OS 디스크 캐시	읽기 전용
데이터 디스크 캐시	읽기 전용
기능 태그	storage:fast

#### 다음에 수행할 작업

사용자를 식별하고 프로비저닝 설정을 정의하기 위해 프로젝트를 생성합니다. [2부: 예제 vRealize Automation Cloud Assembly 프로젝트 생성](#) 항목을 참조하십시오.

## 2부: 예제 vRealize Automation Cloud Assembly 프로젝트 생성

예제 vRealize Automation Cloud Assembly 프로젝트는 프로비저닝을 수행할 수 있는 사용자를 지정하고 프로비저닝 가능한 양을 구성합니다.

프로젝트는 사용자 및 프로비저닝 설정을 정의합니다.

- 사용자 및 사용자 역할의 사용 권한 수준
- 클라우드 영역에 프로비저닝할 때의 배포 우선 순위
- 클라우드 영역당 최대 배포 인스턴스 수

**절차**

- 1 **인프라 > 구성 > 프로젝트**로 이동합니다.
- 2 **새 프로젝트**를 클릭하고 **WordPress**를 이름으로 입력합니다.
- 3 **사용자**를 클릭하고 **사용자 추가**를 클릭합니다.
- 4 사용자의 이메일 주소와 역할을 추가합니다.

사용자를 성공적으로 추가하려면 VMware Cloud Services 관리자가 해당 사용자를 위해 vRealize Automation Cloud Assembly에 대한 액세스 권한을 부여해야 합니다.

여기에 나와 있는 주소는 예시일 뿐입니다.

- chris.ladd@ourco.com, 멤버
- kerry.mott@ourco.com, 멤버
- pat.tubb@ourco.com, 관리자

- 5 **프로비저닝**을 클릭하고 **클라우드 영역 추가**를 클릭합니다.
- 6 사용자가 배포할 수 있는 클라우드 영역을 추가합니다.

프로젝트 클라우드 영역 설정	샘플 값
클라우드 영역	OurCo-AWS-US-East
프로비저닝 우선 순위	1
인스턴스 제한	5
클라우드 영역	OurCo-AWS-US-West
프로비저닝 우선 순위	1
인스턴스 제한	5
클라우드 영역	OurCo-Azure-East-US
프로비저닝 우선 순위	0
인스턴스 제한	1

- 7 **생성**을 클릭합니다.
- 8 **인프라 > 구성 > 클라우드 영역**으로 이동하여 이전에 생성한 영역을 엽니다.
- 9 **프로젝트**를 클릭하고, WordPress가 영역에 프로비저닝할 수 있는 프로젝트인지 확인합니다.
- 10 생성한 다른 영역을 확인합니다.

**다음에 수행할 작업**

기본 클라우드 템플릿을 생성합니다.

**3부: 예제 vRealize Automation Cloud Assembly 템플릿 설계 및 배포**

다음으로, 일반 클라우드 템플릿 형식으로 예제 애플리케이션인 WordPress 사이트를 정의합니다. 템플릿은 설계 변경 없이 다양한 클라우드 벤더에 배포할 수 있습니다.

이 예는 **WordPress** 애플리케이션 서버, **MySQL** 데이터베이스 서버 및 지원 리소스로 구성됩니다. 처음에는 템플릿에 리소스 몇 개만 포함되지만 기존 리소스를 수정하고 더 많은 리소스를 추가하면서 그 크기가 증가합니다.

다음은 클라우드 관리자가 설정한 인프라인 **1부: 예제 vRealize Automation Cloud Assembly 인프라 구성**의 값입니다.

- 클라우드 계정 2개(AWS 및 Azure)
- 클라우드 영역 환경 3개
  - 개발 - OurCo-AWS-US-East
  - 테스트 - OurCo-AWS-US-West
  - 운영 - OurCo-Azure-East-US
- 각 영역에 대해 소규모, 중간 규모 및 대규모 계산 리소스가 포함된 버전 매핑
- 각 영역에 대해 구성된 **Ubuntu** 이미지 매핑
- 각 영역에 대해 내부 및 외부 서브넷이 포함된 네트워크 프로파일
- 배포할 스토리지: 개발 및 테스트 영역에 대한 일반 스토리지와 운영 영역에 대한 빠른 스토리지
- 예제 프로젝트에는 세 가지 클라우드 영역 환경과 설계를 생성할 수 있는 사용자가 모두 포함되어 있습니다.

## 사전 요구 사항

절차를 원활하게 수행하려면 인프라 값에 익숙해야 합니다. 이 예제에서는 개발 및 테스트 환경에서는 **AWS**를 사용하고 운영 환경에서는 **Azure**를 사용합니다. 자체 클라우드 템플릿을 생성할 때는 예시에 나와 있는 값을 일반적으로 클라우드 관리자가 설정한 실제 값으로 대체하십시오.

## 절차

### 1 기본 클라우드 템플릿 생성

vRealize Automation Cloud Assembly 설계 예제에서는 최소한의 **WordPress** 리소스만 포함된(예: 애플리케이션 서버 1개만 포함) 클라우드 템플릿으로 작업을 시작합니다.

### 2 기본 클라우드 템플릿 테스트

설계하는 동안, 필수 항목부터 시작하여 클라우드 템플릿을 구축한 다음, 템플릿이 커지면 배포하고 테스트하는 경우가 많습니다. 이 예제는 vRealize Automation Cloud Assembly에 내장된 진행 중인 테스트 중 일부를 보여줍니다.

### 3 클라우드 템플릿 확장

예제 애플리케이션의 기본 vRealize Automation Cloud Assembly 템플릿을 생성하고 테스트한 후에는 개발, 테스트를 거쳐 최종적으로 운영 환경에 배포할 수 있는 여러 계층의 애플리케이션으로 확장합니다.

## 기본 클라우드 템플릿 생성

vRealize Automation Cloud Assembly 설계 예제에서는 최소한의 WordPress 리소스만 포함된(예: 애플리케이션 서버 1개만 포함) 클라우드 템플릿으로 작업을 시작합니다.

vRealize Automation Cloud Assembly는 인프라를 코드로 구현하는 도구입니다. 먼저 설계 캔버스로 리소스를 끌어 와서 작업을 시작합니다. 그런 다음 캔버스의 오른쪽에 있는 코드 편집기를 사용하여 세부 정보를 완성합니다.

코드 편집기를 사용하면 코드를 직접 입력하고, 잘라내고, 붙여 넣을 수 있습니다. 코드 편집이 불편하면 캔버스에서 리소스를 선택하고 코드 편집기 **속성** 탭을 클릭한 다음 그 곳에 값을 입력하면 됩니다. 입력한 값이 마치 직접 입력한 것처럼 코드에 나타납니다.

### 절차

- 1 **설계 > 클라우드 템플릿**으로 이동하여 **새로 만들기 > 빈 캔버스**를 클릭합니다.
- 2 클라우드 템플릿 이름을 **Wordpress-BP**로 지정합니다.
- 3 **WordPress** 프로젝트를 선택하고 **생성**을 클릭합니다.
- 4 클라우드 템플릿 설계 페이지의 왼쪽에 있는 리소스에서 클라우드 애그노스틱 시스템 2개를 캔버스로 끌어 옵니다.

이 두 시스템은 WordPress 애플리케이션 서버(WebTier)와 MySQL 데이터베이스 서버(DBTier)로 사용됩니다.

- 5 오른쪽에서 시스템 YAML 코드를 편집하여 이름, 이미지, 버전 및 제약 조건 태그를 추가합니다.

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

- 6 클라우드 애그노스틱 네트워크를 캔버스로 끌어 와서 해당 코드를 편집합니다.

```
WP-Network-Private:
```

```

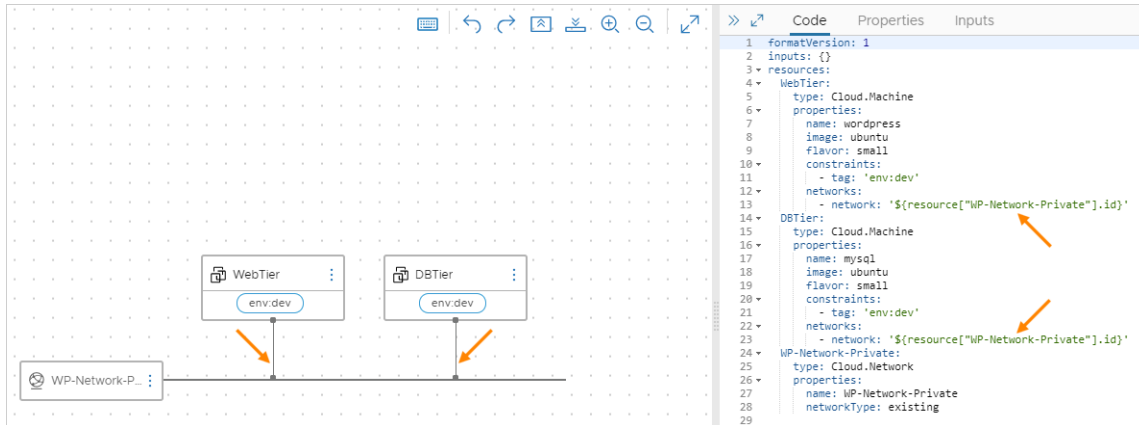
type: Cloud.Network
properties:
  name: WP-Network-Private
  networkType: existing

```

## 7 시스템을 네트워크에 연결합니다.

캔버스에서 네트워크 블록 위로 마우스를 가져간 후 선이 블록과 만나는 지점을 클릭하여 누른 상태에서 시스템 블록으로 끈 다음 마우스 버튼을 놓습니다.

연결 줄을 생성할 때 네트워크 코드는 편집기에서 시스템에 자동으로 추가됩니다.



## 8 사용자 입력 프롬프트를 추가합니다.

일부 경우에는 여러 옵션을 사용하도록 예제 인프라가 설정되었습니다. 예를 들면 다음과 같습니다.

- 개발, 테스트 및 운영을 위한 클라우드 영역 환경
- 소규모, 중간 규모 및 대규모 시스템을 위한 버전 매핑

특정 옵션을 클라우드 템플릿에 직접 설정할 수도 있지만, 템플릿 배포 시 사용자가 옵션을 선택할 수 있게 하는 방법이 더 좋습니다. 사용자에게 입력을 요청하면 하드 코딩된 템플릿 여러 개를 사용하는 대신 다양한 방식으로 배포할 수 있는 하나의 템플릿을 생성할 수 있습니다.

- a 배포 시 사용자가 시스템 크기 및 대상 환경을 선택할 수 있도록 코드에 `inputs` 섹션을 생성합니다. 선택할 수 있는 값을 정의합니다.

```
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 코드의 `resources` 섹션에 사용자의 선택을 요청하는 `${input.input-name}` 코드를 추가합니다.

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

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

- 9 마지막으로 다음의 예시를 사용하여 WebTier 및 DBTier 코드를 향상시킵니다. WP-Network-Private 코드는 추가적으로 변경할 필요가 없습니다.

향상된 기능에는 데이터베이스 서버로의 로그인 액세스 및 배포 시점의 cloudConfig 초기화 스크립트가 포함됩니다.

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

구성 요소	예
	<pre> 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 &amp;&amp; cd /var/www/html &amp;&amp;     wget https://wordpress.org/latest.tar.gz &amp;&amp; tar -xzf /var/www/html/     latest.tar.gz -C /var/www/html/mywordpresssite --strip-components 1     - i=0; while [ \$i -le 10 ]; do mysql --connect-timeout=3 -h \$     {DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" &amp;&amp;     break    sleep 15; i=\$((i+1)); done     - mysql -u root -pmysqlpassword -h \${DBTier.networks[0].address}     -e "create database wordpress_blog;"     - mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/     html/mywordpresssite/wp-config.php     - sed -i -e s/"define( 'DB_NAME',     'database_name_here' );"/"define( 'DB_NAME',     'wordpress_blog' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed     -i -e s/"define( 'DB_USER', 'username_here' );"/"define( 'DB_USER',     'root' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed -i -e     s/"define( 'DB_PASSWORD', 'password_here' );"/"define( 'DB_PASSWORD',     'mysqlpassword' );"/ /var/www/html/mywordpresssite/wp-config.php &amp;&amp; sed     -i -e s/"define( 'DB_HOST', 'localhost' );"/"define( 'DB_HOST', '\$     {DBTier.networks[0].address}' );"/ /var/www/html/mywordpresssite/wp-     config.php     - service apache2 reload </pre>

## 예제: 완료된 기본 클라우드 템플릿 코드 예제

```

formatVersion: 1
inputs:
  env:
    type: string
    enum:
      - env:dev
      - env:prod
      - env:test
    default: env:dev
    title: Environment
    description: Target Environment
  size:
    type: string
    enum:
      - small
      - medium
      - large
    description: Size of Nodes
    title: Tier Machine Size
  username:
    type: string
    minLength: 4
    maxLength: 20

```

```

    pattern: '[a-z]+'
    title: Database Username
    description: Database Username
  userpassword:
    type: string
    pattern: '[a-z0-9A-Z@#&$]+'
    encrypted: true
    title: Database Password
    description: Database Password
  resources:
    WebTier:
      type: Cloud.Machine
      properties:
        name: wordpress
        image: ubuntu
        flavor: '${input.size}'
        constraints:
          - tag: '${input.env}'
        networks:
          - network: '${resource["WP-Network-Private"].id}'
            assignPublicIpAddress: true
      cloudConfig: |
        #cloud-config
        repo_update: true
        repo_upgrade: all
        packages:
          - apache2
          - php
          - php-mysql
          - libapache2-mod-php
          - 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
        - service apache2 reload
    DBTier:
      type: Cloud.Machine
      properties:

```

```

name: mysql
image: ubuntu
flavor: '${input.size}'
constraints:
  - tag: '${input.env}'
networks:
  - network: '${resource["WP-Network-Private"].id}'
    assignPublicIpAddress: true
remoteAccess:
  authentication: usernamePassword
  username: '${input.username}'
  password: '${input.userpassword}'
cloudConfig: |
  #cloud-config
  repo_update: true
  repo_upgrade: all
  packages:
    - mysql-server
  runcmd:
    - sed -e '/bind-address/ s/^#/#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
    - service mysql restart
    - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
    - mysql -e "FLUSH PRIVILEGES;"
  attachedDisks: []
WP-Network-Private:
  type: Cloud.Network
  properties:
    name: WP-Network-Private
    networkType: existing

```

## 다음에 수행할 작업

구문을 검사하고 배포하여 클라우드 템플릿을 테스트합니다.

## 기본 클라우드 템플릿 테스트

설계하는 동안, 필수 항목부터 시작하여 클라우드 템플릿을 구축한 다음, 템플릿이 커지면 배포하고 테스트하는 경우가 많습니다. 이 예제는 vRealize Automation Cloud Assembly에 내장된 진행 중인 테스트 중 일부를 보여줍니다.

배포가 원하는 방식으로 작동하는지 확인하기 위해 클라우드 템플릿을 여러 번 테스트하고 배포할 수 있습니다. 점차적으로 더 많은 리소스를 추가하면서 다시 테스트하고 다시 배포합니다.

## 사전 요구 사항

기본 클라우드 템플릿을 생성합니다. [기본 클라우드 템플릿 생성 항목](#)을 참조하십시오.

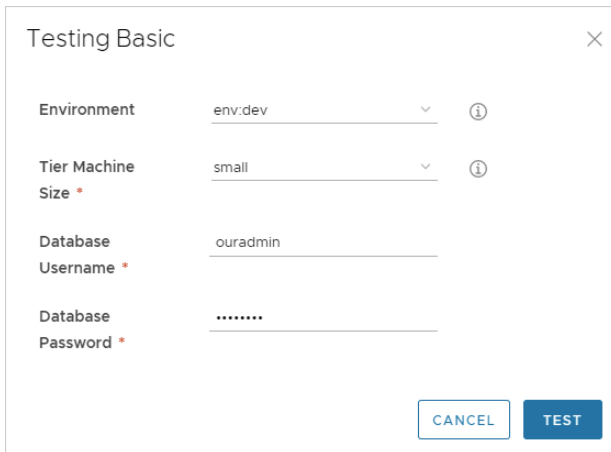
## 절차

- 1 **클라우드 템플릿**을 클릭하고 WordPress-BP 클라우드 템플릿을 엽니다.

설계 캔버스 및 코드 편집기에 기본 클라우드 템플릿이 나타납니다.

- 2 클라우드 템플릿 구문, 배치 및 기본 유효성을 검사하려면 왼쪽 아래에서 **테스트**를 클릭합니다.

### 3 입력 값을 입력하고 **테스트**를 클릭합니다.

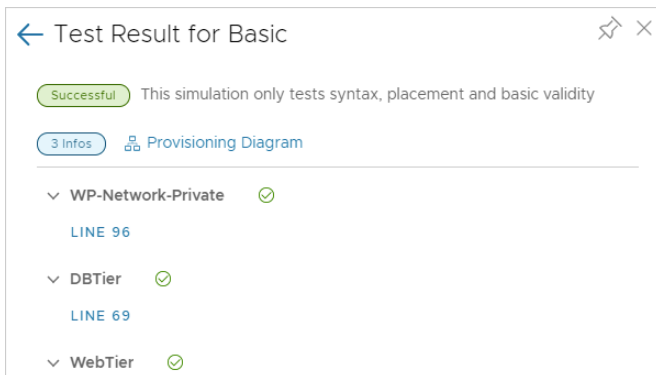


The 'Testing Basic' dialog box contains the following fields:

- Environment:** env:dev (dropdown menu)
- Tier Machine Size:** small (dropdown menu)
- Database Username:** ouradmin (text input)
- Database Password:** ..... (password input)

At the bottom right, there are two buttons: **CANCEL** and **TEST**.

이 테스트는 시뮬레이션일 뿐이며 실제로 가상 시스템이나 다른 리소스를 배포하지는 않습니다.



The 'Test Result for Basic' dialog box shows the following information:

- Status:** Successful (green pill) with the text: This simulation only tests syntax, placement and basic validity.
- Info:** 3 Infos (blue pill) and a link to the Provisioning Diagram (blue icon and text).
- Results:**
  - WP-Network-Private (checked) with a link to LINE 96.
  - DBTier (checked) with a link to LINE 69.
  - WebTier (checked).

테스트에는 **프로비저닝 다이어그램**에 대한 링크가 포함되어 있으며 여기에서 시뮬레이션된 배포 흐름을 검사하고 발생 항목을 확인할 수 있습니다. 시뮬레이션은 잠재적인 문제(예: 클라우드 템플릿의 경성 제약 조건과 일치하는 리소스 기능이 정의되어 있지 않음)를 노출합니다. 다음 예제 오류에서는 기능 태그 env:dev의 클라우드 영역을 정의된 인프라에서 찾을 수 없습니다.



시뮬레이션에 성공한다고 해서 템플릿을 오류 없이 배포할 수 있다고 보장되는 것은 아닙니다.

- 4 템플릿이 시뮬레이션을 통과한 후 왼쪽 아래에서 **배포**를 클릭합니다.
- 5 **새 배포 생성**을 선택합니다.
- 6 배포 이름을 **WordPress for OurCo**로 지정하고 **다음**을 클릭합니다.
- 7 입력 값을 입력하고 **배포**를 클릭합니다.
- 8 템플릿이 성공적으로 배포되었는지 확인하려면 **배포 > 배포** 아래를 확인합니다.

배포가 실패한 경우, 해당 이름을 클릭한 후 **기록** 탭을 클릭하여, 문제 해결을 도와 줄 메시지를 봅니다.

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

일부 기록 항목의 맨 오른쪽에 **프로비저닝 다이어그램** 링크가 있을 수도 있습니다. 다이어그램은 시뮬레이션된 것과 유사하며, 프로비저닝 프로세스에서 vRealize Automation Cloud Assembly 결정 시점의 순서도를 검사할 수 있습니다.

**인프라 > 작업 > 요청** 아래에서 더 많은 순서도를 확인할 수 있습니다.

- 9 애플리케이션이 작동하는지 확인하려면 브라우저에서 **WordPress** 시작 페이지를 엽니다.
  - a **WordPress** 서버가 완전히 생성되고 초기화될 때까지 기다립니다.  
환경에 따라 초기화에 30분 이상이 소요될 수 있습니다.
  - b 사이트 FQDN 또는 IP 주소를 찾으려면 **배포 > 배포 > 토폴로지**로 이동합니다.
  - c 캔버스에서 **WebTier**를 클릭하면 오른쪽 패널에 IP 주소가 표시됩니다.
  - d **WordPress** 시작 페이지에 대한 전체 URL의 일부로 IP 주소를 입력합니다.  
이 예제에서 전체 URL은 다음과 같습니다.  
`http://{IP-address}/mywordpresssite`  
또는  
`http://{IP-address}/mywordpresssite/wp-admin/install.php`
- 10 브라우저에서 **WordPress**를 검사한 후 애플리케이션 작업이 더 필요하면 템플릿을 변경한 후 **기존 배포 업데이트** 옵션을 사용하여 다시 배포합니다.
- 11 클라우드 템플릿의 버전 관리를 고려합니다. 버전을 관리하면 변경 내용 때문에 배포가 실패할 경우에 작동하는 버전으로 되돌릴 수 있습니다.
  - a 클라우드 템플릿 설계 페이지에서 **버전**을 클릭합니다.
  - b 버전 생성 페이지에서 **WP-1.0**을 입력합니다.  
버전 이름에는 공백을 입력하지 마십시오.
  - c **생성**을 클릭합니다.  
검토하거나 특정 버전으로 되돌리려면 설계 페이지에서 **버전 기록** 탭을 클릭합니다.
- 12 이제 기본 배포가 가능하므로 애플리케이션 및 데이터베이스 서버의 CPU와 메모리를 늘려서 첫 번째 배포 시 기능 향상을 테스트합니다.  
둘 모두에서 노드를 보통 크기로 업데이트합니다. 동일한 템플릿을 사용하여, 배포 시 **중형**을 선택한 후 애플리케이션을 다시 배포한 후 다시 확인합니다.

#### 다음에 수행할 작업

훨씬 더 많은 리소스를 추가하여 클라우드 템플릿을 운영 가능한 애플리케이션으로 확장합니다.

### 클라우드 템플릿 확장

예제 애플리케이션의 기본 vRealize Automation Cloud Assembly 템플릿을 생성하고 테스트한 후에는 개발, 테스트를 거쳐 최종적으로 운영 환경에 배포할 수 있는 여러 계층의 애플리케이션으로 확장합니다. 클라우드 템플릿을 확장하려면 다음과 같은 향상된 기능을 추가합니다.

- 용량 증가를 위해 애플리케이션 서버를 클러스터링하는 옵션
- 애플리케이션 서버 앞의 공용 네트워크 및 로드 밸런서

## ■ 아카이브 스토리지가 있는 백업 서버

### 사전 요구 사항

기본 클라우드 템플릿을 생성하여 테스트합니다. [기본 클라우드 템플릿 생성 및 기본 클라우드 템플릿 테스트 항목을 참조하십시오.](#)

### 절차

#### 1 클라우드 템플릿을 클릭하고 WordPress-BP 클라우드 템플릿을 엽니다.

설계 캔버스 및 코드 편집기에 기본 템플릿이 나타납니다.

#### 2 코드 예시 및 그림을 참조하여 내용을 추가하거나 변경합니다.

GUI를 사용하여 로드 밸런서와 같은 새로운 리소스를 캔버스에 끈 다음 코드 편집기에서 구성을 완료합니다.

a WordPress 애플리케이션 서버를 클러스터로 만들려면 count 입력 프롬프트를 추가합니다.

b 클라우드 애그노스틱 로드 밸런서를 추가합니다.

c 로드 밸런서를 WordPress 애플리케이션 서버 클러스터에 연결합니다.

d 클라우드 애그노스틱 백업 시스템을 추가합니다.

e 백업 시스템을 전용/내부 네트워크에 연결합니다.

f 클라우드 애그노스틱 공용/외부 네트워크를 추가합니다.

g 로드 밸런서를 공용 네트워크에 연결합니다.

h 아카이브 디스크로 사용하기 위해 클라우드 애그노스틱 스토리지 볼륨을 추가합니다.

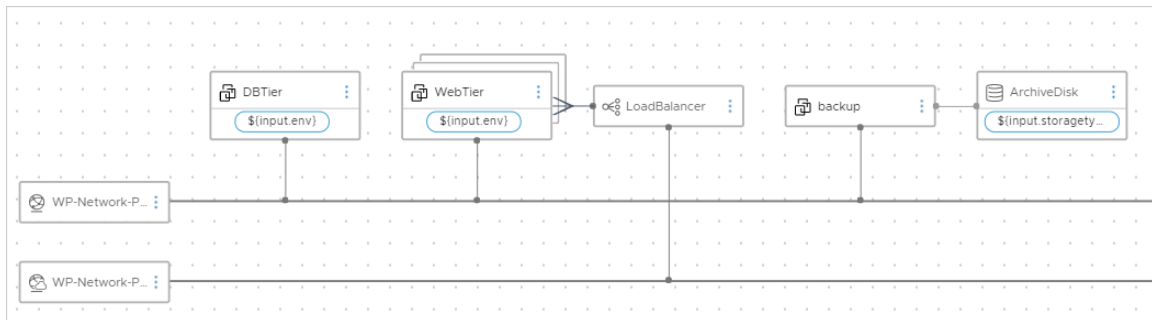
i 아카이브 디스크를 백업 시스템에 연결합니다.

j 아카이브 디스크 속도에 대한 입력 프롬프트를 추가합니다.

#### 3 기본 클라우드 템플릿에서와 동일한 방식으로 배포, 테스트 및 변경합니다.

기존 배포를 업데이트하거나 새 인스턴스를 배포하여 배포를 비교할 수 있습니다.

목표는 운영 배포에 사용할 수 있는 신뢰할 수 있고 반복 가능한 템플릿에 도달하는 것입니다.



## 예제: 완료된 확장 클라우드 템플릿 코드 예제

```

formatVersion: 1
inputs:
  env:
    type: string
    enum:
      - env:dev
      - env:prod
      - env:test
    default: env:dev
    title: Environment
    description: Target Environment
  size:
    type: string
    enum:
      - small
      - medium
      - large
    description: Size of Nodes
    title: Tier Machine Size
  username:
    type: string
    minLength: 4
    maxLength: 20
    pattern: '[a-z]+'
    title: Database Username
    description: Database Username
  userpassword:
    type: string
    pattern: '[a-z0-9A-Z@#&$]+'
    encrypted: true
    title: Database Password
    description: Database Password
  count:
    type: integer
    default: 2
    maximum: 5
    minimum: 2
    title: WordPress Cluster Size
    description: WordPress Cluster Size (Number of Nodes)
  storagetype:
    type: string
    enum:
      - storage:general
      - storage:fast
    description: Archive Storage Disk Type
    title: Archive Disk Type
resources:
  WebTier:
    type: Cloud.Machine
    properties:
      name: wordpress
      image: ubuntu

```

```

    flavor: '${input.size}'
    count: '${input.count}'
    constraints:
      - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all
      packages:
        - apache2
        - php
        - php-mysql
        - libapache2-mod-php
        - 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
        - service apache2 reload
  DBTier:
    type: Cloud.Machine
    properties:
      name: mysql
      image: ubuntu
      flavor: '${input.size}'
      constraints:
        - tag: '${input.env}'
      networks:
        - network: '${resource["WP-Network-Private"].id}'
          assignPublicIpAddress: true
      remoteAccess:
        authentication: usernamePassword
        username: '${input.username}'
        password: '${input.userpassword}'
      cloudConfig: |
        #cloud-config

```

```

    repo_update: true
    repo_upgrade: all
    packages:
      - mysql-server
    runcmd:
      - sed -e '/bind-address/ s/^#*#/' -i /etc/mysql/mysql.conf.d/mysqld.cnf
      - service mysql restart
      - mysql -e "GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'mysqlpassword';"
      - mysql -e "FLUSH PRIVILEGES;"
    attachedDisks: []
  LoadBalancer:
    type: Cloud.LoadBalancer
    properties:
      name: myapp-lb
      network: '${resource["WP-Network-Public"].id}'
      instances:
        - '${WebTier.id}'
      routes:
        - protocol: HTTP
          port: '80'
          instanceProtocol: HTTP
          instancePort: '80'
          healthCheckConfiguration:
            protocol: HTTP
            port: '80'
            urlPath: /mywordpresssite/wp-admin/install.php
            intervalSeconds: 6
            timeoutSeconds: 5
            unhealthyThreshold: 2
            healthyThreshold: 2
      internetFacing: true
  WP-Network-Private:
    type: Cloud.Network
    properties:
      name: WP-Network-Private
      networkType: existing
  WP-Network-Public:
    type: Cloud.Network
    properties:
      name: WP-Network-Public
      networkType: public
  backup:
    type: Cloud.Machine
    properties:
      name: backup
      flavor: '${input.size}'
      image: ubuntu
      networks:
        - network: '${resource["WP-Network-Private"].id}'
      attachedDisks:
        - source: '${resource.ArchiveDisk.id}'
  ArchiveDisk:
    type: Cloud.Volume
    properties:
      name: ArchiveDisk

```

```
capacityGb: 5
constraints:
  - tag: '${input.storageType}'
```

### 다음에 수행할 작업

자체 인프라를 정의하고 자체 클라우드 템플릿을 생성합니다.

[장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축](#) 및 [장 6 vRealize Automation Cloud Assembly 배포 설계 항목](#)을 참조하십시오.

## 자습서: vRealize Automation에 대해 VMware Cloud on AWS 구성

이 vRealize Automation 자습서는 VMware Cloud on AWS 환경에 배포할 리소스 인프라 및 클라우드 템플릿 설정을 정의하는 프로세스를 설명합니다.

이 절차를 수행하려면 [VMware Cloud on AWS 시작 설명서](#)의 "소프트웨어 정의 데이터 센터 배포 및 관리"에 설명된 대로 관리자가 귀사의 VMware Cloud on AWS SDDC 데이터 센터를 이미 구성한 상태여야 합니다.

VMware Cloud on AWS에 대한 환경을 구성하는 프로세스를 이해하려면 순차적 설정을 확인하십시오. 여기에 나와 있는 값은 사용 사례 예시에만 해당합니다. 클라우드 인프라 및 배포 요구 사항에 맞게 고유한 값으로 대체하거나 예시의 값에서 추론할 수 있는 부분에 대해 생각해 보십시오.



관련 정보는 [Cloud Assembly용 VMware Cloud on AWS를 구성하는 방법](#) 비디오를 참조하십시오.

### 절차

#### 1 vRealize Automation에서 기본 VMware Cloud on AWS 워크플로 구성

이 사용 사례는 VMware Cloud on AWS 환경에 배포하기 위해 리소스 인프라와 해당 클라우드 템플릿을 정의하는 프로세스를 보여줍니다.

#### 2 vRealize Automation의 VMware Cloud on AWS 워크플로에서 격리된 네트워크 구성

이 절차에서는 vRealize Automation에서 VMware Cloud on AWS 배포에 대해 격리된 네트워크를 추가합니다.

## vRealize Automation에서 기본 VMware Cloud on AWS 워크플로 구성

이 사용 사례는 VMware Cloud on AWS 환경에 배포하기 위해 리소스 인프라와 해당 클라우드 템플릿을 정의하는 프로세스를 보여줍니다.

이 절차에서 기존 VMware Cloud on AWS 환경의 리소스에 대한 클라우드 템플릿 배포를 지원하는 인프라를 구성합니다.

## 사전 요구 사항

- vRealize Automation Cloud Assembly에서 VMware Cloud on AWS 클라우드 계정을 생성하고 구성하려면 우선 기존 VMware Cloud on AWS SDDC 환경에서 조직의 일부여야 합니다. VMware Cloud on AWS 서비스 구성에 대한 자세한 내용은 [VMware Cloud on AWS 설명서](#)를 참조하십시오.
- vCenter에 있는 기존 VMware Cloud on AWS 호스트 SDDC와 vRealize Automation Cloud Assembly에 있는 VMware Cloud on AWS 클라우드 계정 간에 필요한 연결을 용이하게 하려면 네트워크 연결을 제공하고 VPN 또는 유사한 네트워킹 수단을 사용하여 방화벽 규칙을 추가해야 합니다. vRealize Automation에서 VMware Cloud on AWS 클라우드 계정에 연결할 VMware Cloud on AWS SDDC 준비의 내용을 참조하십시오.

## 절차

- 1 [vRealize Automation에서 VMware Cloud on AWS 클라우드 계정에 연결할 VMware Cloud on AWS SDDC 준비](#)  
vRealize Automation 환경에서 VMware Cloud on AWS 클라우드 계정을 사용하는 경우 vCenter의 SDDC와 vRealize Automation의 VMware Cloud on AWS 클라우드 계정 간에 통신을 지원하도록 네트워크 연결을 생성하고 규칙을 구성해야 합니다.
- 2 [샘플 워크플로 내의 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성](#)  
이 단계에서는 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정을 생성합니다.
- 3 [vRealize Automation에서 VMware Cloud on AWS 배포에 대한 클라우드 영역 생성](#)  
이 단계에서는 클라우드 영역을 생성하고 vRealize Automation에서 VMware Cloud on AWS로 작업할 때 CloudAdmin 사용자가 액세스할 수 있는 계산 리소스를 지정합니다.
- 4 [vRealize Automation에서 VMware Cloud on AWS 배포를 위한 네트워크 및 스토리지 프로파일 구성](#)  
이 단계에서는 네트워크 프로파일 및 스토리지 프로파일을 구성하여 vRealize Automation의 VMware Cloud on AWS CloudAdmin 사용자가 사용할 수 있는 리소스를 지정합니다.
- 5 [vRealize Automation에서 VMware Cloud on AWS 배포를 지원하는 프로젝트 생성](#)  
이 단계에서는 VMware Cloud on AWS 배포에 사용할 수 있는 리소스를 제어하는 데 사용하는 vRealize Automation 프로젝트를 정의합니다.
- 6 [vRealize Automation에서 VMware Cloud on AWS 배포를 지원하도록 클라우드 템플릿 설계에 vCenter 시스템 리소스 정의](#)  
이 단계에서는 vCenter 시스템 리소스를 설계 캔버스로 끌어와서 vRealize Automation에서 VMware Cloud on AWS 배포에 대한 설정을 추가합니다.

## vRealize Automation에서 VMware Cloud on AWS 클라우드 계정에 연결할 VMware Cloud on AWS SDDC 준비

vRealize Automation 환경에서 VMware Cloud on AWS 클라우드 계정을 사용하는 경우 vCenter의 SDDC와 vRealize Automation의 VMware Cloud on AWS 클라우드 계정 간에 통신을 지원하도록 네트워크 연결을 생성하고 규칙을 구성해야 합니다.

SDDC 통신을 지원하는 데 필요한 연결 및 규칙을 구성합니다.

vCenter의 기존 VMware Cloud on AWS 호스트 SDDC와 vRealize Automation의 VMware Cloud on AWS 클라우드 계정 사이에 필요한 연결을 용이하게 하려면 VPN 또는 유사한 네트워킹 수단을 사용하여 두 요소 간에 네트워크 연결을 제공해야 합니다.

### 1 공용 인터넷 또는 AWS Direct Connect를 통해 VPN 연결을 구성합니다.

온-프레미스 데이터 센터에 대한 VPN 연결 구성 및 VMware Cloud on AWS에 대한 AWS Direct Connect 구성에 대한 자세한 내용은 [VMware Cloud on AWS 설명서](#)에서 "VMware Cloud on AWS 네트워킹 및 보안"을 참조하십시오.

### 2 관리 네트워크의 개인 IP 주소에서 vCenter Server FQDN을 확인할 수 있는지 확인합니다.

vCenter Server FQDN 확인 주소 설정에 대한 자세한 내용은 [VMware Cloud on AWS 설명서](#)에서 "VMware Cloud on AWS 네트워킹 및 보안"을 참조하십시오.

### 3 필요한 방화벽 규칙을 구성합니다.

통신을 지원하도록 SDDC의 VMware Cloud on AWS 콘솔에서 관리 게이트웨이 방화벽 규칙을 구성해야 합니다. 규칙은 **관리 게이트웨이** 방화벽 규칙 섹션에 있어야 합니다. 방화벽 규칙은 SDDC 콘솔에서 **네트워킹 및 보안** 탭의 옵션을 사용하여 생성합니다.

- HTTPS(TCP 443) 서비스에 대한 ESXi로의 네트워크 트래픽을 vRealize Automation 장치/서버 또는 vRealize Automation 로드 밸런서 VIP의 검색된 IP 주소로 제한합니다.
- ICMP(All ICMP), SSO(TCP 7444) 및 HTTPS(TCP 443) 서비스에 대한 vCenter로의 네트워크 트래픽을 vRealize Automation 장치/서버 또는 vRealize Automation 로드 밸런서 VIP의 검색된 IP 주소로 제한합니다.
- HTTPS(TCP 443) 서비스에 대한 NSX-T Manager로의 네트워크 트래픽을 vRealize Automation 장치/서버 또는 vRealize Automation 로드 밸런서 VIP의 검색된 IP 주소로 제한합니다.

필요한 방화벽 규칙은 다음 표에 요약되어 있습니다.

**표 2-1. 필요한 관리 게이트웨이 방화벽 규칙 요약**

이름	소스	대상	서비스
vCenter	온-프레미스 데이터 센터의 CIDR 블록	vCenter	임의(모든 트래픽)
vCenter ping	임의	vCenter	ICMP(모든 ICMP)
NSX Manager	온-프레미스 데이터 센터의 CIDR 블록	NSX Manager	임의(모든 트래픽)
온-프레미스에서 ESXi로 ping	온-프레미스 데이터 센터의 CIDR 블록	ESXi 관리만	ICMP(모든 ICMP)
온-프레미스에서 ESXi로 원격 콘솔 및 프로비저닝	온-프레미스 데이터 센터의 CIDR 블록	ESXi 관리만	TCP 902

표 2-1. 필요한 관리 게이트웨이 방화벽 규칙 요약 (계속)

이름	소스	대상	서비스
온-프레미스에서 SDDC VM으로	온-프레미스 데이터 센터의 CIDR 블록	SDDC 논리적 네트워크의 CIDR 블록	임의(모든 트래픽)
SDDC VM에서 온-프레미스로	SDDC 논리적 네트워크의 CIDR 블록	온-프레미스 데이터 센터의 CIDR 블록	임의(모든 트래픽)

관련 정보는 [VMware Cloud on AWS 설명서](#)에서 "VMware Cloud on AWS 네트워킹 및 보안" 및 "VMware Cloud on AWS 작업 가이드" 를 참조하십시오.

필요한 게이트웨이 액세스 및 방화벽 규칙을 구성한 후에는 VMware Cloud on AWS 클라우드 계정 생성 프로세스를 계속할 수 있습니다. [샘플 워크플로](#) 내의 [vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성](#)의 내용을 참조하십시오.

## 샘플 워크플로 내의 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성

이 단계에서는 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정을 생성합니다.

관련 정보는 [VMware Cloud on AWS 설명서](#)를 참조하십시오.

달리 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

### 사전 요구 사항

- 이 절차는 vCenter의 대상 SDDC에 대한 VMware Cloud on AWS CloudAdmin 자격 증명을 비롯해 필요한 관리자 자격 증명에 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했다고 가정합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목](#)을 참조하십시오.
- 이 절차에서는 사용자에게 클라우드 관리자 사용자 역할이 있다고 가정합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- vCenter의 기존 VMware Cloud on AWS 호스트 SDDC와 vRealize Automation의 VMware Cloud on AWS 클라우드 계정 사이에 필요한 연결을 용이하게 하려면, VPN 또는 유사한 네트워킹 수단을 사용하여 네트워크 연결 및 방화벽 규칙을 제공해야 합니다. [vRealize Automation에서 VMware Cloud on AWS 클라우드 계정에 연결할 VMware Cloud on AWS SDDC 준비](#)의 내용을 참조하십시오. 외부 HTTP 인터넷 프록시를 사용 중인 경우에는 IPv4에 대해 해당 프록시를 구성해야 합니다.
- 외부 인터넷에 액세스할 수 없는 경우 인터넷 서버 프록시를 구성합니다. [vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법](#)의 내용을 참조하십시오.

### 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택합니다.
- 2 **클라우드 계정 추가**를 클릭하고 VMware Cloud on AWS를 선택한 후 값을 입력합니다.

샘플 값 및 지원 정보는 다음 표에 나와 있습니다.

설정	샘플 값 및 지침	설명
VMC API 토큰	<ol style="list-style-type: none"> <li>1 <b>VMC API 토큰</b> 줄 끝에 있는 "i" 도움말 아이콘을 클릭하고 도움말 텍스트 상자에서 <b>API 토큰 페이지</b>를 클릭하여 조직의 <b>내 계정</b> 페이지에서 <b>API 토큰</b> 탭을 엽니다.</li> <li>2 <b>토큰 생성</b>을 클릭하여 <b>새 API 토큰 생성</b> 옵션을 표시합니다.</li> <li>3 새 토큰 이름(예: <b>myinitials_mytoken</b>)을 입력합니다.</li> <li>4 <b>토큰 TTL</b>을 <b>만료되지 않음</b>으로 설정합니다.  만료되도록 설정된 토큰을 생성하는 경우, 토큰이 만료되면 vRealize Automation에서 VMware Cloud on AWS 작업이 중지되며, 새 토큰으로 클라우드 계정을 업데이트할 때까지 계속 작동하지 않습니다.</li> <li>5 <b>범위 정의</b> 섹션에서 <b>모든 역할</b>을 선택합니다.   <div data-bbox="501 863 884 1052" data-label="Image"> </div> </li> <li>6 <b>생성</b>을 클릭합니다.</li> <li>7 [생성된 토큰] 페이지에서 <b>복사</b>를 클릭하고 <b>계속</b>을 클릭합니다.</li> <li>8 <b>새 클라우드 계정</b> 페이지로 돌아가서 복사한 토큰을 <b>VMC API 토큰</b> 행에 붙여넣고 <b>API 토큰 적용</b>을 클릭합니다.   <div data-bbox="501 1297 842 1507" data-label="Image"> </div> </li> </ol>	<p>새 토큰을 생성하거나 연결된 <b>API 토큰</b> 페이지에서 조직에 대한 기존 토큰을 사용할 수 있습니다.</p> <p><b>범위 정의</b> 섹션에서 API 토큰에 필요한 최소 역할은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>■ <b>조직 역할</b> <ul style="list-style-type: none"> <li>■ 조직 멤버</li> <li>■ 조직 소유자</li> </ul> </li> <li>■ <b>서비스 역할 - VMware Cloud on AWS</b> <ul style="list-style-type: none"> <li>■ 관리자</li> <li>■ NSX Cloud 관리자</li> <li>■ NSX Cloud 감사자</li> </ul> </li> </ul> <p><b>참고</b> 생성된 토큰을 복사, 다운로드 또는 인쇄합니다. 이 페이지에서 나가면 생성된 토큰을 검색할 수 없습니다.</p> <p>생성 또는 제공된 토큰을 적용하여 조직의 VMware Cloud on AWS 구독에서 사용할 수 있는 SDDC 환경에 연결하고 SDDC 이름 목록을 채웁니다.</p> <p>vRealize Automation 및 VMware Cloud on AWS 서비스가 서로 다른 조직에 있으면 VMware Cloud on AWS 조직으로 전환한 다음 토큰을 생성해야 합니다.</p> <p>API 토큰에 대한 자세한 내용은 <a href="#">API 토큰 생성</a>을 참조하십시오.</p>
SDDC 이름	<p>이 예에서는 <b>Datacenter:Datacenter-abz</b>를 선택합니다.</p> <p>유효한 SDDC 이름이 vCenter 및 NSX-T FQDN 항목에 자동으로 채워집니다. 클라우드 프록시가 SDDC에 이미 배포되어 있다면 클라우드 프록시 값도 자동으로 채워집니다.</p>	<p>VMware Cloud on AWS 구독의 사용 가능한 SDDC 목록에서 선택합니다. SDDC 목록은 VMware Cloud on AWS API 토큰을 기반으로 합니다.</p> <p>NSX-V SDDC는 vRealize Automation에서 지원되지 않기 때문에 사용 가능한 SDDC 목록에 표시되지 않습니다.</p>

설정	샘플 값 및 지침	설명
vCenter IP 주소/ FQDN	주소는 선택한 SDDC에 기반하여 자동으로 채워집니다.	지정된 SDDC에 있는 vCenter Server의 IP 주소 또는 FQDN을 입력합니다.  IP 주소는 기본적으로 개인 IP 주소로 설정됩니다. SDDC에 액세스하는 데 사용되는 네트워크 연결 유형에 따라 기본 주소는 지정된 SDDC에 있는 NSX Manager 서버의 IP 주소와 다를 수 있습니다.
NSX Manager IP 주소/FQDN	주소는 선택한 SDDC에 기반하여 자동으로 채워집니다.	지정된 SDDC에 있는 NSX Manager의 IP 주소 또는 FQDN을 지정합니다.  IP 주소는 기본적으로 개인 IP 주소로 설정됩니다. SDDC에 액세스하는 데 사용되는 네트워크 연결 유형에 따라 기본 주소는 지정된 SDDC에 있는 NSX Manager 서버의 IP 주소와 다를 수 있습니다.  VMware Cloud on AWS 클라우드 계정은 NSX-T를 지원합니다.
vCenter 사용자 이름 및 암호	cloudadmin@vmc.local이라는 사용자가 이 자동으로 채워집니다.	지정된 SDDC의 vCenter 사용자 이름을 입력합니다(기본값과 다른 경우).  지정된 사용자에게 CloudAdmin 자격 증명이 필요합니다. 사용자에게 CloudGlobalAdmin 자격 증명이 필요하지 않습니다.  사용자 암호를 입력합니다.
검증	<b>검증</b> 을 클릭합니다.  Error updating endpoint <Name>: Endpoint already exists을 수신하면 클라우드 계정이 해당 SDDC에 이미 연결된 것입니다.	검증 기능은 지정된 vCenter에 대한 사용자의 액세스 권한을 확인하고 vCenter가 실행 중인지 확인합니다.
이름 및 설명	클라우드 계정 이름으로 <b>OurCo-VMC</b> 를 입력합니다.  클라우드 계정 설명으로 <b>Sample deployment for VMC</b> 를 입력합니다.	
이러한 데이터 센터로의 프로비저닝 허용	이 정보는 읽기 전용입니다.	지정된 VMware Cloud on AWS SDDC 환경에 있는 사용 가능한 데이터 센터가 나열됩니다.
클라우드 영역 생성	확인란을 선택 취소합니다. 이 예의 경우 나중에 워크플로에서 클라우드 영역을 생성합니다.	<a href="#">vRealize Automation Cloud Assembly</a> 클라우드 영역에 대해 알아보기 항목을 참조하십시오.
기능 태그	비워 둡니다. 이 워크플로는 기능 태그를 사용하지 않습니다.	조직의 태그 전략에 따라 태그를 사용하십시오. 태그를 사용하여 <a href="#">vRealize Automation Cloud Assembly</a> 리소스 및 배포를 관리하는 방법 및 태그 지정 전략 생성 항목을 참조하십시오.

vSphere에 배포된 VM과 마찬가지로 VMware Cloud on AWS에 배포할 VM에 대해 시스템 태그를 구성할 수 있습니다. 초기 배포 후에 시스템 태그를 업데이트할 수도 있습니다. 이러한 시스템 태그를 사용하면 배포 중에 vRealize Automation에서 VM을 적절한 NSX-T 보안 그룹에 동적으로 할당할 수 있습니다. 관련 정보는 [vRealize Automation 클라우드 템플릿에서 보안 그룹 리소스 사용 항목을 참조하십시오](#).

### 3 추가를 클릭합니다.

#### 결과

시스템 및 볼륨 같은 리소스 데이터가 VMware Cloud on AWS SDDC 데이터 센터에서 수집되어 vRealize Automation **인프라** 탭의 **리소스** 섹션에 나열됩니다.

#### 다음에 수행할 작업

[vRealize Automation에서 VMware Cloud on AWS 배포에 대한 클라우드 영역 생성](#).

### vRealize Automation에서 VMware Cloud on AWS 배포에 대한 클라우드 영역 생성

이 단계에서는 클라우드 영역을 생성하고 vRealize Automation에서 VMware Cloud on AWS로 작업할 때 CloudAdmin 사용자가 액세스할 수 있는 계산 리소스를 지정합니다.

VMware Cloud on AWS의 두 가지 기본 관리자 자격 증명은 CloudGlobalAdmin과 CloudAdmin입니다. vRealize Automation Cloud Assembly는 CloudAdmin 사용자를 지원하도록 설계되었습니다. VMware Cloud on AWS CloudAdmin 사용자가 사용할 수 있는 리소스에 배포합니다. VMware Cloud on AWS CloudGlobalAdmin 자격 증명이 필요한 리소스에는 배포하지 마십시오.

클라우드 영역은 프로젝트 클라우드 템플릿이 시스템, 네트워크 및 스토리지를 배포하는 계산 리소스를 식별합니다. [vRealize Automation Cloud Assembly 클라우드 영역에 대해 알아보기](#) 항목을 참조하십시오.

달리 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

#### 사전 요구 사항

- 샘플 워크플로 내의 [vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성](#) 절차를 완료합니다.
- 이 절차에서는 사용자에게 vCenter의 대상 SDDC에 대한 VMware Cloud on AWS CloudAdmin 자격 증명을 포함하여 필요한 관리자 자격 증명이 있다고 가정합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 이 절차에서는 사용자에게 클라우드 관리자 사용자 역할이 있다고 가정합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.

#### 절차

- 1 **인프라 > 구성 > 클라우드 영역**을 선택합니다.

## 2 새 클라우드 영역을 클릭하고 VMware Cloud on AWS 환경에 대한 값을 입력합니다.

설정	샘플 값
계정/지역	OurCo-VMC/Datacenter:Datacenter-abz 이전 단계인 샘플 워크플로 내의 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성에서 정의한 클라우드 계정 및 연결된 지역입니다.
이름	VMC_cloud_zone-1
설명	VMware Cloud on AWS 리소스만
배치 정책	기본값
기능 태그	비워 둡니다. 이 워크플로는 기능 태그를 사용하지 않습니다.

## 3 계산 탭을 클릭합니다.

- 4 아래 영역 1에 표시된 대로, CloudAdmin 사용자가 사용할 수 있는 계산 리소스를 찾아서 선택합니다. 이 예에서는 Cluster 1/ Compute-ResourcePool이라는 리소스를 사용합니다.

Cluster 1/ Compute-ResourcePool은 VMware Cloud on AWS에 대한 기본 계산 리소스입니다.

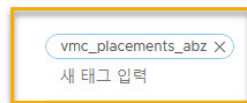


- 5 위의 영역 2에 표시된 대로 태그 이름 vmc\_placements\_abz를 추가합니다.

태그

개체 1개 선택됨

태그 추가



태그 제거

태그 없음 ⓘ

6 **필터 태그** 섹션에 `vmc_placements_abz`를 입력하여 이 클라우드 영역에서 사용되는 계산 리소스를 필터링합니다.

7 **저장**을 클릭합니다.

이름	계정/지역	유형	태그
<input type="checkbox"/> ComputeClusterA	LK-TEST 测试表示A中正在部署的Ubuntu / NSX62-Scale-DC	common title cluster	Cluster-ComputeClusterA
<input checked="" type="checkbox"/> ComputeClusterA-New	nsx-测试表示A中正在部署的Ubuntu / NSX62-DataCenter	common title cluster	ComputeClusterA
<input type="checkbox"/> ComputeClusterA / Scale	270_VC_account 测试表示A中正在部署的Ubuntu / NSX62-Scale-DC	ResourcePool	ComputeClusterA

이 예에서는 이름이 Cluster 1/ Compute-ResourcePool인 계산 리소스만 CloudAdmin 사용자에게 제공됩니다.

다음에 수행할 작업

vRealize Automation에서 VMware Cloud on AWS 배포를 위한 네트워크 및 스토리지 프로파일 구성.

## vRealize Automation에서 VMware Cloud on AWS 배포를 위한 네트워크 및 스토리지 프로파일 구성

이 단계에서는 네트워크 프로파일 및 스토리지 프로파일을 구성하여 vRealize Automation의 VMware Cloud on AWS CloudAdmin 사용자가 사용할 수 있는 리소스를 지정합니다.

이미지 및 버전 값도 필요하지만 VMware Cloud on AWS 사용자 자격 증명과 관련된 고유 항목은 없습니다. 이 예에서는 클라우드 템플릿을 정의할 때 버전 값은 `small`을 사용하고 이미지 값은 `ubuntu-16`을 사용합니다.

매핑 및 프로파일에 대한 일반적인 내용은 [장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축](#) 항목을 참조하십시오.

다들 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

### 사전 요구 사항

- 클라우드 영역을 생성합니다. vRealize Automation에서 VMware Cloud on AWS 배포에 대한 클라우드 영역 생성 항목을 참조하십시오.
- 이 절차에서는 사용자에게 vCenter의 대상 SDDC에 대한 VMware Cloud on AWS CloudAdmin 자격 증명을 포함하여 필요한 관리자 자격 증명이 있다고 가정합니다. vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.
- 이 절차에서는 사용자에게 클라우드 관리자 사용자 역할이 있다고 가정합니다. vRealize Automation 사용자 역할이란? 항목을 참조하십시오.

## 절차

1 VMware Cloud on AWS 배포에 대한 네트워크 프로파일을 정의합니다.

a 인프라 > 구성 > 네트워크 프로파일을 선택하고 새 네트워크 프로파일을 클릭합니다.

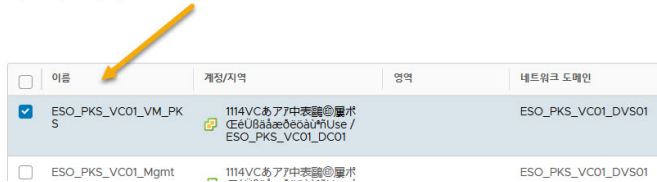
설정	샘플 값
계정/지역	OurCo-VMC/Datacenter:Datacenter-abz
이름	vmc-network1
설명	VMware Cloud on AWS CloudAdmin 자격 증명이 있는 클라우드 템플릿 관리자가 액세스할 수 있는 네트워크가 포함되어 있습니다.

**참고** 샘플 워크플로 내의 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성에서 생성한 VMware Cloud on AWS 클라우드 계정 및 이와 일치하는 SDDC 데이터 센터를 선택합니다.

b 네트워크 탭을 클릭하고 네트워크 추가를 클릭합니다.

c CloudAdmin 자격 증명이 있는 VMware Cloud on AWS 사용자가 배포할 수 있는 네트워크(예: sddc-cgw-network-1)를 선택합니다.

네트워크 추가



<input type="checkbox"/>	이름	계정/지역	영역	네트워크 도메인
<input checked="" type="checkbox"/>	ESO_PKS_VC01_VM_PKS	1114VC아7中審監監術 CEU8a8ae0e0auiUse / ESO_PKS_VC01_DC01		ESO_PKS_VC01_DVS01
<input type="checkbox"/>	ESO_PKS_VC01_Mgmt	1114VC아7中審監監術		ESO_PKS_VC01_DVS01

2 네트워크 프로파일을 저장합니다.

### 3 VMware Cloud on AWS 배포에 대한 스토리지 프로파일을 정의합니다.

CloudAdmin 사용자가 액세스할 수 있는 데이터스토어/클러스터를 대상으로 하는 스토리지 프로파일을 구성합니다.

a 인프라 > 구성 > 스토리지 프로파일을 선택하고 새 스토리지 프로파일을 클릭합니다.

설정	샘플 값
계정/지역	OurCo-VMC/Datacenter:Datacenter-abz 샘플 워크플로 내의 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성에서 생성한 VMware Cloud on AWS 클라우드 계정 및 이와 일치하는 SDDC 데이터 센터를 선택합니다.
이름	vmc-storage1
설명	VMware Cloud on AWS CloudAdmin 자격 증명이 있는 클라우드 템플릿 관리자가 배포할 수 있는 데이터스토어 클러스터가 포함되어 있습니다.

b 데이터스토어/클러스터 드롭다운 메뉴에서 WorkloadDatastore 데이터스토어를 선택합니다.



vRealize Automation Cloud Assembly의 VMware Cloud on AWS인 경우 스토리지 정책에 **WorkloadDatastore** 데이터스토어를 사용하여 VMware Cloud on AWS 배포를 지원해야 합니다.

### 4 스토리지 프로파일을 저장합니다.

다음에 수행할 작업

vRealize Automation에서 VMware Cloud on AWS 배포를 지원하는 프로젝트 생성.

### vRealize Automation에서 VMware Cloud on AWS 배포를 지원하는 프로젝트 생성

이 단계에서는 VMware Cloud on AWS 배포에 사용할 수 있는 리소스를 제어하는 데 사용하는 vRealize Automation 프로젝트를 정의합니다.

프로젝트에 대한 자세한 내용은 배포 시간에 vRealize Automation Cloud Assembly 프로젝트가 작동하는 방식 항목을 참조하십시오.

다들 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

#### 사전 요구 사항

- vRealize Automation에서 VMware Cloud on AWS 배포를 위한 네트워크 및 스토리지 프로파일 구성 절차를 완료합니다.

- 이 절차에서는 사용자에게 vCenter의 대상 SDDC에 대한 VMware Cloud on AWS CloudAdmin 자격 증명을 포함하여 필요한 관리자 자격 증명이라고 가정합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 이 절차에서는 사용자에게 클라우드 관리자 사용자 역할이라고 가정합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.

## 절차

1 **인프라 > 관리 > 프로젝트**를 선택합니다.

2 새 **프로젝트**를 클릭하고 VMC\_proj-1\_abz를 프로젝트 이름으로 입력합니다.

3 **사용자**를 클릭하고 **사용자 추가**를 클릭합니다.

사용자에게 조직의 VMware Cloud on AWS 구독에 대한 CloudAdmin 자격 증명が必要です.

- chris.gray@ourco.com, 관리자
- kerry.white@ourco.com, Member

4 **프로비저닝**을 클릭한 다음 **클라우드 영역 추가**를 클릭합니다.

5 이전 단계에서 구성한 클라우드 영역을 추가합니다.

설정	샘플 값
클라우드 영역	VMC_cloud_zone-1 이 클라우드 영역은 이전 단계인 <a href="#">vRealize Automation에서 VMware Cloud on AWS</a> 배포에 대한 클라우드 영역 생성에서 생성했습니다.
프로비저닝 우선 순위	1
인스턴스 제한	3

6 이 예에서는 다른 옵션을 무시합니다.

## 다음에 수행할 작업

VMware Cloud on AWS 환경에 배포할 클라우드 템플릿을 생성합니다. [vRealize Automation에서 VMware Cloud on AWS](#) 배포를 지원하도록 클라우드 템플릿 설계에 [vCenter 시스템 리소스 정의](#) 항목을 참조하십시오.

## vRealize Automation에서 VMware Cloud on AWS 배포를 지원하도록 클라우드 템플릿 설계에 vCenter 시스템 리소스 정의

이 단계에서는 vCenter 시스템 리소스를 설계 캔버스로 끌어와서 vRealize Automation에서 VMware Cloud on AWS 배포에 대한 설정을 추가합니다.

사용 가능한 VMware Cloud on AWS 리소스에 배포할 수 있는 클라우드 템플릿 설계를 생성합니다.

달리 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

## 사전 요구 사항

- 이 절차에서는 사용자에게 클라우드 템플릿 설계자 자격 증명이 있다고 가정합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 이 절차에서는 vCenter(Datacenter:Datacenter-abz)의 대상 SDDC에 대해 VMware Cloud on AWS CloudAdmin 자격 증명이 있다고 가정합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 앞의 섹션에 설명된 대로 리소스 인프라 및 프로젝트를 구성합니다.

## 절차

- 1 **설계** 탭을 클릭한 다음, **새로 만들기**를 클릭합니다.

설정	샘플 값
이름	vmc-bp_abz
설명	1
프로젝트	VMC_proj-1_abz 이전에 생성한 프로젝트이며 이전에 생성한 클라우드 영역을 지원합니다. 이제 프로젝트가 클라우드 영역과 연결되며, 이 영역은 차례로 이전에 생성한 VMware Cloud on AWS 클라우드 계정/지역과 연결됩니다.

- 2 vSphere 시스템 리소스를 캔버스로 씁니다.

- 3 시스템 리소스에서 다음(굵게 표시됨) 클라우드 템플릿 리소스 코드를 편집합니다.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: ubuntu-1604
      cpuCount: 1
      totalMemoryMB: 1024
      folderName: Workloads
```

image에는 배포 요구 사항에 적합한 값을 지정할 수 있습니다.

VMware Cloud on AWS 배포를 지원하려면 클라우드 템플릿 설계 코드에 `folderName: Workloads` 문을 추가해야 합니다. `folderName: Workloads` 설정은 VMware Cloud on AWS SDDC 환경에서 CloudAdmin 자격 증명을 지원하며 필수입니다.

참고: 위의 코드 샘플에 나와 있는 `folderName: Workloads` 설정이 필요하지만 위에 나와 있는 대로 클라우드 템플릿 코드에 직접 추가하거나 연결된 클라우드 영역 또는 프로젝트에 추가할 수 있습니다. 설정이 이러한 세 위치 중 하나 이상에 지정된 경우 우선 순위는 다음과 같습니다.

- 프로젝트 설정은 클라우드 템플릿 설정과 클라우드 영역 설정을 재정의합니다.

- 클라우드 템플릿 설정은 클라우드 영역 설정을 재정의합니다.

참고: 필요에 따라 `cpuCount` 및 `totalMemoryMB` 설정을 `flavor`(크기 조정) 항목으로 교체할 수 있습니다. 아래를 참조하세요.

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

클라우드 영역의 폴더 값이 **Workloads**로 설정된 경우 클라우드 영역 폴더 값을 재정의하려는 경우가 아니면 클라우드 템플릿에 `folderName` 속성을 설정할 필요가 없습니다.

#### 다음에 수행할 작업

네트워크 격리를 추가하여 기본 VMware Cloud on AWS 워크플로를 확장합니다. [vRealize Automation의 VMware Cloud on AWS 워크플로에서 격리된 네트워크 구성](#) 항목을 참조하십시오.

## vRealize Automation의 VMware Cloud on AWS 워크플로에서 격리된 네트워크 구성

이 절차에서는 vRealize Automation에서 VMware Cloud on AWS 배포에 대해 격리된 네트워크를 추가합니다.

VMware Cloud on AWS 클라우드 계정을 정의할 때 VMware Cloud on AWS 서비스에 구성된 NSX-T 설정을 사용할 수 있습니다. VMware Cloud on AWS 서비스에서 NSX-T 설정을 구성하는 방법에 대한 자세한 내용은 VMware Cloud on AWS [제품 설명서](#)를 참조하십시오.

vRealize Automation는 VMware Cloud on AWS에서 NSX-T를 지원합니다. VMware Cloud on AWS에는 NSX-V가 지원되지 않습니다.

vRealize Automation는 VMware Cloud on AWS 배포에 대한 네트워크 분리를 지원합니다. VMware Cloud on AWS에는 다른 네트워크 방법이 지원되지 않습니다.

기본 VMware Cloud on AWS 워크플로에 대한 확장은 클라우드 템플릿에 사용할 격리된 네트워크를 생성하는 다음과 같은 방법을 설명합니다.

- 주문형 네트워크 기반 격리 구성.
- 주문형 보안 그룹 기반 격리 구성.

## 사전 요구 사항

이 절차는 기본 VMware Cloud on AWS 워크플로를 확장합니다. 여기에는 [자습서: vRealize Automation에 대해 VMware Cloud on AWS 구성](#) 워크플로에 구성한 것과 동일한 클라우드 계정 및 지역, 클라우드 영역, 프로젝트 및 네트워크 프로파일이 사용됩니다.

## 절차

- 1 [vRealize Automation에서 VMware Cloud on AWS 배포에 대해 격리된 네트워크 정의](#)  
다음 절차 중 하나를 사용하여 VMware Cloud on AWS 배포에 대한 네트워크 격리를 구성할 수 있습니다.
- 2 [vRealize Automation에서 VMware Cloud on AWS의 네트워크 격리를 지원하도록 클라우드 템플릿의 네트워크 구성 요소 정의](#)  
이 단계에서는 네트워크 시스템 구성 요소를 vRealize Automation 클라우드 템플릿 캔버스로 끌어와서, 격리된 네트워크 배포에 대한 설정을 대상 VMware Cloud on AWS 환경에 추가합니다.

## vRealize Automation에서 VMware Cloud on AWS 배포에 대해 격리된 네트워크 정의

다음 절차 중 하나를 사용하여 VMware Cloud on AWS 배포에 대한 네트워크 격리를 구성할 수 있습니다.

- [vRealize Automation에서 주문형 네트워크 기반 격리 구성](#)
- [vRealize Automation에서 주문형 보안 그룹 기반 격리 구성](#)

## vRealize Automation에서 주문형 네트워크 기반 격리 구성

네트워크 프로파일에 주문형 네트워크 설정을 지정하고 사용하여 VMware Cloud on AWS 배포 요구 사항에 맞게 네트워크 격리를 구성할 수 있습니다.

보안 그룹을 사용하거나 주문형 네트워크 설정을 사용하여 격리된 네트워크를 지정할 수 있습니다. 이 예에서는 네트워크 프로파일에 주문형 네트워크 설정을 지정하여 네트워크 격리를 구성합니다. 나중에, 클라우드 템플릿의 네트워크에 액세스하고 VMware Cloud on AWS 배포에 클라우드 템플릿을 사용합니다.

다리 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

## 사전 요구 사항

- [vRealize Automation에서 기본 VMware Cloud on AWS 워크플로 구성](#) 워크플로를 완료합니다.
- [vRealize Automation의 VMware Cloud on AWS 워크플로에서 격리된 네트워크 구성](#) 항목을 검토합니다.
- 이 절차에서는 사용자에게 vCenter의 대상 SDDC에 대한 VMware Cloud on AWS CloudAdmin 자격 증명을 포함하여 필요한 관리자 자격 증명 이 있다고 가정합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조](#)하십시오.
- 이 절차에서는 사용자에게 클라우드 관리자 사용자 역할이 있다고 가정합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.

## 절차

- 1 기본 VMware Cloud on AWS 워크플로에 사용한 네트워크 프로파일(예: vmc-network1)을 엽니다. [vRealize Automation](#)에서 [VMware Cloud on AWS](#) 배포를 위한 네트워크 및 스토리지 프로파일 구성의 내용을 참조하십시오.
- 2 **네트워크** 탭에서 아무것도 선택할 필요가 없습니다.
- 3 **네트워크 정책** 탭을 클릭합니다.
- 4 **주문형 네트워크 생성** 옵션 선택하고 기본 cgw 네트워크 도메인을 선택합니다. 적절한 CIDR 및 서브넷 크기를 지정합니다.
- 5 **저장**을 클릭합니다.

이 네트워크 프로파일을 사용하면 시스템이 기본 네트워크 도메인의 네트워크에 배포됩니다. 네트워크는 전용 또는 아웃바운드 네트워크 액세스를 사용하여 다른 네트워크와 격리됩니다.

## 다음에 수행할 작업

클라우드 템플릿에 네트워크 구성 요소를 구성합니다. [vRealize Automation](#)에서 [VMware Cloud on AWS](#)의 네트워크 격리를 지원하도록 클라우드 템플릿의 네트워크 구성 요소 정의 항목을 참조하십시오.

## vRealize Automation에서 주문형 보안 그룹 기반 격리 구성

네트워크 프로파일에 주문형 보안 그룹을 지정하고 사용하여 VMware Cloud on AWS 배포 요구 사항에 맞게 네트워크 격리를 구성할 수 있습니다.

보안 그룹을 사용하거나 주문형 네트워크 설정을 사용하여 격리된 네트워크를 지정할 수 있습니다. 이 예에서는 네트워크 프로파일에 주문형 보안 그룹을 지정하여 네트워크 격리를 구성합니다. 나중에, 클라우드 템플릿에 네트워크를 지정하고 VMware Cloud on AWS 배포에 클라우드 템플릿을 사용합니다.

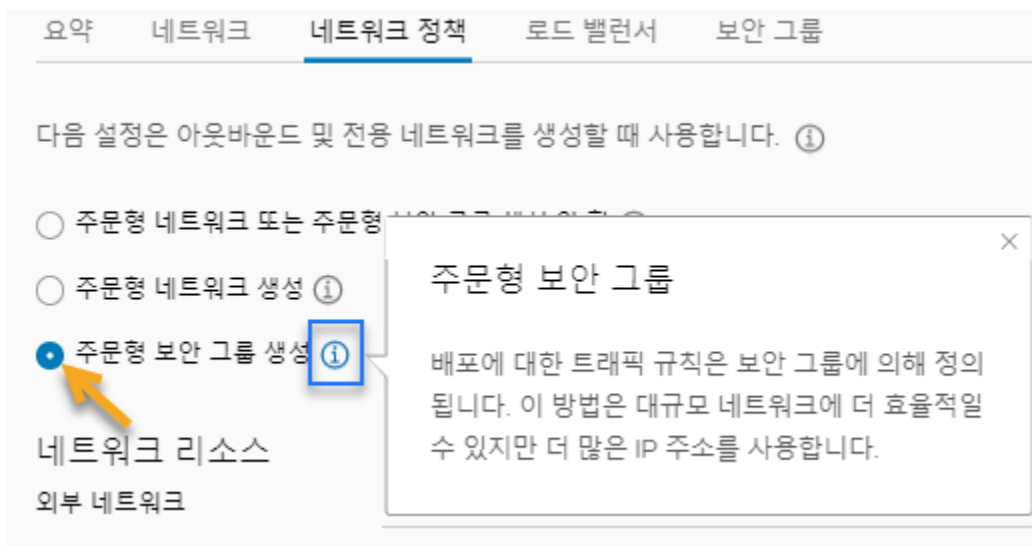
달리 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

## 사전 요구 사항

- [vRealize Automation](#)에서 기본 [VMware Cloud on AWS](#) 워크플로 구성 워크플로를 완료합니다.
- [vRealize Automation](#)의 [VMware Cloud on AWS](#) 워크플로에서 격리된 네트워크 구성 항목을 검토합니다.
- 이 절차에서는 사용자에게 vCenter의 대상 SDDC에 대한 VMware Cloud on AWS CloudAdmin 자격 증명을 포함하여 필요한 관리자 자격 증명이 있다고 가정합니다. [vRealize Automation](#)에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.
- 이 절차에서는 사용자에게 클라우드 관리자 사용자 역할이 있다고 가정합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.

## 절차

- 1 기본 VMware Cloud on AWS 워크플로에 사용한 네트워크 프로파일(예: vmc-network1)을 엽니다. [vRealize Automation](#)에서 [VMware Cloud on AWS](#) 배포를 위한 네트워크 및 스토리지 프로파일 구성의 내용을 참조하십시오.
- 2 기본 VMware Cloud on AWS 워크플로에 사용한 기존 네트워크를 선택합니다(예: sddc-cgw-network-1). [vRealize Automation](#)에서 [VMware Cloud on AWS](#) 배포를 위한 네트워크 및 스토리지 프로파일 구성의 내용을 참조하십시오.
- 3 **네트워크 정책** 탭을 클릭합니다.
- 4 **주문형 보안 그룹 생성** 옵션을 선택합니다.



- 5 **저장**을 클릭합니다.

이 네트워크 프로파일을 사용하면 시스템이 선택한 네트워크에 배포되고 새 보안 그룹 정책에 의해 격리됩니다. 새 보안 정책에는 전용 또는 아웃바운드 네트워크 액세스가 허용됩니다.

## 다음에 수행할 작업

클라우드 템플릿에 네트워크 구성 요소를 구성합니다. [vRealize Automation](#)에서 [VMware Cloud on AWS](#)의 네트워크 격리를 지원하도록 클라우드 템플릿의 네트워크 구성 요소 정의 항목을 참조하십시오.

## vRealize Automation에서 VMware Cloud on AWS의 네트워크 격리를 지원하도록 클라우드 템플릿의 네트워크 구성 요소 정의

이 단계에서는 네트워크 시스템 구성 요소를 vRealize Automation 클라우드 템플릿 캔버스로 끌어와서, 격리된 네트워크 배포에 대한 설정을 대상 VMware Cloud on AWS 환경에 추가합니다.

이전에 생성한 클라우드 템플릿에 네트워크 격리를 추가합니다. 클라우드 템플릿은 VMware Cloud on AWS 환경에 대한 배포를 지원하는 프로젝트 및 클라우드 영역과 이미 연결되어 있으며 격리를 위해 구성된 네트워크 및 네트워크 프로파일과도 연결되어 있습니다.

다들 명시되지 않는 한, 이 절차에서 입력하는 단계 값은 예시 워크플로에만 해당됩니다.

## 사전 요구 사항

- vRealize Automation에서 주문형 보안 그룹 기반 격리 구성 또는 vRealize Automation에서 주문형 네트워크 기반 격리 구성 절차를 완료합니다.
- 이 절차에서는 사용자에게 클라우드 템플릿 설계자 자격 증명이 있다고 가정합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 이 절차에서는 vCenter의 대상 SDDC에 대해 VMware Cloud on AWS CloudAdmin 자격 증명이 있다고 가정합니다. vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.

## 절차

- 1 이전 워크플로에서 생성한 클라우드 템플릿을 엽니다. vRealize Automation에서 VMware Cloud on AWS 배포를 지원하도록 클라우드 템플릿 설계에 vCenter 시스템 리소스 정의의 내용을 참조하십시오.
- 2 클라우드 템플릿 설계 페이지의 왼쪽에 있는 구성 요소에서 네트워크 구성 요소를 캔버스 위로 끌어 놓습니다.
- 3 네트워크 구성 요소 YAML 코드를 편집하여 네트워크 유형을 `private` 또는 `outbound`(굵은 글꼴로 표시)로 지정합니다.

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

또는

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

## 다음에 수행할 작업

클라우드 템플릿을 배포하거나 단을 준비가 되었습니다.

# Tutorial: vRealize Automation에 대한 제공자별 외부 IPAM 통합 구성

외부 IPAM 제공자를 사용하여 클라우드 템플릿 배포에 대한 IP 주소 할당을 관리할 수 있습니다. 이 자습서에서는 Infoblox를 외부 IPAM 제공자로 사용하여 vRealize Automation에서 외부 IPAM 통합을 구성하는 방법을 설명합니다.

이 절차에서는 기존 IPAM 제공자 패키지(이 경우 Infoblox 패키지)와 기존 실행 환경을 사용하여 제공자별 IPAM 통합 지점을 만듭니다. 외부 IPAM 제공자의 IP 주소 할당을 지원하도록 기존 네트워크를 구성하고 네트워크 프로파일을 생성합니다. 마지막으로, 네트워크 및 네트워크 프로파일과 일치하는 클라우드 템플릿을 생성하고 외부 IPAM 제공자로부터 가져온 IP 값을 사용하여 네트워크 시스템을 배포합니다.

IPAM 제공자 패키지를 확보하고 구성하는 방법과 IPAM 제공자 통합을 지원하기 위해 클라우드 확장성 프록시에 액세스할 수 있는 실행 환경을 구성하는 방법에 대한 정보는 참조로 포함됩니다.

보이는 값은 예시 값입니다. 사용자 환경에서 이러한 값을 그대로 사용해서는 안 됩니다. 조직의 요구 사항에 맞게 고유한 값으로 대체하거나 예시의 값에서 추론할 수 있는 부분에 대해 생각해 보십시오.



Infoblox IPAM 통합 워크플로를 보여 주는 비디오 형식의 유사한 vRealize Automation 시나리오를 참조하려면 [vRealize Automation 8.1/vRealize Automation Cloud와 Infoblox IPAM 플러그인 1.1 통합](#)을 참조하십시오.

## 절차

- 1 **vRealize Automation와 통합하기 위해 Infoblox 애플리케이션에서 필요한 확장 가능 특성 추가**  
Infoblox 웹 사이트 또는 VMware Marketplace에서 vRealize Automation와 통합하기 위해 Infoblox 제공자 패키지(infoblox.zip)를 다운로드하고 배포하려면, 그 전에 Infoblox에서 필요한 확장성 특성을 추가해야 합니다.
- 2 **vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포**  
vRealize Automation에서 외부 IPAM 통합 지점을 정의하려면, 먼저 구성된 IPAM 제공자 패키지가 필요합니다.
- 3 **vRealize Automation에서 IPAM 통합 지점에 대한 실행 환경 생성**  
vRealize Automation에서 외부 IPAM 통합 지점을 정의하려면 먼저 IPAM 제공자와 vRealize Automation 간에 중개자 역할을 하는 기존 실행 환경에 액세스하거나 새로 생성해야 합니다. 실행 환경은 일반적으로 Amazon Web Services 또는 Microsoft Azure 클라우드 계정이거나 클라우드 확장성 프록시에 연결된 온-프레미스 작업 기반 확장성 통합 지점입니다.
- 4 **vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가**  
vRealize Automation은 외부 IPAM 제공자와의 통합을 지원합니다. 이 예에서는 Infoblox를 외부 IPAM 제공자로 사용합니다.
- 5 **vRealize Automation에서 기존 네트워크에 대해 외부 IPAM을 사용하도록 네트워크 및 네트워크 프로파일 구성**  
내부 vRealize Automation가 아닌, 외부 IPAM 제공자가 관리하고 외부 IPAM 제공자로부터 가져오는 IP 주소 값을 사용하도록 기존 네트워크를 정의할 수 있습니다.
- 6 **vRealize Automation에서 외부 IPAM 제공자 범위 할당을 사용하는 클라우드 템플릿 정의 및 배포**  
외부 IPAM 제공자로부터 IP 주소 할당을 가져오고 관리하는 클라우드 템플릿을 정의할 수 있습니다. 이 예에서는 Infoblox를 외부 IPAM 제공자로 사용합니다.

## 7 vRealize Automation 클라우드 템플릿에서 IPAM 통합을 위해 Infoblox 특정 속성 및 확장 가능한 특성 사용

Infoblox용 외부 IPAM 통합을 포함하는 vRealize Automation 프로젝트에 대해 Infoblox 특정 속성을 사용할 수 있습니다.

## 8 vRealize Automation에서 Infoblox 필터를 사용하여 네트워크 데이터 수집을 제어

Infoblox의 경우 데이터 수집 네트워크 수를 vRealize Automation 작업에 필요한 네트워크로만 제한할 수 있습니다. 이렇게 하면 전송되는 데이터의 양이 줄어들고 시스템 성능이 향상됩니다.

# vRealize Automation과 통합하기 위해 Infoblox 애플리케이션에서 필요한 확장 가능 특성 추가

Infoblox 웹 사이트 또는 VMware Marketplace에서 vRealize Automation과 통합하기 위해 Infoblox 제 공자 패키지(infoblox.zip)를 다운로드하고 배포하려면, 그 전에 Infoblox에서 필요한 확장성 특성을 추가해야 합니다.

이 절차는 vRealize Automation Cloud Assembly와 Infoblox 통합을 위해 외부 IPAM 통합 지점을 생성하는 경우에 적용됩니다.

infoblox.zip 다운로드를 사용하려면, 먼저 조직 계정 관리자 자격 증명을 사용하여 Infoblox 계정에 로그인하고 다음과 같은 Infoblox 확장 가능 특성을 미리 생성해야 합니다.

- VMware NIC index
- VMware resource ID

### 사전 요구 사항

- Infoblox 계정이 있고 조직의 Infoblox 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다.
- Infoblox WAPI 버전이 지원되는지 확인합니다. Infoblox와 IPAM 통합은 Infoblox WAPI 버전 v2.7에 따라 다릅니다. WAPI v2.7을 지원하는 모든 Infoblox 장치가 지원됩니다.
- vRealize Automation 클라우드 템플릿에서 IPAM 통합을 위해 Infoblox 특정 속성 및 확장 가능한 특성 사용 항목을 검토합니다.

### 절차

- 1 관리자 자격 증명을 사용하여 Infoblox 계정에 로그인합니다.

이것은 **인프라 > 연결 > 통합 >** 메뉴 시퀀스를 사용하여 vRealize Automation Cloud Assembly에서 외부 IPAM 통합 지점을 생성할 때 지정하는 것과 동일한 관리자 이름과 암호 자격 증명입니다.

- 2 Infoblox 설명서에 설명된 절차를 사용하여 Infoblox 애플리케이션에서 다음 필수 확장 가능 특성을 생성합니다.

- VMware NIC index - 유형 정수
- VMware resource ID - 유형 문자열

이 절차는 Infoblox 설명서 항목 [About Extensible Attributes](#)의 "Adding Extensible Attributes" 섹션에 설명되어 있습니다. [Managing Extensible Attributes](#) 항목도 참조하십시오.

#### 다음에 수행할 작업

필요한 특성을 추가하면 vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포에 설명된 대로 Infoblox 패키지를 다운로드하고 배포하는 프로세스를 재개할 수 있습니다.

## vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포

vRealize Automation에서 외부 IPAM 통합 지점을 정의하려면, 먼저 구성된 IPAM 제공자 패키지가 필요합니다.

제공자별 통합 패키지는 IPAM 제공자의 웹 사이트, [VMware Solution Exchange](#) 마켓플레이스 또는 vRealize Automation [마켓플레이스](#) 탭에서 다운로드할 수 있습니다.

---

**참고** 이 예에서는 VMware에서 제공하는 Infoblox 패키지인 Infoblox.zip을 사용합니다. 이 패키지는 [VMware Marketplace](#)에서 다음과 같은 버전으로 다운로드할 수 있습니다.

- **Infoblox 플러그인 버전 1.4** - vRealize Automation 8.3.x와 호환되며 이전 버전의 모든 기능을 제공합니다. 이 버전에서는 두 개의 NIC에 대해 DNS 접미사가 다른 동일한 호스트 이름을 사용할 수 있습니다. 자세한 내용은 플러그인 릴리스 정보를 참조하십시오.
- **Infoblox 플러그인 버전 1.3** - vRealize Automation 8.3.x와 호환되며 추가 네트워크 데이터 수집 필터를 제공합니다. vRealize Automation에서 Infoblox 필터를 사용하여 네트워크 데이터 수집을 제어의 내용을 참조하십시오.

Infoblox v1.3 플러그인은 vRealize Automation 8.1 또는 8.2와 함께 사용할 수 있지만, KB 문서 [vRealize Automation 8.x와 Infoblox 1.3 호환성\(82142\)](#)에 설명된 대로 일부 상황에서만 주의하여 사용할 수 있습니다.

- **vRA Cloud Infoblox 플러그인 버전 1.2** - vRealize Automation 8.1.x 및 8.2.x와 호환
- **vRA Cloud Infoblox 플러그인 버전 1.1** - vRealize Automation 8.1.x와 호환
- **vRA Cloud Infoblox 플러그인 버전 1.0** - vRealize Automation 8.0.1.x와 호환(글로벌 네트워크에 대한 인터넷 연결 사용 또는 사용 안 함).
- **vRA Cloud Infoblox 플러그인 버전 0.4** - vRealize Automation 8.0.0.x 및 8.0.1.x와 호환(글로벌 네트워크에 대한 인터넷 연결이 있는 경우).

Infoblox와 IPAM 통합은 Infoblox WAPI 버전 v2.7에 따라 다릅니다. WAPI v2.7을 지원하는 모든 Infoblox 장치가 지원됩니다.

---

다른 IPAM 제공자를 위한 IPAM 통합 패키지를(마켓플레이스에 아직 없는 경우) 만드는 방법에 대한 자세한 내용은 [IPAM SDK를 사용하여 vRealize Automation에 대한 제공자별 외부 IPAM 통합 패키지를 생성하는 방법](#) 항목을 참조하십시오.

IPAM 제공자 패키지에는 메타데이터 및 기타 구성과 함께 패키지로 구성된 스크립트가 포함되어 있습니다. 스크립트에는 vRealize Automation가 외부 IPAM 제공자와 함께 수행하는 작업에 사용되는 소스 코드가 포함되어 있습니다. 예제 작업에는 Allocate an IP address for a virtual machine, Fetch a list of IP ranges from the provider 및 Update the MAC address of a host record in the provider가 포함됩니다.

### 사전 요구 사항

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 외부 IPAM 제공자(예: [Infoblox](#) 또는 [Bluecat](#))의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다.
- Infoblox를 외부 IPAM 제공자로 사용하는 경우, 계속하기 전에 Infoblox 계정에 필요한 확장 가능 특성을 추가했는지 확인합니다. [vRealize Automation와 통합하기 위해 Infoblox 애플리케이션에서 필요한 확장 가능 특성 추가](#) 항목을 참조하십시오.

---

**참고** Infoblox 플러그인의 Python 요소가 SSL 핸드셰이크를 처리하는 방식에서 파생된 인증서 체인 문제가 존재합니다. 문제 및 해당하는 필수 작업에 대한 자세한 내용은 기술 자료 문서 [인증 프로세스 중 vRA Cloud Infoblox 플러그인에서 인증서 체인 오류 발생\(88057\)](#)을 참조하십시오.

---

### 절차

- 1 올바른 Infoblox 플러그인 페이지로 이동합니다(예: [Infoblox 플러그인 버전 1.4](#)).
- [VMware Marketplace](#)에서 사용할 수 있는 Infoblox 플러그인 옵션은 위 항목을 참조하십시오.
- 2 로그인하고 플러그인 패키지를 다운로드합니다.
- 3 Infoblox에 필요한 확장 가능 특성을 추가하지 않았다면 지금 추가합니다. [vRealize Automation와 통합하기 위해 Infoblox 애플리케이션에서 필요한 확장 가능 특성 추가](#)의 내용을 참조하십시오.

### 결과

이제 vRealize Automation에서 [Infoblox에 대한 외부 IPAM 통합 추가](#)에 설명된 대로 **통합 > 통합 추가 > IPAM > 제공자 관리 > 패키지 가져오기** 메뉴 시퀀스를 사용하여 배포를 위한 패키지를 사용할 수 있습니다.

## vRealize Automation에서 IPAM 통합 지점에 대한 실행 환경 생성

vRealize Automation에서 외부 IPAM 통합 지점을 정의하려면 먼저 IPAM 제공자와 vRealize Automation 간에 중개자 역할을 하는 기존 실행 환경에 액세스하거나 새로 생성해야 합니다. 실행 환경은 일반적으로 Amazon Web Services 또는 Microsoft Azure 클라우드 계정이거나 클라우드 확장성 프록시에 연결된 온-프레미스 작업 기반 확장성 통합 지점입니다.

외부 IPAM 통합에는 실행 환경이 필요합니다. IPAM 통합 지점을 정의할 때 사용 가능한 실행 환경을 지정하여 vRealize Automation Cloud Assembly와 IPAM 제공자 간에 연결을 생성합니다.

IPAM 통합은 Amazon Web Services Lambda, Microsoft Azure Functions와 같은 FaaS(Function-as-a-Service) 제공자 또는 ABX(Action-Based Extensibility) 온-프레미스 내장형 통합 지점이 지원하는 실행 환경에서 다운로드한 제공자별 스크립트 또는 플러그인 세트를 사용합니다. 실행 환경은 외부 IPAM 제공자(예: Infoblox)에 연결하는 데 사용됩니다.

---

**참고** Infoblox IPAM 통합 지점에는 ABX(Action-Based Extensibility) 온-프레미스 내장형 통합 지점이 필요합니다.

---

런타임 환경 유형마다 장점과 단점이 있습니다.

- **ABX(Action-Based Extensibility) 통합 지점**
  - 무료이며, 추가 벤더 사용 비용 없음
  - 공개적으로 액세스할 수 없는 NAT/방화벽 뒤에 있는 온-프레미스 데이터 센터에 상주하는 IPAM 벤더 장치에 연결 가능(예: Infoblox)
  - 상용 클라우드 벤더보다 느리고 성능 안정성이 약간 낮음
- **Amazon Web Services**
  - 관련 벤더 FaaS 연결/사용 비용이 있음
  - 공개적으로 액세스할 수 없는 NAT/방화벽 뒤에 있는 온-프레미스 데이터 센터에 상주하는 IPAM 벤더 장치에 연결 불가능
  - 빠르고 매우 안정적인 성능
- **Microsoft Azure**
  - 관련 벤더 FaaS 연결/사용 비용이 있음
  - 공개적으로 액세스할 수 없는 NAT/방화벽 뒤에 있는 온-프레미스 데이터 센터에 상주하는 IPAM 벤더 장치에 연결 불가능
  - 빠르고 매우 안정적인 성능

#### 사전 요구 사항

- 클라우드 관리자 자격 증명이 있는지 확인합니다. vRealize Automation에서 클라우드 계정 작업에 필요한 **자격 증명** 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. vRealize Automation **사용자 역할이란?** 항목을 참조하십시오.
- 외부 IPAM 제공자(예: Infoblox 또는 Bluecat)의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다.

- IPAM 제공자(예: Infoblox 또는 BlueCat)를 위해 배포된 통합 패키지에 액세스할 수 있는지 확인합니다. 배포된 패키지는 처음에 IPAM 제공자 웹 사이트 또는 vRealize Automation Cloud Assembly 마켓플레이스에서 .zip 다운로드로 가져온 후 vRealize Automation Cloud Assembly에서 배포됩니다. 제공자 패키지 .zip 파일을 배포하고 IPAM 통합 페이지에서 **제공자** 값으로 사용할 수 있도록 설정하는 방법에 대한 자세한 내용은 [vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포 항목](#)을 참조하십시오.

## 절차

- 1 IPAM 통합 실행 환경으로 사용할 온-프레미스 FaaS 기반 확장성 작업을 생성하려면 **확장성 > 라이브러리 > 작업**을 선택합니다.
- 2 새 작업을 클릭하고 작업 이름과 설명을 입력한 후 프로젝트를 지정합니다.
- 3 FaaS 제공자 드롭다운 메뉴에서 **온-프레미스**를 선택합니다.
- 4 양식을 작성하여 확장성 작업을 정의합니다.



실행 환경에 대한 관련 정보는 이 [Infoblox IPAM 플러그인 1.1 통합 블로그 비디오](#)(약 24분 후 표시됨)를 참조하십시오.

## vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가

vRealize Automation은 외부 IPAM 제공자와의 통합을 지원합니다. 이 예에서는 Infoblox를 외부 IPAM 제공자로 사용합니다.

제공자별 IPAM 통합 지점을 사용하여 클라우드 템플릿 배포를 위한 IP 주소 및 관련 네트워크 특성을 가져오고 관리할 수 있습니다.

이 예시에서는 외부 IPAM 제공자를 사용하여 조직의 계정에 대한 액세스를 지원하기 위해 외부 IPAM 통합 지점을 생성합니다. 이 예시 워크플로에서 IPAM 제공자는 Infoblox이고 제공자별 통합 패키지가 이미 있습니다. 이러한 지점은 Infoblox 통합에만 해당되지만 다른 외부 IPAM 제공자에 대한 IPAM 통합을 생성하는 경우 참조로 사용할 수 있습니다.

제공자별 통합 패키지는 IPAM 제공자의 웹 사이트, [VMware Solution Exchange](#) 마켓플레이스 또는 vRealize Automation Cloud Assembly **마켓플레이스** 탭에서 구할 수 있습니다.

이 예에서는 VMware에서 제공하는 Infoblox 패키지인 Infoblox.zip을 사용합니다. 이 패키지는 VMware Solution Exchange 마켓플레이스에서 다음과 같은 버전으로 다운로드할 수 있습니다.

- [vRA Cloud Infoblox 플러그인 버전 1.1](#) - vRealize Automation 8.1 이상 지원
- [vRA Cloud Infoblox 플러그인 버전 1.0](#) - vRealize Automation 8.0.1 지원
- [vRA Cloud Infoblox 플러그인 버전 0.1](#) - vRealize Automation 8.0 지원

## 사전 요구 사항

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목](#)을 참조하십시오.

- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 외부 IPAM 제공자의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다.
- IPAM 제공자를 위해 배포된 통합 패키지에 액세스할 수 있는지 확인합니다. 배포된 패키지는 처음에 IPAM 제공자 웹 사이트 또는 VMware Solution Exchange 마켓플레이스에서 .zip 다운로드로 가져온 후 vRealize Automation에 배포됩니다.

제공자 패키지 .zip 파일을 다운로드 및 배포하고 IPAM 통합 페이지에서 **제공자** 값으로 사용할 수 있도록 설정하는 방법에 대한 자세한 내용은 [vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포](#) 항목을 참조하십시오.

- IPAM 제공자에 대해 구성된 실행 환경에 액세스할 수 있는지 확인합니다. 실행 환경은 일반적으로 ABX(작업 기반 확장성) 온-프레미스 내장형 통합 지점입니다.

실행 환경 특성에 대한 자세한 내용은 [vRealize Automation에서 IPAM 통합 지점에 대한 실행 환경 생성](#) 항목을 참조하십시오.

- Infoblox 애플리케이션에서 필요한 확장 가능 특성을 사용하도록 설정합니다. [vRealize Automation](#) 와 통합하기 위해 [Infoblox 애플리케이션에서 필요한 확장 가능 특성](#) 추가의 내용을 참조하십시오.
- 외부 인터넷에 액세스할 수 없는 경우 인터넷 서버 프록시를 구성할 수 있습니다. [vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법](#)의 내용을 참조하십시오.
- Infoblox IPAM 제품에 액세스하고 사용하는 데 필요한 사용자 자격 증명이 있는지 확인합니다. 예를 들어 Infoblox 장치에서 [관리] 탭을 열고 관리자, 그룹 및 역할 항목을 사용자 지정합니다. 관리자 또는 슈퍼유저 권한이 있는 그룹 또는 DHCP, DNS, IPAM 및 그리드 사용 권한이 있는 사용자 지정 그룹의 멤버여야 합니다. 이러한 설정을 통해 Infoblox 플러그인에서 사용할 수 있는 모든 기능에 액세스가 가능하므로, Infoblox IPAM 통합을 생성하면 이 IPAM 통합을 디자이너가 클라우드 템플릿 및 배포에서 사용할 수 있습니다. 사용자 권한에 대한 자세한 내용은 Infoblox 제품 설명서를 참조하십시오.

## 절차

1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.

2 **IPAM**을 클릭합니다.

3 **제공자** 드롭다운 목록에서 구성된 IPAM 제공자 패키지(예: *Infoblox\_hrg*)를 선택합니다.

목록이 비어 있으면 **제공자 패키지 가져오기**를 클릭하고 기존 제공자 패키지 .zip 파일로 이동한 후 선택합니다. 제공자 .zip 파일이 없으면 IPAM 제공자의 웹 사이트나 vRealize Automation Cloud Assembly 마켓플레이스 탭에서 가져올 수 있습니다.

vCenter에서 제공자 패키지 .zip 파일을 배포하고 [통합] 페이지에서 **제공자** 값으로 사용할 수 있도록 설정하는 방법에 대한 자세한 내용은 [vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포](#) 항목을 참조하십시오.

최신 버전의 벤더 IPAM 통합 패키지를 사용하도록 기존 IPAM 통합을 업그레이드하는 방법에 대한 자세한 내용은 [vRealize Automation](#)에서 최신 외부 IPAM 통합 패키지로 업그레이드하는 방법에서 참조하십시오.

- 4 제공자의 호스트 이름과 같은 다른 필수 필드(있는 경우)와 함께 외부 IPAM 제공자의 계정에 대한 관리자 사용자 이름 및 암호 자격 증명을 입력합니다.

이 예시에서는 다음 단계를 사용하여 Infoblox IPAM 제공자의 호스트 이름을 가져옵니다.

- a 별도의 브라우저 탭에서 Infoblox 관리자 자격 증명을 사용하여 IPAM 제공자 계정에 로그인합니다.
- b 호스트 이름 URL을 복사합니다.
- c IPAM 통합 페이지의 **호스트 이름** 필드에 호스트 이름 URL을 붙여넣습니다.

- 5 **실행 환경** 드롭다운 목록에서 기존 온-프레미스 작업 기반 확장성 통합 지점(예: *Infoblox\_abx\_intg*)을 선택합니다.

실행 환경은 vRealize Automation와 외부 IPAM 제공자 간의 통신을 지원합니다.

---

**참고** Amazon Web Services 또는 Microsoft Azure 클라우드 계정을 통합 실행 환경으로 사용하는 경우에는 IPAM 제공자 장치를 인터넷에서 액세스할 수 있고, NAT 또는 방화벽이 뒤에 있지 않으며, 공개적으로 확인할 수 있는 DNS 이름이 있는지 확인하십시오. IPAM 제공자에 액세스할 수 없는 경우 Amazon Web Services Lambda 또는 Microsoft Azure 함수가 연결할 수 없으며 통합이 실패합니다. 관련 정보는 [vRealize Automation](#)에서 IPAM 통합 지점에 대한 실행 환경 생성 항목을 참조하십시오.

---

IPAM 프레임워크는 ABX(Action-Based Extensibility) 온-프레미스 내장형 실행 환경만 지원합니다.

---

**참고** Infoblox IPAM 통합 지점에는 ABX(Action-Based Extensibility) 온-프레미스 내장형 통합 지점이 필요합니다.

---

구성된 클라우드 계정 또는 통합 지점을 사용하면 연결된 클라우드 확장성 프로시를 통해 vRealize Automation와 IPAM 제공자(이 예에서는 Infoblox) 간의 통신이 가능합니다. 이미 생성된 제공자를 선택하거나 새로 생성할 수 있습니다.

실행 환경을 생성하는 방법에 대한 자세한 내용은 [vRealize Automation](#)에서 IPAM 통합 지점에 대한 실행 환경 생성 항목을 참조하십시오.

- 6 **검증**을 클릭합니다.

이 예에서는 실행 환경에 대해 온-프레미스 작업 기반 확장성 통합을 사용하기 때문에 유효성 검사 작업을 볼 수 있습니다.

- a **확장성** 탭을 클릭합니다.
- b **작업 > 작업 실행**을 클릭하고 필터에서 **모든 실행** 또는 **통합 실행**을 선택하여 끝점 검증 작업이 시작되어 실행 중임을 확인합니다.

- 7 IPAM 제공자의 자체 서명된 인증서를 신뢰하라는 메시지가 표시되면 **수락**을 클릭합니다.

자체 서명된 인증서를 수락하면 검증 작업을 계속하여 완료할 수 있습니다.

- 8 이 IPAM 통합 지점의 **이름**(예: *Infoblox\_Integration*)과 **설명**(예: *팀 HRG에 ABX 통합을 사용하는 Infoblox IPAM*)을 입력합니다.

- 9 **추가**를 클릭하여 새로운 외부 IPAM 통합 지점을 저장합니다.

데이터 수집 작업은 모방됩니다. 네트워크 및 IP 범위는 IPAM 제공자가 수집한 데이터입니다. 다음과 같이 데이터 수집 작업을 볼 수 있습니다.

- a **확장성** 탭을 클릭합니다.

- b **작업 > 작업 실행**을 클릭하고 데이터 수집 작업이 시작되어 실행되고 있는지 확인합니다. 작업 실행 콘텐츠를 열어서 볼 수 있습니다.

## 결과

이제 네트워크 및 네트워크 프로파일에 제공자별 외부 IPAM 통합을 사용할 수 있습니다.

## vRealize Automation에서 기존 네트워크에 대해 외부 IPAM을 사용하도록 네트워크 및 네트워크 프로파일 구성

내부 vRealize Automation가 아닌, 외부 IPAM 제공자가 관리하고 외부 IPAM 제공자로부터 가져오는 IP 주소 값을 사용하도록 기존 네트워크를 정의할 수 있습니다.

조직의 외부 IPAM 제공자 계정에 정의한 기존 IP 설정에 액세스하도록 네트워크를 정의할 수 있습니다. 이 단계는 이전 단계에서 생성한 Infoblox 제공자 통합을 확장합니다.

이 예시에서는 vCenter에서 데이터를 수집한 기존 네트워크로 네트워크 프로파일을 구성합니다. 그런 다음 외부 IPAM 제공자(이 경우 Infoblox)에서 IP 정보를 가져오도록 해당 네트워크를 구성합니다. 이 네트워크 프로파일과 일치할 수 있는 vRealize Automation에서 프로비저닝하는 가상 시스템은 외부 IPAM 제공자로부터 IP 및 기타 TCP/IP 관련 설정을 가져옵니다.

네트워크에 대한 자세한 내용은 [vRealize Automation의 네트워크 리소스](#) 항목을 참조하십시오. 네트워크 프로파일에 대한 자세한 내용은 [vRealize Automation에서 네트워크 프로파일을 추가하는 방법](#) 및 [vRealize Automation의 네트워크 프로파일에 대해 알아보기](#) 항목을 참조하십시오.

관련 정보는 [vRealize Automation에서 외부 IPAM 통합에 대해 주문형 네트워크를 지원하도록 네트워크 프로파일을 구성하는 방법](#) 항목을 참조하십시오.

## 사전 요구 사항

다음 일련의 단계는 IPAM 제공자 통합 워크플로의 컨텍스트에서 표시됩니다. [Tutorial: vRealize Automation에 대한 제공자별 외부 IPAM 통합 구성](#)의 내용을 참조하십시오.

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.

- 외부 IPAM 제공자(예: Infoblox 또는 Bluecat)의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다. 이 예시 워크플로에서 IPAM 제공자는 Infoblox입니다.
- IPAM 제공자에 대한 IPAM 통합 지점이 있는지 확인합니다. vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가의 내용을 참조하십시오.

## 절차

- 1 네트워크를 구성하려면 **인프라 > 리소스 > 네트워크**를 클릭합니다.
- 2 **네트워크** 탭에서 IPAM 제공자 통합 지점에 사용할 기존 네트워크를 선택합니다. 이 예시에서 네트워크 이름은 *net.23.117-only-IPAM*입니다.  
나열된 네트워크는 조직에 있는 vCenter에서 vRealize Automation가 데이터를 수집했습니다.
- 3 외부 IPAM 제공자로부터 값을 얻으려면 **계정/지역, 이름 및 네트워크 도메인**을 제외하고, 다음을 비롯한 다른 모든 네트워크 설정이 비어 있는지 확인합니다.
  - 도메인(8단계의 참고 참조)
  - CIDR
  - 기본 게이트웨이
  - DNS 서버
  - DNS 검색 도메인
- 4 **IP 범위** 탭을 클릭하고 **IPAM IP 범위 추가**를 클릭합니다.
- 5 **네트워크** 메뉴에서 방금 구성한 네트워크(예: *net.23.117-only-IPAM*)를 선택합니다.
- 6 **제공자** 메뉴에서 이전에 워크플로에서 생성한 *Infoblox\_Integration* IPAM 통합 지점을 선택합니다.
- 7 지금 보이는 **주소 공간** 드롭다운 메뉴에서 나열된 네트워크 보기 중 하나를 선택합니다.

Infoblox의 주소 공간을 네트워크 보기라고 합니다.

네트워크 보기는 IPAM 제공자 계정에서 가져옵니다. 이 예시에서는 방금 구성한 네트워크 서브넷(예: *net.23.117-only-IPAM*), 이전에 워크플로에서 생성한 *Infoblox\_Integration* 통합 지점 및 이름이 *default*인 주소 공간을 사용합니다.

나열된 주소 공간 값은 외부 IPAM 제공자로부터 가져옵니다.

- 8 선택한 주소 공간에 사용할 수 있는 표시된 네트워크 목록에서 하나 이상의 네트워크(예: 10.23.117.0/24)를 선택합니다.

이 예시에서는 선택한 네트워크에 대한 **도메인** 및 **DNS 서버** 열 값에 Infoblox의 값이 포함되어 있습니다.

**참고** 3단계에서 vRealize Automation에 도메인이 지정된 네트워크를 선택한 다음, 도메인 값이 포함된 외부 IPAM 제공자 주소 공간에서 네트워크를 선택하면, 외부 IPAM 제공자 네트워크의 도메인 값이 vRealize Automation에 지정된 도메인보다 우선합니다. IPAM IP 범위 설정에 도메인 값이 없으면 (위에서 설명한 대로 Cloud Assembly 또는 외부 IPAM 제공자에 지정됨) 프로비저닝이 실패합니다.

Infoblox의 경우 시스템 수준에서 Blueprint 속성 Infoblox.IPAM.Network.dnsSuffix를 사용하여 도메인 값을 덮어쓸 수 있습니다. 관련 정보는 [vRealize Automation 클라우드 템플릿에서 IPAM 통합을 위해 Infoblox 특정 속성 및 확장 가능한 특성 사용 항목](#)을 참조하십시오.

- 9 **추가**를 클릭하여 네트워크에 대한 IPAM IP 범위를 저장합니다.

IP 범위 테이블에 범위가 표시됩니다.

- 10 **IP 주소** 탭을 클릭합니다.

외부 IPAM 제공자의 새 주소 범위를 사용하여 시스템을 프로비저닝한 후에는 **IP 주소** 테이블에 새 레코드가 표시됩니다.

- 11 네트워크를 사용하도록 네트워크 프로파일을 구성하려면 **인프라 > 구성 > 네트워크 프로파일**을 클릭합니다.
- 12 네트워크 프로파일에 이름(예: *Infoblox-NP*)을 지정하고 다음 샘플 설정을 추가합니다.

■ 요약 탭

- vSphere 클라우드 계정/지역을 지정합니다.
  - 네트워크 프로파일에 대한 기능 태그(예: 이름이 *infoblox\_abx*인)를 추가합니다.
- 기능 태그를 기록해 둡니다. 클라우드 템플릿에서 프로비저닝 연결을 수행하기 위해 기능 태그를 클라우드 템플릿 제약 조건 태그로 사용해야 하기 때문입니다.

■ 네트워크 탭

- 이전에 생성한 네트워크를 추가합니다(예: *net.23.117-only-IPAM*).

- 13 **저장**을 클릭하여 이 설정으로 네트워크 프로파일을 저장합니다.

## 결과

이제 클라우드 템플릿 설계에서 Infoblox IPAM 통합에 사용할 기존 네트워크 유형에 대한 네트워크 및 네트워크 프로파일 설정이 구성되었습니다.

## vRealize Automation에서 외부 IPAM 제공자 범위 할당을 사용하는 클라우드 템플릿 정의 및 배포

외부 IPAM 제공자로부터 IP 주소 할당을 가져오고 관리하는 클라우드 템플릿을 정의할 수 있습니다. 이 예에서는 Infoblox를 외부 IPAM 제공자로 사용합니다.

외부 IPAM 통합 워크플로의 마지막 단계에서는, 이전에 정의한 네트워크 및 네트워크 프로파일을 조직의 Infoblox 계정에 연결하는 클라우드 템플릿을 정의하고 배포하여, vRealize Automation Cloud Assembly가 아닌 외부 IPAM 제공자로부터 배포된 VM에 대한 IP 주소 할당을 확보하고 관리합니다.

이 워크플로에서는 Infoblox를 외부 IPAM 제공자로 사용하며 일부 단계에서 예제 값은 Infoblox에 고유합니다. 단, 이러한 절차는 다른 외부 IPAM 통합에도 적용될 수 있습니다.



VMware vRealize Automation과 Infoblox DDI를 사용하여 VM에 대한 IPAM 및 DNS 자동화 Infoblox 블로그는 관련 정보를 제공합니다.

클라우드 템플릿을 배포하고 VM이 시작되면, 배포에서 각 VM에 사용된 IP 주소는 **리소스 > 네트워크** 페이지에 네트워크 항목으로 나타나며, IPAM 제공자 계정의 IPAM 제공자 네트워크 및 호스트 vCenter에 배포된 각 VM에 대한 vSphere Web Client 레코드에 새 호스트 레코드로 나타납니다.

### 사전 요구 사항

다음 일련의 단계는 외부 IPAM 제공자 통합 워크플로의 컨텍스트에 기반합니다. [Tutorial: vRealize Automation에 대한 제공자별 외부 IPAM 통합 구성](#)의 내용을 참조하십시오.

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 외부 IPAM 제공자(예: Infoblox 또는 BlueCat)의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다.
- 호스트 vCenter에 배포된 VM에 대한 vSphere 웹 클라이언트 레코드에서 상태 레코드를 표시하는 데 필요한 호스트 계정 및 역할 요구 사항에 대한 관리자 액세스 권한이 있는지 확인합니다.
- 외부 IPAM 제공자에 대한 IPAM 통합 지점이 있는지 확인합니다. [vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가](#)의 내용을 참조하십시오.
- 의도한 IPAM 통합 지점에 대해 외부 IPAM 통합을 지원하는 vRealize Automation Cloud Assembly 네트워크 및 네트워크 프로파일을 구성했는지 확인합니다. [vRealize Automation에서 기존 네트워크에 대해 외부 IPAM을 사용하도록 네트워크 및 네트워크 프로파일 구성](#)의 내용을 참조하십시오.
- 프로젝트 및 클라우드 영역이 IPAM 통합 지점 및 네트워크 또는 네트워크 프로파일의 태그와 일치하도록 태그가 지정되어 있는지 확인합니다. 필요한 경우 사용자 지정 리소스 이름 지정을 지원하도록 프로젝트를 구성합니다.

클라우드 템플릿에서 프로젝트 및 클라우드 영역의 역할 및 기타 인프라 요소의 역할에 대한 자세한 내용은 [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트 항목을 참조하십시오](#). 태그 지정에 대한 자세한 내용은 [태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#) 항목을 참조하십시오.

프로젝트의 설정을 사용한, 사용자에게 맞게 VM 이름 지정에 대한 자세한 내용은 [vRealize Automation Cloud Assembly를 사용하여 배포된 리소스의 이름을 사용자 지정하는 방법](#) 항목을 참조하십시오.

## 절차

- 1 **클라우드 템플릿 > 새로 만들기**를 클릭하고 **새 클라우드 템플릿** 페이지에 다음 정보를 입력하고 **생성**을 클릭합니다.

- **이름** = ipam-bpa
- **설명** = Infoblox IPAM 통합을 사용하는 클라우드 템플릿
- **프로젝트** = 123VC

- 2 이 예에서는 클라우드 애그노스틱 시스템 구성 요소와 클라우드 애그노스틱 네트워크 구성 요소를 클라우드 템플릿 캔버스에 추가하고 두 구성 요소를 연결합니다.

- 3 클라우드 템플릿 코드를 편집하여 네트워크 프로파일에 추가한 기능 태그와 일치하는 네트워크 구성 요소에 제약 조건 태그를 추가합니다. 이 예에서는 해당 태그 값이 *infoblox\_abx*입니다.

- 4 클라우드 템플릿 코드를 편집하여 네트워크 할당 유형을 *static*으로 지정합니다.

외부 IPAM 제공자를 사용하는 경우에는 `assignment: static` 설정이 필요합니다.

이 예에서 지정된 IP 주소 10.23.117.4는 연결된 네트워크 프로파일의 네트워크에 대해 선택한 외부 IPAM 주소 공간에서 현재 사용 가능한 것으로 알려져 있습니다. `assignment: static` 설정은 필수이지만 `address: value` 설정은 아닙니다. 특정 주소 값에서 외부 IP 주소 선택을 시작하도록 선택할 수 있지만 그렇게 할 필요는 없습니다. `address: value` 설정을 지정하지 않으면 외부 IPAM 제공자는 외부 IPAM 네트워크에서 사용 가능한 다음 주소를 선택합니다.

- 5 다음 예제와 비교하여 클라우드 템플릿 코드를 확인합니다.

```
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}'
```

클라우드 템플릿에서 DNS 및 DHCP 설정을 지정하는 데 사용할 수 있는 Infoblox 속성의 예는

[vRealize Automation 클라우드 템플릿에서 IPAM 통합을 위해 Infoblox 특정 속성 및 확장 가능한 특성 사용](#)의 내용을 참조하십시오.

- 6 [클라우드 템플릿] 페이지에서 **배포**를 클릭하고 배포 이름을 *Infoblox-1*로 지정한 후 **배포 유형** 페이지에서 **배포**를 클릭합니다.
- 7 클라우드 템플릿이 배포되는 동안 **확장성** 탭을 클릭하고 **작업 > 작업 실행**을 선택하여 *Infoblox\_AllocateIP\_n* 확장성 작업이 실행 중인지 확인합니다.  
  
확장성 작업이 완료되고 시스템이 프로비저닝되면 *Infoblox\_Update\_n* 작업이 MAC 주소를 Infoblox로 전파합니다.
- 8 Infoblox 계정에 로그인하여 열면 연결된 10.23.117.0/24 네트워크의 IPAM 주소에 대한 새 호스트 레코드를 볼 수 있습니다. Infoblox의 DNS 탭을 열어서 새 DNS 호스트 레코드를 볼 수도 있습니다.
- 9 VM이 프로비저닝되고 있는지 확인하려면 호스트 vCenter 및 vSphere Web Client에 로그인하여 프로비저닝된 시스템을 찾아서 DNS 이름과 IP 주소를 확인합니다.  
  
프로비저닝된 VM이 시작된 후에는 *Infoblox\_AllocateIP* 확장성 작업에 의해 MAC 주소가 Infoblox에 전파됩니다.
- 10 vRealize Automation Cloud Assembly에서 새 네트워크 레코드를 보려면 **인프라 > 리소스 > 네트워크**를 선택하고 **IP 주소** 탭을 클릭하여 엽니다.
- 11 배포를 삭제하면 배포에서 VM의 IPAM 주소가 해제되어 해당 IP 주소를 외부 IPAM 제공자가 다른 할당에 다시 사용할 수 있습니다. vRealize Automation Cloud Assembly에서 이 이벤트에 대한 확장성 작업은 *Infoblox\_Deallocate*입니다.

## vRealize Automation 클라우드 템플릿에서 IPAM 통합을 위해 Infoblox 특정 속성 및 확장 가능한 특성 사용

Infoblox용 외부 IPAM 통합을 포함하는 vRealize Automation 프로젝트에 대해 Infoblox 특정 속성을 사용할 수 있습니다.

클라우드 템플릿 설계 및 배포에서 Infoblox IPAM 통합에 다음 Infoblox 속성을 사용할 수 있습니다. 클라우드 템플릿 배포 중 IP 주소 할당을 추가적으로 제어하기 위해 vRealize Automation에서 사용할 수 있습니다. 이러한 속성의 사용은 선택 사항입니다.

이러한 속성은 vRealize Automation용 Infoblox 플러그인의 최신 버전에 포함되어 제공됩니다. Infoblox 플러그인 버전 및 vRealize Automation의 IPAM 통합을 위한 Infoblox 플러그인 최신 버전을 얻을 수 있는 위치에 대한 자세한 내용은 [vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포](#)에서 확인하십시오.

- Infoblox.IPAM.createFixedAddress

이 속성을 사용하면 Infoblox 내에서 고정 주소 레코드를 생성할 수 있습니다. 가능한 값은 True 및 False입니다. 기본적으로 호스트 레코드가 생성됩니다. 기본값은 False입니다.

- `Infoblox.IPAM.Network.dnsView`

이 속성을 사용하면 Infoblox 내에서 호스트 레코드를 생성할 때 DNS 보기를 사용할 수 있습니다.

- `Infoblox.IPAM.Network.enableDns`

Infoblox에서 IP를 할당할 때 이 속성을 사용하면 DNS 레코드도 생성할 수 있습니다. 가능한 값은 True 및 False입니다. 기본값은 True입니다.

- `Infoblox.IPAM.Network.enableDhcp`

이 옵션을 True로 설정하여 호스트 주소에 대해 DHCP 구성을 사용하도록 설정할 수 있습니다.

- `Infoblox.IPAM.Network.dnsSuffix`

이 속성을 사용하면 Infoblox 네트워크의 도메인 DHCP 옵션을 새 항목으로 덮어쓸 수 있습니다. 이 기능은 Infoblox 네트워크에 도메인 DHCP 옵션이 설정되어 있지 않거나 도메인 DHCP 옵션을 덮어써야 하는 경우에 유용합니다. 기본값은 null(빈 문자열)입니다.

Infoblox와 같은 외부 IPAM 제공자를 사용하는 경우 시스템을 프로비저닝할 때 DNS 접미사를 지정해야 합니다. DNS 접미사는 필수이지만 다음 방법 중 하나로 지정할 수 있습니다.

- vRealize Automation의 vSphere 네트워크 서버넷에서 DNS 접미사를 지정합니다.
- 클라우드 템플릿의 시스템 리소스 코드에 `Infoblox.IPAM.Network.dnsSuffix` 속성을 지정합니다.

아래의 `Infoblox.IPAM.Network.hostnameNicSuffix` 섹션에 예가 나와 있습니다.

`Infoblox.IPAM.Network.dnsSuffix`는 `Infoblox.IPAM.Network.enableDns`가 True로 설정된 경우에만 적용됩니다.

- `Infoblox.IPAM.Network.hostnameNicSuffix`

이 속성을 사용하여 호스트 이름을 생성할 때 NIC 인덱스 접미사를 지정할 수 있습니다.

이 방법을 사용하면 NIC가 둘 이상인 시스템을 프로비저닝할 수 있으며 이 경우 각 NIC의 호스트 이름이 사용자 지정된 접미사로 구분됩니다. 다음 예에 나와 있는 것처럼 2개의 NIC가 있는 시스템(예: *my-machine*)을 프로비저닝하여 첫 번째 NIC의 호스트 이름 접미사는 *-nic1*이고 다른 NIC의 호스트 이름 접미사는 *-nic2*가 되도록 할 수 있습니다.

예에 나와 있는 것처럼 DNS 접미사를 지정할 수도 있습니다.

`Infoblox.IPAM.Network.dnsSuffix` 속성이 `test.local` 값과 함께 사용되어 첫 번째 NIC의 이름은 *my-machine-nic1.test.local*이고 다른 NIC의 이름은 *my-machine-nic2.test.local*이 됩니다.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Machine_1:
```

```

type: Cloud.Machine
properties:
  Infoblox.IPAM.Network.dnsSuffix: test.local
  Infoblox.IPAM.Network0.hostnameNicSuffix: -nic1
  Infoblox.IPAM.Network1.hostnameNicSuffix: -nic2
  image: ubuntu
  flavor: small
  networks:
    - network: '${resource.Cloud_Network_1.id}'
      deviceIndex: 0
    - network: '${resource.Cloud_Network_2.id}'
      deviceIndex: 1
Cloud_Network_1:
  type: Cloud.Network
  properties:
    networkType: existing
Cloud_Network_2:
  type: Cloud.Network
  properties:
    networkType: existing

```

이 속성은 Infoblox 플러그인 버전 1.3에 도입되었습니다. [vRealize Automation](#)에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포의 내용을 참조하십시오.

- 확장성 구독을 사용하여 속성을 지정할 수도 있습니다.

이 사용 사례와 관련된 Infoblox 확장 가능 특성에 대한 자세한 내용은 [vRealize Automation](#)와 통합하기 위해 Infoblox 애플리케이션에서 필요한 확장 가능 특성 추가에서 참조하십시오.

## 클라우드 템플릿의 서로 다른 시스템 NIC에서 Infoblox 속성 사용

다음 Infoblox 속성은 클라우드 템플릿의 각 시스템 NIC에 대해 다른 값을 지원할 수 있습니다.

- Infoblox.IPAM.Network.enableDhcp
- Infoblox.IPAM.Network.dnsView
- Infoblox.IPAM.Network.enableDns
- Infoblox.IPAM.Network.hostnameNicSuffix

예를 들어 각 NIC에 대해 다른 Infoblox.IPAM.Network.dnsView 값을 사용하려면 각 NIC에 대해 Infoblox.IPAM.Network<nicIndex>.dnsView 항목을 사용합니다. 다음 샘플은 두 NIC에 대해 다른 Infoblox.IPAM.Network.dnsView를 보여 줍니다.

```

formatVersion: 1
inputs: {}
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      Infoblox.IPAM.Network0.dnsView: default
      Infoblox.IPAM.Network1.dnsView: my-net
    image: ubuntu
    flavor: small

```

```

networks:
  - network: '${resource.Cloud_Network_1.id}'
    deviceIndex: 0
  - network: '${resource.Cloud_Network_2.id}'
    deviceIndex: 1
Cloud_Network_1:
  type: Cloud.Network
  properties:
    networkType: existing
Cloud_Network_2:
  type: Cloud.Network
  properties:
    networkType: existing

```

기본적으로 Infoblox 통합은 Infoblox의 기본 DNS 보기에 DNS 호스트 레코드를 생성합니다. Infoblox 관리자가 사용자 지정 DNS 보기를 생성한 경우, 시스템 구성 요소의 Infoblox.IPAM.Network.dnsView 속성을 사용하여 기본 통합 동작을 덮어쓰고 명명된 보기를 지정할 수 있습니다. 예를 들어, 다음 속성을 Cloud\_Machine\_1 구성 요소에 추가하여 Infoblox에서 명명된 DNS 보기를 지정할 수 있습니다.

```

Cloud_Machine_1:
  type: Cloud.Machine
  properties:
    image: ubuntu
    flavor: small
    Infoblox.IPAM.Network.dnsView: <dns-view-name>

```

DNS 보기 구성 및 사용에 대한 자세한 내용은 Infoblox 제품 설명서의 [DNS 보기](#)를 참조하십시오.

Infoblox 통합 워크플로의 예는 [vRealize Automation](#)에서 외부 IPAM 제공자 범위 할당을 사용하는 클라우드 템플릿 정의 및 배포의 내용을 참조하십시오.

## Infoblox 속성을 지정하는 방법

vRealize Automation Cloud Assembly에서 다음 방법 중 하나를 사용하여 Infoblox 속성을 지정할 수 있습니다.

- **인프라 > 관리 > 프로젝트** 페이지에서 **사용자 지정 속성** 섹션을 사용하여 프로젝트의 속성을 지정할 수 있습니다. 이 방법을 사용하면 지정된 속성이 이 프로젝트의 범위에서 프로비저닝된 모든 시스템에 적용됩니다.
- 클라우드 템플릿의 각 시스템 구성 요소에서 속성을 지정할 수 있습니다. Infoblox.IPAM.Network.dnsView 속성의 사용을 설명하는 샘플 클라우드 템플릿 코드는 다음과 같습니다.

```

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

```

## vRealize Automation에서 Infoblox 필터를 사용하여 네트워크 데이터 수집을 제어

Infoblox의 경우 데이터 수집 네트워크 수를 vRealize Automation 작업에 필요한 네트워크로만 제한할 수 있습니다. 이렇게 하면 전송되는 데이터의 양이 줄어들고 시스템 성능이 향상됩니다.

vRealize Automation은 외부 IPAM 시스템에서 10분마다 데이터를 수집합니다. Infoblox의 경우, vRealize Automation 작업에서 사용되는 네트워크의 하위 집합만 검색하고 데이터를 수집하도록 여러 가지 방법으로 필터링할 수 있습니다.

Infoblox 생성 IP 주소를 사용하는 네트워크에 대한 데이터 수집을 필터링하려면 [IPAM 통합] 탭에서 다음 속성을 사용합니다. 필터 속성은 Infoblox에 대한 외부 IPAM 통합 지점을 생성하거나 편집할 때 사용할 수 있습니다.

이러한 필터는 vRealize Automation 8.3 이상 및 [Infoblox v1.3 플러그인](#)에서만 사용할 수 있습니다.

[Infoblox v1.3 플러그인](#)은 vRealize Automation 8.1 또는 8.2와 함께 사용할 수 있지만, KB 문서 [vRealize Automation 8.x와 Infoblox 1.3 호환성\(82142\)](#)에 설명된 대로 일부 상황에서만 주의하여 사용할 수 있습니다.

- Infoblox.IPAM.NetworkContainerFilter

네트워크 컨테이너에 대한 필터.

- Infoblox.IPAM.NetworkFilter

네트워크에 대한 필터.

- Infoblox.IPAM.RangeFilter

IP 주소 범위에 대한 필터.

이러한 옵션은 "vRA Cloud Infoblox 플러그인 버전1.3" 에서 사용할 수 있습니다.

이러한 데이터 수집 필터를 이미 데이터가 수집된 네트워크에 적용할 때는 주의해야 합니다. 일부 네트워크의 데이터가 수집되지 않도록 필터를 적용하는 경우 수집되지 않는 네트워크는 불필요한 것으로 간주되며 vRealize Automation에서 삭제됩니다. 예외는 vRealize Automation 서브넷에 연결된 네트워크입니다. 이전에 데이터를 수집한 네트워크에서 이후에 검색 및 데이터를 수집하지 않으면(예를 들어, 데이터 수집 작업에서 필터링으로 제외되었기 때문에) vRealize Automation 데이터베이스에서 삭제됩니다. 하지만, 이전에 데이터를 수집한 네트워크가 vRealize Automation에서 사용 중인 경우에는 삭제되지 않습니다.

이러한 필터는 여러 네트워크 개체에 대한 검색 요청에서 쿼리 매개 변수로 적용됩니다. Infoblox가 지원하는 검색 매개 변수를 사용할 수 있습니다. 정규식 또는 정확한 일치로 하는 CIDR 또는 확장 가능한 특성을 기준으로 필터링합니다. 이 형식에는 Infoblox WAPI 필터링 형식([Infoblox WAPI 설명서 참조](#))이 사용됩니다. CIDR 또는 확장 가능한 특성으로 필터링하는 방법은 다음 예에 나와 있습니다.

- 네트워크 및 네트워크 컨테이너에 대한 CIDR을 기반으로 필터링합니다. 예:
  - 정확히 일치 - Infoblox.IPAM.NetworkFilter: network=192.168.0.0
  - 확장 가능 특성에 따라 일치 - Infoblox.IPAM.NetworkFilter: network~=192.168
- IP 주소 범위에 대한 CIDR을 기반으로 필터링합니다. 예:
 

정규식 및 네트워크 보기 이름으로 일치 - Infoblox.IPAM.RangeFilter:

```
network~=192.168.&network_view=my_view
```
- 네트워크, IP 범위 및 네트워크 컨테이너에 대한 확장 가능 특성을 기반으로 필터링합니다.
 

구문은 *filter\_name=\*ext\_attr=ext\_attr\_value* 형식을 사용합니다. 예:

  - 정확히 일치 - \*Building=Data Center
  - 정규식으로 일치, '~' 사용 - \*Building~=\*Center
  - 대/소문자 구분 일치, ':' 사용 - \*Building:=data center
  - 일치하는 항목 제외, '!' 사용 - \*Building!=Data Center
  - 정규식으로 일치(대/소문자 구분 및 제외를 결합할 수 있음): \*Building! ~:=Data Cent / \*Building~:=center
- 위 필터링 방법의 구문을 사용하여 CIDR 및 확장 가능 특성을 기반으로 필터링합니다. 예:
 

```
network=192.168.&*Building=Data Center
```

이러한 속성에서 확장 가능한 특성 및 정규식을 사용하는 방법에 대한 자세한 내용은 [Infoblox 검색 매개 변수에 지원되는 표현식 및 Infoblox REST API 참조 가이드](#)를 참조하십시오.

# 조직에 vRealize Automation Cloud Assembly 설정

## 3

Cloud Assembly 관리자는 사용자 역할을 이해하고 클라우드 계정 벤더 및 통합 애플리케이션과의 연결을 설정해야 합니다.

클라우드 계정 및 통합을 구성하는 경우 Cloud Assembly와 해당 대상 시스템 간의 통신을 구성하게 됩니다.

본 장은 다음 항목을 포함합니다.

- vRealize Automation 사용자 역할이란?
- vRealize Automation Cloud Assembly에 클라우드 계정 추가
- vRealize Automation를 다른 애플리케이션과 통합
- vRealize Automation Cloud Assembly의 온보딩 계획이란?
- vRealize Automation Cloud Assembly 환경에 대한 고급 구성

## vRealize Automation 사용자 역할이란?

vRealize Automation에는 몇 가지 수준의 사용자 역할이 있습니다. 이렇게 다양한 수준을 통해 클라우드 템플릿, 카탈로그 항목 및 파이프라인을 생성하거나 소비하는 조직, 프로젝트, 서비스에 대한 액세스 권한 및 사용자 인터페이스의 개별 부분을 사용하거나 볼 수 있는 기능을 제어합니다. 이렇게 다양한 수준을 통해 클라우드 관리자의 운영 요구에 필요한 수준의 세분성을 적용할 수 있는 다양한 도구가 제공됩니다.

## 일반 역할 설명

사용자 역할은 다양한 수준에서 정의됩니다. 서비스 수준 역할은 각 서비스에 대해 정의됩니다.

서비스 역할에 대한 자세한 내용은 아래 표에 나와 있습니다.

역할	일반 사용 권한	역할이 정의된 위치
조직 소유자	콘솔에 액세스하고 조직에 사용자를 추가할 수 있습니다. 조직 소유자는 서비스 역할이 없으면 서비스에 액세스할 수 없습니다. <a href="#">조직 사용자 역할</a> 에 대한 자세한 내용	조직 콘솔
조직 멤버	콘솔에 액세스할 수 있습니다. 조직 멤버는 서비스 역할이 없으면 서비스에 액세스할 수 없습니다. <a href="#">조직 사용자 역할</a> 에 대한 자세한 내용	조직 콘솔
서비스 관리자	콘솔에 액세스할 수 있고 서비스에서 전체 보기, 업데이트 및 삭제 권한을 갖습니다. <ul style="list-style-type: none"> <li>■ <a href="#">Cloud Assembly</a> 서비스 역할</li> <li>■ <a href="#">Service Broker</a> 서비스 역할</li> <li>■ <a href="#">Code Stream</a> 서비스 역할</li> <li>■ <a href="#">vRA Migration Assistant</a> 서비스 역할</li> <li>■ <a href="#">Orchestrator</a> 서비스 역할</li> <li>■ <a href="#">SaltStack Config</a> 서비스 역할</li> </ul>	조직 콘솔
서비스 사용자	제한된 권한으로 콘솔 및 서비스에 액세스할 수 있습니다. 서비스 멤버의 사용자 인터페이스는 제한됩니다. 프로젝트 멤버 자격에 따라 멤버가 보거나 수행할 수 있는 작업이 달라집니다. <ul style="list-style-type: none"> <li>■ <a href="#">Cloud Assembly</a> 서비스 역할</li> <li>■ <a href="#">Service Broker</a> 서비스 역할</li> <li>■ <a href="#">Code Stream</a> 서비스 역할</li> </ul>	조직 콘솔
서비스 뷰어	보기 전용 모드에서 콘솔 및 서비스에 액세스할 수 있습니다. <ul style="list-style-type: none"> <li>■ <a href="#">Cloud Assembly</a> 서비스 역할</li> <li>■ <a href="#">Service Broker</a> 서비스 역할</li> <li>■ <a href="#">Code Stream</a> 서비스 역할</li> <li>■ <a href="#">vRA Migration Assistant</a> 서비스 역할</li> <li>■ <a href="#">Orchestrator</a> 서비스 역할</li> </ul>	조직 콘솔
실행자(vRealize Automation Code Stream에만 해당)	콘솔에 액세스하고 파이프라인 실행을 관리할 수 있습니다. <a href="#">Code Stream</a> 서비스 역할	조직 콘솔

역할	일반 사용 권한	역할이 정의된 위치
Orchestrator 워크플로 디자이너 (Orchestrator 전용)	자신의 vRealize Orchestrator 클라이언트 콘텐츠를 생성, 실행, 편집 및 삭제할 수 있습니다. 자신의 콘텐츠를 할당된 그룹에 추가할 수 있습니다. vRealize Orchestrator 클라이언트의 관리 및 문제 해결 기능에 액세스할 수 없습니다. <a href="#">Orchestrator 서비스 역할</a>	조직 콘솔
프로젝트 역할	프로젝트 역할에 따라 프로젝트 리소스를 보고 관리할 수 있습니다. 프로젝트 역할에는 관리자, 멤버 및 뷰어가 포함됩니다. <a href="#">vRealize Automation의 조직 및 서비스 사용자 역할</a>	vRealize Automation Cloud Assembly, vRealize Automation Service Broker 및 vRealize Automation Code Stream
사용자 지정 역할	권한은 모든 서비스에 대한 vRealize Automation Cloud Assembly에 의해 정의됩니다. 사용자에게 최소한 관련 서비스의 서비스 뷰어 역할이 있어야 서비스에 액세스할 수 있습니다. 사용자 지정 역할은 서비스 역할 보다 우선합니다. <a href="#">vRealize Automation의 사용자 지정 사용자 역할</a>	vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker

## vRealize Automation의 조직 및 서비스 사용자 역할

vRealize Automation Cloud Assembly, vRealize Automation Service Broker, vRealize Automation Code Stream 서비스에 대해 정의된 조직 및 서비스 사용자 역할에 따라 각 서비스에서 사용자가 보고 수행하는 작업이 결정됩니다.

### 조직 사용자 역할

사용자 역할은 조직 소유자가 vRealize Automation 콘솔에서 조직에 대해 정의합니다. 역할에는 조직 역할과 서비스 역할이라는 두 가지 유형이 있습니다.

조직 역할은 전역적이며 조직의 모든 서비스에 적용됩니다. 조직 수준 역할은 조직 소유자 또는 조직 멤버 역할입니다.

조직 역할에 대한 자세한 내용은 [vRealize Automation 관리](#)를 참조하십시오.

서비스별 사용 권한인 vRealize Automation Cloud Assembly 서비스 역할은 콘솔의 조직 수준에서도 할당됩니다.

### 서비스 역할

이러한 서비스 역할은 조직 소유자가 할당합니다.

이 문서에는 다음 서비스에 대한 정보가 포함되어 있습니다.

#### ■ Cloud Assembly 서비스 역할

- Service Broker 서비스 역할
- Code Stream 서비스 역할
- vRA Migration Assistant 서비스 역할
- Orchestrator 서비스 역할
- SaltStack Config 서비스 역할

## Cloud Assembly 서비스 역할

vRealize Automation Cloud Assembly 서비스 역할은 사용자가 vRealize Automation Cloud Assembly에서 보고 수행할 수 있는 작업을 결정합니다. 이러한 서비스 역할은 조직 소유자가 콘솔에서 정의합니다.

**표 3-1. vRealize Automation Cloud Assembly 서비스 역할 설명**

역할	설명
Cloud Assembly 관리자	전체 사용자 인터페이스 및 API 리소스에 대해 읽기 및 쓰기 권한이 있는 사용자입니다. 클라우드 계정 추가, 새 프로젝트 생성 및 프로젝트 관리자 할당 등을 볼 수 있고 수행할 수 있는 유일한 사용자 역할입니다.
Cloud Assembly 사용자	Cloud Assembly 관리자 역할이 없는 사용자입니다. vRealize Automation Cloud Assembly 프로젝트에서는 관리자가 프로젝트에 사용자를 프로젝트 멤버, 관리자 또는 뷰어로 추가합니다. 관리자는 프로젝트 관리자를 추가할 수도 있습니다.
Cloud Assembly 뷰어	정보를 볼 수 있는 읽기 권한이 있지만 값을 생성, 업데이트 또는 삭제할 수 없는 사용자입니다. 모든 프로젝트에서 읽기 전용 역할입니다.  뷰어 역할이 있는 사용자는 관리자가 사용할 수 있는 모든 정보를 볼 수 있습니다. 이들을 프로젝트 관리자 또는 프로젝트 멤버로 만들지 않는 한 아무 작업도 수행할 수 없습니다. 사용자가 프로젝트와 연관된 경우에는 역할과 관련된 권한이 있습니다. 프로젝트 뷰어는 관리자 또는 멤버 역할이 수행하는 방식으로 권한을 확장하지 않습니다.

서비스 역할 외에도 vRealize Automation Cloud Assembly에는 프로젝트 역할이 있습니다. 모든 서비스에서 모든 프로젝트를 사용할 수 있습니다.

프로젝트 역할은 vRealize Automation Cloud Assembly에 정의되며 프로젝트마다 다를 수 있습니다.

다음 표에는 다양한 서비스 및 프로젝트 역할이 볼 수 있고 수행할 수 있는 내용이 있습니다. 서비스 관리자에게는 사용자 인터페이스의 모든 영역에 대한 모든 권한이 있습니다.

프로젝트 역할에 대한 설명은 사용자에게 부여할 권한을 결정하는 데 도움이 됩니다.

- 프로젝트 관리자는 서비스 관리자가 생성한 인프라를 활용하여, 프로젝트의 멤버가 개발 작업에 필요한 리소스를 사용할 수 있도록 보장합니다.
- 프로젝트 멤버는 프로젝트 내에서 작업하며 클라우드 템플릿을 설계 및 배포합니다.

- 프로젝트 뷰어는 클라우드 템플릿 다운로드와 같은 비파괴적인 작업을 수행할 수 있는 몇 가지 경우를 제외하고 읽기 전용 액세스로 제한됩니다.

표 3-2. vRealize Automation Cloud Assembly 서비스 역할 및 프로젝트 역할

UI 컨텍스트	작업	Cloud Assembly 관리자	Cloud Assembly 뷰어	Cloud Assembly 사용자 프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자 또는 멤버여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
Cloud Assembly 액세스						
콘솔	vRA 콘솔에서 Cloud Assembly를 보고 열 수 있음	예	예	예	예	예
인프라						
	[인프라] 탭을 보고 열기	예	예	예	예	예
구성 - 프로젝트	프로젝트 생성	예				
	프로젝트 요약, 프로비저닝, Kubernetes, 통합 및 테스트 프로젝트 구성에서 값 업데이트 또는 삭제.	예				
	사용자 및 그룹을 추가하고 프로젝트에서 역할을 할당합니다.	예		예. 프로젝트.		
	프로젝트 보기	예	예	예. 프로젝트	예. 프로젝트	예. 프로젝트
구성 - 클라우드 영역	클라우드 영역 생성, 업데이트 또는 삭제	예				
	클라우드 영역 보기	예	예			
구성 - Kubernetes 영역	Kubernetes 영역 생성, 업데이트 또는 삭제	예				
	Kubernetes 영역 보기	예	예			
구성 - 버전	버전 생성, 업데이트 또는 삭제	예				
	버전 보기	예	예			
구성 - 이미지 매핑	이미지 매핑 생성, 업데이트 또는 삭제	예				
	이미지 매핑 보기	예	예			
구성 - 네트워크 프로파일	네트워크 프로파일 생성, 업데이트 또는 삭제	예				
	이미지 네트워크 프로파일 보기	예	예			

표 3-2. vRealize Automation Cloud Assembly 서비스 역할 및 프로젝트 역할 (계속)

UI 컨텍스트	작업	Cloud Assembly 관리자	Cloud Assembly 뷰어	Cloud Assembly 사용자 프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자 또는 멤버여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
구성 - 스토리지 프로파일	스토리지 프로파일 생성, 업데이트 또는 삭제	예				
	이미지 스토리지 프로파일 보기	예	예			
구성 - 가격 책정 카드	가격 책정 카드 생성, 업데이트 또는 삭제	예				
	가격 책정 카드 보기	예	예			
구성 - 태그	태그 생성, 업데이트 또는 삭제	예				
	태그 보기	예	예			
리소스 - 계산	검색된 계산 리소스에 태그 추가	예				
	검색된 계산 리소스 보기	예	예			
리소스 - 네트워크	네트워크 태그, IP 범위 및 IP 주소 수정	예				
	검색된 네트워크 리소스 보기	예	예			
리소스 - 보안	검색된 보안 그룹에 태그 추가	예				
	검색된 보안 그룹 보기	예	예			
리소스 - 스토리지	검색된 스토리지에 태그 추가	예				
	스토리지 보기	예	예			
리소스 - 시스템	시스템 추가 및 삭제	예				
	시스템 보기	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
리소스 - 볼륨	검색된 스토리지 볼륨 삭제	예				
	검색된 스토리지 볼륨 보기	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트.
리소스 - Kubernetes	Kubernetes 클러스터 배포 또는 추가, 네임스페이스 생성 또는 추가	예				
	Kubernetes 클러스터 및 네임스페이스 보기	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
작업 - 요청	배포 요청 기록 삭제	예				

표 3-2. vRealize Automation Cloud Assembly 서비스 역할 및 프로젝트 역할 (계속)

UI 컨텍스트	작업	Cloud Assembly 관리자	Cloud Assembly 뷰어	Cloud Assembly 사용자 프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자 또는 멤버여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
	배포 요청 기록 보기	예	예	예. 프로젝트	예. 프로젝트	예. 프로젝트
작업 - 이벤트 로그	이벤트 로그 보기	예	예	예. 프로젝트	예. 프로젝트	예. 프로젝트
연결 - 클라우드 계정	클라우드 계정 생성, 업데이트 또는 삭제	예				
	클라우드 계정 보기	예	예			
연결 - 통합	통합 생성, 업데이트 또는 삭제	예				
	통합 보기	예	예			
온보딩	온보딩 계획 생성, 업데이트 또는 삭제	예				
	온보딩 계획 보기	예	예			예. 프로젝트
<b>마켓플레이스</b>						
	[마켓플레이스] 탭을 보고 열기	예	예			
	[설계] 탭에서 다운로드된 클라우드 템플릿 사용	예		예. 프로젝트와 연결된 경우.	예. 프로젝트와 연결된 경우.	
마켓플레이스 - 클라우드 템플릿	클라우드 템플릿 다운로드	예				
	클라우드 템플릿 보기	예	예			
마켓플레이스 - 이미지	이미지 다운로드	예				
	이미지 보기	예	예			
마켓플레이스 - 다운로드	다운로드한 모든 항목의 로그 보기	예	예			
<b>확장성</b>						
	[확장성] 탭을 보고 열기	예	예			예
이벤트	확장성 이벤트 보기	예	예			

표 3-2. vRealize Automation Cloud Assembly 서비스 역할 및 프로젝트 역할 (계속)

UI 컨텍스트	작업	Cloud Assembly 관리자	Cloud Assembly 뷰어	Cloud Assembly 사용자 프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자 또는 멤버여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
구독	확장성 구독 생성, 업데이트 또는 삭제	예				
	구독 비활성화	예				
	구독 보기	예	예			
라이브러리 - 이벤트 항목	이벤트 항목 보기	예	예			
라이브러리 - 작업	확장성 작업 생성, 업데이트 또는 삭제	예				
	확장성 작업 보기	예	예			
라이브러리 - 워크플로	확장성 워크플로 보기	예	예			
작업 - 작업 실행	확장성 작업 실행 취소 또는 삭제	예				
	확장성 작업 실행 보기	예	예			예. 프로젝트
활동 - 워크플로 실행	확장성 워크플로 실행 보기	예	예			
<b>설계</b>						
설계	[설계] 탭을 열고 클라우드 템플릿 목록 보기	예	예	예. 프로젝트	예. 프로젝트	예. 프로젝트
클라우드 템플릿	클라우드 템플릿 생성, 업데이트 및 삭제	예		예. 프로젝트	예. 프로젝트	
	클라우드 템플릿 보기	예	예	예. 프로젝트	예. 프로젝트	예. 프로젝트
	클라우드 템플릿 다운로드	예	예	예. 프로젝트	예. 프로젝트	예. 프로젝트
	클라우드 템플릿 업로드	예		예. 프로젝트	예. 프로젝트	
	클라우드 템플릿 배포	예		예. 프로젝트	예. 프로젝트	
	클라우드 템플릿 버전 지정 및 복원	예		예. 프로젝트	예. 프로젝트	
	카탈로그에 클라우드 템플릿 릴리스	예		예. 프로젝트	예. 프로젝트	

표 3-2. vRealize Automation Cloud Assembly 서비스 역할 및 프로젝트 역할 (계속)

UI 컨텍스트	작업	Cloud Assembly 관리자	Cloud Assembly 뷰어	Cloud Assembly 사용자 프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자 또는 멤버여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
사용자 지정 리소스	사용자 지정 리소스 생성, 업데이트 또는 삭제	예				
	사용자 지정 리소스 보기	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
사용자 지정 작업	사용자 지정 작업 생성, 업데이트 또는 삭제	예				
	사용자 지정 작업 보기	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
<b>배포</b>						
	[배포] 탭 보기 및 열기	예	예	예	예	예
	배포 세부 정보, 배포 기록 및 문제 해결 정보를 포함한 배포 보기.	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
	정책에 따라 배포에서 2일차 작업 실행.	예		예, 프로젝트	예, 프로젝트	

## Service Broker 서비스 역할

vRealize Automation Service Broker 서비스 역할은 사용자가 vRealize Automation Service Broker에서 보고 수행할 수 있는 작업을 결정합니다. 이러한 서비스 역할은 조직 소유자가 콘솔에서 정의합니다.

표 3-3. Service Broker 서비스 역할 설명

역할	설명
Service Broker 관리자	전체 사용자 인터페이스 및 API 리소스에 대해 읽기/쓰기 액세스 권한을 갖고 있어야 합니다. 새 프로젝트 생성 및 프로젝트 관리자 할당을 비롯한 모든 작업을 수행할 수 있는 유일한 사용자 역할입니다.
Service Broker 사용자	vRealize Automation Service Broker 관리자 역할이 없는 모든 사용자입니다.  vRealize Automation Service Broker 프로젝트에서는 관리자가 프로젝트에 사용자를 프로젝트 멤버, 관리자 또는 뷰어로 추가합니다. 관리자는 프로젝트 관리자를 추가할 수도 있습니다.
Service Broker 뷰어	정보를 볼 수 있는 읽기 권한이 있지만 값을 생성, 업데이트 또는 삭제할 수 없는 사용자입니다.  뷰어 역할이 있는 사용자는 관리자가 사용할 수 있는 모든 정보를 볼 수 있습니다. 이들을 프로젝트 관리자 또는 프로젝트 멤버로 만들지 않는 한 아무 작업도 수행할 수 없습니다. 사용자가 프로젝트와 연관된 경우에는 역할과 관련된 권한이 있습니다. 프로젝트 뷰어는 관리자 또는 멤버 역할이 수행하는 방식으로 권한을 확장하지 않습니다.

서비스 역할 외에도 vRealize Automation Service Broker에는 프로젝트 역할이 있습니다. 모든 서비스에서 모든 프로젝트를 사용할 수 있습니다.

프로젝트 역할은 vRealize Automation Service Broker에 정의되며 프로젝트마다 다를 수 있습니다.

다음 표에는 다양한 서비스 및 프로젝트 역할이 볼 수 있고 수행할 수 있는 내용이 있습니다. 서비스 관리자에게는 사용자 인터페이스의 모든 영역에 대한 모든 권한이 있습니다.

프로젝트 역할에 대한 다음 설명을 사용하면 사용자에게 부여할 사용 권한을 결정하는 데 도움이 됩니다.

- 프로젝트 관리자는 서비스 관리자가 생성한 인프라를 활용하여, 프로젝트의 멤버가 개발 작업에 필요한 리소스를 사용할 수 있도록 보장합니다.
- 프로젝트 멤버는 프로젝트 내에서 작업하며 클라우드 템플릿을 설계 및 배포합니다.
- 프로젝트 뷰어는 읽기 전용 액세스로 제한됩니다.

표 3-4. Service Broker 서비스 역할 및 프로젝트 역할

UI 컨텍스트	작업	Service Broker 관리자	Service Broker 뷰어	Service Broker 사용자		
				프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
<b>Service Broker 액세스</b>						
콘솔	콘솔에서 Service Broker를 보고 열 수 있음	예	예	예	예	예
<b>인프라</b>						

표 3-4. Service Broker 서비스 역할 및 프로젝트 역할 (계속)

UI 컨텍스트	작업	Service Broker 관리자	Service Broker 뷰어	Service Broker 사용자 프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
	[인프라] 탭을 보고 열기	예	예			
구성 - 프로젝트	프로젝트 생성	예				
	프로젝트 요약, 프로비저닝, Kubernetes, 통합 및 테스트 프로젝트 구성에서 값 업데이트 또는 삭제.	예				
	사용자 및 그룹을 추가하고 프로젝트에서 역할을 할당합니다.	예		예, 프로젝트.		
	프로젝트 보기	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
구성 - 클라우드 영역	클라우드 영역 생성, 업데이트 또는 삭제	예				
	클라우드 영역 보기	예	예			
구성 - Kubernetes 영역	Kubernetes 영역 생성, 업데이트 또는 삭제	예				
	Kubernetes 영역 보기	예	예			
연결 - 클라우드 계정	클라우드 계정 생성, 업데이트 또는 삭제	예				
	클라우드 계정 보기	예	예			
연결 - 통합	통합 생성, 업데이트 또는 삭제	예				
	통합 보기	예	예			
작업 - 요청	배포 요청 기록 삭제	예				
	배포 요청 기록 보기	예				
작업 - 이벤트 로그	이벤트 로그 보기	예				
<b>컨텐츠 및 정책</b>						
	[컨텐츠 및 정책] 탭을 보고 열기	예	예			
컨텐츠 소스	컨텐츠 소스 생성, 업데이트 또는 삭제	예				
	컨텐츠 소스 보기	예	예			
컨텐츠 공유	공유 컨텐츠 추가 또는 제거	예				

표 3-4. Service Broker 서비스 역할 및 프로젝트 역할 (계속)

UI 컨텍스트	작업	Service Broker 관리자	Service Broker 뷰어	Service Broker 사용자 프로젝트 관련 작업을 보고 수행하려면 사용자가 프로젝트 관리자여야 합니다.		
				프로젝트 관리자	프로젝트 멤버	프로젝트 뷰어
컨텐츠	공유 콘텐츠 보기	예	예			
	양식 사용자 지정 및 항목 구성	예				
	컨텐츠 보기	예	예			
정책 - 정의	정책 정의 생성, 업데이트 또는 삭제	예				
	정책 정의 보기	예	예			
정책 - 적용	적용 로그 보기	예	예			
알림 - 이메일 서버	이메일 서버 구성	예				
<b>카탈로그</b>						
	[카탈로그] 탭 보기 및 열기	예	예	예	예	예
	사용 가능한 카탈로그 항목 보기	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
	카탈로그 항목 요청	예		예, 프로젝트	예, 프로젝트	
<b>배포</b>						
	[배포] 탭 보기 및 열기	예	예	예	예	예
	배포 세부 정보, 배포 기록 및 문제 해결 정보를 포함한 배포 보기.	예	예	예, 프로젝트	예, 프로젝트	예, 프로젝트
	정책에 따라 배포에서 2일차 작업 실행	예		예, 프로젝트	예, 프로젝트	
<b>승인</b>						
	[승인] 탭 보기 및 열기	예	예	예	예	예
	승인 요청에 응답	예		Service Broker 사용자 역할만	Service Broker 사용자 역할만	Service Broker 사용자 역할만

## Code Stream 서비스 역할

vRealize Automation Code Stream 서비스 역할은 사용자가 vRealize Automation Code Stream에서 보고 수행할 수 있는 작업을 결정합니다. 이러한 역할은 조직 소유자가 콘솔에서 정의합니다. 모든 서비스에서 모든 프로젝트를 사용할 수 있습니다.

표 3-5. Code Stream 서비스 역할 설명

역할	설명
Code Stream 관리자	전체 사용자 인터페이스 및 API 리소스에 대해 읽기 및 쓰기 권한이 있는 사용자입니다. 이 사용자 역할은 프로젝트 생성, 끝점 통합, 트리거 추가, 파이프라인 및 사용자 지정 대시보드 생성, 끝점 및 변수를 제한된 리소스로 표시, 제한된 리소스를 사용하는 파이프라인 실행, 이 파이프라인을 vRealize Automation Service Broker에 게시하도록 요청하는 등의 작업을 포함한 모든 작업을 보고 수행할 수 있는 유일한 사용자 역할입니다.
Code Stream 개발자	파이프라인을 사용할 수 있지만 제한된 끝점 또는 변수로 작업할 수 없는 사용자입니다. 파이프라인에 제한된 끝점이나 변수가 포함된 경우 이 사용자는 제한된 끝점이나 변수를 사용하는 파이프라인 작업에 대한 승인을 받아야 합니다.
Code Stream Executor	파이프라인을 실행하고 사용자 작업을 승인하거나 거부할 수 있는 사용자입니다. 이 사용자는 파이프라인 실행을 재개, 일시 중지 및 취소할 수 있지만 파이프라인을 수정할 수는 없습니다.
Code Stream 사용자	vRealize Automation Code Stream에 액세스할 수 있지만 vRealize Automation Code Stream에서 다른 권한은 없는 사용자입니다.
Code Stream 뷰어	파이프라인, 끝점, 파이프라인 실행, 대시보드를 볼 수 있는 읽기 권한이 있지만 생성, 업데이트 또는 삭제할 수 없는 사용자입니다. 서비스 뷰어 역할도 있는 사용자는 관리자가 사용할 수 있는 모든 정보를 볼 수 있습니다. 이들을 프로젝트 관리자 또는 프로젝트 멤버로 만들지 않는 한 아무 작업도 수행할 수 없습니다. 사용자가 프로젝트와 연관된 경우에는 역할과 관련된 권한이 있습니다. 프로젝트 뷰어는 관리자 또는 멤버 역할이 수행하는 방식으로 권한을 확장하지 않습니다.

서비스 역할 외에도 vRealize Automation Code Stream에는 프로젝트 역할이 있습니다. 모든 프로젝트에서 모든 서비스에서 사용할 수 있습니다.

프로젝트 역할은 vRealize Automation Code Stream에 정의되며 프로젝트마다 다를 수 있습니다.

다음 표에는 다양한 서비스 및 프로젝트 역할이 볼 수 있고 수행할 수 있는 내용이 있습니다. 서비스 관리자에게는 사용자 인터페이스의 모든 영역에 대한 모든 권한이 있습니다.

프로젝트 역할에 대한 다음 설명을 사용하면 사용자에게 부여할 권한을 결정하는 데 도움이 됩니다.

- 프로젝트 관리자는 서비스 관리자가 생성한 인프라를 활용하여, 프로젝트의 멤버가 개발 작업에 필요한 리소스를 사용할 수 있도록 보장합니다. 프로젝트 관리자는 멤버를 추가할 수 있습니다.
- 서비스 역할이 있는 프로젝트 멤버는 서비스를 사용할 수 있습니다.
- 프로젝트 뷰어는 프로젝트를 볼 수 있지만 생성, 업데이트 또는 삭제할 수 없습니다.

제한된 항목을 제외한 모든 작업은 엔티티(제한된 변수와 끝점은 제외)에 대한 생성, 읽기, 업데이트 및 삭제 작업을 수행할 수 있는 권한이 이 역할에 있다는 의미입니다.

표 3-6. vRealize Automation Code Stream 서비스 역할 기능

UI 컨텍스트	기능	Code Stream 관리자 역할	Code Stream 개발자 역할	Code Stream Executor 역할	Code Stream 뷰어 역할	Code Stream 사용자 역할
파이프라인	파이프라인 보기	예	예	예	예	

표 3-6. vRealize Automation Code Stream 서비스 역할 기능 (계속)

UI 컨텍스트	기능	Code Stream 관리자 역할	Code Stream 개발자 역할	Code Stream Executor 역할	Code Stream 뷰어 역할	Code Stream 사용자 역할
	파이프라인 생성	예	예			
	파이프라인 실행	예	예	예		
	제한된 끝점 또는 변수를 포함하는 파이프라인 실행	예				
	파이프라인 업데이트	예	예			
	파이프라인 삭제	예	예			
<b>파이프라인 실행</b>						
	파이프라인 실행 보기	예	예	예	예	
	파이프라인 실행 재개, 일시 중지 및 취소	예	예	예		
	제한된 리소스에 대한 승인을 위해 중지되는 파이프라인 재개	예				
<b>사용자 지정 통합</b>						
	사용자 지정 통합 생성	예	예			
	사용자 지정 통합 읽기	예	예	예	예	
	사용자 지정 통합 업데이트	예	예			
<b>끝점</b>						
	실행 보기	예	예	예	예	
	실행 생성	예	예			
	실행 업데이트	예	예			
	실행 삭제	예	예			
<b>리소스를 제한된 리소스로 표시</b>						
	끝점 또는 변수를 제한된 리소스로 표시	예				
<b>대시보드</b>						
	대시보드 보기	예	예	예	예	

표 3-6. vRealize Automation Code Stream 서비스 역할 기능 (계속)

UI 컨텍스트	기능	Code Stream 관리자 역할	Code Stream 개발자 역할	Code Stream Executor 역할	Code Stream 뷰어 역할	Code Stream 사용자 역할
	대시보드 생성	예	예			
	대시보드 업데이트	예	예			
	대시보드 삭제	예	예			

## vRA Migration Assistant 서비스 역할

vRA Migration Assistant 서비스 역할은 vRA Migration Assistant 및 Cloud Assembly에서 볼 수 있고 수행할 수 있는 작업을 결정합니다. 이러한 서비스 역할은 조직 소유자가 콘솔에서 정의합니다.

표 3-7. vRealize Automation Migration Assistant 서비스 역할 설명

역할	설명
Migration Assistant 관리자	vRA Migration Assistant 및 Cloud Assembly에서 전체 보기, 업데이트 및 삭제 권한이 있는 사용자입니다. 이 역할에는 하나 이상의 Cloud Assembly 뷰어 역할도 있어야 합니다.
Migration Assistant 뷰어	정보를 볼 수 있는 읽기 권한이 있지만 vRA Migration Assistant 또는 Cloud Assembly에서 값을 생성, 업데이트 또는 삭제할 수 없는 사용자입니다. 이 역할에는 하나 이상의 Cloud Assembly 뷰어 역할도 있어야 합니다.

## Orchestrator 서비스 역할

Orchestrator 서비스 역할은 vRealize Orchestrator 클라이언트에서 볼 수 있고 수행할 수 있는 작업을 결정합니다. 이러한 서비스 역할은 조직 소유자가 콘솔에서 정의합니다.

표 3-8. vRealize Orchestrator 서비스 역할 설명

역할	설명
Orchestrator 관리자	vRealize Orchestrator에서 전체 보기, 업데이트 및 삭제 권한이 있는 사용자입니다. 관리자는 특정 그룹에서 생성된 콘텐츠에 액세스할 수도 있습니다.
Orchestrator 뷰어	모든 그룹 및 그룹 콘텐츠를 비롯한 기능 및 콘텐츠를 볼 수 있는 읽기 권한이 있지만 값을 생성, 업데이트, 실행, 삭제하거나 콘텐츠를 내보낼 수는 없는 사용자입니다.
Orchestrator 워크플로 디자이너	자신의 vRealize Orchestrator 클라이언트 콘텐츠를 생성, 실행, 편집 및 삭제할 수 있는 사용자입니다. 이들은 할당된 그룹에 자신의 콘텐츠를 추가할 수 있습니다. vRealize Orchestrator 클라이언트의 관리 및 문제 해결 기능에 액세스할 수 없는 워크플로 디자이너입니다.

## SaltStack Config 서비스 역할

SaltStack Config 서비스 역할은 vRealize Automation에서 볼 수 있고 수행할 수 있는 작업을 결정합니다. 이러한 서비스 역할은 조직 소유자가 콘솔에서 정의합니다.

표 3-9. vRealize Automation SaltStack Config 서비스 역할 설명

역할	설명
SaltStack Config 관리자	Cloud Assembly와의 통합이 구성된 경우 콘솔에서 SaltStack Config 타일에 액세스할 수 있는 사용자입니다. SaltStack Config 인스턴스에 로그인하려면, SaltStack Config에 정의된 SaltStack 관리자 권한이 사용자에게 있어야 합니다. Cloud Assembly 관리자 역할도 사용자에게 있어야 합니다.

## vRealize Automation의 사용자 지정 사용자 역할

vRealize Automation Cloud Assembly 관리자는 사용자가 vRealize Automation에서 보고 수행할 수 있는 작업을 정의하는 사용자 지정 사용자 역할을 생성할 수 있습니다. 그런 다음, 해당 역할에 사용자를 할당할 수 있습니다.

### 사용자 지정 사용자 역할 권한

vRealize Automation Cloud Assembly를 사용하면 보다 세분화된 사용자 역할을 정의하고 해당 역할에 사용자를 할당할 수 있습니다. 사용자 지정 역할에는 보기 및 관리라는 두 가지 범주가 있습니다.

- **보기.** 이 사용 권한을 가진 역할에 할당된 사용자는 사용자 인터페이스의 선택된 섹션에 있는 모든 프로젝트에 대한 모든 항목을 볼 수 있습니다. 이 역할은 계정, 구성 또는 할당된 값을 확인해야 하는 사용자에게 유용합니다.
- **관리.** 이 사용 권한을 가진 역할에 할당된 사용자는 모든 항목을 볼 수 있으며 사용자 인터페이스의 선택된 섹션의 모든 프로젝트에 대해 전체 추가, 편집 및 삭제 권한을 갖습니다.

이러한 권한은 다른 역할에 의해 부여되고 프로젝트 멤버 자격을 통해 제한되지 않는 권한을 확장합니다. 예를 들어, 인프라의 일부를 관리하도록 프로젝트 관리자의 권한을 확장하거나 서비스 뷰어에게 승인 요청을 검토하고 응답할 수 있는 권한을 제공할 수 있습니다.

사용자 역할을 정의하고 사용자를 할당하려면, 서비스 관리자 역할을 사용하여 vRealize Automation Cloud Assembly 또는 vRealize Automation Service Broker를 열고 **인프라 > 관리 > 사용자 지정 역할**을 선택합니다. vRealize Automation Code Stream에서 사용자 지정 역할을 구성할 수는 없지만 모든 서비스에 역할이 적용됩니다.

표 3-10. 사용자 지정 역할

사용자 인터페이스	사용 권한	설명
인프라		
	클라우드 계정 보기.	클라우드 계정 보기.
	클라우드 계정 관리	클라우드 계정 생성, 업데이트 또는 삭제.

표 3-10. 사용자 지정 역할 (계속)

사용자 인터페이스	사용 권한	설명
	이미지 매핑 보기	이미지 매핑 보기.
	이미지 매핑 관리	이미지 매핑 생성, 업데이트 또는 삭제.
	버전 매핑 보기	버전 매핑 보기.
	버전 매핑 관리	버전 매핑 생성, 업데이트 또는 삭제.
	클라우드 영역 보기	클라우드 영역 보기.
	클라우드 영역 관리	클라우드 영역 생성, 업데이트 또는 삭제.
	시스템 보기	시스템을 봅니다.
	요청 보기	활동 요청을 봅니다.
	요청 관리	목록에서 요청을 삭제합니다.
	통합 보기	통합을 봅니다.
	통합 관리	통합을 생성, 업데이트 또는 삭제합니다.
	프로젝트 보기	프로젝트 보기.
	프로젝트 관리	프로젝트 생성. 사용자를 추가하고 프로젝트에서 역할을 할당합니다. 프로젝트 요약, 사용자, 프로비저닝, Kubernetes, 통합 및 테스트 프로젝트 구성에서 값 업데이트 또는 삭제.
	온보딩 계획 보기	온보딩 계획 보기
	온보딩 계획 관리	온보딩 계획을 생성, 업데이트, 실행 또는 삭제
<b>카탈로그</b>		
	컨텐츠 보기	
	컨텐츠 관리	컨텐츠 소스 생성, 업데이트 또는 삭제. 컨텐츠 공유. 카탈로그 아이콘 및 요청 양식을 포함하여 컨텐츠 사용자 지정.
<b>정책</b>		
	정책 보기	정책 정의 보기.
	정책 관리	정책 정의 생성, 업데이트 또는 삭제.
<b>배포</b>		
	배포 보기	배포 세부 정보, 배포 기록 및 문제 해결 정보를 포함한 모든 배포 보기.

표 3-10. 사용자 지정 역할 (계속)

사용자 인터페이스	사용 권한	설명
	배포 관리	모든 배포를 보고 2일차 정책에서 관리자가 배포 및 배포 구성 요소에 대해 실행하도록 허용하는 2일차 작업 실행.
<b>클라우드 템플릿</b>		
	클라우드 템플릿 보기	클라우드 템플릿 보기.
	클라우드 템플릿 관리	클라우드 템플릿 생성, 업데이트, 테스트, 삭제, 버전 지정, 공유 및 클라우드 템플릿 버전 릴리스/릴리스 취소.
	클라우드 템플릿 편집	클라우드 템플릿 생성, 업데이트, 테스트, 버전 지정, 공유 및 클라우드 템플릿 버전 릴리스/릴리스 취소. 이 역할에는 클라우드 템플릿을 삭제할 수 있는 사용 권한이 없습니다.
	클라우드 템플릿 배포	프로젝트에서 클라우드 템플릿 테스트 및 배포.
	인라인 클라우드 템플릿 콘텐츠 배포	담당자가 연결된 프로젝트에서 클라우드 템플릿 배포. 프로젝트 역할은 관리자, 멤버 또는 뷰어일 수 있습니다.
<b>XaaS</b>		
	사용자 지정 리소스 보기	사용자 지정 리소스 보기.
	사용자 지정 리소스 관리	사용자 지정 리소스 생성, 업데이트 또는 삭제
	리소스 작업 보기	사용자 지정 작업 보기.
	리소스 작업 관리	사용자 지정 작업 생성, 업데이트 또는 삭제
<b>확장성</b>		
	확장성 리소스 보기	이벤트, 구독, 이벤트 항목, 작업, 워크플로, 작업 실행 및 워크플로 실행 보기.
	확장성 리소스 관리	확장성 구독 생성, 업데이트, 삭제 및 비활성화. 확장성 작업 생성, 업데이트 또는 삭제. 확장성 작업 실행 취소 또는 삭제.
<b>파이프라인</b>		
	파이프라인 관리	파이프라인, 끝점, 변수 및 트리거 구성 생성, 편집 및 삭제. 제한된 모델은 제외됩니다.

표 3-10. 사용자 지정 역할 (계속)

사용자 인터페이스	사용 권한	설명
	제한된 파이프라인 관리	파이프라인, 끝점, 변수 및 트리거 구성 생성, 편집 및 삭제. 제한된 모델이 포함됩니다.
	사용자 지정 통합 관리	사용자 지정 통합 추가, 편집 및 삭제.
	파이프라인 실행	파이프라인 모델 실행 및 트리거를 실행하고, 실행 및 트리거를 일시 중지, 취소, 재개 또는 다시 실행합니다.
	제한된 파이프라인 실행	파이프라인 모델 실행 및 트리거를 실행하고, 실행 및 트리거를 일시 중지, 취소, 재개 또는 다시 실행합니다. 제한된 끝점 및 변수를 해결합니다.
	실행 관리	파이프라인 모델 실행 및 트리거를 실행하고, 실행 및 트리거를 일시 중지, 취소, 재개 또는 다시 실행합니다. 제한된 끝점 및 변수를 해결합니다. 실행 삭제.
<b>승인</b>		
	승인 관리	승인 요청을 승인하거나 거부할 수 있는 [승인] 탭을 봅니다. 이 역할이 있는 승인자는 정책의 승인자가 아닌 한 승인 요청에 대한 이메일 알림을 수신하지 않습니다.

## 사용 사례: 사용자 역할을 통해 vRealize Automation에서 액세스를 제어하는 방법

클라우드 관리자는 사용자가 vRealize Automation에서 수행할 수 있는 작업을 제어합니다. 관리 목표 및 애플리케이션 개발 팀의 책임에 따라 다양한 방법으로 이러한 목표를 지원하는 사용자 역할을 구성할 수 있습니다.

다음 vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker 예는 세 가지 사용 사례를 기반으로 합니다. 이러한 예는 사용자 역할의 적용을 설명하는 데 충분한 지침만 제공합니다.

이러한 사용 사례의 대상 청중은 서비스 관리자로 간주되기도 하는 클라우드 관리자입니다.

사용 사례는 서로를 기반으로 구축됩니다. 사용 사례 3으로 바로 이동할 준비가 되었다면 사용 사례 1 및 2를 검토하여 지정된 방법으로 역할을 구성하는 이유를 보다 잘 이해할 수 있습니다.

사용 사례의 목적은 인프라 구성, 프로젝트 관리, 클라우드 템플릿 생성 및 배포 작업에 대한 자세한 정보를 제공하지 않고 사용자 역할을 시연하는 것입니다.

시작하기 전에 vRealize Automation 콘솔에서 클라우드 관리자가 구성한 사용자 역할의 수준을 이해해야 합니다.

#### ■ 조직 역할

조직 역할은 콘솔에 액세스할 수 있는 사용자를 제어합니다.

조직 소유자는 서비스의 모든 사용자에게 최소한 조직 멤버 역할이 할당되어 있는지 확인해야 합니다.

역할	설명
조직 소유자	관리자는 사용자를 추가하고, 사용자의 역할을 변경하고, 조직에서 사용자를 제거할 수 있습니다. 소유자는 사용자가 액세스할 수 있는 서비스를 관리합니다.
조직 멤버	일반 사용자는 조직 콘솔에 로그인할 수 있습니다. 서비스에 액세스하려면 조직 소유자가 사용자에게 서비스 역할을 할당해야 합니다.

#### ■ 서비스 역할

서비스 역할은 할당된 서비스에 액세스할 수 있는 사용자를 제어합니다.

조직 소유자는 서비스에 액세스해야 하는 사용자에게 적절한 역할이 할당되었는지 확인해야 합니다. 역할을 사용하여 사용자가 각 서비스에서 수행할 수 있는 작업을 제어합니다.

**표 3-11. vRealize Automation Cloud Assembly 서비스 역할 설명**

역할	설명
Cloud Assembly 관리자	전체 사용자 인터페이스 및 API 리소스에 대해 읽기 및 쓰기 권한이 있는 사용자입니다. 클라우드 계정 추가, 새 프로젝트 생성 및 프로젝트 관리자 할당 등을 볼 수 있고 수행할 수 있는 유일한 사용자 역할입니다.
Cloud Assembly 사용자	Cloud Assembly 관리자 역할이 없는 사용자입니다. vRealize Automation Cloud Assembly 프로젝트에서는 관리자가 프로젝트에 사용자를 프로젝트 멤버, 관리자 또는 뷰어로 추가합니다. 관리자는 프로젝트 관리자를 추가할 수도 있습니다.
Cloud Assembly 뷰어	정보를 볼 수 있는 읽기 권한이 있지만 값을 생성, 업데이트 또는 삭제할 수 없는 사용자입니다. 모든 프로젝트에서 읽기 전용 역할입니다. 뷰어 역할이 있는 사용자는 관리자가 사용할 수 있는 모든 정보를 볼 수 있습니다. 이들을 프로젝트 관리자 또는 프로젝트 멤버로 만들지 않는 한 아무 작업도 수행할 수 없습니다. 사용자가 프로젝트와 연관된 경우에는 역할과 관련된 권한이 있습니다. 프로젝트 뷰어는 관리자 또는 멤버 역할이 수행하는 방식으로 권한을 확장하지 않습니다.

**표 3-12. Service Broker 서비스 역할 설명**

역할	설명
Service Broker 관리자	전체 사용자 인터페이스 및 API 리소스에 대해 읽기/쓰기 액세스 권한을 갖고 있어야 합니다. 새 프로젝트 생성 및 프로젝트 관리자 할당을 비롯한 모든 작업을 수행할 수 있는 유일한 사용자 역할입니다.
Service Broker 사용자	vRealize Automation Service Broker 관리자 역할이 없는 모든 사용자입니다. vRealize Automation Service Broker 프로젝트에서는 관리자가 프로젝트에 사용자를 프로젝트 멤버, 관리자 또는 뷰어로 추가합니다. 관리자는 프로젝트 관리자를 추가할 수도 있습니다.
Service Broker 뷰어	정보를 볼 수 있는 읽기 권한이 있지만 값을 생성, 업데이트 또는 삭제할 수 없는 사용자입니다. 뷰어 역할이 있는 사용자는 관리자가 사용할 수 있는 모든 정보를 볼 수 있습니다. 이들을 프로젝트 관리자 또는 프로젝트 멤버로 만들지 않는 한 아무 작업도 수행할 수 없습니다. 사용자가 프로젝트와 연관된 경우에는 역할과 관련된 권한이 있습니다. 프로젝트 뷰어는 관리자 또는 멤버 역할이 수행하는 방식으로 권한을 확장하지 않습니다.

표 3-13. Code Stream 서비스 역할 설명

역할	설명
Code Stream 관리자	전체 사용자 인터페이스 및 API 리소스에 대해 읽기 및 쓰기 권한이 있는 사용자입니다. 이 사용자 역할은 프로젝트 생성, 끝점 통합, 트리거 추가, 파이프라인 및 사용자 지정 대시보드 생성, 끝점 및 변수를 제한된 리소스로 표시, 제한된 리소스를 사용하는 파이프라인 실행, 이 파이프라인을 vRealize Automation Service Broker에 게시하도록 요청하는 등의 작업을 포함한 모든 작업을 보고 수행할 수 있는 유일한 사용자 역할입니다.
Code Stream 개발자	파이프라인을 사용할 수 있지만 제한된 끝점 또는 변수로 작업할 수 없는 사용자입니다. 파이프라인에 제한된 끝점이나 변수가 포함된 경우 이 사용자는 제한된 끝점이나 변수를 사용하는 파이프라인 작업에 대한 승인을 받아야 합니다.
Code Stream Executor	파이프라인을 실행하고 사용자 작업을 승인하거나 거부할 수 있는 사용자입니다. 이 사용자는 파이프라인 실행을 재개, 일시 중지 및 취소할 수 있지만 파이프라인을 수정할 수는 없습니다.
Code Stream 사용자	vRealize Automation Code Stream에 액세스할 수 있지만 vRealize Automation Code Stream에서 다른 권한은 없는 사용자입니다.
Code Stream 뷰어	파이프라인, 끝점, 파이프라인 실행, 대시보드를 볼 수 있는 읽기 권한이 있지만 생성, 업데이트 또는 삭제할 수 없는 사용자입니다. 서비스 뷰어 역할도 있는 사용자는 관리자가 사용할 수 있는 모든 정보를 볼 수 있습니다. 이들을 프로젝트 관리자 또는 프로젝트 멤버로 만들지 않는 한 아무 작업도 수행할 수 없습니다. 사용자가 프로젝트와 연관된 경우에는 역할과 관련된 권한이 있습니다. 프로젝트 뷰어는 관리자 또는 멤버 역할이 수행하는 방식으로 권한을 확장하지 않습니다.

#### ■ 프로젝트 멤버 자격 역할

프로젝트 멤버 자격은 사용할 수 있는 인프라 리소스 및 클라우드 템플릿을 결정합니다.

프로젝트 멤버 자격은 서비스 관리자 역할이 있는 사용자가 서비스에 정의합니다. 서비스 관리자는 각 프로젝트에서 하나 이상의 프로젝트에 액세스해야 하는 사용자에게 적절한 프로젝트 역할이 할당되었는지 확인해야 합니다.

표 3-14. 프로젝트 역할

역할	설명
프로젝트 관리자	프로젝트 관리자는 자체 프로젝트를 관리하고, 해당 프로젝트와 관련된 클라우드 템플릿을 생성 및 배포하고, 모든 프로젝트 멤버에 대한 프로젝트 배포를 관리할 수 있습니다.
프로젝트 멤버	프로젝트 멤버는 자체 프로젝트와 연결된 클라우드 템플릿을 생성 및 배포하고 자체 배포를 관리하며 공유 배포를 관리할 수 있습니다.
프로젝트 뷰어	프로젝트 뷰어는 해당 프로젝트 리소스, 클라우드 템플릿 및 배포에 대해 읽기 전용 액세스 권한이 있는 프로젝트의 멤버입니다.

#### ■ 사용자 지정 역할

사용자 지정 역할은 멤버 및 뷰어 역할을 구체화하기 위해 vRealize Automation Cloud Assembly에서 생성됩니다.

이러한 사용 사례에서 제공된 절차는 사용자 역할을 강조하기 위한 것으로, vRealize Automation 설정을 위한 세부적이거나 최종적인 절차는 아닙니다.

역할을 구성하는 동안, API 작업을 실행하는 사용자에게 여기에서 할당한 역할이 적용된다는 점에 유의하십시오.

### 사전 요구 사항

- 조직 소유자 역할이 있는지 확인합니다. 콘솔에 로그인할 때 **ID 및 액세스 관리** 탭이 표시되어야 합니다. 그렇지 않은 경우 조직 소유자에게 문의합니다.
  - 사용자 vRealize Automation에 추가되었는지 확인합니다.
- vRealize Automation을 설치할 때 Active Directory 사용자가 프로세스의 일부로 추가됩니다.
- 다양한 역할에 대한 자세한 작업 및 역할 목록은 [vRealize Automation의 조직 및 서비스 사용자 역할](#)의 내용을 참조하십시오.

### 절차

#### 1 사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정

vRealize Automation 클라우드 관리자는 인프라 리소스에 대한 액세스 및 예산을 관리해야 합니다. 자신을 비롯해 다른 두 사람을 관리자로 추가합니다. 이 소규모 팀은 인프라를 생성할 수 있으며, 클라우드 템플릿을 사용하는 팀의 비즈니스 목표와 일치하는 클라우드 템플릿을 개발할 수 있습니다. 그런 다음 자신과 소규모 관리자 팀에서 관리자가 아닌 소비자를 위한 클라우드 템플릿을 배포합니다. 관리자가 아닌 사람의 vRealize Automation 액세스는 허용하지 않습니다.

#### 2 사용자 역할 사용 사례 2: 대규모 개발자 팀 및 카탈로그를 지원하는 vRealize Automation 사용자 역할 설정

vRealize Automation 조직 소유자는 인프라 리소스에 대한 액세스 및 예산을 관리해야 합니다. 소비자에게 제공할 준비가 될 때까지 다른 프로젝트의 템플릿을 반복적으로 생성하고 배포하는 클라우드 템플릿 개발자 팀이 있습니다. 그런 다음 배포 가능한 리소스를 카탈로그의 소비자에게 전달합니다.

#### 3 사용자 역할 사용 사례 3: vRealize Automation 사용자 지정 사용자 역할을 설정하여 시스템 역할을 구체화하기

vRealize Automation 조직 소유자 또는 서비스 관리자는 조직 및 서비스 시스템 역할을 사용하여 사용자 액세스를 관리합니다. 하지만, 선택한 사용자에게 사용자 지정 역할을 생성하고 작업을 수행하거나 시스템 역할 외부에 있는 콘텐츠를 보려고 할 수도 있습니다.

### 사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정

vRealize Automation 클라우드 관리자는 인프라 리소스에 대한 액세스 및 예산을 관리해야 합니다. 자신을 비롯해 다른 두 사람을 관리자로 추가합니다. 이 소규모 팀은 인프라를 생성할 수 있으며, 클라우드 템플릿을 사용하는 팀의 비즈니스 목표와 일치하는 클라우드 템플릿을 개발할 수 있습니다. 그런 다음 자신과

소규모 관리자 팀에서 관리자가 아닌 소비자를 위한 클라우드 템플릿을 배포합니다. 관리자가 아닌 사람의 vRealize Automation 액세스는 허용하지 않습니다.

이 사용 사례에서 귀하는 조직 소유자이며 서비스 관리자 역할을 수행하는 소규모 팀이 있습니다.

다음 절차에서는 한 명의 사용자를 따라 프로세스 전체를 진행합니다. 여러 사용자에 대해 각 단계를 수행할 수 있습니다.

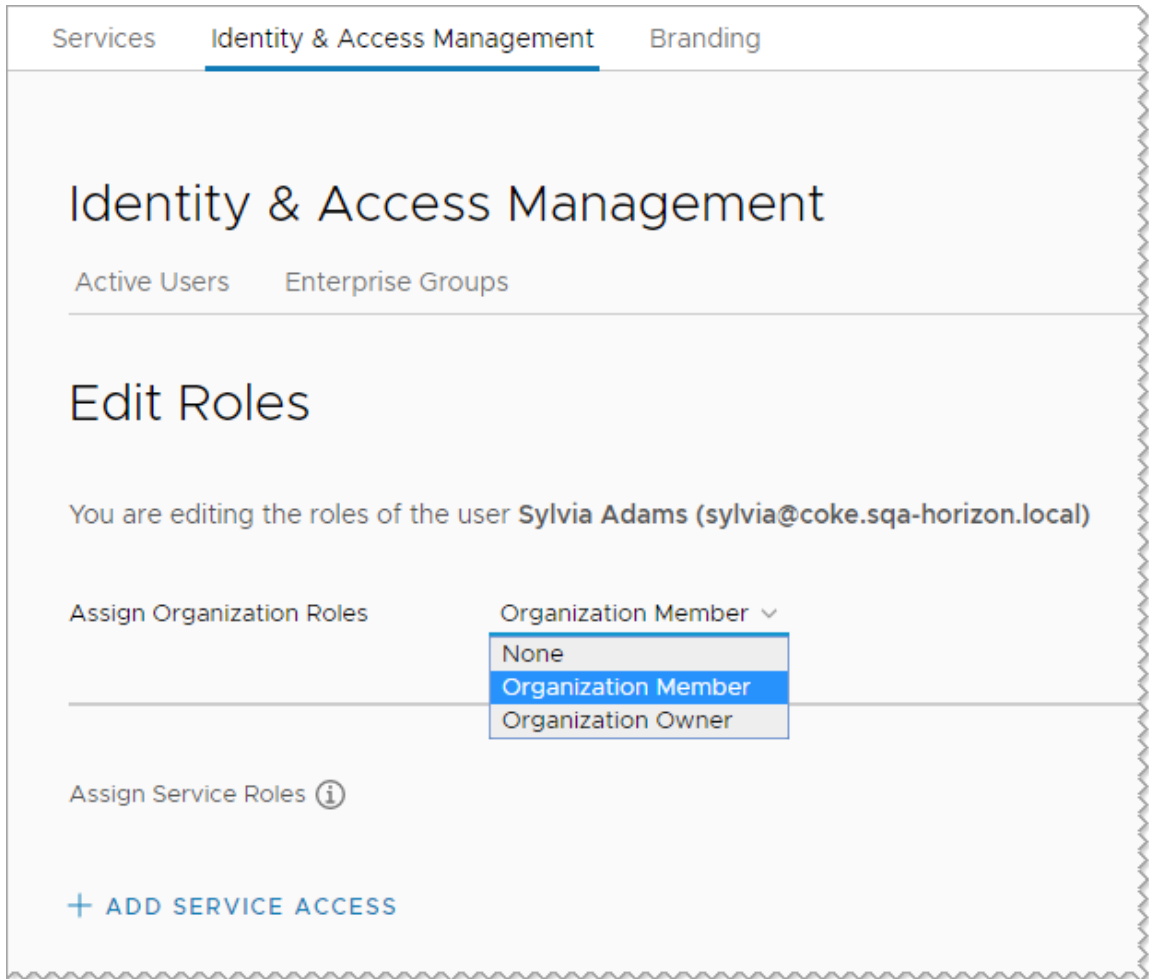
### 사전 요구 사항

- 사용 사례 소개에 규정되어 있는 모든 사전 요구 사항을 충족하는지 확인합니다. [사용 사례: 사용자 역할을 통해 vRealize Automation에서 액세스를 제어하는 방법](#)의 내용을 참조하십시오.

### 절차

- 1 조직 역할을 할당합니다. **ID 및 액세스 관리**를 클릭합니다.
  - a vRealize Automation 콘솔에 로그인합니다.
  - b **ID 및 액세스 관리**를 클릭합니다.

- c 사용자 이름을 선택하고 **역할 편집**을 클릭합니다.
- d **조직 역할 할당** 드롭다운 메뉴에서 **조직 멤버**를 선택합니다.



조직 멤버 역할은 사용자가 콘솔 그리고 사용자에게 추가된 모든 서비스에 액세스할 수 있도록 합니다. 조직 사용자를 관리할 수는 없습니다.

이 사용자에 대해 [역할 편집] 페이지를 열어 두고 다음 단계를 계속 진행합니다.

- 2 이 시나리오에서 자신을 비롯해 다른 관리자 하나 또는 둘에 Cloud Assembly 관리자 역할을 할당합니다.

서비스 관리자 역할은 인프라, 프로젝트, 클라우드 템플릿 및 배포를 추가, 편집, 삭제할 수 있는 전체 권한을 갖습니다. 한 사람에 대해 관리자 역할을 정의하고 다른 한 사람에 대해 사용자 역할을 정의하는 것은 시나리오 2에서 다룹니다. 이 예에서는 Sylvia를 사용합니다.

- a 서비스 액세스 추가를 클릭합니다.
- b 다음 값을 사용하여 사용자를 구성합니다.

서비스	역할
vRealize Automation Cloud Assembly	vRealize Automation Cloud Assembly 관리자

- 3 리소스를 그룹화하고 다른 비즈니스 그룹에 대한 리소스 청구를 관리하는 데 사용하는 프로젝트를 Cloud Assembly에서 생성합니다.

- a 콘솔에서 **서비스** 탭을 클릭한 다음 **Cloud Assembly**를 클릭합니다.
- b **인프라 > 프로젝트 > 새 프로젝트**를 선택합니다.

이 사용자 역할 사용 사례는 완전히 정의된 시스템을 생성하는 것이 아니라 사용자 역할을 구현할 수 있는 방법의 예를 제공하는 데 중점을 두었습니다.

인프라 구성에 대한 자세한 내용은 **리소스 인프라 구축**을 참조하십시오. 프로젝트에 대한 자세한 내용은 **프로젝트 추가 및 관리**를 참조하십시오.

- c 프로젝트 이름으로 **WebAppTeam**을 입력합니다.
- d **사용자**를 클릭한 후 **사용자 추가**를 클릭합니다.

- e 인프라 및 클라우드 템플릿을 구축하고 관리하는 데 도움을 줄 수 있는 개인의 이메일 주소를 입력합니다.

예를 들어 `tony@mycompany.com`, `syliya@mycompany.com`과 같이 입력합니다.

- f **역할 할당** 드롭다운 메뉴에서 **관리자**를 선택합니다.

vRealize Automation Cloud Assembly 관리자인 이 두 사용자에게는 클라우드 계정, 인프라 및 모든 프로젝트에 대한 관리자 액세스 권한이 이미 있습니다. 이 단계는 이후 시나리오에서 사용되는 역할을 이해하는 데 도움이 됩니다. 이후 시나리오에서는 권한이 다른 프로젝트 관리자 및 프로젝트 멤버 역할을 정의합니다.

- g **프로비저닝** 탭을 클릭하고 하나 이상의 클라우드 영역을 추가합니다.

다른 미리 알림. 이 사용 사례는 사용자 역할에 대한 것입니다.

#### 4 WebAppTeam 프로젝트를 테스트할 수 있도록 간단한 클라우드 템플릿을 개발합니다.

이 클라우드 템플릿 섹션은 간략하게 설명되어 있습니다. 클라우드 템플릿을 생성하는 방법이 아닌 프로젝트에서 정의한 사용자 및 사용자 역할에 중점을 두고 있습니다.

- a **클라우드 템플릿 > 새로 만들기**를 선택합니다.
- b 새 클라우드 템플릿 이름으로 **WebApp**을 입력합니다.
- c **프로젝트**로는 **WebAppTeam**을 선택합니다.

New Cloud Template

Name \* WebApp

Description

Project \* WebAppTeam

Cloud template sharing in Service Broker

☒ Share only with this project

☐ Allow an administrator to share with any project in this organization

CANCEL CREATE

- d **이 프로젝트와만 공유**를 선택합니다.

이 설정을 사용하면 프로젝트 멤버만 클라우드 템플릿을 사용할 수 있습니다. 다른 팀에 클라우드 템플릿을 제공할 준비가 되면 [관리자가 이 조직의 모든 프로젝트와 공유하도록 허용]을 선택할 수 있습니다. 클라우드 템플릿을 다른 프로젝트와 공유한다는 것은 동일한 기본 템플릿의 중복 인스턴스를 유지할 필요가 없다는 것을 의미합니다. 카탈로그 소비자가 운영 인프라 리소스에 배포할 수 있도록 클라우드 템플릿을 개발 프로젝트에서 운영 프로젝트로 이동할 수 있습니다.

- e **생성**을 클릭합니다.
- f 클라우드 템플릿 디자이너에서 **클라우드 애그노스틱 > 시스템** 구성 요소를 캔버스로 끌어 옵니다.

클라우드 템플릿 구성에 대한 자세한 내용은 [배포 설계](#)를 참조하십시오.

g 배포를 클릭합니다.

h 소비자에게 제공할 준비가 될 때까지 클라우드 템플릿에서 계속 반복합니다.

i 버전을 클릭하고 클라우드 템플릿의 릴리스와 버전을 지정합니다.

5 가장 일반적인 방법을 사용하여 사용자에게 로그인 정보를 전송합니다.

## 결과

이 사용 사례에서는 두 동료 조직 멤버를 만들었습니다. 그런 다음, Sylvia를 vRealize Automation Cloud Assembly 관리자로 만들었습니다. Tony는 WebApp 프로젝트 관리자로 만들었습니다. 이 사용자 역할 구성은 셀프 서비스 액세스 또는 카탈로그를 제공하는 대신 소비자에게 배포된 애플리케이션을 제공하는 소규모 팀에만 적용됩니다.

## 사용자 역할 사용 사례 2: 대규모 개발자 팀 및 카탈로그를 지원하는 vRealize Automation 사용자 역할 설정

vRealize Automation 조직 소유자는 인프라 리소스에 대한 액세스 및 예산을 관리해야 합니다. 소비자에게 제공할 준비가 될 때까지 다른 프로젝트의 템플릿을 반복적으로 생성하고 배포하는 클라우드 템플릿 개발자 팀이 있습니다. 그런 다음 배포 가능한 리소스를 카탈로그의 소비자에게 전달합니다.

이 사용 사례에서는 사용 사례 1이 관리자 전용 사용 사례라는 것을 알고 있다고 가정합니다. 이제 더 많은 팀과 더 큰 목표를 지원하도록 시스템을 확장하려고 합니다.

- 개발자가 개발 중에 자체 애플리케이션 클라우드 템플릿을 생성하고 배포할 수 있도록 합니다. 자신을 관리자로 추가한 다음 서비스 사용자 역할과 서비스 뷰어 역할을 모두 가진 사용자를 더 추가합니다. 다음으로 사용자를 프로젝트 멤버로 추가합니다. 프로젝트 멤버는 자체 클라우드 템플릿을 개발하고 배포할 수 있습니다.
- 개발자가 아닌 사용자가 배포할 수 있도록 클라우드 템플릿을 카탈로그에 게시합니다. 이제 Service Broker에 대한 사용자 역할을 할당합니다. Service Broker는 클라우드 템플릿 소비자를 위한 카탈로그를 제공합니다. 또한 이를 사용하여 리스 및 사용 권한을 포함한 정책을 생성할 수 있지만 해당 기능은 이 사용자 역할 사용 사례에 속하지 않습니다.

## 사전 요구 사항

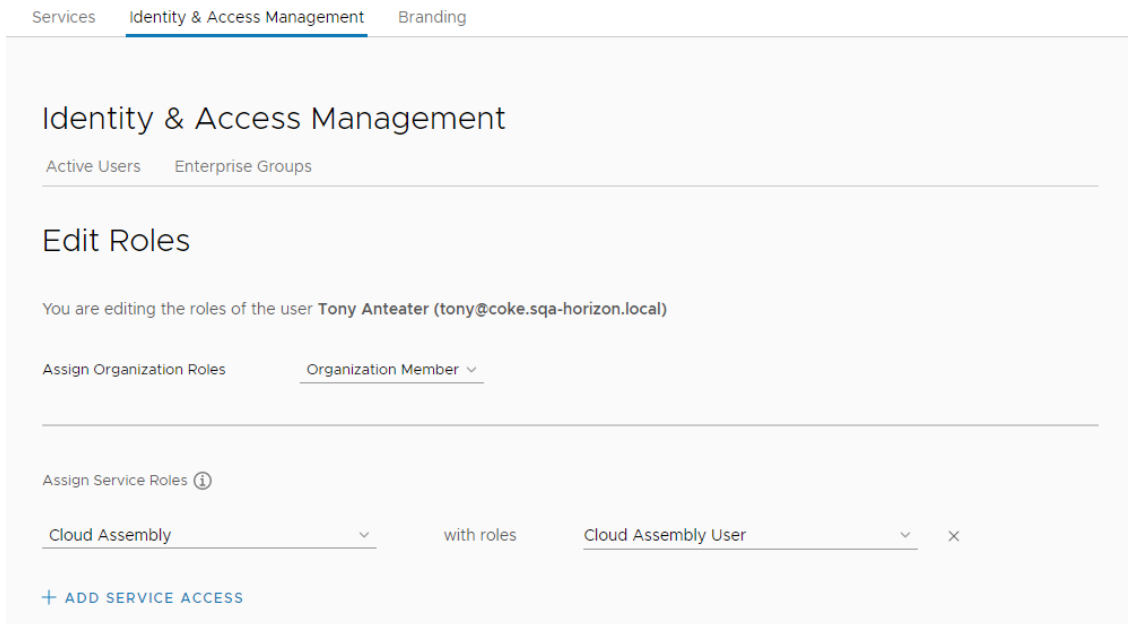
- 첫 번째 사용 사례를 검토합니다. [사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정](#)의 내용을 참조하십시오.
- 사용자에게 부여하려는 사용 권한에 따라 다음 사용자를 식별합니다.
  - vRealize Automation Cloud Assembly 사용자 및 뷰어가 되는 클라우드 템플릿 개발자
  - vRealize Automation Service Broker 관리자
  - vRealize Automation Service Broker 사용자로 카탈로그 소비자가 되는 개발자가 아닌 사용자

## 절차

- 1 조직 멤버 역할을 클라우드 템플릿 개발자에게 할당합니다.

지침이 필요한 경우 사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정을 참조하십시오.

- 2 vRealize Automation Cloud Assembly 서비스 멤버 역할을 클라우드 템플릿 개발자에게 할당합니다.
  - a 서비스 액세스 추가를 클릭합니다.



- b 다음 값을 사용하여 사용자를 구성합니다.

서비스	역할
vRealize Automation Cloud Assembly	vRealize Automation Cloud Assembly 사용자
vRealize Automation Cloud Assembly	vRealize Automation Cloud Assembly 뷰어

이 사용 사례에서 개발자는 배포 가능한 클라우드 템플릿을 구축하고 있는지 확인하기 위해 인프라를 볼 수 있어야 합니다. 다음 단계에서 프로젝트 관리자 및 프로젝트 멤버로 할당되는 사용자는 인프라를 볼 수 없습니다. 서비스 뷰어는 인프라가 구성된 방식을 볼 수 있지만 변경할 수는 없습니다. 클라우드 관리자는 제어권을 유지하면서 해당 사용자에게 클라우드 템플릿 개발에 필요한 정보에 대한 액세스 권한을 제공해야 합니다.

- 3 vRealize Automation Cloud Assembly에서 리소스 사용자를 그룹화하는 데 사용하는 프로젝트를 생성합니다.

이 사용 사례에서는 두 개의 프로젝트를 생성합니다. 첫 번째 프로젝트는 PersonnelAppDev이고 두 번째 프로젝트는 PayrollAppDev입니다.

- a 콘솔에서 서비스 탭을 클릭한 다음 Cloud Assembly를 클릭합니다.

- b 인프라 > 프로젝트 > 새 프로젝트를 선택합니다.

- c 이름으로 **PersonnelAppDev**를 입력합니다.
- d **사용자**를 클릭한 후 **사용자 추가**를 클릭합니다.
- e 프로젝트 멤버를 추가하고 프로젝트 관리자를 할당합니다.

프로젝트 역할	설명
프로젝트 사용자	프로젝트 멤버는 프로젝트에서 기본 개발자 사용자 역할입니다. 프로젝트는 클라우드 템플릿을 배포하여 개발 작업을 테스트할 준비가 되었을 때 사용할 수 있는 클라우드 리소스를 결정합니다.
프로젝트 관리자	프로젝트 관리자는 프로젝트에 대한 사용자를 추가하고 제거하여 개발자를 지원합니다. 프로젝트를 삭제할 수도 있습니다. 프로젝트를 생성하려면 서비스 관리자 권한이 있어야 합니다.

- f 프로젝트 멤버로 추가할 사용자의 경우 각 사용자의 이메일 주소를 쉼표로 구분하여 입력하고 **역할 할당** 드롭다운 메뉴에서 **사용자**를 선택합니다.

예를 들어 tony@mycompany.com,sylvia@mycompany.com과 같이 입력할 수 있습니다.

- g 지정된 관리자의 경우 **역할 할당** 드롭다운 메뉴에서 **관리자**를 선택하고 필요한 이메일 주소를 제공합니다.
- h **프로비저닝** 탭을 클릭하고 하나 이상의 클라우드 영역을 추가합니다.  
이 프로젝트에 속하는 클라우드 템플릿 개발자가 템플릿을 배포하는 경우 클라우드 영역에서 사용할 수 있는 리소스에 배포됩니다. 클라우드 영역 리소스가 프로젝트 개발 팀 템플릿의 요구 사항과 일치하는지 확인해야 합니다.
- i 프로세스를 반복하여 필요한 사용자와 관리자가 있는 PayrollAppDev 프로젝트를 추가합니다.

- 4 서비스 사용자에게 필요한 로그인 정보를 제공하고 각 프로젝트의 멤버가 다음 작업을 수행할 수 있는지 확인합니다.
  - a vRealize Automation Cloud Assembly를 엽니다.
  - b 모든 프로젝트에서 인프라를 참조합니다.
  - c 멤버로 속해 있는 프로젝트에 대한 클라우드 템플릿을 생성합니다.
  - d 프로젝트에 정의된 클라우드 영역 리소스에 클라우드 템플릿을 배포합니다.
  - e 배포를 관리합니다.
- 5 조직 멤버 역할을 클라우드 템플릿 개발자에게 할당합니다.  
 지침이 필요한 경우 [사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정](#)을 참조하십시오.
- 6 작업에 따라 카탈로그 관리자, 카탈로그 소비자 및 클라우드 템플릿 개발자에게 역할을 할당합니다.
  - a **서비스 액세스 추가**를 클릭합니다.
  - b 다음 값을 사용하여 카탈로그 관리자를 구성합니다.  
 이 역할은 클라우드 관리자이거나 애플리케이션 개발 팀의 다른 사람일 수 있습니다.

서비스	역할
vRealize Automation Service Broker	vRealize Automation Service Broker 관리자

- c 다음 값을 사용하여 클라우드 템플릿 소비자를 구성합니다.

서비스	역할
vRealize Automation Service Broker	vRealize Automation Service Broker 사용자

### Identity & Access Management

Active Users   Enterprise Groups

## Edit Roles

You are editing the roles of the user **Gloria Martinez (gloria@coke.sqa-horizon.local)**

Assign Organization Roles   Organization Member ▾

---

Assign Service Roles ⓘ

Service Broker ▾

with roles

Service Broker User ▾

×

[+ ADD SERVICE ACCESS](#)

- d 다음 값을 사용하여 클라우드 템플릿 개발자를 구성합니다.

서비스	역할
Cloud AssemblyvRealize Automation Cloud Assembly	vRealize Automation Cloud Assembly 사용자

- 7 vRealize Automation Cloud Assembly에서 리소스 및 사용자를 그룹화하는 데 사용하는 프로젝트를 생성합니다.

이 사용 사례에서는 두 개의 프로젝트를 생성합니다. 첫 번째 프로젝트는 **PersonnelAppDev**이고 두 번째 프로젝트는 **PayrollAppDev**입니다.

지침이 필요한 경우 [사용자 역할 사용 사례 2: 대규모 개발자 팀 및 카탈로그를 지원하는 vRealize Automation 사용자 역할 설정](#)을 참조하십시오.

- 8 각 프로젝트 팀에 대한 클라우드 템플릿을 생성하고 릴리스합니다.

지침이 필요한 경우 [사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정](#)을 참조하십시오.

- 9 vRealize Automation Cloud Assembly 클라우드 템플릿을 vRealize Automation Service Broker로 가져옵니다.

vRealize Automation Service Broker 관리자 역할이 있는 사용자로 로그인해야 합니다.

- vRealize Automation Service Broker 관리자 역할이 있는 사용자로 로그인합니다.
- 콘솔에서 vRealize Automation Service Broker를 클릭합니다.

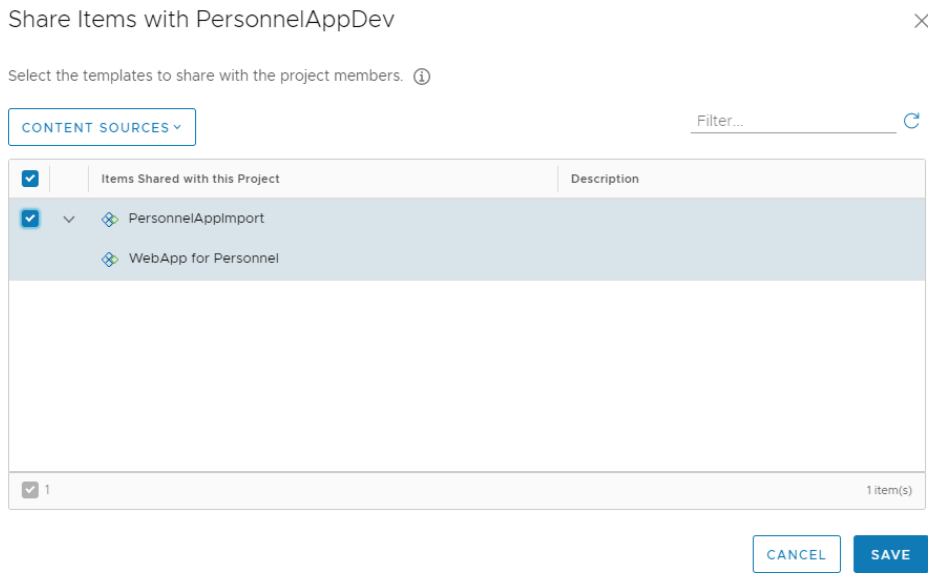
- c **컨텐츠 및 정책 > 컨텐츠 소스**를 선택하고 **새로 만들기**를 클릭합니다.

The screenshot shows the 'New Content Source' configuration page. The left-hand navigation pane is open, highlighting 'Content Sources'. The main area contains the following fields and sections:

- Type \***: VMware Cloud Templates
- Name \***: PersonnelAppImport
- Description**: (Empty text box)
- Get cloud templates from**:
  - Source project \***: PersonnelAppDev
  - VALIDATE** button
- Deploy cloud templates to**:
  - Cloud templates will be deployed to cloud zones in projects they are shared with.
  - CREATE & IMPORT** button
  - CANCEL** button

- d **Cloud Assembly 클라우드 템플릿**을 선택합니다.
- e 이름으로 **PersonnelAppImport**를 입력합니다.
- f **소스 프로젝트** 드롭다운 메뉴에서 **PersonnelAppDev**를 선택하고 **검증**을 클릭합니다.
- g 소스가 검증되면 **생성 및 가져오기**를 클릭합니다.
- h **PayrollAppImport**를 컨텐츠 소스 이름으로 사용하여 **PayrollAppDev**대해 반복합니다.
- 10 가져온 클라우드 템플릿을 프로젝트와 공유합니다.
- 클라우드 템플릿은 프로젝트와 이미 연결되어 있지만 카탈로그에서 사용할 수 있도록 vRealize Automation Service Broker에서 공유합니다.
- a vRealize Automation Service Broker 관리자 역할이 있는 사용자로 계속 작업합니다.
- b vRealize Automation Service Broker에서 **컨텐츠 및 정책 > 컨텐츠 공유**를 선택합니다.
- c 카탈로그에서 클라우드 템플릿을 배포할 수 있어야 하는 사용자가 포함된 **PersonnelAppDev** 프로젝트를 선택합니다.

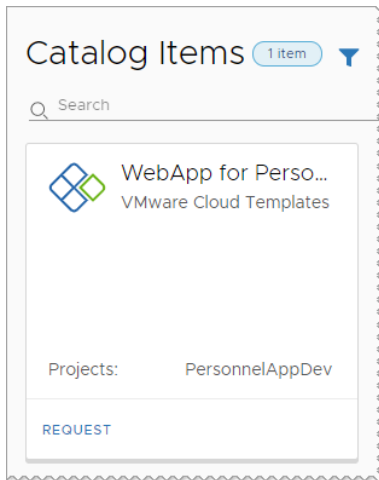
- d 항목 추가를 클릭한 다음 프로젝트 멤버와 공유할 PersonnelApp 클라우드 템플릿을 선택합니다.



- e 저장을 클릭합니다.

- 11 vRealize Automation Service Broker 카탈로그에서 프로젝트 멤버에 대해 클라우드 템플릿을 사용할 수 있는지 확인합니다.

- a 프로젝트 멤버가 로그인하고 카탈로그 탭을 클릭하도록 요청합니다.

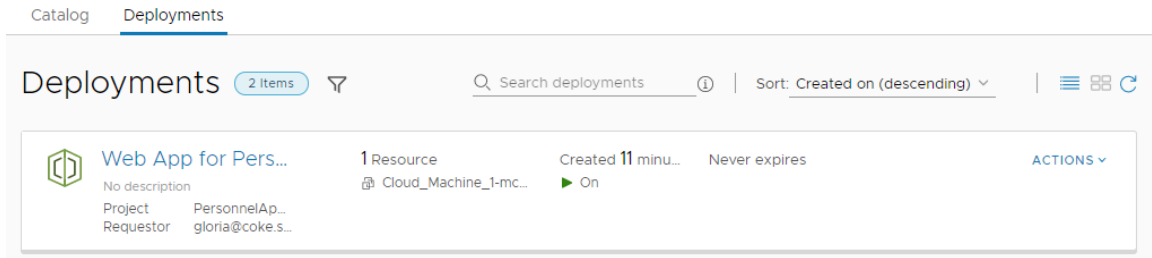


- b PersonnelApp 클라우드 템플릿 카드에서 [요청]을 클릭합니다.

- c 양식 작성을 완료하고 제출을 클릭합니다.

12 프로젝트 멤버가 배포 프로세스를 모니터링할 수 있는지 확인합니다.

- a 프로젝트 멤버가 **배포** 탭을 클릭하고 해당 프로비저닝 요청을 찾으도록 요청합니다.



- b 클라우드 템플릿을 배포할 때 요청하는 사용자가 애플리케이션에 액세스할 수 있는지 확인합니다.

13 추가 프로젝트에 대해 이 프로세스를 반복합니다.

## 결과

이 사용 사례에서는 개발자에게 클라우드 템플릿 개발을 위임할 필요를 인식하고 조직 멤버를 더 추가합니다. 그들을 vRealize Automation Cloud Assembly 사용자로 설정했습니다. 그런 다음 클라우드 템플릿을 생성하고 배포할 수 있도록 해당 멤버를 관련 프로젝트의 멤버로 설정했습니다. 프로젝트 멤버는 사용자가 관리하는 인프라를 보거나 변경할 수 없지만 자신들이 설계하는 인프라의 제약 조건을 이해할 수 있도록 전체 서비스 뷰어 사용 권한을 제공했습니다.

이 사용 사례에서는 vRealize Automation Service Broker 관리자 및 사용자를 포함하여 다양한 역할을 가진 사용자를 구성합니다. 그런 다음 개발자가 아닌 사용자에게 vRealize Automation Service Broker 카탈로그를 제공합니다.

## 다음에 수행할 작업

사용자 지정 역할을 정의하고 사용자에게 할당하는 방법에 대한 자세한 내용은 [사용자 역할 사용 사례 3: vRealize Automation 사용자 지정 사용자 역할을 설정하여 시스템 역할을 구체화하기](#) 항목을 참조하십시오.

## 사용자 역할 사용 사례 3: vRealize Automation 사용자 지정 사용자 역할을 설정하여 시스템 역할을 구체화하기

vRealize Automation 조직 소유자 또는 서비스 관리자는 조직 및 서비스 시스템 역할을 사용하여 사용자 액세스를 관리합니다. 하지만, 선택한 사용자에게 사용자 지정 역할을 생성하고 작업을 수행하거나 시스템 역할 외부에 있는 콘텐츠를 보려고 할 수도 있습니다.

이 시나리오에서는 사용 사례 2에 정의된 서비스 사용자 및 뷰어, 프로젝트 멤버 및 뷰어 역할을 이해하고 있다고 가정합니다. 이러한 역할은 사용 사례 1에서 사용된 서비스 및 프로젝트 관리자 역할보다 제한적입니다. 이제 일부 사용자가 일부 기능에 대해서는 전체 관리 권한을 가지지만 다른 기능에 대해서는 보기 권한을 가지고, 또다른 기능 집합에 대해서는 보기 권한조차 없도록 하려는 몇 가지 로컬 사용 사례를 식별했습니다. 사용자 지정 역할을 사용하여 이러한 사용 권한을 정의합니다.

이 사용 사례는 세 가지 가능한 로컬 사용 사례를 기반으로 합니다. 이 절차에서는 다음 사용자 지정 역할에 대한 사용 권한을 생성하는 방법을 보여줍니다.

- 제한된 인프라 관리자. 서비스 관리자가 아닌 일부 서비스 사용자에게 보다 광범위한 인프라 사용 권한이 있기를 원합니다. 관리자는 이러한 사용자가 클라우드 영역, 이미지, 버전을 설정할 때 도와주기를 원합니다. 또한 검색된 리소스를 온보딩하고 관리할 수 있기를 원합니다. 하지만 클라우드 계정 또는 통합을 추가할 수 없으며 이러한 끝점에 대한 인프라를 정의할 수만 있어야 합니다.
- 확장성 개발자. 일부 서비스 사용자에게 프로젝트 팀 및 기타 프로젝트에서 클라우드 템플릿 개발의 일환으로 확장성 작업 및 구독을 사용할 수 있는 전체 권한이 있기를 원합니다. 또한 여러 프로젝트에 대한 사용자 지정 리소스 유형 및 사용자 지정 작업을 개발할 수 있어야 합니다.
- XaaS 개발자. 일부 서비스 사용자가 여러 프로젝트에 대한 사용자 지정 리소스 유형 및 사용자 지정 작업을 개발할 수 있는 전체 권한을 갖기를 원합니다.
- 배포 문제 해결자. 프로젝트 관리자에게 실패한 배포에 대해 문제를 해결하고 근본 원인 분석을 수행하는 데 필요한 사용 권한이 있기를 원합니다. 이미지 및 버전 매핑과 같이 비파괴적이거나 비용이 낮은 범주에 대한 관리 권한을 부여합니다. 또한 프로젝트 관리자에게 실패한 배포 문제 해결 역할의 일부로 승인 및 2일차 정책을 설정할 수 있는 사용 권한이 있기를 원합니다.

## 사전 요구 사항

- [vRealize Automation 사용자 역할이란?](#)에서 vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker 서비스 역할 및 프로젝트 역할 포를 검토합니다. 각 서비스 사용자 역할이 해당 서비스에서 보고 수행할 수 있는 작업을 이해해야 합니다.
- [vRealize Automation의 사용자 지정 사용자 역할](#) 설명을 검토하여 사용자의 사용 권한을 구체화하는 방법을 숙지합니다.
- 조직 역할과 서비스 관리자 역할을 이해할 수 있도록 첫 번째 사용 사례를 검토합니다. [사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정](#)의 내용을 참조하십시오.
- 두 번째 사용 사례를 검토하여 서비스 사용자 및 프로젝트 멤버 역할을 이해합니다. [사용자 역할 사용 사례 2: 대규모 개발자 팀 및 카탈로그를 지원하는 vRealize Automation 사용자 역할 설정](#)의 내용을 참조하십시오.
- vRealize Automation Service Broker 관련 내용을 숙지합니다. [카탈로그에 콘텐츠 추가](#)를 참조하십시오.

## 절차

- 1 조직 멤버 역할을 클라우드 템플릿 개발자에게 할당합니다.

지침이 필요한 경우 [사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정](#)를 참조하십시오.

- 2 클라우드 템플릿 개발자 및 카탈로그 소비자에게 vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker 서비스 역할을 할당합니다.

지침이 필요한 경우 [사용자 역할 사용 사례 2: 대규모 개발자 팀 및 카탈로그를 지원하는 vRealize Automation 사용자 역할 설정](#)을 참조하십시오.

- 3 vRealize Automation Cloud Assembly에서 리소스 및 사용자를 그룹화하는 데 사용하는 프로젝트를 생성합니다.

아래의 사용자 지정 역할 단계에는 프로젝트 역할도 포함되어 있습니다.

프로젝트 생성 지침이 필요한 경우 [사용자 역할 사용 사례 2: 대규모 개발자 팀 및 카탈로그를 지원하는 vRealize Automation 사용자 역할 설정](#)을 참조하십시오.

- 4 각 프로젝트 팀에 대한 클라우드 템플릿을 생성하고 릴리스합니다.

지침이 필요한 경우 [사용자 역할 사용 사례 1: 소규모 애플리케이션 개발 팀을 지원하도록 vRealize Automation 사용자 역할 설정](#)을 참조하십시오.

- 5 vRealize Automation Cloud Assembly에 서비스 관리자로 로그인하고 **인프라 > 관리 > 사용자 지정 역할**을 선택합니다.

- 6 제한된 인프라 관리자 역할을 생성합니다.

이 예에서는 다양한 프로젝트에 대한 인프라를 설정하는 전문가인 Tony 사용자가 있지만 그에게 전체 서비스 사용 권한을 부여하지 않으려고 합니다. 대신 Tony는 모든 프로젝트의 작업을 지원하는 코어 인프라를 구축합니다. Tony에게 제한된 인프라 관리 권한을 부여합니다. Tony 또는 외부 계약자에게는 검색된 시스템을 온보딩하고 vRealize Automation를 통해 관리하는 유사한 권한이 있을 수도 있습니다.

- a Tony를 서비스 사용자 및 뷰어로 vRealize Automation Cloud Assembly에 추가합니다.

뷰어 권한을 통해 Tony는 작업 문제 해결에 필요한 기본 클라우드 계정 및 통합을 볼 수 있지만 변경할 수는 없습니다.

- b 프로젝트를 생성하고 Tony를 프로젝트 멤버로 추가합니다.

- c 사용자 지정 역할을 생성하려면 **인프라 > 관리 > 사용자 지정 역할**을 선택하고 **새 사용자 지정 역할**을 클릭합니다.

- d **제한된 인프라 관리자**라는 이름을 입력하고 다음 사용 권한을 선택합니다.

선택하는 사용 권한...	사용자가 수행할 수 있는 작업...
인프라 > 클라우드 영역 관리	클라우드 영역 생성, 업데이트 및 삭제.
인프라 > 버전 매핑 관리	버전 매핑 생성, 업데이트 및 삭제.
인프라 > 이미지 매핑 관리	이미지 매핑 생성, 업데이트 및 삭제.

- e **생성**을 클릭합니다.

- f [사용자 지정 역할] 페이지에서 [제한된 인프라 관리자] 역할을 선택하고 **할당**을 클릭합니다.

- g Tony의 이메일 계정을 입력하고 **추가**를 클릭합니다.

예를 들어, Tony@yourcompany.com을 입력합니다.

정의된 Active Directory 사용자 그룹을 입력할 수도 있습니다.

- h Tony가 로그인 할 때 사용자 지정 역할에 의해 정의된 영역에서 값을 추가, 편집 및 삭제할 수 있는지 확인하도록 합니다.

## 7 확장성 개발자 역할을 생성합니다.

이 예에서는 확장성 작업 및 구독을 사용하여 일상적인 개발 작업을 관리하는 방법을 알고 있는 여러 클라우드 템플릿 개발자, Sylvia와 Igor가 있습니다. 또한 vRealize Orchestrator 사용 경험도 있어 다양한 프로젝트에 대한 사용자 지정 리소스 및 작업 제공을 담당합니다. 그들에게 사용자 지정 리소스 및 작업을 관리하고 확장성 작업 및 구독을 관리하여 확장성을 관리하는 추가 사용 권한을 부여합니다.

- a Sylvia와 Igor를 vRealize Automation Cloud Assembly 사용자로 추가합니다.
- b 그들이 확장성 스킬을 이용하여 작업하는 프로젝트에 멤버로 추가합니다.
- c **확장성 개발자**라는 이름으로 사용자 지정 사용자 역할을 생성하고 다음 사용 권한을 선택합니다.

선택하는 사용 권한...	사용자가 수행할 수 있는 작업...
XaaS > 사용자 지정 리소스 관리	사용자 지정 리소스 생성, 업데이트 또는 삭제.
XaaS > 리소스 작업 관리	사용자 지정 작업 생성, 업데이트 또는 삭제.
확장성 > 확장성 리소스 관리	확장성 작업 및 구독 생성, 업데이트 또는 삭제. 구독을 사용하지 않도록 설정. 작업 실행을 취소하고 삭제.

- d **생성**을 클릭합니다.

- e Sylvia와 Igor에게 확장성 개발자 역할을 할당합니다.

- f Sylvia와 Igor가 사용자 지정 리소스 및 작업을 관리할 수 있고 [확장성] 탭에서 다양한 옵션을 관리할 수 있는지 확인합니다.

## 8 배포 문제 해결자 역할을 생성합니다.

이 예에서는 팀의 배포 실패를 해결할 수 있도록 프로젝트 관리자에게 더 많은 관리 권한을 부여합니다.

- a 프로젝트 관리자인 Shauna, Pratap과 Wei를 vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker 서비스 사용자로 추가합니다.
- b 해당 프로젝트에서 프로젝트 관리자로 추가합니다.

- c **배포 문제 해결자**라는 이름으로 사용자 지정 사용자 역할을 생성하고 다음 사용 권한을 선택합니다.

선택하는 사용 권한...	사용자가 수행할 수 있는 작업...
인프라 > 버전 매핑 관리	버전 매핑 생성, 업데이트 및 삭제.
인프라 > 이미지 매핑 관리	이미지 매핑 생성, 업데이트 및 삭제.
배포 > 배포 관리	모든 배포를 보고, 프로젝트 전체에서 배포 및 배포 구성 요소에 대한 2일차 작업을 모두 실행합니다.
정책 > 정책 관리	정책 정의 생성, 업데이트 또는 삭제.

- d **생성**을 클릭합니다.
- e Shauna, Pratap과 Wei에게 배포 문제 해결자 역할을 할당합니다.
- f vRealize Automation Service Broker에서 버전 매핑, 이미지 매핑 및 정책을 관리할 수 있는지 확인합니다.

## 결과

이 사용 사례에서는 서비스 및 프로젝트 역할을 확장하는 사용자 지정 역할을 비롯한 다양한 역할을 가진 여러 사용자를 구성합니다.

## 다음에 수행할 작업

로컬 사용 사례를 다루는 사용자 지정 역할을 생성합니다.

## vRealize Automation Cloud Assembly에 클라우드 계정 추가

클라우드 계정은 지역 또는 데이터 센터에서 데이터를 수집하고 해당 지역에 클라우드 템플릿을 배포하기 위해 vRealize Automation Cloud Assembly에서 사용하는 구성된 사용 권한입니다.

수집된 데이터에는 나중에 클라우드 영역과 연결하는 지역이 포함됩니다.

나중에 클라우드 영역, 매핑 및 프로파일을 구성할 때 이러한 항목들이 연결된 클라우드 계정을 선택합니다.

클라우드 관리자는 팀 멤버가 작업하는 프로젝트에 대해 클라우드 계정을 생성합니다. 네트워크 및 보안, 계산, 스토리지 및 태그 콘텐츠 같은 리소스 정보는 클라우드 계정에서 데이터 수집됩니다.

**참고** 클라우드 계정에 이미 지역에 배포되어 있는 연결된 시스템이 있는 경우 온보딩 계획을 사용하여 그러한 시스템을 vRealize Automation Cloud Assembly 관리로 가져올 수 있습니다. [vRealize Automation Cloud Assembly의 온보딩 계획이란?](#)의 내용을 참조하십시오.

배포에 사용되는 클라우드 계정을 제거하면 해당 배포에 속하는 리소스가 관리되지 않습니다.

## vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명

vRealize Automation에서 클라우드 계정을 구성하고 작업하려면 다음과 같은 자격 증명이 있는지 확인합니다.

### 필수 클라우드 계정 자격 증명

수행할 작업	필요 사항
vRealize Automation Cloud Assembly에 가입 및 로그인	<p>VMware ID가 필요합니다.</p> <ul style="list-style-type: none"> <li>■ 회사 이메일 주소를 사용하여 <a href="#">My VMware</a> 계정을 설정합니다.</li> </ul>
vRealize Automation 서비스에 연결	<p>HTTPS 포트 443이 송신 트래픽을 위해 열려 있고 방화벽을 통해 다음에 액세스할 수 있어야 합니다.</p> <ul style="list-style-type: none"> <li>■ *.vmwareidentity.com</li> <li>■ gaz.csp-vidm-prod.com</li> <li>■ *.vmware.com</li> </ul> <p>포트 및 프로토콜에 대한 자세한 내용은 <a href="#">VMware 포트 및 프로토콜</a>을 참조하십시오.</p> <p>필수 포트 및 프로토콜에 대한 관련 정보는 <a href="#">포트 요구 사항</a>을 참조하십시오.</p>

수행할 작업	필요 사항
AWS(Amazon Web Services) 클라우드 계정 추가	<p>읽기 및 쓰기 권한을 가진 고급 사용자 계정을 제공합니다. 사용자 계정은 AWS IAM(Identity and Access Management) 시스템에서 전원 액세스 정책(PowerUserAccess)의 멤버여야 합니다.</p> <ul style="list-style-type: none"> <li>■ 20자리 액세스 키 ID 및 해당하는 비밀 액세스 키</li> </ul> <p>외부 HTTP 인터넷 프록시를 사용 중인 경우에는 IPv4에 대해 해당 프록시를 구성해야 합니다.</p> <p>vRealize Automation ABX(Action-Based Extensibility) 및 외부 IPAM 통합에는 추가 권한이 필요할 수 있습니다.</p> <p>자동 크기 조정 기능을 허용하려면 다음 AWS 권한이 제안됩니다.</p> <ul style="list-style-type: none"> <li>■ 자동 크기 조정 작업: <ul style="list-style-type: none"> <li>■ autoscaling:DescribeAutoScalingInstances</li> <li>■ autoscaling:AttachInstances</li> <li>■ autoscaling&gt;DeleteLaunchConfiguration</li> <li>■ autoscaling:DescribeAutoScalingGroups</li> <li>■ autoscaling&gt;CreateAutoScalingGroup</li> <li>■ autoscaling:UpdateAutoScalingGroup</li> <li>■ autoscaling&gt;DeleteAutoScalingGroup</li> <li>■ autoscaling:DescribeLoadBalancers</li> </ul> </li> <li>■ 자동 크기 조정 리소스: <ul style="list-style-type: none"> <li>■ *</li> </ul> <p>모든 크기 조정 리소스 권한을 제공합니다.</p> </li> </ul> <p>AWS STS(Security Token Service) 기능이 AWS ID 및 액세스를 위해 제한된 권한의 임시 자격 증명을 지원하도록 하려면 다음 권한이 필요합니다.</p> <ul style="list-style-type: none"> <li>■ AWS STS 리소스: <ul style="list-style-type: none"> <li>■ *</li> </ul> <p>모든 STS 리소스 권한을 제공합니다.</p> </li> </ul> <p>EC2 기능을 허용하려면 다음과 같은 AWS 권한이 필요합니다.</p> <ul style="list-style-type: none"> <li>■ EC2 작업: <ul style="list-style-type: none"> <li>■ ec2:AttachVolume</li> <li>■ ec2:AuthorizeSecurityGroupIngress</li> <li>■ ec2&gt;DeleteSubnet</li> <li>■ ec2&gt;DeleteSnapshot</li> <li>■ ec2:DescribeInstances</li> <li>■ ec2&gt;DeleteTags</li> <li>■ ec2:DescribeRegions</li> <li>■ ec2:DescribeVolumesModifications</li> <li>■ ec2&gt;CreateVpc</li> <li>■ ec2:DescribeSnapshots</li> <li>■ ec2:DescribeInternetGateways</li> <li>■ ec2&gt;DeleteVolume</li> <li>■ ec2:DescribeNetworkInterfaces</li> <li>■ ec2:StartInstances</li> <li>■ ec2:DescribeAvailabilityZones</li> <li>■ ec2&gt;CreateInternetGateway</li> <li>■ ec2&gt;CreateSecurityGroup</li> <li>■ ec2:DescribeVolumes</li> </ul> </li> </ul>

수행할 작업	필요 사항
	<ul style="list-style-type: none"> <li>■ ec2:CreateSnapshot</li> <li>■ ec2:ModifyInstanceAttribute</li> <li>■ ec2:DescribeRouteTables</li> <li>■ ec2:DescribeInstanceType</li> <li>■ ec2:DescribeInstanceTypeOfferings</li> <li>■ ec2:DescribeInstanceStatus</li> <li>■ ec2:DetachVolume</li> <li>■ ec2:RebootInstances</li> <li>■ ec2:AuthorizeSecurityGroupEgress</li> <li>■ ec2:ModifyVolume</li> <li>■ ec2:TerminateInstances</li> <li>■ ec2:DescribeSpotFleetRequestHistory</li> <li>■ ec2:DescribeTags</li> <li>■ ec2:CreateTags</li> <li>■ ec2:RunInstances</li> <li>■ ec2:DescribeNatGateways</li> <li>■ ec2:StopInstances</li> <li>■ ec2:DescribeSecurityGroups</li> <li>■ ec2:CreateVolume</li> <li>■ ec2:DescribeSpotFleetRequests</li> <li>■ ec2:DescribeImages</li> <li>■ ec2:DescribeVpcs</li> <li>■ ec2&gt;DeleteSecurityGroup</li> <li>■ ec2&gt;DeleteVpc</li> <li>■ ec2:CreateSubnet</li> <li>■ ec2:DescribeSubnets</li> <li>■ ec2:RequestSpotFleet</li> </ul>
	<p><b>참고</b> vRealize Automation ABX(Action-Based Extensibility) 또는 외부 IPAM 통합에는 SpotFleet 요청 권한이 필요하지 않습니다.</p>
	<ul style="list-style-type: none"> <li>■ EC2 리소스:             <ul style="list-style-type: none"> <li>■ *</li> </ul> <p>모든 EC2 리소스 권한을 제공합니다.</p> <p>탄력적 로드 밸런싱 기능을 허용하려면 다음과 같은 AWS 권한이 필요합니다.</p> </li> <li>■ 로드 밸런서 작업:             <ul style="list-style-type: none"> <li>■ elasticloadbalancing:DeleteLoadBalancer</li> <li>■ elasticloadbalancing:DescribeLoadBalancers</li> <li>■ elasticloadbalancing:RemoveTags</li> <li>■ elasticloadbalancing:CreateLoadBalancer</li> <li>■ elasticloadbalancing:DescribeTags</li> <li>■ elasticloadbalancing:ConfigureHealthCheck</li> <li>■ elasticloadbalancing:AddTags</li> <li>■ elasticloadbalancing:CreateTargetGroup</li> <li>■ elasticloadbalancing&gt;DeleteLoadBalancerListeners</li> </ul> </li> </ul>

## 수행할 작업

### 필요 사항

- elasticloadbalancing:DeregisterInstancesFromLoadBalancer
- elasticloadbalancing:RegisterInstancesWithLoadBalancer
- elasticloadbalancing:CreateLoadBalancerListeners
- 로드 밸런서 리소스:
  - \*

모든 로드 밸런서 리소스 권한을 제공합니다.

다음과 같은 AWS IAM(Identity and Access Management) 권한을 사용하도록 설정할 수 있지만 필수는 아닙니다.

- iam:SimulateCustomPolicy
- iam:GetUser
- iam:ListUserPolicies
- iam:GetUserPolicy
- iam:ListAttachedUserPolicies
- iam:GetPolicyVersion
- iam:ListGroupsForUser
- iam:ListGroupPolicies
- iam:GetGroupPolicy
- iam:ListAttachedGroupPolicies
- iam:ListPolicyVersions

수행할 작업	필요 사항
Microsoft Azure 클라우드 계정 추가	<p>Microsoft Azure 인스턴스를 구성하고, 사용 가능한 구독 ID를 제공하는 올바른 Microsoft Azure 구독을 얻습니다.</p> <p>Microsoft Azure 제품 설명서의 방법: 포털을 사용하여 리소스에 액세스할 수 있는 Azure AD 애플리케이션 및 서비스 사용자 생성에 설명된 대로 Active Directory 애플리케이션을 생성합니다.</p> <p>외부 HTTP 인터넷 프록시를 사용 중인 경우에는 IPv4에 대해 해당 프록시를 구성해야 합니다.</p> <p>다음 정보를 기록해 둡니다.</p> <ul style="list-style-type: none"> <li>■ 구독 ID <p>Microsoft Azure 구독에 액세스할 때 사용합니다.</p> </li> <li>■ 테넌트 ID <p>Microsoft Azure 계정에 생성하는 Active Directory 애플리케이션에 대한 권한 부여 끝점입니다.</p> </li> <li>■ 클라이언트 애플리케이션 ID <p>Microsoft Azure 개별 계정에서 Microsoft Active Directory에 대한 액세스를 제공합니다.</p> </li> <li>■ 클라이언트 애플리케이션 비밀 키 <p>클라이언트 애플리케이션 ID와 쌍으로 연결할 고유 비밀 키입니다.</p> </li> </ul> <p>Microsoft Azure 클라우드 계정을 생성하고 검증하기 위해 다음과 같은 사용 권한이 필요합니다.</p> <ul style="list-style-type: none"> <li>■ Microsoft Compute <ul style="list-style-type: none"> <li>■ Microsoft.Compute/virtualMachines/extensions/write</li> <li>■ Microsoft.Compute/virtualMachines/extensions/read</li> <li>■ Microsoft.Compute/virtualMachines/extensions/delete</li> <li>■ Microsoft.Compute/virtualMachines/deallocate/action</li> <li>■ Microsoft.Compute/virtualMachines/delete</li> <li>■ Microsoft.Compute/virtualMachines/powerOff/action</li> <li>■ Microsoft.Compute/virtualMachines/read</li> <li>■ Microsoft.Compute/virtualMachines/restart/action</li> <li>■ Microsoft.Compute/virtualMachines/start/action</li> <li>■ Microsoft.Compute/virtualMachines/write</li> <li>■ Microsoft.Compute/availabilitySets/write</li> <li>■ Microsoft.Compute/availabilitySets/read</li> <li>■ Microsoft.Compute/availabilitySets/delete</li> <li>■ Microsoft.Compute/disks/delete</li> <li>■ Microsoft.Compute/disks/read</li> <li>■ Microsoft.Compute/disks/write</li> </ul> </li> <li>■ Microsoft Network <ul style="list-style-type: none"> <li>■ Microsoft.Network/loadBalancers/backendAddressPools/join/action</li> <li>■ Microsoft.Network/loadBalancers/delete</li> <li>■ Microsoft.Network/loadBalancers/read</li> <li>■ Microsoft.Network/loadBalancers/write</li> <li>■ Microsoft.Network/networkInterfaces/join/action</li> <li>■ Microsoft.Network/networkInterfaces/read</li> <li>■ Microsoft.Network/networkInterfaces/write</li> <li>■ Microsoft.Network/networkInterfaces/delete</li> <li>■ Microsoft.Network/networkSecurityGroups/join/action</li> <li>■ Microsoft.Network/networkSecurityGroups/read</li> </ul> </li> </ul>

수행할 작업	필요 사항
	<ul style="list-style-type: none"> <li>■ Microsoft.Network/networkSecurityGroups/write</li> <li>■ Microsoft.Network/networkSecurityGroups/delete</li> <li>■ Microsoft.Network/publicIPAddresses/delete</li> <li>■ Microsoft.Network/publicIPAddresses/join/action</li> <li>■ Microsoft.Network/publicIPAddresses/read</li> <li>■ Microsoft.Network/publicIPAddresses/write</li> <li>■ Microsoft.Network/virtualNetworks/read</li> <li>■ Microsoft.Network/virtualNetworks/subnets/delete</li> <li>■ Microsoft.Network/virtualNetworks/subnets/join/action</li> <li>■ Microsoft.Network/virtualNetworks/subnets/read</li> <li>■ Microsoft.Network/virtualNetworks/subnets/write</li> <li>■ Microsoft.Network/virtualNetworks/write</li> <li>■ Microsoft Resources <ul style="list-style-type: none"> <li>■ Microsoft.Resources/subscriptions/resourcegroups/delete</li> <li>■ Microsoft.Resources/subscriptions/resourcegroups/read</li> <li>■ Microsoft.Resources/subscriptions/resourcegroups/write</li> </ul> </li> <li>■ Microsoft Storage <ul style="list-style-type: none"> <li>■ Microsoft.Storage/storageAccounts/delete</li> <li>■ Microsoft.Storage/storageAccounts/listKeys/action</li> <li>■ Microsoft.Storage/storageAccounts/read</li> <li>■ Microsoft.Storage/storageAccounts/write</li> </ul> </li> <li>■ Microsoft Web <ul style="list-style-type: none"> <li>■ Microsoft.Web/sites/read</li> <li>■ Microsoft.Web/sites/write</li> <li>■ Microsoft.Web/sites/delete</li> <li>■ Microsoft.Web/sites/config/read</li> <li>■ Microsoft.Web/sites/config/write</li> <li>■ Microsoft.Web/sites/config/list/action</li> <li>■ Microsoft.Web/sites/publishxml/action</li> <li>■ Microsoft.Web/serverfarms/write</li> <li>■ Microsoft.Web/serverfarms/delete</li> <li>■ Microsoft.Web/sites/hostruntime/functions/keys/read</li> <li>■ Microsoft.Web/sites/hostruntime/host/read</li> <li>■ Microsoft.web/sites/functions/masterkey/read</li> </ul> </li> </ul> <p>ABX(Action-Based Extensibility)가 있는 Microsoft Azure를 사용하는 경우 최소 사용 권한 외에 다음과 같은 사용 권한이 필요합니다.</p> <ul style="list-style-type: none"> <li>■ Microsoft.Web/sites/read</li> <li>■ Microsoft.Web/sites/write</li> <li>■ Microsoft.Web/sites/delete</li> <li>■ Microsoft.Web/sites/*/action</li> <li>■ Microsoft.Web/sites/config/read</li> <li>■ Microsoft.Web/sites/config/write</li> <li>■ Microsoft.Web/sites/config/list/action</li> <li>■ Microsoft.Web/sites/publishxml/action</li> </ul>

## 수행할 작업

### 필요 사항

- 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
- Microsoft.Web/apimanagementaccounts/apis/read
- Microsoft.Authorization/roleAssignments/read
- Microsoft.Authorization/roleAssignments/write
- Microsoft.Authorization/roleAssignments/delete

확장 기능과 함께 ABX(Action-Based Extensibility)가 있는 Microsoft Azure를 사용하는 경우 다음 사용 권한도 필요합니다.

- Microsoft.Compute/virtualMachines/extensions/write
- Microsoft.Compute/virtualMachines/extensions/read
- Microsoft.Compute/virtualMachines/extensions/delete

수행할 작업	필요 사항
GCP(Google Cloud Platform) 클라우드 계정 추가	<p>Google Cloud Platform 클라우드 계정이 Google Cloud Platform 계산 엔진과 상호 작용합니다.</p> <p>Google Cloud Platform 클라우드 계정을 생성하고 유효성을 검사하려면 프로젝트 관리자 및 소유자 자격 증명이 필요합니다.</p> <p>외부 HTTP 인터넷 프록시를 사용 중인 경우에는 IPv4에 대해 해당 프록시를 구성해야 합니다.</p> <p>계산 엔진 서비스를 사용하도록 설정해야 합니다. vRealize Automation에서 클라우드 계정을 생성하는 경우 계산 엔진을 초기화할 때 생성한 서비스 계정을 사용합니다.</p> <p>사용자가 수행할 수 있는 작업에 따라 다음과 같은 계산 엔진 권한도 필요합니다.</p> <ul style="list-style-type: none"> <li>■ <b>roles/compute.admin</b> <p>모든 계산 엔진 리소스에 대한 모든 권한을 제공합니다.</p> </li> <li>■ <b>roles/iam.serviceAccountUser</b> <p>서비스 계정으로 실행되도록 구성된 가상 시스템 인스턴스를 관리하는 사용자에게 액세스를 제공합니다. 다음 리소스 및 서비스에 대한 액세스 권한을 부여합니다.</p> <ul style="list-style-type: none"> <li>■ <b>compute.*</b></li> <li>■ <b>resourceManager.projects.get</b></li> <li>■ <b>resourceManager.projects.list</b></li> <li>■ <b>serviceUsage.quotas.get</b></li> <li>■ <b>serviceUsage.services.get</b></li> <li>■ <b>serviceUsage.services.list</b></li> </ul> </li> <li>■ <b>roles/compute.imageUser</b> <p>이미지에 대한 다른 사용 권한 없이 이미지를 나열하고 읽을 수 있는 권한을 제공합니다. 프로젝트 수준에서 <b>compute.imageUser</b> 역할을 부여하면 사용자가 프로젝트의 모든 이미지를 나열할 수 있습니다. 또한 사용자는 프로젝트의 이미지를 기반으로 인스턴스 및 영구 디스크와 같은 리소스를 생성할 수 있습니다.</p> <ul style="list-style-type: none"> <li>■ <b>compute.images.get</b></li> <li>■ <b>compute.images.getFromFamily</b></li> <li>■ <b>compute.images.list</b></li> <li>■ <b>compute.images.useReadOnly</b></li> <li>■ <b>resourceManager.projects.get</b></li> <li>■ <b>resourceManager.projects.list</b></li> <li>■ <b>serviceUsage.quotas.get</b></li> <li>■ <b>serviceUsage.services.get</b></li> <li>■ <b>serviceUsage.services.list</b></li> </ul> </li> <li>■ <b>roles/compute.instanceAdmin</b> <p>가상 시스템 인스턴스를 생성, 수정 및 삭제할 수 있는 권한을 제공합니다. 여기에는 디스크를 생성, 수정 및 삭제하고 쉼드된 VMBETA 설정을 구성할 수 있는 권한이 포함됩니다.</p> <p>가상 시스템 인스턴스(네트워크 또는 보안 설정이나 서비스 계정으로 실행되는 인스턴스는 제외)를 관리하는 사용자의 경우, 개별 인스턴스나 인스턴스를 포함하는 조직, 폴더 또는 프로젝트에 이 역할을 부여합니다.</p> <p>서비스 계정으로 실행되도록 구성된 가상 시스템 인스턴스를 관리하는 사용자에게는 <b>roles/iam.serviceAccountUser</b> 역할도 필요합니다.</p> <ul style="list-style-type: none"> <li>■ <b>compute.acceleratorTypes</b></li> <li>■ <b>compute.addresses.get</b></li> <li>■ <b>compute.addresses.list</b></li> <li>■ <b>compute.addresses.use</b></li> <li>■ <b>compute.autoscalers</b></li> </ul> </li> </ul>

## 수행할 작업

## 필요 사항

- 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.useExternalIp
- 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
- compute.subnetworks.useExternalIp
- compute.targetPools.get
- compute.targetPools.list

수행할 작업	필요 사항
	<ul style="list-style-type: none"> <li>■ compute.zoneOperations.get</li> <li>■ compute.zoneOperations.list</li> <li>■ compute.zones</li> <li>■ resourcemanager.projects.get</li> <li>■ resourcemanager.projects.list</li> <li>■ serviceusage.quotas.get</li> <li>■ serviceusage.services.get</li> <li>■ serviceusage.services.list</li> <li>■ roles/compute.instanceAdmin.v1</li> </ul> <p>계산 엔진 인스턴스, 인스턴스 그룹, 디스크, 스냅샷 및 이미지에 대한 모든 권한을 제공합니다. 모든 계산 엔진 네트워킹 리소스에 대한 읽기 액세스도 제공합니다.</p> <hr/> <p><b>참고</b> 인스턴스 수준에서 사용자에게 이 역할을 부여하면 해당 사용자가 새 인스턴스를 생성할 수 없습니다.</p> <hr/> <ul style="list-style-type: none"> <li>■ compute.acceleratorTypes</li> <li>■ compute.addresses.get</li> <li>■ compute.addresses.list</li> <li>■ compute.addresses.use</li> <li>■ compute.autoscalers</li> <li>■ compute.backendBuckets.get</li> <li>■ compute.backendBuckets.list</li> <li>■ compute.backendServices.get</li> <li>■ compute.backendServices.list</li> <li>■ compute.diskTypes</li> <li>■ compute.disks</li> <li>■ compute.firewalls.get</li> <li>■ compute.firewalls.list</li> <li>■ compute.forwardingRules.get</li> <li>■ compute.forwardingRules.list</li> <li>■ compute.globalAddresses.get</li> <li>■ compute.globalAddresses.list</li> <li>■ compute.globalAddresses.use</li> <li>■ compute.globalForwardingRules.get</li> <li>■ compute.globalForwardingRules.list</li> <li>■ compute.globalOperations.get</li> <li>■ compute.globalOperations.list</li> <li>■ compute.healthChecks.get</li> <li>■ compute.healthChecks.list</li> <li>■ compute.httpHealthChecks.get</li> <li>■ compute.httpHealthChecks.list</li> <li>■ compute.httpsHealthChecks.get</li> <li>■ compute.httpsHealthChecks.list</li> <li>■ compute.images</li> <li>■ compute.instanceGroupManagers</li> <li>■ compute.instanceGroups</li> </ul>

## 수행할 작업

## 필요 사항

- 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
- compute.targetHttpsProxies.list
- compute.targetInstances.get

수행할 작업	필요 사항
	<ul style="list-style-type: none"> <li>■ compute.targetInstances.list</li> <li>■ compute.targetPools.get</li> <li>■ compute.targetPools.list</li> <li>■ compute.targetSslProxies.get</li> <li>■ compute.targetSslProxies.list</li> <li>■ compute.targetTcpProxies.get</li> <li>■ compute.targetTcpProxies.list</li> <li>■ compute.targetVpnGateways.get</li> <li>■ compute.targetVpnGateways.list</li> <li>■ compute.urlMaps.get</li> <li>■ compute.urlMaps.list</li> <li>■ compute.vpnTunnels.get</li> <li>■ compute.vpnTunnels.list</li> <li>■ compute.zoneOperations.get</li> <li>■ compute.zoneOperations.list</li> <li>■ compute.zones</li> <li>■ resourceManager.projects.get</li> <li>■ resourceManager.projects.list</li> <li>■ serviceusage.quotas.get</li> <li>■ serviceusage.services.get</li> <li>■ serviceusage.services.list</li> </ul>
NSX-T 클라우드 계정 추가	<p>계정에 다음과 같은 읽기 및 쓰기 권한을 제공합니다.</p> <ul style="list-style-type: none"> <li>■ NSX-T 엔터프라이즈 관리자 역할 및 액세스 자격 증명</li> <li>■ NSX-T IP 주소 또는 FQDN</li> </ul> <p>관리자에게는 이 페이지의 다음 "vCenter 기반 클라우드 계정에 대한 vSphere 에이전트 요구 사항" 섹션에 설명된 대로 vCenter Server에 대한 액세스 권한 "도" 필요합니다.</p>
NSX-V 클라우드 계정 추가	<p>계정에 다음과 같은 읽기 및 쓰기 권한을 제공합니다.</p> <ul style="list-style-type: none"> <li>■ NSX-V 엔터프라이즈 관리자 역할 및 액세스 자격 증명</li> <li>■ NSX-V IP 주소 또는 FQDN</li> </ul> <p>관리자에게는 이 페이지의 다음 "vCenter 기반 클라우드 계정에 대한 vSphere 에이전트 요구 사항" 섹션에 설명된 대로 vCenter Server에 대한 액세스 권한 "도" 필요합니다.</p>

수행할 작업	필요 사항
vCenter 클라우드 계정 추가	<p>계정에 다음과 같은 읽기 및 쓰기 권한을 제공합니다.</p> <ul style="list-style-type: none"> <li>■ vCenter IP 주소 또는 FQDN</li> </ul> <p>관리자에게는 이 페이지의 다음 "vCenter 기반 클라우드 계정에 대한 vSphere 에이전트 요구 사항" 섹션에 설명된 대로 vCenter Server에 대한 액세스 권한 "도" 필요합니다.</p>
VMC(VMware Cloud on AWS) 클라우드 계정 추가	<p>계정에 다음과 같은 읽기 및 쓰기 권한을 제공합니다.</p> <ul style="list-style-type: none"> <li>■ cloudadmin@vmc.local 계정 또는 CloudAdmin 그룹의 모든 사용자 계정</li> <li>■ NSX 엔터프라이즈 관리자 역할 및 액세스 자격 증명</li> <li>■ 조직의 VMware Cloud on AWS SDDC 환경에 대한 NSX 클라우드 관리자 액세스 권한</li> <li>■ 조직의 VMware Cloud on AWS SDDC 환경에 대한 관리자 액세스 권한</li> <li>■ 조직의 VMware Cloud on AWS 서비스에서 VMware Cloud on AWS 환경의 VMware Cloud on AWS API 토큰</li> <li>■ vCenter IP 주소 또는 FQDN</li> </ul> <p>관리자에게는 이 페이지의 "vSphere 기반 클라우드 계정에 대한 vCenter 에이전트 요구 사항" 섹션에 나열된 모든 사용 권한이 있는 대상 VMware Cloud on AWS SDDC에 사용되는 vCenter에 대한 액세스 권한 "도" 필요합니다.</p> <p>VMware Cloud on AWS 클라우드 계정을 생성하고 사용하는 데 필요한 사용 권한에 대한 자세한 내용은 VMware Cloud on AWS <a href="#">제품 설명서</a>에서 "VMware Cloud on AWS 데이터 센터 관리"를 참조하십시오.</p>

## vCenter 기반 클라우드 계정에 대한 vSphere 에이전트 요구 사항

다음 표에는 VMware Cloud on AWS 및 vCenter 클라우드 계정을 관리하는 데 필요한 사용 권한이 나열되어 있습니다. 끝점을 호스팅하는 클러스터만이 아닌 vCenter Server의 모든 클러스터에 해당 사용 권한이 설정되어야 합니다.

NSX-V, NSX-T, vCenter 및 VMware Cloud on AWS를 포함하여 모든 vCenter Server 기반 클라우드 계정에 대해 관리자는 vSphere 끝점 자격 증명 또는 호스트 vCenter Server에 대한 관리 액세스 권한을 제공하는 자격 증명을 가지고 있어야 하며, 이러한 자격 증명을 사용하여 에이전트 서비스가 vCenter에서 실행됩니다.

vSphere 에이전트 요구 사항에 대한 자세한 내용은 [VMware vSphere 제품 설명서](#)를 참조하십시오.

**표 3-15. vSphere 에이전트가 vCenter Server 인스턴스를 관리하는 데 필요한 사용 권한**

특성 값	사용 권한
데이터스토어	<ul style="list-style-type: none"> <li>■ 공간 할당</li> <li>■ 데이터스토어 찾아보기</li> <li>■ 하위 수준 파일 작업</li> </ul>
데이터스토어 클러스터	데이터스토어 클러스터 구성
폴더	<ul style="list-style-type: none"> <li>■ 폴더 생성</li> <li>■ 폴더 삭제</li> </ul>
글로벌	<ul style="list-style-type: none"> <li>■ 사용자 지정 특성 관리</li> <li>■ 사용자 지정 특성 설정</li> </ul>
네트워크	네트워크 할당

표 3-15. vSphere 에이전트가 vCenter Server 인스턴스를 관리하는 데 필요한 사용 권한 (계속)

특성 값	사용 권한
사용 권한	사용 권한 수정
리소스	<ul style="list-style-type: none"> <li>■ VM을 리소스 풀에 할당</li> <li>■ 전원이 꺼진 가상 시스템 마이그레이션</li> <li>■ 전원이 켜진 가상 시스템 마이그레이션</li> </ul>
컨텐츠 라이브러리	<p>컨텐츠 라이브러리에 대한 사용 권한을 할당하려면 관리자가 사용 권한을 사용자에게 글로벌 사용 권한으로 부여해야 합니다. 관련 정보는 <a href="#">VMware vSphere 설명서의 "vSphere 가상 시스템 관리"</a>에서 <a href="#">컨텐츠 라이브러리에 대한 사용 권한의 계층적 상속</a>을 참조하십시오.</p> <ul style="list-style-type: none"> <li>■ 라이브러리 항목 추가</li> <li>■ 로컬 라이브러리 생성</li> <li>■ 구독 라이브러리 생성</li> <li>■ 라이브러리 항목 삭제</li> <li>■ 로컬 라이브러리 삭제</li> <li>■ 구독 라이브러리 삭제</li> <li>■ 파일 다운로드</li> <li>■ 라이브러리 항목 제거</li> <li>■ 구독 라이브러리 제거</li> <li>■ 구독 정보 검색</li> <li>■ 스토리지 읽기</li> <li>■ 라이브러리 항목 동기화</li> <li>■ 구독 라이브러리 동기화</li> <li>■ 유형 검사</li> <li>■ 구성 설정 업데이트</li> <li>■ 파일 업데이트</li> <li>■ 라이브러리 업데이트</li> <li>■ 라이브러리 항목 업데이트</li> <li>■ 로컬 라이브러리 업데이트</li> <li>■ 구독 라이브러리 업데이트</li> <li>■ 구성 설정 보기</li> </ul>
태그	<ul style="list-style-type: none"> <li>■ vSphere 태그 할당 또는 할당 취소</li> <li>■ vSphere 태그 생성</li> <li>■ vSphere 태그 범주 생성</li> <li>■ vSphere 태그 삭제</li> <li>■ vSphere 태그 범주 삭제</li> <li>■ vSphere 태그 편집</li> <li>■ vSphere 태그 범주 편집</li> <li>■ 범주에 대한 UsedBy 필드 수정</li> <li>■ 태그에 대한 UsedBy 필드 수정</li> </ul>

표 3-15. vSphere 에이전트가 vCenter Server 인스턴스를 관리하는 데 필요한 사용 권한 (계속)

특성 값	사용 권한
vApp	<ul style="list-style-type: none"> <li>■ 가져오기</li> <li>■ vApp 애플리케이션 구성</li> </ul> <p>OVF 템플릿에 대해 그리고 콘텐츠 라이브러리에서 VM을 프로비저닝하려면 vApp.Import 애플리케이션 구성이 필요합니다.</p> <p>클라우드 구성 스크립팅에 대해 cloud-init를 사용할 때 vApp.vApp 애플리케이션 구성이 필요합니다. 이 설정을 사용하면 제품 정보 및 속성 같은 vApp의 내부 구조를 수정할 수 있습니다.</p>
가상 시스템 인벤토리	<ul style="list-style-type: none"> <li>■ 기존 항목에서 생성</li> <li>■ 새로 생성</li> <li>■ 이동</li> <li>■ 제거</li> </ul>
가상 시스템 상호 작용	<ul style="list-style-type: none"> <li>■ CD 미디어 구성</li> <li>■ 콘솔 상호 작용</li> <li>■ 디바이스 연결</li> <li>■ 전원 끄기</li> <li>■ 전원 켜기</li> <li>■ 재설정</li> <li>■ 일시 중단</li> <li>■ 도구 설치</li> </ul>
가상 시스템 - 구성	<ul style="list-style-type: none"> <li>■ 기존 디스크 추가</li> <li>■ 새 디스크 추가</li> <li>■ 디스크 제거</li> <li>■ 고급</li> <li>■ CPU 수 변경</li> <li>■ 리소스 변경</li> <li>■ 가상 디스크 확장</li> <li>■ 디스크 변경 내용 추적</li> <li>■ 메모리</li> <li>■ 디바이스 설정 수정</li> <li>■ 이름 바꾸기</li> <li>■ 주석 설정</li> <li>■ 설정</li> <li>■ 스왑 파일 배치</li> </ul>

표 3-15. vSphere 에이전트가 vCenter Server 인스턴스를 관리하는 데 필요한 사용 권한 (계속)

특성 값	사용 권한
가상 시스템 - 프로비저닝	<ul style="list-style-type: none"> <li>■ 사용자 지정</li> <li>■ 템플릿 복제</li> <li>■ 가상 시스템 복제</li> <li>■ 템플릿 배포</li> <li>■ 사용자 지정 규격 읽기</li> </ul>
가상 시스템 상태	<ul style="list-style-type: none"> <li>■ 스냅샷 생성</li> <li>■ 스냅샷 제거</li> <li>■ 스냅샷으로 되돌리기</li> </ul>

## vRealize Automation Cloud Assembly에서 사용할 Microsoft Azure 구성

vRealize Automation Cloud Assembly에서 Microsoft Azure 클라우드 계정을 생성하려면 몇 가지 정보를 수집하고 구성을 수행해야 합니다.

### 절차

- 1 Microsoft Azure 구독 및 테넌트 ID를 찾아서 기록해둡니다.
  - 구독 ID - Azure Portal의 왼쪽 도구 모음에 있는 [구독] 아이콘을 클릭하여 구독 ID를 확인합니다.
  - 테넌트 ID - Azure Portal에서 [도움말] 아이콘을 클릭하고 [진단 표시]를 선택합니다. 테넌트를 검색하여 찾으려면 ID를 기록해둡니다.
- 2 새 스토리지 계정 및 리소스 그룹을 생성하여 시작할 수 있습니다. 또는, 나중에 Blueprint에서 생성할 수 있습니다.
  - 스토리지 계정 - 다음 절차를 사용하여 계정을 구성합니다.
    - 1 Azure Portal의 사이드바에서 스토리지 계정 아이콘을 찾습니다. 올바른 구독이 선택되었는지 확인하고 **추가**를 클릭합니다. Azure 검색 필드에서 스토리지 계정을 검색할 수도 있습니다.
    - 2 스토리지 계정에 대한 필수 정보를 입력합니다. 구독 ID가 필요합니다.
    - 3 기존 리소스 그룹을 사용할지 아니면 새로 생성할지 선택합니다. 리소스 그룹 이름을 기록해둡니다. 나중에 필요할 수 있습니다.

**참고** 스토리지 계정의 위치를 저장합니다. 나중에 필요할 수 있습니다.

- 3 가상 네트워크를 생성합니다. 또는 적합한 기존 네트워크가 있는 경우 해당 네트워크를 선택할 수 있습니다.

네트워크를 생성하는 경우 [기존 리소스 그룹 사용]을 선택하고 이전 단계에서 생성한 그룹을 지정해야 합니다. 또한 이전에 지정한 것과 동일한 위치를 선택합니다. 개체가 사용할 적용 가능한 모든 구성 요소 간에 위치가 일치하지 않으면 Microsoft Azure에서 가상 시스템이나 기타 개체가 배포되지 않습니다.

- a 왼쪽 패널에서 [가상 네트워크] 아이콘을 찾아서 클릭하거나 가상 네트워크를 검색합니다. 올바른 구독을 선택하고 **추가**를 클릭합니다.
  - b 새 가상 네트워크의 고유 이름을 입력하고 나중을 위해 기록해둡니다.
  - c **주소 공간** 필드에 가상 네트워크에 적합한 IP 주소를 입력합니다.
  - d 올바른 구독이 선택되었는지 확인하고 **추가**를 클릭합니다.
  - e 나머지 기본 구성 정보를 입력합니다.
  - f 필요에 따라 다른 옵션을 수정할 수 있지만 대부분의 구성에서 기본값을 그대로 두어도 됩니다.
  - g **생성**을 클릭합니다.
- 4 vRA가 인증할 수 있도록 Azure Active Directory 애플리케이션을 설정합니다.
- a Azure 왼쪽 메뉴에서 **Active Directory** 아이콘을 찾아서 클릭합니다.
  - b **애플리케이션 등록**을 클릭하고 **추가**를 선택합니다.
  - c Azure 이름 유효성 검사를 준수하는 애플리케이션 이름을 입력합니다.
  - d 웹앱/API를 애플리케이션 유형으로 둡니다.
  - e [로그온 URL]은 사용에 적합한 것이면 무엇이든 가능합니다.
  - f **생성**을 클릭합니다.
- 5 Cloud Assembly에서 애플리케이션을 인증하기 위한 비밀 키를 생성합니다.
- a Azure에서 애플리케이션 이름을 클릭합니다.  
나중에 사용할 수 있도록 애플리케이션 ID를 기록해둡니다.
  - b 다음 창에서 **모든 설정**을 클릭하고 설정 목록에서 [키]를 선택합니다.
  - c 새 키에 대한 설명을 입력하고 기간을 선택합니다.
  - d **저장**을 클릭하고 키 값을 안전한 위치에 복사합니다. 키 값은 나중에 검색할 수 없습니다.
  - e 왼쪽 메뉴에서 애플리케이션에 대해 **API 사용 권한**을 선택하고 **사용 권한 추가**를 클릭하여 새 사용 권한을 생성합니다.
  - f [API 선택] 페이지에서 Azure 서비스 관리를 선택합니다.
  - g **위임된 사용 권한**을 클릭합니다.
  - h [사용 권한 선택]에서 user\_impersonation을 선택하고 **사용 권한 추가**를 클릭합니다.

- 6 Active Directory 애플리케이션이 Azure 구독에 연결할 수 있도록 권한을 부여합니다. 그래야 가상 시스템을 배포하고 관리할 수 있습니다.
  - a 왼쪽 메뉴에서 구독 아이콘을 클릭하고 새 구독을 선택합니다.  
이름 텍스트를 클릭해야 패널이 미끄러질 수도 있습니다.
  - b [액세스 제어(IAM)] 옵션을 선택하여 구독에 대한 사용 권한을 살펴봅니다.
  - c [역할 할당 추가] 머리글 아래에서 **추가**를 클릭합니다.
  - d [역할] 드롭다운에서 [참가자]를 선택합니다.
  - e [액세스 할당] 드롭다운의 기본 선택 항목을 그대로 둡니다.
  - f [선택] 상자에 애플리케이션의 이름을 입력합니다.
  - g **저장**을 클릭합니다.
  - h 다른 역할을 추가하여 새 애플리케이션에 소유자, 참가자 및 독자 역할이 포함되도록 합니다.
  - i **저장**을 클릭합니다.

#### 다음에 수행할 작업

Microsoft Azure 명령줄 인터페이스 도구를 설치해야 합니다. 이러한 도구는 Windows 및 Mac 운영 체제 모두에서 무료로 사용할 수 있습니다. 이러한 도구의 다운로드 및 설치에 대한 자세한 내용은 Microsoft 설 명서를 참조하십시오.

명령줄 인터페이스가 설치되어 있으면 새 구독에 인증해야 합니다.

- 1 터미널 창을 열고 Microsoft Azure 로그인을 입력합니다. 인증할 수 있는 URL과 짧은 코드가 제공됩 니다.
- 2 브라우저에서 디바이스의 애플리케이션으로부터 받은 코드를 입력합니다.
- 3 인증 코드를 입력하고 **계속**을 클릭합니다.
- 4 Azure 계정 및 로그인을 선택합니다.  
여러 구독이 있는 경우 `azure account set <subscription-name>` 명령을 사용하여 올바른 구독이 선택되도록 합니다.
- 5 계속 진행하기 전에 `azure provider register microsoft.compute` 명령을 사용하여 Microsoft.Compute 제공자를 새 Azure 구독에 등록해야 합니다.

명령이 시간 초과되고 처음 실행할 때 오류가 생성되면 다시 실행합니다.

구성을 완료하면 `azure vm image list` 명령을 사용하여 사용 가능한 가상 시스템 이미지 이름을 검색할 수 있습니다. 원하는 이미지를 선택하여 제공된 URN을 기록해두면 나중에 Blueprint에서 사용할 수 있습 니다.

## vRealize Automation에서 Microsoft Azure 클라우드 계정 생성

클라우드 관리자는 팀에서 vRealize Automation 클라우드 템플릿을 배포할 계정 지역에 대한 Microsoft Azure 클라우드 계정을 생성할 수 있습니다.

Microsoft Azure 클라우드 계정이 vRealize Automation에서 작동하는 방식에 대한 사용 사례 예시를 보려면 자습서: [vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트를 참조하십시오](#).

### 사전 요구 사항

- 필요한 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오](#).
- 필요한 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- vRealize Automation에서 사용할 Microsoft Azure 계정을 구성합니다. [vRealize Automation Cloud Assembly에서 사용할 Microsoft Azure 구성의 내용을 참조하십시오](#).
- 외부 인터넷에 액세스할 수 없는 경우 인터넷 서버 프록시를 구성합니다. [vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법의 내용을 참조하십시오](#).

### 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭합니다.
- 2 Microsoft Azure 계정 유형을 선택하고 자격 증명과 기타 값을 입력합니다.
- 3 **검증**을 클릭합니다.  
계정과 연결된 계정 지역이 수집됩니다.
- 4 이 리소스를 프로비저닝할 지역을 선택합니다.
- 5 효율성을 위해 **선택한 지역에 대한 클라우드 영역 생성**을 클릭합니다.
- 6 태그 지정 전략을 지원하기 위해 태그를 추가해야 하는 경우에는 기능 태그를 입력합니다. 태그를 사용하여 [vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#) 및 [태그 지정 전략 생성 항목을 참조하십시오](#).



기능 태그 및 제약 조건 태그가 배포 배치를 제어하는 데 도움이 되는 방법에 대한 자세한 내용은 [제약 조건 태그 및 배치](#) 비디오 자습서를 참조하십시오.

- 7 **저장**을 클릭합니다.

### 결과

계정이 vRealize Automation에 추가되고 선택한 지역을 지정된 클라우드 영역에서 사용할 수 있습니다.

### 다음에 수행할 작업

이 클라우드 계정에 대한 인프라 리소스를 생성합니다.

## vRealize Automation에서 Amazon Web Services 클라우드 계정 생성

클라우드 관리자는 팀에서 vRealize Automation 클라우드 템플릿을 배포할 계정 지역에 대한 AWS(Amazon Web Services) 클라우드 계정을 생성할 수 있습니다.

인증된 사용자의 경우 AWS 클라우드 계정은 AWS GovCloud 구성에 대한 액세스를 지원합니다. 이 구성은 프로젝트 구성, 태그 및 인프라와 관련하여 대부분의 표준 vRealize Automation 클라우드 계정 기능을 지원합니다. Cloud Assembly 클라우드 템플릿에서 이 구성은 AWS PaaS(Platform as a Service) 속성 사용을 지원합니다.

다음 절차에서는 AWS 클라우드 계정을 구성하는 방법을 설명합니다.

### 사전 요구 사항

- 필요한 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.](#)
- 필요한 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 필요한 AWS 관리자 자격 증명이 있는지 확인합니다.
- 외부 인터넷에 액세스할 수 없는 경우 인터넷 서버 프록시를 구성합니다. [vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법](#)의 내용을 참조하십시오.

### 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭합니다.
- 2 AWS 계정 유형을 선택하고 자격 증명과 기타 값을 입력합니다.
- 3 **검증**을 클릭합니다.  
계정과 연결된 계정 지역이 수집됩니다.
- 4 이 리소스를 프로비저닝할 지역을 선택합니다.
- 5 효율성을 위해 **선택한 지역에 대한 클라우드 영역 생성**을 클릭합니다.
- 6 태그 지정 전략을 지원하기 위해 태그를 추가해야 하는 경우에는 기능 태그를 입력합니다. 태그를 사용하여 [vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#) 및 [태그 지정 전략 생성](#) 항목을 참조하십시오.



기능 태그 및 제약 조건 태그가 배포 배치를 제어하는 데 도움이 되는 방법에 대한 자세한 내용은 [제약 조건 태그 및 배치](#) 비디오 자습서를 참조하십시오.

- 7 **추가**를 클릭합니다.

### 결과

계정이 vRealize Automation에 추가되고 선택한 지역을 지정된 클라우드 영역에서 사용할 수 있습니다.

## 다음에 수행할 작업

이 클라우드 계정에 대한 인프라 리소스를 구성합니다.

## vRealize Automation에서 Google Cloud Platform 클라우드 계정 생성

클라우드 관리자는 팀에서 vRealize Automation 클라우드 템플릿을 배포할 계정 지역에 대한 GCP(Google Cloud Platform) 클라우드 계정을 생성할 수 있습니다.

### 사전 요구 사항

- 필요한 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.](#)
- 필요한 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- Google Cloud Platform JSON 보안 키에 액세스할 수 있는지 확인합니다.
- Google Cloud Platform 인스턴스에 대한 필수 보안 정보가 있는지 확인합니다. 인스턴스에서 또는 Google 설명서에서 이러한 정보의 대부분을 얻을 수 있습니다.
- 외부 인터넷에 액세스할 수 없는 경우 인터넷 서버 프록시를 구성합니다. [vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법](#)의 내용을 참조하십시오.

### 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭합니다.
- 2 Google Cloud Platform 계정 유형을 선택하고 적절한 자격 증명 및 관련 정보를 입력합니다. 소스 GCP 계정 계산 엔진을 초기화할 때 생성한 서비스 계정을 사용합니다.

위의 **사전 요구 사항** 섹션에 명시된 바와 같이, 자격 증명 요구 사항은 [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#)에서 사용할 수 있습니다. vRealize Automation에서 클라우드 계정을 성공적으로 생성하려면 소스 GCP 계정에 계산 엔진 서비스가 사용되도록 설정되어 있어야 합니다.

vRealize Automation에서 프로젝트 ID는 Google Cloud Platform 끝점의 일부입니다. 클라우드 계정을 생성할 때 이를 지정합니다. 프로젝트별 전용 이미지의 데이터를 수집하는 동안 vRealize Automation GCP 어댑터는 Google Cloud Platform API를 쿼리합니다.

- 3 **검증**을 클릭합니다.  
계정과 연결된 계정 지역이 수집됩니다.
- 4 이 리소스를 프로비저닝할 지역을 선택합니다.
- 5 효율성을 위해 **선택한 지역에 대한 클라우드 영역 생성**을 클릭합니다.

- 6 태그 지정 전략을 지원 하는 태그가 필요한 경우에는 기능 태그를 입력합니다. [태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법 및 태그 지정 전략 생성 항목](#)을 참조하십시오.



기능 태그 및 제약 조건 태그가 배포 배치를 제어하는 데 도움이 되는 방법에 대한 자세한 내용은 [제약 조건 태그 및 배치 비디오](#) 자습서를 참조하십시오.

- 7 **추가**를 클릭합니다.

## 결과

계정이 vRealize Automation에 추가되고 선택한 지역을 지정된 클라우드 영역에서 사용할 수 있습니다.

## 다음에 수행할 작업

이 클라우드 계정에 대한 인프라 리소스를 생성합니다.

## vRealize Automation에서 vCenter 클라우드 계정 생성

vRealize Automation 클라우드 템플릿을 배포하려는 계정 지역에 대한 vCenter 클라우드 계정을 추가할 수 있습니다.

네트워크 및 보안을 위해 vCenter 클라우드 계정을 NSX-T 또는 NSX-V 클라우드 계정에 연결할 수 있습니다.

NSX-T 클라우드 계정은 하나 이상의 vCenter 클라우드 계정에 연결할 수 있습니다. 그러나 NSX-V 클라우드 계정은 하나의 vCenter 클라우드 계정에만 연결할 수 있습니다.

## 사전 요구 사항

- 필요한 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목](#)을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 클라우드 계정 지원을 위해 포트와 프로토콜을 올바르게 구성했는지 확인합니다. [vRealize Automation 제품 설명서](#)의 "vRealize Easy Installer를 사용하여 vRealize Automation 설치"에서 "vRealize Automation의 포트 및 프로토콜" 항목 및 "vRealize Automation 참조 아키텍처 가이드"에서 "포트 요구 사항" 항목을 참조하십시오.

## 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭합니다.
- 2 vCenter 계정 유형을 선택하고 vCenter Server 호스트 IP 주소를 입력합니다.

### 3 vCenter Server 관리자 자격 증명을 입력하고 **검증**을 클릭합니다.

계정과 연결된 모든 데이터 센터의 데이터가 수집됩니다. 다음 요소에 대한 모든 vSphere 태그와 마찬가지로 다음 요소 데이터가 수집됩니다.

- 시스템
- 클러스터 및 호스트
- 포트 그룹
- 데이터스토어

### 4 이 클라우드 계정에 대한 프로비저닝을 허용하려면 지정된 vCenter Server에서 사용 가능한 데이터 센터를 하나 이상 선택합니다.

### 5 효율성을 위해, 선택한 데이터 센터에 프로비저닝을 위한 클라우드 영역을 생성합니다.

조직의 클라우드 전략에 따라 별도의 단계로 클라우드 영역을 생성할 수도 있습니다.

클라우드 영역에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 클라우드 영역에 대해 알아보기](#) 항목을 참조하십시오.

### 6 기존 NSX 클라우드 계정을 선택합니다.

NSX 계정을 지금 선택하거나 나중에 클라우드 계정을 편집할 때 선택할 수 있습니다.

NSX-V 클라우드 계정에 대한 자세한 내용은 [vRealize Automation에서 NSX-V 클라우드 계정 생성](#) 항목을 참조하십시오.

NSX-T 클라우드 계정에 대한 자세한 내용은 [vRealize Automation에서 NSX-T 클라우드 계정 생성](#) 항목을 참조하십시오.

클라우드 템플릿을 배포한 후 연결 변경을 수행하는 방법에 대한 자세한 내용은 [vRealize Automation에서 NSX 클라우드 계정 연결을 제거하면 어떻게 됩니까?](#) 항목을 참조하십시오.

### 7 태그 지정 전략을 지원하기 위해 태그를 추가하려면 기능 태그를 입력합니다.

태그를 지금 추가하거나 나중에 클라우드 계정을 편집할 때 추가할 수 있습니다. 태그 지정에 대한 자세한 내용은 [태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#) 항목을 참조하십시오.



기능 태그 및 제약 조건 태그가 배포 배치를 제어하는 데 도움이 되는 방법에 대한 자세한 내용은 [제약 조건 태그 및 배치](#) 비디오 자습서를 참조하십시오.

### 8 **저장**을 클릭합니다.

#### 결과

클라우드 계정이 추가되고 선택한 데이터 센터를 지정된 클라우드 영역에서 사용할 수 있습니다. 시스템, 네트워크, 스토리지 및 볼륨과 같은 수집된 데이터가 **인프라** 탭의 **리소스** 섹션에 나열됩니다.

## 다음에 수행할 작업

이 클라우드 계정에 대한 나머지 인프라 리소스를 구성합니다. [장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축](#)의 내용을 참조하십시오.

## vRealize Automation에서 NSX-V 클라우드 계정 생성

네트워크 및 보안을 위해 NSX-V 클라우드 계정을 생성하고 vCenter 클라우드 계정과 연결할 수 있습니다.

NSX-V 클라우드 계정은 하나의 vCenter 클라우드 계정에만 연결할 수 있습니다.

NSX-V와 vCenter 클라우드 계정 간의 연결은 특히 NSX 애플리케이션에서 vRealize Automation 외부에서 구성되어야 합니다. vRealize Automation은 NSX와 vCenter 간의 연결을 생성하지 않습니다.

vRealize Automation에서 NSX에 이미 있는 연결을 지정합니다.

### 사전 요구 사항

- 필요한 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 이 NSX 클라우드 계정에서 사용할 vCenter 클라우드 계정이 있는지 확인합니다. [vRealize Automation에서 vCenter 클라우드 계정 생성](#) 항목을 참조하십시오.
- 클라우드 계정 지원을 위해 포트와 프로토콜을 올바르게 구성했는지 확인합니다. [vRealize Automation 제품 설명서](#)의 "vRealize Easy Installer를 사용하여 vRealize Automation 설치"에서 "vRealize Automation의 포트 및 프로토콜" 항목 및 "vRealize Automation 참조 아키텍처 가이드"에서 "포트 요구 사항" 항목을 참조하십시오.

### 절차

1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭합니다.

2 NSX-V 계정 유형을 선택하고 NSX-V 호스트 IP 주소를 입력합니다.

3 NSX 관리자 자격 증명을 입력하고 **검증**을 클릭합니다.

계정과 연결된 자산이 수집됩니다.

NSX 호스트 IP 주소를 사용할 수 없으면 검증이 실패합니다.

4 가능한 경우, 이 NSX-V 계정과 연결하는 vCenter 클라우드 계정을 나타내는 vCenter 끝점을 선택합니다.

현재 NSX-T 또는 NSX-V 클라우드 계정에 연결되지 않은 vCenter 클라우드 계정만 선택할 수 있습니다.

클라우드 템플릿을 배포한 후 연결 변경을 수행하는 방법에 대한 자세한 내용은 [vRealize Automation에서 NSX 클라우드 계정 연결을 제거하면 어떻게 됩니까?](#) 항목을 참조하십시오.

- 5 태그 지정 전략을 지원하기 위해 태그를 추가하려면 기능 태그를 입력합니다.

나중에 기능 태그를 추가하거나 제거할 수 있습니다. [태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#)의 내용을 참조하십시오.



기능 태그 및 제약 조건 태그가 배포 배치를 제어하는 데 도움이 되는 방법에 대한 자세한 내용은 [제약 조건 태그 및 배치](#) 비디오 자습서를 참조하십시오.

- 6 **저장**을 클릭합니다.

#### 다음에 수행할 작업

이 NSX 클라우드 계정에 연결할 vCenter 클라우드 계정을 생성하거나 편집할 수 있습니다. [vRealize Automation에서 vCenter 클라우드 계정 생성](#)의 내용을 참조하십시오.

이 클라우드 계정에서 사용하는 데이터 센터에 사용할 하나 이상의 클라우드 영역을 생성하고 구성합니다. [vRealize Automation Cloud Assembly 클라우드 영역에 대해 알아보기](#) 항목을 참조하십시오.

이 클라우드 계정에 대한 인프라 리소스를 구성합니다. [장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축](#)의 내용을 참조하십시오.

## vRealize Automation에서 NSX-T 클라우드 계정 생성

네트워크 및 보안을 위해 NSX-T 클라우드 계정을 생성하고 하나 이상의 vCenter 클라우드 계정에 연결할 수 있습니다.

NSX-T 클라우드 계정은 하나 이상의 vCenter 클라우드 계정에 연결할 수 있습니다. 그러나 NSX-V 클라우드 계정은 하나의 vCenter 클라우드 계정에만 연결할 수 있습니다.

NSX-T와 하나 이상의 vCenter 클라우드 계정 간의 연결은 특히 NSX 애플리케이션에서 vRealize Automation 외부에서 구성되어야 합니다. vRealize Automation은 NSX와 vCenter 간의 연결을 생성하지 않습니다. vRealize Automation에서 NSX에 이미 있는 하나 이상의 구성 연결을 지정합니다.

NSX-T Manager API 메서드 또는 NSX-T 정책 API 메서드를 지원하도록 NSX-T 클라우드 계정을 정의할 수 있습니다. 두 메서드에 대한 자세한 내용은 [NSX-T Data Center 제품 설명서](#)의 "NSX-T Data Center 관리 가이드"에 있는 "NSX Manager 개요"와 같은 항목에서 확인할 수 있습니다. 정보는 아래의 단계 순서에서도 제공됩니다.

NSX-T 클라우드 계정을 생성한 후에는 한 API 메서드에서 다른 API 메서드로 변환할 수 없습니다. 대신 클라우드 계정을 삭제하고 다른 API 모드를 사용하여 다시 생성해야 합니다.

배포에서 Fault Tolerance 및 고가용성을 용이하게 하기 위해 각 NSX-T 데이터 센터 끝점은 NSX Manager 3개의 클러스터를 나타냅니다.

- vRealize Automation은 NSX Manager 중 하나를 가리킬 수 있습니다. 이 옵션을 사용하면 하나의 NSX Manager가 vRealize Automation에서 API 호출을 받습니다.

- vRealize Automation은 클러스터의 가상 IP를 가리킬 수 있습니다. 이 옵션을 사용하면 하나의 NSX Manager가 VIP의 제어를 담당합니다. 이 NSX Manager는 vRealize Automation에서 API 호출을 수신합니다. 장애가 발생할 경우 클러스터의 다른 노드가 VIP의 제어를 담당하고 vRealize Automation에서 API 호출을 수신합니다.

NSX의 VIP 구성에 대한 자세한 내용은 [VMware NSX-T Data Center 설명서](#)의 "NSX-T Data Center 설치 가이드"에서 "클러스터의 VIP(가상 IP) 주소 구성"을 참조하십시오.

- vRealize Automation은 로드 밸런서 VIP를 가리켜서 세 NSX Manager로 호출을 로드 밸런싱할 수 있습니다. 이 옵션을 사용하면 세 NSX Manager 모두 vRealize Automation에서 API 호출을 수신합니다.

타사 로드 밸런서 또는 NSX-T 로드 밸런서에서 VIP를 구성할 수 있습니다.

대규모 환경에서는 이 옵션을 사용하여 vRealize Automation API 호출을 세 NSX Manager로 분할하는 것이 좋습니다.

#### 사전 요구 사항

- 필요한 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation](#)에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 이 NSX 클라우드 계정에서 사용할 vCenter 클라우드 계정이 있는지 확인합니다. [vRealize Automation](#)에서 vCenter 클라우드 계정 생성 항목을 참조하십시오.
- 클라우드 계정 지원을 위해 포트와 프로토콜을 올바르게 구성했는지 확인합니다. [vRealize Automation 제품 설명서](#)의 "vRealize Easy Installer를 사용하여 vRealize Automation 설치"에서 "vRealize Automation의 포트 및 프로토콜" 항목 및 "vRealize Automation 참조 아키텍처 가이드"에서 "포트 요구 사항" 항목을 참조하십시오.

#### 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭합니다.
- 2 NSX-T 계정 유형을 선택하고 NSX-T 끝점 관리자 인스턴스 또는 VIP에 대한 호스트 IP 주소를 입력합니다. (NSX Manager 및 VIP 옵션과 관련하여 예상되는 동작에 대한 자세한 내용은 위를 참조하십시오.)
- 3 NSX 사용자 이름 및 암호 관리자 자격 증명을 입력하고 **검증**을 클릭합니다.  
계정과 연결된 자산이 수집됩니다.  
NSX 호스트 IP 주소를 사용할 수 없으면 검증이 실패합니다.
- 4 **NSX-T API 메서드**에서 **Manager 메서드** 또는 **정책 메서드**를 선택합니다.
  - Manager API 메서드

이전 버전의 vRealize Automation에서 온보딩 또는 마이그레이션된 기존 NSX-T 끝점 또는 클라우드 계정은 Manager API 메서드 NSX-T 클라우드 계정으로 취급됩니다.

Manager API 메서드는 NSX-T 2.4, NSX-T 3.0 및 NSX-T 3.1 이상에서 지원됩니다.

지금 NSX-T Manager API 메서드를 사용 중인 경우에는 vRealize Automation가 Manager API를 정책 API 마이그레이션 경로에 도입할 때까지 Manager API 메서드를 계속 사용하는 것이 좋습니다.

NSX-T에 대한 일부 vRealize Automation 옵션에는 클라우드 템플릿의 가상 시스템 NIC 구성 요소에 태그를 추가하는 것을 포함하여, NSX-T 3.0 이상이 필요합니다.

#### ■ 정책 API 메서드(기본값)

정책 API 메서드는 NSX-T 3.0 및 NSX-T 3.1 이상에서 사용할 수 있습니다. 이 옵션을 사용하면 vRealize Automation가 NSX-T 정책 API에서 제공되는 추가 기능을 사용할 수 있습니다.

vRealize Automation 8.2에 정책 API 메서드를 도입하기 전에 생성된 기존 NSX-T 클라우드 계정이 있는 경우 Manager API 메서드를 사용합니다. vRealize Automation에서 Manager API에서 정책 API로의 마이그레이션 도구를 사용할 수 있게 될 때까지 기다리는 것이 좋습니다. 기다리지 않으려면 기존 NSX-T 클라우드 계정을 정책 API 메서드를 지정하는 새 NSX-T 클라우드 계정으로 바꿔야 합니다.

- 5 연결에서 이 NSX-T 클라우드 계정과 연결할 하나 이상의 vCenter 클라우드 계정을 추가합니다. 기존 vCenter 클라우드 계정 연결을 제거할 수도 있습니다.

현재 vRealize Automation에 NSX-T 또는 NSX-V 클라우드 계정에 연결되지 않은 vCenter 클라우드 계정만 선택할 수 있습니다.

vRealize Automation에서 여러 vCenter에 대한 NSX-T 매핑으로 수행할 수 있는 작업의 내용을 참조하십시오.

클라우드 템플릿을 배포한 후 연결 변경을 수행하는 방법 또는 클라우드 템플릿을 배포한 후 클라우드 계정을 삭제하는 방법에 대한 자세한 내용은 vRealize Automation에서 NSX 클라우드 계정 연결을 제거하면 어떻게 됩니까? 항목을 참조하십시오.

- 6 태그 지정 전략을 지원하기 위해 태그를 추가하려면 기능 태그를 입력합니다.

나중에 기능 태그를 추가하거나 제거할 수 있습니다. 태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법의 내용을 참조하십시오.



기능 태그 및 제약 조건 태그가 배포 배치를 제어하는 데 도움이 되는 방법에 대한 자세한 내용은 제약 조건 태그 및 배치 비디오 자습서를 참조하십시오.

- 7 저장을 클릭합니다.

#### 다음에 수행할 작업

이 NSX 클라우드 계정에 연결할 vCenter 클라우드 계정을 생성하거나 편집할 수 있습니다. vRealize Automation에서 vCenter 클라우드 계정 생성의 내용을 참조하십시오.

이 클라우드 계정에서 사용하는 데이터 센터에 사용할 하나 이상의 클라우드 영역을 생성하고 구성합니다. [vRealize Automation Cloud Assembly 클라우드 영역에 대해 알아보기](#) 항목을 참조하십시오.

이 클라우드 계정에 대한 인프라 리소스를 구성합니다. [장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축](#)의 내용을 참조하십시오.

## vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성

클라우드 관리자는 팀에서 vRealize Automation 클라우드 템플릿을 배포할 계정 지역에 대한 VMware Cloud on AWS 클라우드 계정을 생성할 수 있습니다.

VMware Cloud on AWS는 vRealize Automation에서 고유한 구성 절차가 필요합니다. 클라우드 계정에 대한 API 토큰 값 설정 및 클라우드 프록시에 대한 게이트웨이 방화벽 규칙 설정을 비롯하여, VMware Cloud on AWS에 대해 vRealize Automation를 올바르게 구성하려면 [자습서: vRealize Automation에 대해 VMware Cloud on AWS 구성 워크플로](#)를 참조하십시오.

### 사전 요구 사항

- vCenter의 대상 SDDC에 대한 VMware Cloud on AWS CloudAdmin 자격 증명을 비롯해 필요한 VMware Cloud on AWS 관리자 자격 증명에 있는지 그리고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목](#)을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 외부 인터넷에 액세스할 수 없는 경우 인터넷 서버 프록시를 구성합니다. [vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법](#)의 내용을 참조하십시오.
- SDDC에서 필요한 액세스 및 방화벽 규칙을 구성했는지 확인합니다. [vRealize Automation에서 VMware Cloud on AWS 클라우드 계정에 연결할 VMware Cloud on AWS SDDC 준비](#)의 내용을 참조하십시오.

### 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭한 후 VMware Cloud on AWS 계정 유형을 선택합니다.
- 2 사용 가능한 SDDC에 액세스하려면 조직의 **VMC API 토큰**을 추가합니다.  
새 토큰을 생성하거나 연결된 **API 토큰** 페이지에서 조직에 대한 기존 토큰을 사용할 수 있습니다. 자세한 내용은 [샘플 워크플로 내의 vRealize Automation에서 VMware Cloud on AWS 클라우드 계정 생성](#)의 내용을 참조하십시오.
- 3 배포에 사용할 SDDC를 선택합니다.  
NSX-V SDDC는 지원되지 않으며 목록에 표시되지 않습니다.  
vCenter 및 NSX-T Manager IP 주소/FQDN 값은 SDDC를 기반으로 자동으로 채워집니다.

- 4 기본값인 cloudadmin@vmc.local이 아닌 경우 지정된 SDDC에 대한 vCenter 사용자 이름 및 암호를 입력합니다.
- 5 **검증**을 클릭하여 지정된 vCenter에 대한 액세스 권한을 확인하고 vCenter가 실행 중인지 확인합니다. 계정과 연결된 데이터 센터가 수집됩니다.
- 6 효율성을 위해 선택한 SDDC에 프로비저닝을 위한 클라우드 영역을 생성합니다. 조직의 클라우드 전략에 따라 별도의 단계로 클라우드 영역을 생성할 수도 있습니다.
- 7 태그 지정 전략을 지원하기 위해 태그를 추가하려면 기능 태그를 입력합니다.

나중에 기능 태그를 추가하거나 제거할 수 있습니다. [태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#)의 내용을 참조하십시오.



기능 태그 및 제약 조건 태그가 배포 배치를 제어하는 데 도움이 되는 방법에 대한 자세한 내용은 [제약 조건 태그 및 배치](#) 비디오 자습서를 참조하십시오.

vSphere에 배포된 VM과 마찬가지로 VMware Cloud on AWS에 배포할 VM에 대해 시스템 태그를 구성할 수 있습니다. 초기 배포 후에 시스템 태그를 업데이트할 수도 있습니다. 이러한 시스템 태그를 사용하면 배포 중에 vRealize Automation에서 VM을 적절한 NSX-T 보안 그룹에 동적으로 할당할 수 있습니다. 관련 정보는 [vRealize Automation 클라우드 템플릿에서 보안 그룹 리소스 사용 항목](#)을 참조하십시오.

- 8 **저장**을 클릭합니다.

## 결과

클라우드 계정이 추가되고 선택한 SDDC를 지정된 클라우드 영역에서 사용할 수 있습니다.

## 다음에 수행할 작업

VMware Cloud on AWS에 대해 vRealize Automation를 올바르게 구성하려면 [자습서: vRealize Automation에 대해 VMware Cloud on AWS 구성](#)의 내용을 참조하십시오.

vRealize Automation 외부의 VMware Cloud on AWS에 대한 관련 정보는 [VMware Cloud on AWS 설명서](#)를 참조하십시오.

## VMware Cloud Foundation 클라우드 계정 생성

VCF(VMware Cloud Foundation)를 vRealize Automation Cloud Assembly 내에서 클라우드 계정으로 구성하여 워크로드 도메인을 사용할 수 있습니다.

VCF 클라우드 계정을 사용하면 VCF 워크로드를 Cloud Assembly에 통합하여 포괄적인 하이브리드 클라우드 관리 솔루션을 용이하게 할 수 있습니다. Cloud Assembly는 VCF 클라우드 계정 구성 페이지를 활성화할 수 있는 몇 가지 진입점을 제공합니다. SDDC 통합 [워크로드 도메인] 탭에서 **클라우드 계정 추가** 버튼을 사용하여 이 페이지에 액세스하면 vCenter 및 NSX Manager에 대한 기본 정보를 포함하여 워크로드가 미리 선택되어 있습니다.

## 사전 요구 사항

이 클라우드 계정에 사용할 vRealize Automation Cloud Assembly 통합으로 구성된 VMware SDDC Manager 4.1 이상의 인스턴스가 있어야 합니다. 자세한 내용은 [VMware SDDC Manager 통합 구성 항목](#)을 참조하십시오.

## 절차

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하고 **클라우드 계정 추가**를 클릭합니다.
- 2 VCF 클라우드 계정 유형을 선택하고 **이름**과 **설명**을 입력합니다.
- 3 이 클라우드 계정으로 사용하는 SDDC Manager 인스턴스에 대한 FQDN 및 자격 증명을 입력합니다.  
이 계정으로 사용할 SDDC 관리자 인스턴스를 이미 구성한 경우에는 이 단계를 건너뛸 수 있습니다.
- 4 이 VCF 클라우드 계정에 사용할 하나 이상의 워크로드 도메인을 선택합니다.
- 5 Cloud Assembly에서 vCenter 및 NSX에 대한 Cloud Foundation 관리 서비스 자격 증명을 사용하도록 하려면 **서비스 자격 증명 자동 생성**을 선택합니다. 나중에 이러한 자격 증명을 변경하려면 암호 관리를 위해 VCF 메커니즘을 사용해야 합니다.  
  
이 옵션을 선택하면 7단계와 8단계를 건너뛸 수 있습니다.
- 6 이 클라우드 계정과 연결된 vCenter에 액세스하는 데 필요한 자격 증명을 입력합니다.
- 7 NSX Manager 머리글 아래에서 VCF 클라우드 계정에 대한 자격 증명을 수동으로 입력하려면 NSX 자격 증명을 입력하고, Cloud Assembly에서 NSX 자격 증명을 생성하고 검증하도록 하려면 [서비스 자격 증명 생성 및 검증]을 클릭합니다.
- 8 이 클라우드 계정과 연결된 NSX-T 네트워크에 액세스하는 데 필요한 자격 증명을 입력합니다.
- 9 해당하는 경우 NSX 모드를 선택합니다.
- 10 **검증**을 클릭하여 SDDC Manager에 대한 연결을 확인합니다.
- 11 해당하는 경우 [구성] 머리글 아래에서 프로비저닝할 데이터 센터를 선택합니다. 선택한 데이터 센터에 대한 클라우드 영역을 생성하려면 확인란을 클릭합니다.
- 12 태그 지정 전략을 지원하는 태그를 사용하는 경우에는 기능 태그를 입력합니다. **태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법** 및 **태그 지정 전략 생성 항목**을 참조하십시오.
- 13 **저장**을 클릭합니다.

## 결과

이 클라우드 계정은 지정된 SDDC Manager와 연결된 선택한 워크로드 도메인을 vRealize Automation Cloud Assembly로 가져와서 사용할 수 있도록 합니다.

vRealize Automation을 사용하여 추가 워크로드 도메인을 관리하려면 각 도메인에 대해 이 프로세스를 반복해야 합니다.

## 다음에 수행할 작업

VCF 클라우드 계정을 구성한 후 기본 클라우드 계정 페이지에서 계정을 선택하고 **클라우드 설정**을 클릭하여 클라우드를 구성할 VMware Cloud Foundation 빠른 시작 마법사를 시작할 수 있습니다.

빠른 시작 마법사에 대한 자세한 내용은 [../Getting-Started-Cloud-Assembly/GUID-BDC673B9-D2AD-47BC-93C5-8C500074F931.html](https://docs.vmware.com/en/Getting-Started-Cloud-Assembly/GUID-BDC673B9-D2AD-47BC-93C5-8C500074F931.html) 항목을 참조하십시오.

## vRealize Automation를 다른 애플리케이션과 통합

통합을 통해 vRealize Automation에 외부 시스템을 추가할 수 있습니다.

통합에는 vRealize Orchestrator, 구성 관리 및 기타 외부 시스템(예: GitHub, Ansible, Puppet) 및 Infoblox와 같은 외부 IPAM 제공자가 포함됩니다.

---

**참고** 외부 인터넷에 액세스할 수 없으며 통합에 필요한 경우 인터넷 서버 프록시를 구성할 수 있습니다. [vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법](#)의 내용을 참조하십시오.

---

## vRealize Automation Cloud Assembly에서 Git 통합을 사용하는 방법

vRealize Automation Cloud Assembly는 소스 제어 하에 VMware Cloud Templates 및 작업 스크립트를 관리할 수 있도록 다양한 Git 저장소 버전과의 통합을 지원합니다. 이 기능은 배포와 관련된 프로세스의 감사 및 책임을 용이하게 합니다.

vRealize Automation Cloud Assembly는 다음 목록에 설명된 대로 다양한 버전의 Git 저장소를 지원합니다. 이러한 각 옵션은 별도의 통합입니다.

- GitHub 클라우드, GitHub Enterprise 온-프레미스
- GitLab 클라우드, GitLab Enterprise 온-프레미스
- BitBucket 온-프레미스

vRealize Automation Cloud Assembly로 Git 통합을 설정하려면 지정된 모든 사용자에게 대한 액세스 권한으로 구성된 적절한 로컬 Git 저장소가 있어야 합니다. 또한 클라우드 템플릿을 Git에서 검색할 수 있도록 특정 구조로 저장해야 합니다. GitLab 또는 GitHub와의 통합을 생성하려면 vRealize Automation Cloud Assembly에서 **인프라 > 연결 > 통합**을 선택한 다음 적절한 항목을 선택합니다. 대상 저장소에 대한 URL과 토큰이 필요합니다.

Git 통합이 기존 저장소에 구성되면 선택된 프로젝트와 연결된 모든 클라우드 템플릿을 자격 있는 사용자가 사용할 수 있게 됩니다. 이러한 템플릿은 기존 배포와 함께 사용하거나 새 배포의 기초로 사용할 수 있습니다. 프로젝트를 추가할 때 Git에서 프로젝트가 저장되는 위치와 방법에 관한 몇 가지 속성을 선택해야 합니다.

vRealize Automation Cloud Assembly에서 직접 Git 저장소에 작업을 저장할 수 있습니다. 작업 스크립트 버전을 Git에 직접 지정하거나 vRealize Automation Cloud Assembly에서 버전을 생성할 수도 있습니다. vRealize Automation Cloud Assembly에서 작업 버전을 생성하면 자동으로 Git에 버전으로 저장됩니다. 클라우드 템플릿은 조금 더 복잡합니다. vRealize Automation Cloud Assembly에서 Git 통합에 직접 추가할 수 없기 때문입니다. Git 인스턴스에 직접 저장해야 합니다. 그런 다음, vRealize Automation Cloud Assembly에서 클라우드 템플릿 관리 페이지로 작업할 때 Git에서 검색할 수 있습니다.

## 시작하기 전에

클라우드 템플릿을 GitLab 또는 GitHub에서 검색할 수 있도록 특정 구조로 생성하고 저장해야 합니다.

- GitLab에 올바르게 통합되도록 클라우드 템플릿을 구성하고 저장합니다. 유효한 템플릿만 GitLab으로 가져올 수 있습니다.
  - 클라우드 템플릿에 대해 지정된 폴더를 하나 이상 생성합니다.
  - 모든 클라우드 템플릿은 `blueprint.yaml` 파일 내에 저장되어야 합니다.
  - 템플릿 맨 위에 `name:` 및 `version:` 속성이 포함되어 있는지 확인합니다.
- 해당하는 저장소의 API 키를 추출합니다. Git 계정에서 오른쪽 상단 모서리에 있는 로그인을 선택하고 [설정] 메뉴로 이동합니다. **액세스 토큰**을 선택하고 토큰에 이름을 지정한 후 만료 날짜를 설정합니다. 그런 다음, API를 선택하고 토큰을 생성합니다. 결과 값을 복사하여 저장합니다.

Git 통합에 사용되는 모든 클라우드 템플릿에 대해 다음 지침을 준수해야 합니다.

- 각 클라우드 템플릿은 별도의 폴더에 상주해야 합니다.
- 모든 클라우드 템플릿의 이름은 `blueprint.yaml`로 지정해야 합니다.
- 모든 클라우드 템플릿 YAML 파일은 `name` 및 `version` 필드를 사용해야 합니다.
- 유효한 클라우드 템플릿만 가져올 수 있습니다.
- Git에서 가져온 초안 클라우드 템플릿을 업데이트할 때 해당 콘텐츠가 최상위 버전의 콘텐츠와 다르면 이후 동기화 시 초안이 업데이트되지 않고 새 버전이 생성됩니다. 템플릿을 업데이트하고 Git의 추가 동기화도 허용하려면 마지막 변경 후 새 버전을 생성해야 합니다.
- **vRealize Automation Cloud Assembly에서 GitLab 클라우드 템플릿 통합 구성**  
 이 절차에서는 저장소에서 클라우드 템플릿으로 작업하고 지정된 프로젝트와 연결되어 있는 저장된 템플릿을 자동으로 다운로드할 수 있도록 vRealize Automation Cloud Assembly에서 GitLab 통합을 구성하는 방법을 보여줍니다. GitLab에서 클라우드 템플릿을 사용하려면 적절한 GitLab 인스턴스에 대한 연결을 생성한 다음, 원하는 템플릿을 해당 인스턴스에 저장해야 합니다.
- **vRealize Automation Cloud Assembly에서 GitHub 통합 구성**  
 vRealize Automation Cloud Assembly에서 GitHub 클라우드 기반 저장소 호스팅 서비스를 통합할 수 있습니다.

## ■ vRealize Automation Cloud Assembly에서 Bitbucket 통합 구성

vRealize Automation Cloud Assembly는 ABX 작업 스크립트 및 VMware Cloud Templates에 대한 Git 기반 저장소로 사용하기 위해 Bitbucket과의 통합을 지원합니다.

## vRealize Automation Cloud Assembly에서 GitLab 클라우드 템플릿 통합 구성

이 절차에서는 저장소에서 클라우드 템플릿으로 작업하고 지정된 프로젝트와 연결되어 있는 저장된 템플릿을 자동으로 다운로드할 수 있도록 vRealize Automation Cloud Assembly에서 GitLab 통합을 구성하는 방법을 보여줍니다. GitLab에서 클라우드 템플릿을 사용하려면 적절한 GitLab 인스턴스에 대한 연결을 생성한 다음, 원하는 템플릿을 해당 인스턴스에 저장해야 합니다.

GitLab 통합이 기존 저장소에 구성되면 선택된 프로젝트와 연결된 모든 클라우드 템플릿을 자격 있는 사용자가 사용할 수 있게 됩니다. 이러한 템플릿은 기존 배포와 함께 사용하거나 새 배포의 기초로 사용할 수 있습니다. 프로젝트를 추가할 때 GitLab에서 프로젝트가 저장되는 위치와 방법에 관한 몇 가지 속성을 선택해야 합니다.

---

**참고** 새로운 또는 업데이트된 클라우드 템플릿을 vRealize Automation Cloud Assembly에서 Git 저장소로 푸시할 수 없습니다. 또한 새 템플릿을 vRealize Automation Cloud Assembly에서 저장소로 푸시할 수 없습니다. 클라우드 템플릿을 저장소에 추가하려면 개발자가 Git 인터페이스를 사용해야 합니다.

---

Git에서 가져온 초안 클라우드 템플릿을 업데이트할 때 해당 콘텐츠가 최상위 버전의 콘텐츠와 다르다면 이후 동기화 시 초안이 업데이트되지 않고 새 버전이 생성됩니다. 클라우드 템플릿을 업데이트하고 Git의 추가 동기화도 허용하려면 마지막 변경 후 새 버전을 생성해야 합니다.

GitLab에서 사용할 클라우드 템플릿을 설정하고 필요한 정보를 수집한 후에는 GitLab 인스턴스와 통합을 설정해야 합니다. 그런 다음, 지정된 클라우드 템플릿을 GitLab으로 가져올 수 있습니다. 이 절차에 대한 비디오 데모는 <https://www.youtube.com/watch?v=h0vqo63Sdgg>에서 볼 수 있습니다.

### 사전 요구 사항

- 해당하는 저장소의 API 키를 추출합니다. GitLab 계정에서 오른쪽 상단 모서리에 있는 로그인을 선택하고 [설정] 메뉴로 이동합니다. [액세스 토큰]을 선택하고 토큰에 이름을 지정한 후 만료 날짜를 설정합니다. 그런 다음, API를 선택하고 토큰을 생성합니다. 결과 값을 복사하여 저장합니다.

vRealize Automation Cloud Assembly로 Git 통합을 설정하려면 지정된 모든 사용자에게 대한 액세스 권한으로 구성된 적절한 로컬 Git 저장소가 있어야 합니다. 또한 클라우드 템플릿을 GitLab에서 검색할 수 있도록 특정 구조로 생성하고 저장해야 합니다.

- GitLab에 올바르게 통합되도록 클라우드 템플릿을 구성하고 저장합니다. 유효한 템플릿만 GitLab으로 가져올 수 있습니다. vRealize Automation Cloud Assembly에서 Git 통합을 사용하는 방법의 내용을 참조하십시오.

## 절차

- 1 vRealize Automation Cloud Assembly에서 GitLab 환경과 통합을 설정합니다.
  - a **인프라 > 통합 > 새로 추가**를 선택하고 GitLab을 선택합니다.
  - b GitLab 인스턴스에 대한 **URL**을 입력합니다. SaaS(Software as a Service) GitLab은 대부분의 경우 **gitlab.com**입니다.
  - c 지정된 GitLab 인스턴스에 대한 **토큰**(API 키라고도 함)을 입력합니다. GitLab 인스턴스에서 토큰을 추출하는 방법에 대한 자세한 내용은 위의 사전 요구 사항을 참조하십시오.
  - d 적절한 이름과 설명을 추가합니다.
  - e **검증**을 클릭하여 연결을 확인합니다.
  - f 필요한 경우 기능 태그를 추가합니다. 자세한 내용은 [vRealize Automation Cloud Assembly에서 기능 태그 사용](#)의 내용을 참조하십시오.
  - g **추가**를 클릭합니다.
- 2 적절한 저장소에서 클라우드 템플릿을 수락하도록 GitLab 연결을 구성합니다.
  - a **인프라 > 통합**을 선택하고 적절한 GitLab 통합을 선택합니다.
  - b **프로젝트**를 선택합니다.
  - c **새 프로젝트**를 선택하고 프로젝트의 이름을 생성합니다.
  - d GitLab 내 **저장소** 경로를 입력합니다. 일반적으로 이는 저장소 이름에 추가된 기본 계정의 사용자 이름입니다.
  - e 사용하려는 적절한 GitLab **분기**를 입력합니다.
  - f 해당하는 경우 **폴더** 이름을 입력합니다. 비워 두면 모든 폴더를 사용할 수 있습니다.
  - g 적절한 **유형**을 입력합니다. 해당하는 경우 폴더 이름을 입력합니다. 비워 두면 모든 폴더를 사용할 수 있습니다.
  - h **다음**을 클릭하여 저장소 추가를 완료합니다.

**다음**을 클릭하면 자동화된 동기화 작업이 시작되어 클라우드 템플릿을 플랫폼으로 가져옵니다. 동기화 작업이 완료되면 클라우드 템플릿을 가져왔다는 메시지가 표시됩니다.

## 결과

이제 GitLab에서 클라우드 템플릿을 검색할 수 있습니다.

## vRealize Automation Cloud Assembly에서 GitHub 통합 구성

vRealize Automation Cloud Assembly에서 GitHub 클라우드 기반 저장소 호스팅 서비스를 통합할 수 있습니다.

vRealize Automation Cloud Assembly에서 GitHub 통합을 구성하려면 올바른 GitHub 토큰이 필요합니다. 토큰 생성 및 찾기에 대한 자세한 내용은 [GitHub 설명서](#)를 참조하십시오.

## 사전 요구 사항

- GitHub에 액세스할 수 있어야 합니다.
- GitHub와 올바르게 통합되도록 클라우드 템플릿을 구성하고 저장합니다. 유효한 클라우드 템플릿만 GitHub로 가져올 수 있습니다. vRealize Automation Cloud Assembly에서 Git 통합을 사용하는 방법의 내용을 참조하십시오.

## 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 GitHub를 선택합니다.
- 3 [GitHub 구성] 페이지에 필요한 정보를 입력합니다.
- 4 **검증**을 클릭하여 통합을 확인합니다.
- 5 태그 지정 전략을 지원하기 위해 태그를 추가해야 하는 경우에는 기능 태그를 입력합니다. 태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법 및 태그 지정 전략 생성 항목을 참조하십시오.
- 6 **추가**를 클릭합니다.
- 7 적절한 저장소에서 클라우드 템플릿을 수락하도록 GitLab 연결을 구성합니다.
  - a **인프라 > 통합**을 선택하고 적절한 GitHub 통합을 선택합니다.
  - b **프로젝트**를 선택합니다.
  - c **새 프로젝트**를 선택하고 프로젝트의 이름을 생성합니다.
  - d GitHub 내 **저장소** 경로를 입력합니다. 일반적으로 이는 저장소 이름에 추가된 기본 계정의 사용자 이름입니다.
  - e 사용하려는 적절한 GitHub **분기**를 입력합니다.
  - f 해당하는 경우 **폴더** 이름을 입력합니다. 비워 두면 모든 폴더를 사용할 수 있습니다.
  - g 적절한 **유형**을 입력합니다.
  - h **다음**을 클릭하여 저장소 추가를 완료합니다.

자동화된 동기화 작업이 시작되어 클라우드 템플릿을 플랫폼으로 가져옵니다.

동기화 작업이 완료되면 클라우드 템플릿을 가져왔다는 메시지가 표시됩니다.

## 결과

GitHub는 vRealize Automation Cloud Assembly Blueprint에서 사용할 수 있습니다.

## 다음에 수행할 작업

이제 GitHub에서 클라우드 템플릿을 검색할 수 있습니다.

## vRealize Automation Cloud Assembly에서 Bitbucket 통합 구성

vRealize Automation Cloud Assembly는 ABX 작업 스크립트 및 VMware Cloud Templates에 대한 Git 기반 저장소로 사용하기 위해 Bitbucket과의 통합을 지원합니다.

vRealize Automation Cloud Assembly에서는 Bitbucket 통합을 사용하여 두 가지 유형의 저장소 항목을 사용할 수 있습니다(VMware Cloud Templates 또는 ABX 작업 스크립트). Bitbucket 통합을 사용하기 전에 사용할 프로젝트를 동기화해야 합니다. ABX 작업은 Bitbucket 저장소에 다시 쓰기를 지원하지만 통합에서 클라우드 템플릿을 다시 쓸 수는 없습니다. 새 버전의 클라우드 템플릿 파일을 생성하려면 수동으로 수행해야 합니다.

### 사전 요구 사항

- 배포에 사용하려는 하나 이상의 ABX 또는 클라우드 템플릿 기반 프로젝트를 사용하여 온-프레미스 Bitbucket 서버 배포를 설정합니다. Bitbucket 클라우드는 현재 지원되지 않습니다.
- Bitbucket 통합을 연결할 vRealize Automation Cloud Assembly 프로젝트를 생성하거나 지정합니다.
- Bitbucket 통합에 동기화할 클라우드 템플릿 파일의 이름은 `blueprint.yaml`이어야 합니다.

### 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 Bitbucket을 선택합니다.
- 3 Bitbucket 새 통합의 [요약] 페이지에 요약 정보 및 Bitbucket 자격 증명을 입력합니다.
- 4 통합을 확인하려면 **검증**을 클릭합니다.
- 5 태그 지정 전략을 지원하는 태그를 추가하는 경우에는 기능 태그를 입력합니다. **태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법** 및 **태그 지정 전략 생성 항목**을 참조하십시오.
- 6 **추가**를 클릭합니다.
- 7 Bitbucket 통합에 대한 기본 페이지에서 [프로젝트] 탭을 선택하여 프로젝트를 이 Bitbucket 통합과 연결합니다.
- 8 이 Bitbucket 통합과 연결할 프로젝트를 선택합니다.
- 9 **다음**을 클릭하여 저장소를 Bitbucket 프로젝트에 추가하고, 추가하는 저장소 유형을 지정한 다음 **저장소** 이름과 **분기** 및 **폴더**를 지정합니다.
- 10 **추가**를 클릭합니다.

프로젝트에 하나 이상의 저장소를 추가하려는 경우 **저장소 추가**를 클릭합니다.

## 결과

Bitbucket 통합은 지정된 저장소 구성으로 구성되며, 사용자는 구성된 저장소에 포함된 ABX 작업 및 클라우드 템플릿을 보고 사용할 수 있습니다. Bitbucket 통합에 프로젝트를 추가하는 경우 동기화 작업이 실행되어 지정된 저장소에서 최신 버전의 ABX 작업 스크립트 및 클라우드 템플릿 파일을 끌어옵니다.

[Bitbucket 통합] 페이지의 [기록] 탭에는 통합에 대한 모든 동기화 작업의 기록이 표시됩니다. 기본적으로 파일은 15분마다 자동으로 동기화되지만 파일을 선택하고 언제든지 **동기화**를 클릭하여 수동으로 파일을 동기화할 수 있습니다.

## 다음에 수행할 작업

vRealize Automation Cloud Assembly의 [확장성] 페이지에서 ABX 작업을 사용할 수 있으며 [설계] 페이지에서 클라우드 템플릿을 사용할 수 있습니다. vRealize Automation Cloud Assembly의 [확장성] 영역에 ABX 작업의 변경된 버전을 저장하면 새 버전의 스크립트가 생성되고 저장소에 다시 기록됩니다.

## vRealize Automation에서 외부 IPAM 통합을 구성하는 방법

제공자별 외부 IPAM 통합 지점을 생성하여 클라우드 템플릿 배포에 사용되는 IP 주소를 관리할 수 있습니다. 외부 IPAM 통합 지점을 사용하는 경우 IP 주소는 vRealize Automation가 아니라 지정된 IPAM 제공자에서 가져오고 이 제공자가 관리합니다.

vRealize Automation의 VM 및 클라우드 템플릿 배포에 대한 IP 주소 및 DNS 설정을 관리하기 위해 제공자별 IPAM 통합 지점을 생성할 수 있습니다.

사전 요구 사항을 구성하는 방법과 샘플 워크플로 컨텍스트 내에서 제공자별 외부 IPAM 통합 지점을 생성하는 방법에 대한 자세한 내용은 [vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가 항목](#)을 참조하십시오. 이 워크플로는 Infoblox IPAM 통합용이지만 외부 IPAM 벤더에 대한 참조로 사용할 수 있습니다.

외부 IPAM 파트너 및 벤더가 IPAM 솔루션을 vRealize Automation와 통합할 수 있도록 필요한 자산을 생성하는 방법에 대한 자세한 내용은 [IPAM SDK를 사용하여 vRealize Automation에 대한 제공자별 외부 IPAM 통합 패키지를 생성하는 방법](#) 항목을 참조하십시오.

## 사전 요구 사항

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 외부 IPAM 제공자(예: [Infoblox](#) 또는 [Bluecat](#))의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다.
- IPAM 제공자(예: [Infoblox](#) 또는 [BlueCat](#))에 대한 배포된 통합 패키지에 액세스할 수 있는지 확인합니다. 배포된 패키지는 처음에 IPAM 제공자 또는 vRealize Automation 마켓플레이스에서 .zip 다운로드 후 가져온 후 vRealize Automation에 배포됩니다.
- IPAM 제공자에 대해 구성된 실행 환경에 액세스할 수 있는지 확인합니다.

- ABX(Action-Based Extensibility) 온-프레미스 내장형 실행 환경을 사용 중인 경우 vRealize Automation 네트워크에 송신 트래픽을 gcr.io 및 storage.googleapis.com과 같은 외부 사이트로 전달할 수 있는 HTTP 프록시 서버가 있는지 확인합니다. 자세한 내용은 [vRealize Automation 8.x의 프록시](#) 뒤에서 [Docker 이미지 가져오기\(75180\)](#)를 참조하십시오.
- IPAM 벤더 제품에 액세스하고 사용하는 데 필요한 사용자 자격 증명이 있는지 확인합니다. 필요한 사용자 권한에 대한 자세한 내용은 통합 벤더의 제품 설명서를 참조하십시오.

## 절차

1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.

2 **IPAM**을 클릭합니다.

3 **제공자** 드롭다운 목록에서 구성된 IPAM 제공자 패키지를 선택합니다.

목록이 비어 있으면 **제공자 패키지 가져오기**를 클릭하고 기존 제공자 패키지 .zip 파일로 이동한 후 선택합니다. .zip 파일이 없으면 제공자의 웹 사이트나 vRealize Automation **마켓플레이스** 탭에서 가져올 수 있습니다.

4 제공자의 호스트 이름과 같은 다른 필수 필드(있는 경우)와 함께 외부 IPAM 제공자의 계정에 대한 관리자 사용자 이름 및 암호 자격 증명을 입력합니다.

5 **실행 환경** 드롭다운 목록에서 기존 실행 환경(예: 온-프레미스 작업 기반 확장성 통합 지점)을 선택합니다.

실행 환경은 vRealize Automation과 IPAM 제공자 간의 통신을 지원합니다.

IPAM 프레임워크는 ABX(Action-Based Extensibility) 온-프레미스 내장형 실행 환경만 지원합니다.

---

**참고** Amazon Web Services 또는 Microsoft Azure 클라우드 계정을 통합 실행 환경으로 사용하는 경우에는 IPAM 제공자 장치를 인터넷에서 액세스할 수 있고, NAT 또는 방화벽이 뒤에 있지 않으며, 공개적으로 확인할 수 있는 DNS 이름이 있는지 확인하십시오. IPAM 제공자에 액세스할 수 없는 경우 Amazon Web Services Lambda 또는 Microsoft Azure 함수가 연결할 수 없으며 통합이 실패합니다.

---

6 **검증**을 클릭합니다.

7 외부 IPAM 제공자의 자체 서명된 인증서를 신뢰하라는 메시지가 표시되면 **수락**을 클릭합니다.

자체 서명된 인증서를 수락하면 검증 작업을 계속하여 완료할 수 있습니다.

8 이 IPAM 통합 지점의 이름을 입력하고 **추가**를 클릭하여 새 IPAM 통합 지점을 저장합니다.

데이터 수집 작업은 모방됩니다. 네트워크 및 IP 주소는 외부 IPAM 제공자에서 데이터 수집됩니다.

## vRealize Automation에서 최신 외부 IPAM 통합 패키지로 업그레이드하는 방법

기존 외부 IPAM 통합 지점을 업그레이드하여 최신 버전의 벤더 특정 IPAM 통합 패키지를 받을 수 있습니다.

외부 IPAM 공급자나 VMware는 특정 벤더의 소스 IPAM 통합 패키지를 업그레이드할 수 있습니다. 예를 들어 Infoblox용 외부 IPAM 통합 패키지가 여러 번 업그레이드되었습니다. 명명된 IPAM 통합 지점을 사용하는 기존 vRealize Automation 인프라 설정을 유지하려면 새 IPAM 통합 지점을 생성하는 대신 IPAM 통합 지점을 편집하여 업데이트된 IPAM 통합 패키지를 소싱할 수 있습니다.

### 사전 요구 사항

이 절차에서는 이미 외부 IPAM 통합 지점을 생성했고 최신 버전의 벤더 특정 IPAM 통합 패키지를 사용하도록 해당 통합 지점을 업그레이드하려는 경우를 가정합니다.

외부 IPAM 통합 지점 생성 방법에 대한 자세한 내용은 vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가 항목을 참조하십시오.

- 클라우드 관리자 자격 증명이 있는지 확인합니다. vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. vRealize Automation 사용자 역할이란? 항목을 참조하십시오.
- 외부 IPAM 제공자의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다.
- IPAM 제공자를 위해 배포된 통합 패키지에 액세스할 수 있는지 확인합니다. 배포된 패키지는 처음에 IPAM 제공자 웹 사이트 또는 vRealize Automation 마켓플레이스에서 .zip 다운로드로 가져온 후 vRealize Automation에 배포됩니다.

제공자 패키지 .zip 파일을 다운로드 및 배포하고 IPAM 통합 페이지에서 **제공자** 값으로 사용할 수 있도록 설정하는 방법에 대한 자세한 내용은 vRealize Automation에서 사용할 외부 IPAM 제공자 패키지 다운로드 및 배포 항목을 참조하십시오.

- IPAM 제공자에 대해 구성된 실행 환경에 액세스할 수 있는지 확인합니다. 실행 환경은 일반적으로 ABX(작업 기반 확장성) 온-프레미스 내장형 통합 지점입니다.

실행 환경 특성에 대한 자세한 내용은 vRealize Automation에서 IPAM 통합 지점에 대한 실행 환경 생성 항목을 참조하십시오.

### 절차

- 1 **인프라 > 연결 > 통합 IPAM**을 선택하고 기존 IPAM 통합 지점을 엽니다.
- 2 **제공자 관리**를 클릭합니다.
- 3 업데이트된 IPAM 통합 패키지로 이동하여 가져옵니다.
- 4 **검증**을 클릭한 후 **저장**을 클릭합니다.

## vRealize Automation Cloud Assembly에서 My VMware 통합 구성

My VMware를 vRealize Automation Cloud Assembly와 통합하여 클라우드 템플릿을 위한 VMware Marketplace 액세스와 같은 VMware 관련 작업 및 기능을 지원할 수 있습니다.

조직별로 하나의 My VMware 통합만 생성할 수 있습니다.

## 사전 요구 사항

My VMware에 대한 적절한 사용 권한이 있는 사용자 계정이 있어야 합니다.

- 사용자를 My VMware 계정에 초대하는 것에 대한 자세한 내용은 [KB 2070555](#)를 참조하십시오.
- My VMware 계정에서 사용자 사용 권한을 할당하는 것에 대한 자세한 내용은 [KB 2006977](#)을 참조하십시오.

## 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 [My VMware]를 선택합니다.
- 3 My VMware 구성 페이지에 필요한 정보를 입력합니다.
- 4 태그 지정 전략을 지원하는 태그가 필요한 경우에는 기능 태그를 입력합니다. **태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법 및 태그 지정 전략 생성 항목**을 참조하십시오.
- 5 **추가**를 클릭합니다.

## 결과

My VMware를 클라우드 템플릿에서 사용할 수 있습니다.

## 다음에 수행할 작업

My VMware 구성 요소를 원하는 클라우드 템플릿에 추가합니다.

## Cloud Assembly에서 vRealize Orchestrator 통합 구성

하나 이상의 vRealize Orchestrator 통합을 구성하면 확장성 및 클라우드 템플릿의 일부로 워크플로를 사용할 수 있습니다.

vRealize Automation에는 미리 구성된 내장된 vRealize Orchestrator 인스턴스가 포함됩니다. vRealize Automation Cloud Services 콘솔에서 내장된 vRealize Orchestrator의 클라이언트에 액세스할 수 있습니다.

---

**참고** [https://your\\_vRA\\_FQDN/vco-controlcenter](https://your_vRA_FQDN/vco-controlcenter)로 이동하여 **root**로 로그인하면 내장된 vRealize Orchestrator의 제어 센터에 액세스할 수 있습니다.

---

클라우드 템플릿에 사용되는 XaaS(Anything as a Service) 작업 및 vRealize Automation 확장성 구독에 사용하기 위해 외부 vRealize Orchestrator 인스턴스를 통합할 수도 있습니다.

## 사전 요구 사항

- 클라우드 관리자 자격 증명이 있는지 확인합니다. **vRealize Automation**에서 클라우드 계정 작업에 필요한 **자격 증명**의 내용을 참조하십시오.
- vRealize Orchestrator 8.3로 업그레이드하거나 마이그레이션합니다. "VMware vRealize Orchestrator 업그레이드 및 마이그레이션"을 참조하십시오.

**절차**

- 1 **인프라 > 연결 > 통합**을 선택합니다.
- 2 **통합 추가**를 클릭합니다.
- 3 **vRealize Orchestrator**를 선택합니다.
- 4 vRealize Orchestrator 통합의 이름을 입력합니다.
- 5 (선택 사항) vRealize Orchestrator 통합에 대한 설명을 입력합니다.
- 6 **vRealize Orchestrator URL**에서 외부 vRealize Orchestrator 인스턴스의 FQDN(정규화된 도메인 이름)을 입력합니다.  
  
예를 들어 `https://my_vRO_FQDN.com:443`과 같이 입력합니다.
- 7 통합을 검증하려면 **검증**을 클릭합니다.
- 8 (선택 사항) 메시지가 표시되면 인증서 정보를 검토하고 **수락**을 클릭합니다.
- 9 (선택 사항) 기능 태그를 추가합니다. 기능 태그에 대한 자세한 내용은 [vRealize Automation Cloud Assembly에서 기능 태그 사용 항목](#)을 참조하십시오.

---

**참고** 기능 태그를 사용하여 여러 vRealize Orchestrator 통합을 관리할 수 있습니다. [프로젝트 제약 조건으로 여러 vRealize Orchestrator 통합 관리](#)의 내용을 참조하십시오.

---

- 10 **추가**를 클릭합니다.  
  
vRealize Orchestrator 통합이 저장됩니다.
- 11 통합이 구성되었고 워크플로가 추가되었는지 확인하려면 **확장성 > 라이브러리 > 워크플로**를 선택합니다.

**다음에 수행할 작업**

통합된 외부 vRealize Orchestrator 클라이언트에 액세스:

- 1 vRealize Automation 클라우드 서비스 콘솔로 이동합니다.
- 2 **Orchestrator**를 선택합니다.
- 3 통합된 vRealize Orchestrator 인스턴스에 해당하는 탭을 선택합니다.

---

**참고** 클라우드 관리자 자격 증명이 없는 vRealize Automation Cloud Assembly 사용자는 통합된 vRealize Orchestrator 인스턴스의 탭을 볼 수 없습니다.

---

**프로젝트 제약 조건으로 여러 vRealize Orchestrator 통합 관리**

프로젝트 제약 조건을 사용하여 워크플로 구독에서 사용되는 vRealize Orchestrator 통합을 관리할 수 있습니다.

vRealize Automation Cloud Assembly는 워크플로 구독에서 사용할 수 있는 여러 vRealize Orchestrator 서버의 통합을 지원합니다. 연성 또는 경성 프로젝트 제약 조건으로 프로젝트에서 프로비저닝된 클라우드 템플릿에 사용되는 vRealize Orchestrator 통합을 관리할 수 있습니다. 프로젝트 제약 조건에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 프로젝트 태그 및 사용자 지정 속성 사용 항목](#)을 참조하십시오.

#### 사전 요구 사항

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#)의 내용을 참조하십시오.
- vRealize Automation Cloud Assembly에서 둘 이상의 vRealize Orchestrator 통합을 구성합니다. [Cloud Assembly에서 vRealize Orchestrator 통합 구성](#)의 내용을 참조하십시오.
- vRealize Orchestrator 통합에 기능 태그를 추가합니다. [vRealize Automation Cloud Assembly에서 기능 태그 사용](#)의 내용을 참조하십시오.

#### 절차

- 1 **인프라 > 관리 > 프로젝트**로 이동하여 프로젝트를 선택합니다.
- 2 **프로비저닝** 탭을 선택합니다.
- 3 **확장성 제약 조건** 텍스트 상자에 vRealize Orchestrator 통합의 기능 태그를 입력하고 연성 또는 경성 프로젝트 제약 조건으로 설정합니다.
- 4 **저장**을 클릭합니다.

#### 결과

클라우드 템플릿을 배포하는 경우, vRealize Automation Cloud Assembly는 프로젝트 제약 조건을 사용하여 워크플로 구독에서 사용되는 vRealize Orchestrator 통합을 관리합니다.

#### 다음에 수행할 작업

또는 기능 태그를 사용하여 클라우드 계정 수준에서 여러 vRealize Orchestrator 통합을 관리할 수 있습니다. 자세한 내용은 [클라우드 계정 기능 태그를 사용하여 여러 vRealize Orchestrator 통합 관리](#) 항목을 참조하십시오.

### 클라우드 계정 기능 태그를 사용하여 여러 vRealize Orchestrator 통합 관리

기능 태그를 사용하여 워크플로 구독에서 사용되는 vRealize Orchestrator 통합을 관리할 수 있습니다.

vRealize Automation Cloud Assembly는 워크플로 구독에서 사용할 수 있는 여러 vRealize Orchestrator 서버의 통합을 지원합니다. 클라우드 계정에 기능 태그를 추가하여 워크플로 구독에 사용되는 vRealize Orchestrator 통합을 관리할 수 있습니다.

#### 사전 요구 사항

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#)의 내용을 참조하십시오.

- vRealize Automation Cloud Assembly에서 둘 이상의 vRealize Orchestrator 통합을 구성합니다. 자세한 내용은 [Cloud Assembly에서 vRealize Orchestrator 통합 구성](#) 항목을 참조하십시오.
- vRealize Orchestrator 통합에 기능 태그를 추가합니다. [vRealize Automation Cloud Assembly에서 기능 태그 사용](#)의 내용을 참조하십시오.

## 절차

1 **인프라 > 연결 > 클라우드 계정**으로 이동합니다.

2 클라우드 계정을 선택합니다.

3 사용하려는 vRealize Orchestrator 통합의 기능 태그를 입력합니다.

기능 태그가 연성 제약 조건으로 자동 변환됩니다. 통합을 관리하는 데 경성 제약 조건을 사용하려면 프로젝트 제약 조건을 사용해야 합니다. 자세한 내용은 [프로젝트 제약 조건으로 여러 vRealize Orchestrator 통합 관리](#) 항목을 참조하십시오.

4 **저장**을 클릭합니다.

## 결과

클라우드 템플릿을 배포하는 경우 vRealize Automation Cloud Assembly는 연결된 클라우드 계정의 태그 지정을 사용하여 워크플로 구독에서 사용되는 vRealize Orchestrator 통합을 관리합니다.

## vRealize Automation Cloud Assembly에서 Kubernetes를 사용하는 방법

vRealize Automation Cloud Assembly에는 Kubernetes 리소스를 관리하고 배포하는 몇 가지 옵션이 제공됩니다.

vRealize Automation Cloud Assembly에서 Kubernetes 리소스를 사용하는 기본 옵션은 두 가지입니다. VMware TKGI(Tanzu Kubernetes Grid Integrated Edition)(이전 이름: PKS) 또는 Red Hat OpenShift를 vRealize Automation Cloud Assembly와 통합하여 Kubernetes 리소스를 구성, 관리 및 배포할 수 있습니다. 두 번째 옵션을 사용하면 vCenter 클라우드 계정을 활용하여 감독자 네임스페이스에 액세스하고 vSphere Project Pacific Kubernetes 기반 기능을 사용할 수 있습니다. vRealize Automation Cloud Assembly에서 외부 Kubernetes 리소스를 통합할 수도 있습니다.

### VMware TKGI(Tanzu Kubernetes Grid Integrated Edition) 또는 Openshift 통합을 사용한 작업

TKGI, 외부 클러스터 또는 Openshift 구성의 경우, vRealize Automation Cloud Assembly는 적용 가능한 Kubernetes 클러스터에 사용자가 액세스할 수 있도록 Kubeconfig를 제공합니다.

TKGI 또는 OpenShift 통합을 생성하면 적용 가능한 Kubernetes 클러스터를 vRealize Automation Cloud Assembly에서 사용할 수 있고, Kubernetes 구성 요소를 vRealize Automation Cloud Assembly에 추가 및 생성하여 클러스터 및 컨테이너 애플리케이션의 관리를 지원할 수 있습니다. 이러한 애플리케이션은 Service Broker 카탈로그에서 사용할 수 있는 셀프 서비스 배포의 기반을 형성합니다.

## vSphere Project Pacific Kubernetes 클러스터를 사용한 작업

Project Pacific은 Kubernetes를 제어부로 사용하는 vSphere의 향상된 기능이며, 하나의 인터페이스에서 가상 시스템과 컨테이너를 모두 관리할 수 있습니다. vRealize Automation Cloud Assembly를 통해 사용자는 vSphere에 내장된 Pacific Kubernetes 기능을 활용할 수 있습니다. Pacific 기능에는 감독자 클러스터가 포함된 vSphere 구현을 통해 vCenter 배포와의 통합을 생성하여 액세스할 수 있습니다. Pacific을 사용하면 vCenter에서 기존 가상 시스템과 Kubernetes 클러스터를 모두 관리할 수 있습니다.

Pacific 기반 감독자 네임스페이스의 경우, 해당하는 vSphere SSO에 사용자가 액세스할 수 있어야 합니다. 그래야 감독자 네임스페이스 세부 정보에 대해 제공된 링크에 로그인할 수 있습니다. 그러면 vSphere 인증을 통해 사용자 지정된 Kubectl을 다운로드하여 감독자 네임스페이스를 사용할 수 있습니다.

이 기능을 사용하려면 감독자 네임스페이스가 구성되어 있는 vSphere 클라우드 계정이 포함된 vCenter가 있어야 합니다. 사용자가 로그인한 후에 해당하는 네임스페이스 작업을 시작할 수 있습니다.

- **vRealize Automation Cloud Assembly에서 VMware Tanzu Kubernetes Grid Integrated Edition 통합 구성**

온-프레미스 및 클라우드에서 VMware TKGI(Tanzu Kubernetes Grid Integrated Edition)(이전 이름: PKS) 리소스 연결을 구성하여 vRealize Automation Cloud Assembly의 Kubernetes 통합 및 관리 기능을 지원할 수 있습니다.

- **vRealize Automation Cloud Assembly에서 Red Hat OpenShift 통합 구성**

온-프레미스 및 클라우드에서 Red Hat OpenShift 리소스 연결을 구성하여 vRealize Automation Cloud Assembly의 엔터프라이즈 수준 Kubernetes 통합 및 관리 기능을 지원할 수 있습니다.

- **vRealize Automation Cloud Assembly에서 Kubernetes 영역 구성**

Kubernetes 영역을 사용하면 클라우드 관리자가 vRealize Automation Cloud Assembly 배포에 사용되는 Kubernetes 클러스터 네임스페이스 및 감독자 네임스페이스의 정책 기반 배치를 정의할 수 있습니다. 관리자는 이 페이지를 사용하여 Kubernetes 네임스페이스의 프로비저닝에 사용할 수 있는 클러스터를 지정하고 클러스터에 허용되는 속성을 지정할 수 있습니다.

- **vRealize Automation Cloud Assembly에서 Pacific 감독자 클러스터 및 네임스페이스 사용**

관리자는 기존 Pacific 지원 vSphere 통합의 감독자 네임스페이스를 사용하도록 vRealize Automation Cloud Assembly를 구성하여 사용자가 클라우드 템플릿에 네임스페이스를 배포하고 Service Broker 카탈로그에서 요청할 수 있도록 할 수 있습니다.

- **vRealize Automation Cloud Assembly에서 Kubernetes 클러스터 및 네임스페이스 사용**

클라우드 관리자는 vRealize Automation Cloud Assembly에서 일반 및 Pacific 기반의 배포된 Kubernetes 클러스터 및 네임스페이스의 구성을 추가하고, 보고, 관리할 수 있습니다.

- **vRealize Automation Cloud Assembly에서 클라우드 템플릿에 Kubernetes 구성 요소 추가**

Kubernetes 구성 요소를 vRealize Automation Cloud Assembly 클라우드 템플릿에 추가할 때 클러스터를 추가하거나 사용자가 다양한 구성으로 네임스페이스를 생성하도록 선택할 수 있습니다. 일반적으로 이러한 선택은 액세스 제어 요구 사항, Kubernetes 구성 요소 구성 방법 및 배포 요구 사항에 따라 결정됩니다.

## ■ Kubernetes와 vRealize Automation Cloud Assembly 확장성 사용

vRealize Automation Cloud Assembly는 Kubernetes 클러스터 및 네임스페이스 배포와 관련된 일반적인 작업에 해당하는 이벤트 항목 집합을 제공합니다. 사용자는 이러한 항목을 원하는 대로 구독할 수 있으며 해당 항목은 적절한 시간에 실행됩니다. 구독된 항목과 관련된 이벤트가 발생하면 사용자에게 알림이 전달됩니다. 이벤트 알림을 기반으로 실행되도록 vRO 워크플로를 구성할 수도 있습니다.

## vRealize Automation Cloud Assembly에서 VMware Tanzu Kubernetes Grid Integrated Edition 통합 구성

온-프레미스 및 클라우드에서 VMware TKGI(Tanzu Kubernetes Grid Integrated Edition)(이전 이름: PKS) 리소스 연결을 구성하여 vRealize Automation Cloud Assembly의 Kubernetes 통합 및 관리 기능을 지원할 수 있습니다.

TKGI 통합을 통해 온-프레미스 및 클라우드의 TKGI 인스턴스와 TKGI 및 외부 클러스터에 프로비저닝된 Kubernetes 클러스터를 관리할 수 있습니다. 정책 기반 리소스 배치를 지원하려면 Kubernetes 프로파일을 생성하여 프로젝트와 연결해야 합니다.

### 사전 요구 사항

- UAA 인증을 사용하여 적절히 구성된 TKGI 서버가 있어야 합니다.
- 클라우드 관리자 자격 증명이 있는지 확인합니다. 자세한 내용은 [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.

### 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 VMware Tanzu Kubernetes Grid Integrated Edition을 선택합니다.
- 3 생성하는 TKGI 클라우드 계정에 대한 IP 주소 또는 FQDN과 TKGI 주소를 입력합니다.
  - IP 주소는 TKGI 사용자 인증 서버의 FQDN이나 IP 주소입니다.
  - TKGI 주소는 기본 TKGI 서버의 FQDN이나 IP 주소입니다.
- 4 TKGI 서버가 로컬인지 아니면 공용 클라우드 또는 사설 클라우드에 있는지 선택합니다.
- 5 TKGI 서버 및 기타 관련 정보에 대해 적절한 **사용자 이름** 및 **암호**를 입력합니다.
- 6 태그 지정 전략을 지원하는 태그를 사용하는 경우에는 기능 태그를 입력합니다. 태그를 사용하여 [vRealize Automation Cloud Assembly](#) 리소스 및 배포를 관리하는 방법 및 태그 지정 전략 생성 항목을 참조하십시오.
- 7 **추가**를 클릭합니다.

### 결과

Kubernetes 영역을 생성하여 프로젝트에 할당하거나 외부 Kubernetes 클러스터를 검색하여 프로젝트에 할당할 수 있습니다. 또한 대규모 그룹과 조직 간의 클러스터 관리에 유용한 Kubernetes 네임스페이스를 추가하거나 생성할 수 있습니다.

## 다음에 수행할 작업

적절한 Kubernetes 영역을 생성하거나 선택한 다음 클러스터 또는 네임스페이스를 하나 이상 선택하여 프로젝트에 할당합니다. 그런 다음 클라우드 템플릿을 생성하고 게시하면 Kubernetes를 사용하는 셀프 서비스 배포를 사용자가 생성할 수 있도록 설정됩니다.

## vRealize Automation Cloud Assembly에서 Red Hat OpenShift 통합 구성

온-프레미스 및 클라우드에서 Red Hat OpenShift 리소스 연결을 구성하여 vRealize Automation Cloud Assembly의 엔터프라이즈 수준 Kubernetes 통합 및 관리 기능을 지원할 수 있습니다.

vRealize Automation Cloud Assembly는 OpenShift 버전 3.x와의 통합을 지원합니다.

### 사전 요구 사항

- 적절하게 구성된 Red Hat OpenShift 구현이 있어야 합니다.
- 클라우드 관리자 자격 증명이 있는지 확인합니다. 자세한 내용은 [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- VMware에서는 클라우드 템플릿과 함께 OpenShift 클러스터를 생성하는 데 사용할 수 있는 리소스를 다음 위치에 제공합니다. <https://flings.vmware.com/enterprise-openshift-as-a-service-on-cloud-automation-services>. 이러한 리소스로 생성된 클러스터를 Kubernetes 영역에서 글로벌 클러스터로 사용하여 셀프 서비스 네임스페이스를 생성할 수 있습니다.

### 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 Red Hat OpenShift를 선택합니다.
- 3 OpenShift 서버의 **주소** 및 **위치**를 입력합니다.
- 4 적합한 **자격 증명 유형**을 선택한 후 자격 증명을 입력합니다.  
OpenShift 통합은 OAuth 사용자 이름/암호, 공용 키 또는 전달자 토큰 인증을 지원합니다.
- 5 OpenShift 통합을 위한 적합한 **이름** 및 **설명**을 입력합니다.
- 6 태그 지정 전략을 지원하는 태그를 사용하는 경우에는 적절한 기능 태그를 입력합니다. **태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법** 및 **태그 지정 전략 생성** 항목을 참조하십시오.
- 7 **추가**를 클릭합니다.

### 결과

통합이 생성되면 새로운 Kubernetes 클러스터가 Kubernetes 페이지의 관련 섹션에 나타납니다. Kubernetes 영역을 생성한 후 프로젝트에 할당할 수 있습니다. 또한 대규모 그룹과 조직 간의 클러스터 관리에 유용한 Kubernetes 네임스페이스를 구성할 수 있습니다.

## 다음에 수행할 작업

적절한 Kubernetes 영역을 생성하거나 선택한 다음 클러스터 또는 네임스페이스를 하나 이상 선택하여 프로젝트에 할당합니다. 그런 다음 클라우드 템플릿을 생성하고 게시하면 Kubernetes를 사용하는 셀프 서비스 배포를 사용자가 생성할 수 있도록 설정됩니다.

## vRealize Automation Cloud Assembly에서 Kubernetes 영역 구성

Kubernetes 영역을 사용하면 클라우드 관리자가 vRealize Automation Cloud Assembly 배포에 사용되는 Kubernetes 클러스터 네임스페이스 및 감독자 네임스페이스의 정책 기반 배치를 정의할 수 있습니다. 관리자는 이 페이지를 사용하여 Kubernetes 네임스페이스의 프로비저닝에 사용할 수 있는 클러스터를 지정하고 클러스터에 허용되는 속성을 지정할 수 있습니다.

클라우드 관리자는 Kubernetes 영역을 Cloud Assembly에 대해 구성된 TKGI 클라우드 계정 또는 프로젝트와 연결되지 않은 외부 Kubernetes 클러스터와 연결할 수 있습니다.

Kubernetes 영역을 생성할 때 제공자별 리소스 여러 개를 영역에 할당할 수 있으며, 이러한 리소스는 작업자, 마스터, 사용 가능한 CPU, 메모리의 수와 기타 구성 설정 측면에서 새로 프로비저닝된 클러스터에 설정할 수 있는 속성을 나타냅니다. TKGI 제공자의 경우 TKGI 계획에 해당합니다. 관리자는 여러 클러스터를 Kubernetes 영역에 할당하여 새로 프로비저닝된 Kubernetes 네임스페이스를 배치하는 데 사용할 수도 있습니다. 관리자는 등록되지 않았거나 CMX에 의해 관리되지 않고 미리 선택된 클러스터 제공자를 통해 프로비저닝된 클러스터만 할당할 수 있습니다. 관리자는 여러 Kubernetes 영역을 단일 프로젝트에 할당하여 해당 프로젝트 내에서 발생하는 배치 작업에 모두 사용할 수 있습니다.

클라우드 관리자는 여러 수준에 우선 순위를 지정할 수 있습니다.

- 프로젝트 내 Kubernetes 영역 우선 순위.
- Kubernetes 영역 내 리소스 우선 순위.
- Kubernetes 영역 내 클러스터 우선 순위.

클라우드 관리자는 여러 수준에 태그를 할당할 수도 있습니다.

- Kubernetes 영역별 기능 태그.
- 리소스 할당별 태그.
- 클러스터 할당별 태그.

일반 Kubernetes 네임스페이스로 작업하는 것과 동일한 방식으로 vSphere에서 감독자 네임스페이스를 사용하여 Kubernetes 영역을 생성할 수 있습니다. Kubernetes 영역에 감독자 네임스페이스를 추가하려면 해당 영역을 원하는 Pacific 네임스페이스 리소스가 포함된 vSphere 7 끝점과 연결해야 합니다.

Service Broker에는 Service Broker 관리자가 카탈로그에서 프로비저닝된 Kubernetes 네임스페이스 및 클러스터에 대한 배치 정책을 생성할 수 있도록 기존 Kubernetes 영역에 액세스할 수 있게 하는 Kubernetes 영역 페이지 버전이 포함되어 있습니다.

## 사전 요구 사항

적절한 VMware TKGI(Tanzu Kubernetes Grid Integrated Edition) 배포와의 통합을 구성합니다.

vRealize Automation Cloud Assembly에서 [VMware Tanzu Kubernetes Grid Integrated Edition 통합 구성 항목](#)을 참조하십시오.

## 절차

1 **인프라 > 구성 > Kubernetes 영역**을 선택한 다음, **새 Kubernetes 영역**을 클릭합니다.

2 이 영역을 적용할 TKGI 통합 **계정**의 이름을 입력합니다.

이렇게 하면 영역과 연결된 클라우드 계정 또는 끝점이 정의됩니다. 각 영역에는 끝점을 하나만 할당할 수 있습니다. vSphere에서 감독자 네임스페이스를 사용하는 경우에는 감독자 네임스페이스를 포함하는 vSphere 인스턴스만 선택할 수 있습니다.

3 Kubernetes 영역의 **이름**과 **설명**을 추가합니다.

4 필요한 경우 기능 태그를 추가합니다. 자세한 내용은 [vRealize Automation Cloud Assembly에서 기능 태그 사용](#)의 내용을 참조하십시오.

5 **저장**을 클릭합니다.

6 [주문형] 탭을 클릭하고 클러스터 프로비저닝에 사용할 영역에 대해 TKGI 계획을 적절하게 추가합니다.

하나 이상의 계획을 선택하여 우선 순위를 할당할 수 있습니다. 숫자가 낮을수록 우선 순위가 높습니다. 우선 순위 할당은 태그 기반 선택에 부수적입니다.

7 [클러스터] 탭을 클릭한 다음, **계산 추가** 버튼을 클릭하여 Kubernetes 또는 감독자 클러스터를 영역에 추가합니다. 외부 클러스터로 작업하는 경우 선택하면 vRealize Automation Cloud Assembly에 자동으로 등록됩니다.

vRealize Automation Cloud Assembly의 [Kubernetes 클러스터] 페이지에서 클러스터에 Kubernetes 네임스페이스를 추가할 수 있습니다.

## 결과

Kubernetes 영역은 vRealize Automation Cloud Assembly 배포에서 사용하도록 구성됩니다.

## 다음에 수행할 작업

프로젝트에 Kubernetes 영역을 할당합니다.

1 **인프라 > 관리 > 프로젝트**를 선택한 다음, Kubernetes 영역과 연결할 프로젝트를 선택합니다.

2 [프로젝트] 페이지에서 [Kubernetes 프로비저닝] 탭을 클릭합니다.

3 **Kubernetes 영역 추가**를 클릭하고 방금 생성한 영역을 추가합니다. 해당되는 경우 여러 영역을 사용할 수 있으며 영역의 우선 순위도 설정할 수 있습니다.

4 **저장**을 클릭합니다.

영역을 프로젝트에 할당한 후에는 [설계] 탭의 [클라우드 템플릿] 페이지를 사용하여 Kubernetes 영역 및 프로젝트 구성을 기반으로 배포를 프로비저닝할 수 있습니다. 이 [클라우드 템플릿] 페이지에는 K8S Cluster, K8S 네임스페이스 및 감독자 네임스페이스를 추가하는 옵션이 포함되어 있습니다. 작업 중인 Kubernetes 리소스에 적절한 옵션을 선택합니다.

## vRealize Automation Cloud Assembly에서 Pacific 감독자 클러스터 및 네임스페이스 사용

관리자는 기존 Pacific 지원 vSphere 통합의 감독자 네임스페이스를 사용하도록 vRealize Automation Cloud Assembly를 구성하여 사용자가 클라우드 템플릿에 네임스페이스를 배포하고 Service Broker 카탈로그에서 요청할 수 있도록 할 수 있습니다.

이 작업에서는 vRealize Automation Cloud Assembly를 사용하여 배포에 사용할 감독자 클러스터를 추가하는 방법과 특정 Kubernetes 리소스에 액세스할 수 있는 vRealize Automation Cloud Assembly 프로젝트 및 사용자를 정의하는 네임스페이스를 생성하거나 추가하는 방법을 설명합니다. 이 기능은 VMware TKGI(Tanzu Kubernetes Grid Integrated Edition) 또는 Openshift와 같은 통합이 아닌 적절한 vSphere 클라우드 계정에 의존합니다. 감독자 클러스터는 vSphere와 연결되어 있는 사용자 지정된 Kubernetes 클러스터입니다. 이는 Kubernetes API를 최종 사용자에게 노출하고, 작업자 노드에 대한 플랫폼으로 Linux가 아닌 ESXi를 사용합니다. 일반적으로 개별 가상 시스템보다 네임스페이스에 정책을 적용하는 것이 더 쉽기 때문에 감독자 네임스페이스는 Kubernetes 리소스에 대한 액세스 제어를 용이하게 합니다. 각 감독자 클러스터에 대해 네임스페이스를 여러 개 생성할 수 있습니다.

Pacific 지원 vSphere 인스턴스와 함께 사용하는 경우 Kubernetes 영역은 감독자 네임스페이스를 사용하여 프로비저닝할 수 있는 감독자 클러스터를 정의합니다. 감독자 네임스페이스는 Pacific 지원 vSphere 인스턴스에만 해당됩니다. 일반 Kubernetes 리소스를 Pacific 지원 vSphere 인스턴스에 프로비저닝할 수 없습니다.

프로젝트 뷰어로 지정된 vRealize Automation Cloud Assembly 사용자는 네임스페이스에 대한 보기 전용 액세스 권한이 있는 반면, 프로젝트 멤버는 이를 편집할 수 있습니다.

필요한 경우 네임스페이스에 연결된 감독자 클러스터를 구성할 수 있습니다.

### 사전 요구 사항

- vRealize Automation Cloud Assembly에서 Pacific 네임스페이스를 사용하려면 vSphere 7.x 끝점이 구성되어 있어야 합니다. vSphere는 vCenter 클라우드 계정의 일부로 설치됩니다. [vRealize Automation](#)에서 [vCenter 클라우드 계정 생성](#)의 내용을 참조하십시오.
- Project Pacific은 vSphere 클라우드 계정에서 사용되도록 설정해야 하며 적절한 감독자 네임스페이스를 포함해야 합니다.
- vCenter와 vRealize Automation 배포 모두 동일한 Active Directory를 사용해야 사용자를 동기화할 수 있습니다. 그렇지 않은 경우 프로비저닝은 계속 작동하지만 vRealize Automation 사용자가 네임스페이스에 대한 액세스 권한을 자동으로 얻을 수 없습니다.

## 절차

- 1 vRealize Automation Cloud Assembly에서 **인프라 > 구성 > Kubernetes 영역**을 선택합니다.  
이 페이지에서는 사용이 가능한 관리 클러스터가 표시되며 이 페이지를 사용하여 다른 클러스터를 추가할 수 있습니다. 클러스터를 클릭하여 세부 정보를 볼 수 있습니다.
- 2 새 **Kubernetes 영역**을 선택합니다.
- 3 대상 vSphere 클라우드 계정에 대한 **계정** 세부 정보를 지정합니다.
- 4 텍스트 상자의 검색 아이콘을 클릭하여 모든 vSphere 계정을 보거나 이름으로 계정을 검색합니다.
- 5 새 영역의 **이름**과 **설명**을 입력합니다.
- 6 필요한 경우 기능 태그를 추가합니다. 자세한 내용은 [vRealize Automation Cloud Assembly에서 기능 태그 사용](#)의 내용을 참조하십시오.
- 7 [프로비저닝] 탭을 클릭하여 네임스페이스와 연결될 감독자 클러스터를 선택합니다.
- 8 **계산 추가**를 클릭하여 사용 가능한 감독자 클러스터를 살펴보고 선택합니다.
- 9 **추가**를 클릭합니다.
- 10 **인프라 > 관리 > 프로젝트**를 선택한 다음, Kubernetes 영역과 연결할 프로젝트를 선택합니다.
- 11 [프로젝트] 페이지에서 [Kubernetes 프로비저닝] 탭을 클릭합니다.
- 12 **Kubernetes 영역 추가**를 클릭하고 방금 생성한 영역을 추가합니다. 해당되는 경우 여러 영역을 사용할 수 있으며 영역의 우선 순위도 설정할 수 있습니다.
- 13 **저장**을 클릭합니다.

## 다음에 수행할 작업

네임스페이스가 구성된 후에는 해당 사용자에게 대한 vRealize Automation Cloud Assembly의 **인프라 > 리소스 > Kubernetes** 페이지에 네임스페이스가 표시됩니다. 사용자는 [요약] 탭의 [주소] 링크를 클릭하여 vSphere Kubernetes CLI 도구를 열고 네임스페이스를 관리할 수 있습니다. 감독자 네임스페이스 세부 정보 링크에 액세스하려면 사용자가 클라우드 관리자이거나 지정된 프로젝트 네임스페이스의 멤버여야 합니다. 사용자는 사용자 지정된 Kubectl을 다운로드하여 감독자 네임스페이스를 사용할 수도 있습니다. 사용자는 감독자 네임스페이스에 로그인하여 다른 네임스페이스와 마찬가지로 사용한 다음 클라우드 템플릿을 생성하고 애플리케이션을 배포할 수 있습니다.

클라우드 템플릿에 네임스페이스를 추가하려면 **설계 > 클라우드 템플릿**을 선택하고 기존 클라우드 템플릿을 선택하거나 새로 생성합니다. 그런 다음, 왼쪽 메뉴에서 감독자 네임스페이스 항목을 선택하여 캔버스에 끌어서 놓으면 됩니다.

감독자 네임스페이스가 포함된 클라우드 템플릿을 배포한 후에는 사용자가 **Service Broker** 카탈로그에서 감독자 네임스페이스를 요청할 수도 있습니다. 또한 Cloud Assembly에서 [배포] 페이지를 클릭하여 배포에 대한 정보를 살펴보고, vSphere에서 네임스페이스에 대한 kubectl을 실행하는 명령이 포함된 링크에 액세스할 수 있습니다.

## vRealize Automation Cloud Assembly에서 Kubernetes 클러스터 및 네임스페이스 사용

클라우드 관리자는 vRealize Automation Cloud Assembly에서 일반 및 Pacific 기반의 배포된 Kubernetes 클러스터 및 네임스페이스의 구성을 추가하고, 보고, 관리할 수 있습니다.

클라우드 관리자 권한이 있는 사용자는 **인프라 > 리소스 > Kubernetes** 페이지에서 액세스 권한이 있는 Kubernetes 클러스터 및 네임스페이스를 보고, 추가하고, 관리할 수 있습니다. 이 페이지에는 클러스터, 네임스페이스, 감독자 클러스터 및 감독자 네임스페이스에 대한 탭이 포함되어 있습니다. 이러한 탭 중 하나를 선택하여 유사한 리소스를 보고 관리할 수 있습니다. 일반적으로 이 페이지에서는 배포된 클러스터 및 네임스페이스를 쉽게 관리할 수 있습니다.

- **클러스터:** 클러스터는 하나 이상의 물리적 시스템에서 분산된 Kubernetes 노드 그룹입니다. 이 페이지에는 vRealize Automation Cloud Assembly 인스턴스에서 사용하도록 구성된 프로비저닝된 클러스터와 배포 해제된 클러스터가 표시됩니다. 클러스터를 클릭하면 현재 상태에 대한 정보를 볼 수 있습니다. 클러스터를 배포하는 경우 클라우드 관리자만 액세스할 수 있는 Kubconfig 파일에 대한 링크가 포함됩니다. 이 파일은 네임스페이스 목록을 포함하여 클러스터에 대한 전체 관리자 권한을 부여합니다.

감독자 클러스터는 vSphere 인스턴스에 고유하며 ESXi를 Linux 대신 작업자 노드로 사용합니다.

- **네임스페이스:** 네임스페이스는 관리자가 클러스터 리소스를 분리할 수 있는 방법을 제공하는 가상 클러스터입니다. 대규모 사용자 및 조직 그룹 간의 리소스 관리를 용이하게 합니다. 역할 기반 액세스 제어의 한 가지 형태로, 클라우드 관리자는 사용자가 배포를 요청하면 프로젝트에 네임스페이스를 추가할 수 있게 한 다음, 해당 네임스페이스를 나중에 Kubernetes 클러스터 페이지에서 관리할 수 있습니다. 네임스페이스를 배포할 때 kubeconfig 파일에 대한 링크가 포함됩니다. 이 링크를 통해 유효한 사용자(예: 개발자)가 해당 네임스페이스의 일부 측면을 보고 관리할 수 있습니다.

감독자 클러스터 및 감독자 네임스페이스는 vSphere 인스턴스에만 존재하며 vSphere 개체에 대해 Kubernetes와 유사한 액세스를 제공합니다.

클라우드 관리자는 관리자가 클라우드 템플릿 및 Service Broker에서 Kubernetes 리소스를 프로비저닝한 다음 사용을 위해 특정 프로젝트에 할당할 수 있도록 이 페이지에서 Kubernetes 네임스페이스 또는 클러스터와 연결된 프로젝트를 변경할 수 있습니다. 관리자는 클러스터의 범위를 변경하여 글로벌 클러스터 또는 프로젝트 특정 클러스터로 만들 수 있습니다. 글로벌 클러스터는 모든 Kubernetes 영역에 대해 클러스터 탭으로 표시되며 선택 및 프로비저닝할 수 있습니다. 글로벌 클러스터는 Kubernetes 영역에 추가된 후 카탈로그에서 네임스페이스를 프로비저닝하는 데 사용할 수 있습니다.

새 클러스터 또는 기존 클러스터를 구성하는 경우 마스터 IP 주소와 마스터 호스트 이름 중 어디에 연결할지를 선택해야 합니다.

### vRealize Automation Cloud Assembly에서 일반적인 Kubernetes 클러스터 사용

이 페이지의 옵션을 사용하여 새 클러스터, 기존 클러스터 또는 외부 클러스터를 vRealize Automation Cloud Assembly에 추가할 수 있습니다.

- 1 **인프라 > 리소스 > Kubernetes**를 선택하고 [클러스터] 탭이 활성 상태인지 확인합니다.

vRealize Automation Cloud Assembly 인스턴스에 대해 현재 구성된 클러스터가 있으면 이 페이지에 표시됩니다.

- 2 신규 또는 기존 클러스터를 추가하거나 클러스터를 배포하는 경우 다음 표에 따라 적절한 옵션을 선택합니다.

옵션	설명	세부 정보
배포	vRealize Automation Cloud Assembly에 새 클러스터 추가	이 클러스터가 배포될 TKGI 클라우드 계정과 원하는 계획 및 노드 수를 지정해야 합니다.
기존 추가	프로젝트에서 작동하도록 기존 클러스터를 구성합니다.	TKGI 클라우드 계정, 사용할 클러스터 및 대상 개발자를 위한 적절한 프로젝트를 지정해야 합니다. 공유 범위도 지정해야 합니다. 전역적으로 공유하려면 Kubernetes 영역 및 네임스페이스를 적절하게 구성해야 합니다.
외부 추가	TKGI와 연결되지 않을 수 있는 vanilla Kubernetes 클러스터를 vRealize Automation Cloud Assembly에 추가합니다.	클러스터가 연결된 프로젝트를 지정하고 원하는 클러스터의 IP 주소를 입력한 후 이 클러스터에 연결하는 데 필요한 클라우드 프록시 및 인증서 정보를 선택해야 합니다.

- 3 **추가**를 클릭하여 vRealize Automation Cloud Assembly 내에서 클러스터를 사용할 수 있도록 만듭니다.

### vRealize Automation Cloud Assembly에서 Kubernetes 네임스페이스 사용

클라우드 관리자가 네임스페이스를 사용하면 Kubernetes 클러스터 리소스를 그룹화하고 관리하는 데 유용합니다. 사용자에게 네임스페이스는 배포를 위한 Kubernetes 클러스터의 영역입니다. 관리자와 사용자는 **인프라 > 리소스 > Kubernetes** 페이지에 있는 [네임스페이스] 탭을 사용하여 네임스페이스에 액세스할 수 있습니다.

vRealize Automation Cloud Assembly의 리소스에 Kubernetes 네임스페이스를 추가하는 방법이 여러 가지 있습니다. 다음 절차는 일반적인 방법 중 하나를 요약한 것입니다.

- 1 **인프라 > 리소스 > Kubernetes**를 선택하고 [네임스페이스] 탭을 클릭합니다.
- 2 새 네임스페이스를 추가하려면 **새 네임스페이스**를 클릭합니다. 기존 네임스페이스를 추가하려면 **네임스페이스 추가**를 클릭합니다.
- 3 네임스페이스에 대한 **이름**과 **설명**을 입력합니다.  
이 시점에서 Kubernetes 리소스에 사용할 네임스페이스가 추가되었지만 특정 항목에 연결되어 있지는 않습니다.
- 4 이 네임스페이스에 연결할 **클러스터**를 지정합니다.
- 5 **생성**을 클릭하여 vRealize Automation Cloud Assembly에 네임스페이스를 추가합니다.

### 감독자 클러스터 및 감독자 네임스페이스 사용

vRealize Automation Cloud Assembly의 Kubernetes 페이지에서 감독자 클러스터 및 네임스페이스에 대한 구성을 보고 변경할 수 있습니다.

- 1 vRealize Automation Cloud Assembly에서 **인프라 > 리소스 > Kubernetes**를 선택합니다.

- 2 **감독자 클러스터 추가**를 선택합니다.
- 3 대상 vSphere 클라우드 계정에 대한 계정 세부 정보를 지정합니다.
- 4 감독자 클러스터 텍스트 상자의 검색 아이콘을 클릭하여 모든 감독자 클러스터를 보거나 이름별로 클러스터를 검색합니다.
- 5 원하는 클러스터를 선택하고 **추가**를 클릭합니다.
- 6 감독자 네임스페이스 탭을 선택하고 **새 감독자 네임스페이스** 버튼을 클릭하여 새 네임스페이스를 추가합니다.
- 7 감독자 네임스페이스 탭을 선택하고 **새 감독자 네임스페이스** 버튼을 클릭하여 새 네임스페이스를 추가합니다.
  - a 새 네임스페이스를 생성하는 경우 **이름** 및 **설명**을 추가합니다.
  - b 이 네임스페이스에 연결할 적절한 클라우드 **계정**을 선택합니다.
  - c 이 네임스페이스에 연결할 **감독자 클러스터**를 선택합니다.
  - d 네임스페이스에 연결할 **프로젝트**를 선택합니다.
  - e **생성**을 클릭합니다.
- 8 새 네임스페이스에 대한 관련 세부 정보를 검토합니다.

현재 vSphere의 네임스페이스에 액세스할 수 있는 사용자 및 그룹이 [사용자] 탭에 나열됩니다. 새 사용자 또는 그룹이 프로젝트에 추가된 경우 이 탭의 **사용자 업데이트** 버튼을 클릭하여 목록을 업데이트합니다. 목록은 자동으로 업데이트되지 않으므로 버튼을 사용하여 업데이트해야 합니다.

---

**참고** 사용자 동기화는 vRealize Automation Cloud Assembly 및 vCenter가 공통 Active Directory/LDAP 서비스로 구성된 경우에만 의미가 있습니다.

---

클러스터 또는 네임스페이스가 구성되면 vRealize Automation Cloud Assembly의 **인프라 > 리소스 > Kubernetes** 페이지에는 사용자가 사용할 수 있는 클러스터 및 네임스페이스가 표시됩니다. 개별 네임스페이스 또는 클러스터를 클릭하여 리소스에 대한 통계 및 기타 정보를 보여 주는 여러 개의 탭이 포함된 페이지를 열 수 있으며 다양한 옵션을 구성할 수 있습니다.

관리자는 **Kubernetes** 페이지의 클러스터에 대한 [요약] 탭을 통해 클러스터의 구성을 보거나 경우에 따라 범위 변경을 비롯한 클러스터 구성을 업데이트할 수 있습니다. [공유] 라디오 버튼을 사용하여 [글로벌] (Kubernetes 영역을 통해 공유 가능) 또는 [프로젝트] (단일 프로젝트로 액세스가 제한됨)를 선택할 수 있습니다. [프로젝트]를 선택하는 경우 아래 프로젝트 선택에서 해당하는 프로젝트도 지정해야 합니다.

---

**참고** 공유 구성을 변경하면 클러스터에서 사용할 수 있는 네임스페이스에 영향을 줄 수 있습니다.

---

사용자는 [요약] 탭의 [주소] 링크를 클릭하여 vSphere Kubernetes CLI 도구를 열고 네임스페이스를 관리할 수 있습니다. 감독자 네임스페이스 세부 정보 링크에 액세스하려면 사용자가 클라우드 관리자이거나 지정된 프로젝트 네임스페이스의 멤버여야 합니다. 사용자는 사용자 지정된 **KubectI**를 다운로드하여 감독자 네임스페이스를 사용할 수도 있습니다. 사용자는 감독자 네임스페이스에 로그인하여 다른 네임스페이스와 마찬가지로 사용한 다음 클라우드 템플릿을 생성하고 애플리케이션을 배포할 수 있습니다.

## vRealize Automation Cloud Assembly에서 클라우드 템플릿에 Kubernetes 구성 요소 추가

Kubernetes 구성 요소를 vRealize Automation Cloud Assembly 클라우드 템플릿에 추가할 때 클러스터를 추가하거나 사용자가 다양한 구성으로 네임스페이스를 생성하도록 선택할 수 있습니다. 일반적으로 이러한 선택은 액세스 제어 요구 사항, Kubernetes 구성 요소 구성 방법 및 배포 요구 사항에 따라 결정됩니다.

Kubernetes 구성 요소를 vRealize Automation Cloud Assembly의 클라우드 템플릿에 추가하려면 **설계 > 클라우드 템플릿**을 선택하고 **새로 만들기**를 클릭한 다음 왼쪽 메뉴에서 **Kubernetes** 옵션을 찾아서 확장합니다. 그런 다음 원하는 항목(클러스터 또는 **KBS 네임스페이스 중 하나**)을 캔버스로 끌어서 선택합니다.

프로젝트와 연결된 **Kubernetes** 클러스터를 클라우드 템플릿에 추가하는 것은 **Kubernetes** 리소스를 유효한 사용자가 사용할 수 있게 만드는 가장 간단한 방법입니다. 다른 **Cloud Assembly** 리소스와 마찬가지로 클러스터에서 태그를 사용하여 배포 위치를 제어할 수 있습니다. 클러스터 배포의 할당 단계에서 태그를 사용하여 영역 및 **VMware TKGI(Tanzu Kubernetes Grid Integrated Edition)** 계획을 선택할 수 있습니다.

이러한 방식으로 클러스터를 추가하면 유효한 모든 사용자가 자동으로 사용할 수 있게 됩니다.

### 클라우드 템플릿 예

첫 번째 클라우드 템플릿 예는 태그 지정으로 제어되는 간단한 **Kubernetes** 배포에 대한 템플릿을 보여줍니다. 새 **Kubernetes** 영역 페이지에 구성된 두 개의 배포 계획으로 **Kubernetes** 영역이 생성되었습니다. 이 경우 `placement:tag`라는 태그가 영역에 기능으로 추가되었으며, 클라우드 템플릿의 유사 제약 조건을 일치시키는 데 사용되었습니다. 태그로 구성된 영역이 둘 이상인 경우, 우선 순위 번호가 가장 낮은 영역이 선택됩니다.

```
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
```

두 번째 클라우드 템플릿 예는 배포를 요청할 때 사용자가 원하는 클러스터 호스트 이름을 입력할 수 있도록 `$(input.hostname)`이라는 변수를 사용하여 템플릿을 설정하는 방법을 보여줍니다. 태그는 클러스터 배포의 리소스 할당 단계에서 영역과 **TKGI** 계획을 선택하는 데 사용할 수도 있습니다.

```
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
```

네임스페이스를 사용하여 클러스터 사용량을 관리하려는 경우, 배포를 요청할 때 사용자가 입력하는 네임스페이스 이름을 대체하도록 클라우드 템플릿에 *name: \${input.name}*이라는 변수를 설정할 수 있습니다. 이러한 종류의 배포에는 다음 예와 같은 템플릿을 생성합니다.

```
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}
```

사용자는 **인프라 > 리소스 > Kubernetes 클러스터** 페이지에서 액세스할 수 있는 kubeconfig 파일을 통해 배포된 클러스터를 관리할 수 있습니다. 원하는 클러스터의 페이지에서 카드를 찾은 후 **Kubeconfig**를 클릭합니다.

## VMware Cloud Templates의 감독자 네임스페이스

다음은 vRealize Automation Cloud Assembly 클라우드 템플릿의 기본 감독자 네임스페이스에 대한 스키마입니다.

```
{
  "title": "Supervisor namespace schema",
  "description": "Request schema for provisioning of Supervisor namespace resource",
  "type": "object",
  "properties": {
    "name": {
      "title": "Name",
      "description": "Alphabetic (a-z and 0-9) string with maximum length of 63 characters. The character '-' is allowed anywhere except the first or last position of the identifier.",
      "type": "string",
      "pattern": "^[a-z0-9]([a-z0-9-]{1,61}[a-z0-9])?$",
      "ignoreOnUpdate": true
    },
    "description": {
      "title": "Description",
      "description": "An optional description of this Supervisor namespace.",
      "type": "string",
      "ignoreOnUpdate": true
    },
    "constraints": {
```

```

    "title": "Constraints",
    "description": "To target the correct resources, blueprint constraints are matched
against infrastructure capability tags. Constraints must include the key name. Options
include value, negative [!], and hard or soft requirement.",
    "type": "array",
    "recreateOnUpdate": true,
    "items": {
      "type": "object",
      "properties": {
        "tag": {
          "title": "Tag",
          "description": "Constraint definition in syntax `[!]tag_key[:tag_value]
[:hard|:soft]` \nExamples:\n```\n!location:eu:hard\n location:us:soft\n!pci\n```,
          "type": "string",
          "recreateOnUpdate": true
        }
      }
    }
  },
  "limits": {
    "title": "Limits",
    "description": "Defines namespace resource limits such as pods, services, etc.",
    "type": "array",
    "recreateOnUpdate": false,
    "items": {
      "type": "object",
      "properties": {
        "stateful_set_count": {
          "title": "stateful_set_count",
          "description": "This represents the new value for 'statefulSetCount' option which
is the maximum number of StatefulSets in the namespace.",
          "type": "integer",
          "recreateOnUpdate": false
        },
        "deployment_count": {
          "title": "deployment_count",
          "description": "This represents the new value for 'deploymentCount' option which
is the maximum number of deployments in the namespace.",
          "type": "integer",
          "recreateOnUpdate": false
        },
        "cpu_limit_default": {
          "title": "cpu_limit_default",
          "description": "This represents the new value for the default CPU limit (in Mhz)
for containers in the pod. If specified, this limit should be at least 10 Mhz.",
          "type": "integer",
          "recreateOnUpdate": false
        },
        "config_map_count": {
          "title": "config_map_count",
          "description": "This represents the new value for 'configMapCount' option which
is the maximum number of ConfigMaps in the namespace.",
          "type": "integer",
          "recreateOnUpdate": false
        }
      }
    }
  },

```

```

    "pod_count": {
      "title": "pod_count",
      "description": "This represents the new value for 'podCount' option which is the
maximum number of pods in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "job_count": {
      "title": "job_count",
      "description": "This represents the new value for 'jobCount' option which is the
maximum number of jobs in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "secret_count": {
      "title": "secret_count",
      "description": "This represents the new value for 'secretCount' option which is
the maximum number of secrets in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "cpu_limit": {
      "title": "cpu_limit",
      "description": "This represents the new value for 'limits.cpu' option which is
equivalent to the maximum CPU limit (in MHz) across all pods in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "cpu_request_default": {
      "title": "cpu_request_default",
      "description": "This represents the new value for the default CPU request (in
Mhz) for containers in the pod. If specified, this field should be at least 10 MHz.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "memory_limit_default": {
      "title": "memory_limit_default",
      "description": "This represents the new value for the default memory limit (in
mebibytes) for containers in the pod.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "memory_limit": {
      "title": "memory_limit",
      "description": "This represents the new value for 'limits.memory' option which is
equivalent to the maximum memory limit (in mebibytes) across all pods in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "memory_request_default": {
      "title": "memory_request_default",
      "description": "This represents the new value for the default memory request (in
mebibytes) for containers in the pod.",
      "type": "integer",
      "recreateOnUpdate": false
    }
  }

```

```

    },
    "service_count": {
      "title": "service_count",
      "description": "This represents the new value for 'serviceCount' option which is
the maximum number of services in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "replica_set_count": {
      "title": "replica_set_count",
      "description": "This represents the new value for 'replicaSetCount' option which
is the maximum number of ReplicaSets in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "replication_controller_count": {
      "title": "replication_controller_count",
      "description": "This represents the new value for 'replicationControllerCount'
option which is the maximum number of ReplicationControllers in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "storage_request_limit": {
      "title": "storage_request_limit",
      "description": "This represents the new value for 'requests.storage' which is the
limit on storage requests (in mebibytes) across all persistent volume claims from pods in the
namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "persistent_volume_claim_count": {
      "title": "persistent_volume_claim_count",
      "description": "This represents the new value for 'persistentVolumeClaimCount'
option which is the maximum number of PersistentVolumeClaims in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    },
    "daemon_set_count": {
      "title": "daemon_set_count",
      "description": "This represents the new value for 'daemonSetCount' option which
is the maximum number of DaemonSets in the namespace.",
      "type": "integer",
      "recreateOnUpdate": false
    }
  },
  "additionalProperties": false
}
}
},
"required": [
  "name"
]
}

```

VMware Cloud Templates는 감독자 네임스페이스에 제한을 사용하도록 지원합니다. 제한을 사용하면 배포된 시스템에서 네임스페이스에 허용되는 최대 포트 수뿐만 아니라 CPU 및 메모리의 리소스 사용량을 제어할 수 있습니다.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_SV_Namespace_1:
    type: Cloud.SV.Namespace
    properties:
      name: '${env.deploymentName}'
      limits:
        - cpu_limit: 1000
          cpu_request_default: 800
          memory_limit: 2000
          memory_limit_default: 1500
          pod_count: 200
```

## Kubernetes와 vRealize Automation Cloud Assembly 확장성 사용

vRealize Automation Cloud Assembly는 Kubernetes 클러스터 및 네임스페이스 배포와 관련된 일반적인 작업에 해당하는 이벤트 항목 집합을 제공합니다. 사용자는 이러한 항목을 원하는 대로 구독할 수 있으며 해당 항목은 적절한 시간에 실행됩니다. 구독된 항목과 관련된 이벤트가 발생하면 사용자에게 알림이 전달됩니다. 이벤트 알림을 기반으로 실행되도록 vRO 워크플로를 구성할 수도 있습니다.

다음 항목은 vRealize Automation Cloud Assembly의 **확장성 > 라이브러리 > 이벤트 항목** 페이지에서 구독할 수 있습니다. 이러한 항목을 보려면 이벤트 항목 검색 텍스트 상자에서 Kubernetes를 검색합니다.

- Kubernetes 클러스터 할당
- Kubernetes 클러스터 사후 프로비저닝
- Kubernetes 클러스터 사후 제거
- Kubernetes 클러스터 프로비저닝
- Kubernetes 클러스터 제거
- Kubernetes 네임스페이스 할당
- Kubernetes 네임스페이스 사후 프로비저닝
- Kubernetes 네임스페이스 사후 제거
- Kubernetes 네임스페이스 제거
- Kubernetes 네임스페이스 할당
- Kubernetes 감독자 네임스페이스 할당
- Kubernetes 감독자 네임스페이스 사후 프로비저닝
- Kubernetes 감독자 네임스페이스 사후 제거

- Kubernetes 감독자 네임스페이스 제거
- Kubernetes 감독자 네임스페이스 할당

항목 중 하나를 클릭하면 수집 및 전송되는 모든 정보를 표시하는 해당 항목에 대한 스키마를 볼 수 있습니다. Kubernetes 네임스페이스와 감독자 네임스페이스 모두에 대한 네임스페이스 항목이 있습니다. 이 스키마 정보를 사용하여 다양한 알림 및 관리 및 보고 작업을 설정할 수 있습니다.

**확장성 > 라이브러리 > 작업** 페이지에서 CMX 관련 작업에 대한 작업 스크립트를 설정할 수 있습니다. 작업 스크립트는 다양한 용도로 사용할 수 있습니다(예: Kubernetes 클러스터 프로비저닝의 DNS 레코드 생성). DNS 레코드를 생성하는 경우에는 Kubernetes 클러스터 사후 프로비저닝 항목의 `masternodeips` 필드를 작업 스크립트에서 REST 명령과 함께 사용하여 DNS 레코드를 생성할 수 있습니다.

[구독] 페이지에서는 이벤트 항목과 작업 스크립트 간의 관계를 정의합니다. vRealize Automation Cloud Assembly의 [구독] 페이지에서 이러한 구성 요소를 살펴보고 관리할 수 있습니다.

자세한 내용은 [확장성을 사용하여 애플리케이션 수명 주기를 연장 및 자동화하는 방법](#)에서 vRealize Automation Cloud Assembly 확장성 설명서를 참조하십시오.

## vRealize Automation Cloud Assembly의 구성 관리

vRealize Automation Cloud Assembly는 배포의 구성 및 편차를 관리할 수 있도록 Puppet Enterprise, Ansible 오픈 소스 및 Ansible Tower와의 통합을 지원합니다.

### Puppet 통합

Puppet 기반 구성 관리를 통합하려면 vSphere 워크로드가 있는 공용 또는 사설 클라우드에 유효한 Puppet Enterprise 인스턴스가 설치되어 있어야 합니다. 이 외부 시스템과 vRealize Automation Cloud Assembly 인스턴스 간에 연결을 설정해야 합니다. 그런 다음 Puppet 구성 관리를 해당 Blueprint에 추가하여 vRealize Automation Cloud Assembly에서 사용 가능하게 만들 수 있습니다.

vRealize Automation Cloud Assembly Blueprint 서비스 Puppet 제공자는 배포된 계산 리소스에 Puppet 에이전트를 설치, 구성 및 실행합니다. Puppet 제공자는 다음 사전 요구 사항을 충족하는 SSH 및 WinRM 연결을 지원합니다.

- SSH 연결:
  - 사용자 이름은 슈퍼 사용자 또는 NOPASSWD로 명령을 실행할 수 있는 `sudo` 권한을 가진 사용자여야 합니다.
  - 해당 사용자에게 대해 `requiretty`를 사용하지 않도록 설정해야 합니다.
  - 배포 계산 리소스에서 `cURL`을 사용할 수 있어야 합니다.
- WinRM 연결:
  - 배포 계산 리소스에서 PowerShell 2.0을 사용할 수 있어야 합니다.
  - vRealize Orchestrator 설명서에 명시된 대로 Windows 템플릿을 구성해야 합니다.

DevOps 관리자는 Puppet Master에 대한 연결을 관리하고 특정 배포에 Puppet 역할 또는 구성 규칙을 적용하는 작업을 수행합니다. 배포 후에는 구성 관리를 지원하도록 구성된 가상 시스템이 지정된 Puppet Master에 등록됩니다.

가상 시스템이 배포되면 사용자가 외부 시스템으로 Puppet Master를 추가 또는 삭제하거나 Puppet Master에 할당된 프로젝트를 업데이트할 수 있습니다. 아울러, 해당 사용자는 시스템이 더 이상 사용되지 않는 경우 Puppet Master에서 배포된 가상 시스템의 등록을 해제할 수 있습니다.

## Ansible 오픈 소스 통합

Ansible 통합을 설정할 때 Ansible 설치 지침에 따라 Ansible 오픈 소스를 설치합니다. 설치에 대한 자세한 내용은 Ansible 설명서를 참조하십시오.

Ansible은 기본적으로 호스트 키 검사를 사용합니다. known\_hosts 파일에 다른 키를 사용하여 호스트를 다시 설치하면 오류 메시지가 나타납니다. known\_hosts 파일에 호스트가 나열되지 않으면 시작 시 키를 입력해야 합니다. /etc/ansible/ansible.cfg 또는 ~/.ansible.cfg 파일에서 다음 설정을 사용하여 호스트 키 확인을 사용하지 않도록 설정할 수 있습니다.

```
[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
```

호스트 키 검사 오류를 방지하려면 host\_key\_checking 및 record\_host\_keys를 False로 설정하고 ssh\_args에 추가 옵션 UserKnownHostsFile=/dev/null을 설정합니다. 또한 인벤토리가 처음에 비어 있는 경우 Ansible은 호스트 목록이 비어 있음을 경고합니다. 이로 인해 플레이 북 구문 검사가 실패합니다.

Ansible Vault를 사용하면 암호 또는 키와 같은 중요한 정보를 일반 텍스트가 아닌 암호화된 파일에 저장할 수 있습니다. Vault는 암호로 암호화됩니다. vRealize Automation Cloud Assembly에서, Ansible은 Vault를 사용하여 호스트 시스템에 대한 ssh 암호와 같은 데이터를 암호화합니다. 이는 Vault 암호 경로가 설정된 것으로 가정합니다.

ansible.cfg 파일을 수정하여 다음 형식으로 암호 파일의 위치를 지정할 수 있습니다.

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

Ansible이 암호를 자동으로 검색하도록 ANSIBLE\_VAULT\_PASSWORD\_FILE 환경 변수를 설정할 수도 있습니다. 예를 들어 ANSIBLE\_VAULT\_PASSWORD\_FILE=~/.vault\_pass.txt와 같이 설정할 수 있습니다.

vRealize Automation Cloud Assembly는 Ansible 인벤토리 파일을 관리하므로 vRealize Automation Cloud Assembly 사용자가 인벤토리 파일에 대한 **rwX** 액세스 권한이 있는지 확인해야 합니다.

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

vRealize Automation Cloud Assembly 오픈 소스 통합으로 루트가 아닌 사용자를 사용하려는 경우 vRealize Automation Cloud Assembly 오픈 소스 제공자가 사용하는 명령을 실행하려면 사용자에게 일련의 사용 권한이 필요합니다. 다음 명령은 사용자의 **sudoers** 파일에서 설정해야 합니다.

```
Defaults:myuser !requiretty
```

**askpass** 애플리케이션이 지정되지 않은 관리 그룹에 사용자가 속하지 않은 경우 사용자의 **sudoers** 파일에서 다음 명령을 설정합니다.

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

Ansible 통합을 설정하는 동안 오류가 발생하거나 다른 문제가 발생하면, Ansible 제어 시스템의 'cat~/var/tmp/vmware/provider/user\_defined\_script/\$(ls -t ~/var/tmp/vmware/provider/user\_defined\_script/ | head -1)/'에 있는 **log.txt** 파일을 참조하십시오.

## Ansible Tower 통합

지원되는 운영 체제 유형

- Red Hat Enterprise Linux 8.0 이상 64비트(x86)는 Ansible Tower 3.5 이상만 지원합니다.
- Red Hat Enterprise Linux 7.4 이상 64비트(x86).
- CentOS 7.4 이상 64비트(x86).

다음은 Ansible 타워 설치 중에 생성되는 샘플 인벤토리 파일입니다. vRealize Automation Cloud Assembly 통합 사용을 위해 이를 수정해야 할 수 있습니다.

```
[root@cava-env8-dev-001359 ansible-tower-setup-bundle-3.5.2-1.el8]# pwd

/root/ansible-tower-install/ansible-tower-setup-bundle-3.5.2-1.el8

[root@cava-env8-dev-001359 ansible-tower-setup-bundle-3.5.2-1.el8]# cat inventory

[tower]

localhost ansible_connection=local


[database]
```

```
[all:vars]

admin_password='VMware1!'

pg_host=''

pg_port=''

pg_database='awx'

pg_username='awx'

pg_password='VMware1!'

rabbitmq_port=5672

rabbitmq_vhost=tower

rabbitmq_username=tower

rabbitmq_password='VMware1!'

rabbitmq_cookie=cookiemonster

# Needs to be true for fqdns and ip addresses

rabbitmq_use_long_name=false

# Isolated Tower nodes automatically generate an RSA key for authentication;

# To disable this behavior, set this value to false

# isolated_key_generation=true
```

## vRealize Automation Cloud Assembly에서 Puppet Enterprise 통합 구성

vRealize Automation Cloud Assembly는 Puppet Enterprise 구성 관리와의 통합을 지원합니다.

Cloud Assembly에 외부 시스템으로 Puppet Enterprise를 추가하면 기본적으로 모든 프로젝트에서 Puppet Enterprise를 사용할 수 있는데 이를 특정 프로젝트로 제한할 수 있습니다.

Puppet Enterprise를 통합을 추가하려면 마스터의 Puppet 마스터 이름과 마스터의 호스트 이름 또는 IP 주소가 있어야 합니다.

오류 또는 정보 용도로 Puppet 로그를 확인해야 하는 경우, 다음 위치에서 해당 로그를 찾을 수 있습니다.

설명	로그 위치
관련 이벤트 생성 및 설치에 대한 로그	배포된 시스템 <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/   head -1)/`</code> 에 로그가 있습니다.  전체 로그는 <b>log.txt</b> 파일을 참조하십시오. 세부 Puppet 에이전트 로그에 대한 자세한 내용은 <a href="https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging">https://puppet.com/docs/puppet/4.8/services_agent_unix.html#logging</a> 을 참조하십시오.
Puppet 삭제 및 실행 관련 작업에 대한 로그	PE, <code>~/var/tmp/vmware/provider/user_defined_script/\$(ls -t ~/var/tmp/vmware/provider/user_defined_script/   head -1)/`</code> 에 로그가 있습니다. 전체 로그는 <b>log.txt</b> 파일을 참조하십시오.

## 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 Puppet을 선택합니다.
- 3 Puppet 구성 페이지에 필요한 정보를 입력합니다.
- 4 **검증**을 클릭하여 통합을 확인합니다.
- 5 **추가**를 클릭합니다.

## 결과

Puppet을 클라우드 템플릿에서 사용할 수 있습니다.

## 다음에 수행할 작업

Puppet 구성 요소를 원하는 클라우드 템플릿에 추가합니다.

- 1 Cloud Assembly의 클라우드 템플릿에서 클라우드 템플릿 메뉴의 [컨텐츠 관리] 머리글 아래에 있는 Puppet을 선택하고 Puppet 구성 요소를 캔버스로 끌어다 놓습니다.
- 2 오른쪽 창에서 Puppet 속성을 입력합니다.

속성	설명
마스터	이 클라우드 템플릿에 사용할 Puppet 기본 시스템의 이름을 입력합니다.
환경	Puppet 기본 시스템의 환경을 선택합니다.

속성	설명
역할	이 클라우드 템플릿에 사용할 Puppet 역할을 선택합니다.
에이전트 실행 간격	Puppet 에이전트가 이 클라우드 템플릿과 관련하여 배포된 가상 시스템에 적용할 구성 세부 정보를 Puppet 기본 시스템에서 풀링하는 주기입니다.

3 Puppet 구성 속성에 대한 YAML 코드를 보려면 오른쪽 창에서 [코드] 탭을 클릭합니다.

## vRealize Automation Cloud Assembly에서 Ansible 오픈 소스 통합 구성

vRealize Automation Cloud Assembly는 Ansible 오픈 소스 구성 관리와의 통합을 지원합니다. 통합을 구성한 후에는 Ansible 구성 요소를 새 배포 또는 기존 배포에 추가할 수 있습니다.

Ansible 오픈 소스를 vRealize Automation Cloud Assembly와 통합하는 경우 구성 관리 자동화를 위해 새 시스템이 프로비저닝될 때 지정된 순서로 하나 이상의 Ansible 플레이북을 실행하도록 Ansible 오픈 소스를 구성할 수 있습니다. 배포의 클라우드 템플릿에서 원하는 플레이북을 지정합니다.

Ansible 통합을 설정할 때 리소스 관리용 정보를 정의하는 인벤토리 파일 경로를 비롯해 Ansible Open 소스 호스트 시스템을 지정해야 합니다. 또한 Ansible 오픈 소스 인스턴스 액세스를 위해 이름과 암호를 제공해야 합니다. 나중에 Ansible 구성 요소를 배포에 추가할 때 키 기반 인증을 사용하도록 연결을 업데이트할 수 있습니다.

기본적으로 Ansible에서는 ssh를 사용하여 물리적 시스템에 연결합니다. Windows 시스템을 osType Windows 속성과 함께 클라우드 템플릿에 지정된 대로 사용하는 경우 connection\_type 변수가 자동으로 winrm으로 설정됩니다.

처음에는 Ansible 통합에서 통합에 제공된 사용자/암호 또는 사용자/키 자격 증명을 사용하여 Ansible 제어 시스템에 연결합니다. 연결에 성공하면 클라우드 템플릿에 제공된 플레이 북의 구문에 대한 유효성이 검사됩니다.

유효성 검사에 성공하면 Ansible 제어 시스템의 ~/var/tmp/vmware/provider/user\_defined\_script/에 실행 폴더가 생성됩니다. 이 위치에서 스크립트를 실행하여 인벤토리에 호스트를 추가하고, 인증 모드 설정을 비롯한 호스트 vars 파일을 생성하여 호스트에 연결하고, 마지막으로 플레이 북을 실행합니다. 이 시점에 클라우드 템플릿에 제공된 자격 증명을 사용하여 Ansible 제어 시스템에서 호스트에 연결합니다.

Ansible 통합은 IP 주소를 사용하지 않는 물리적 시스템을 지원합니다. AWS, Azure 및 GCP와 같은 공용 클라우드에서 프로비저닝된 시스템의 경우, 생성된 리소스의 주소 속성은 시스템이 공용 네트워크에 연결되어 있는 경우에만 시스템의 공용 IP 주소로 채워집니다. 공용 네트워크에 연결되어 있지 않은 시스템의 경우 Ansible 통합은 시스템에 연결되어 있는 네트워크에서 IP 주소를 찾습니다. 여러 개의 네트워크가 연결되어 있다면 최소 deviceIndex(즉 시스템에 연결된 NIC(네트워크 인터페이스 카드)의 인덱스)가 있는 네트워크를 찾습니다. deviceIndex 속성이 Blueprint에 지정되어 있지 않은 경우 통합은 연결된 첫 번째 네트워크를 사용합니다.

vRealize Automation Cloud Assembly에서 통합을 위한 Ansible 오픈 소스 구성에 대한 자세한 내용은 [vRealize Automation Cloud Assembly의 구성 관리 항목](#)을 참조하십시오.

## 사전 요구 사항

- Ansible 제어 시스템은 Ansible 버전 2.6.0 이상을 사용해야 합니다.
- 사용자에게 Ansible 인벤토리 파일이 있는 디렉토리에 대한 읽기/쓰기 액세스 권한이 있어야 합니다. 또한 사용자에게 인벤토리 파일(이미 있는 경우)에 대한 읽기/쓰기 액세스 권한이 있어야 합니다.
- sudo 옵션과 함께 루트가 아닌 사용자를 사용하는 경우 sudoers 파일에 다음이 설정되어 있는지 확인합니다.

```
Defaults:user_name !requiretty
```

및

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

- /etc/ansible/ansible.cfg 또는 ~/.ansible.cfg에 host\_key\_checking = False를 설정하여 호스트 키 검사가 사용되지 않도록 설정합니다.
- 다음 줄을 /etc/ansible/ansible.cfg 또는 ~/.ansible.cfg 파일에 추가하여 Vault 암호가 설정되도록 해야 합니다.

```
vault password_file = /path/to/password_file
```

Vault 암호 파일에는 일반 텍스트 형식의 암호가 포함되어 있으며, 클라우드 템플릿 또는 배포에서 다음 예에 표시된 것과 같이 ACM과 노드 간에 사용할 사용자 이름과 암호 조합을 제공하는 경우에만 사용됩니다.

```
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
```

- 플레이 북을 실행하는 동안 호스트 키 실패를 방지하려면 /etc/ansible/ansible config에 다음 설정을 포함하는 것이 좋습니다.

```
[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.
```

## 절차

- 1 인프라 > 연결 > 통합을 선택하고 통합 추가를 클릭합니다.

- 2 [Ansible]을 클릭합니다.

[Ansible 구성] 페이지가 나타납니다.

- 3 Ansible 오픈 소스 인스턴스에 대한 호스트 이름, 인벤토리 파일 경로 및 기타 필수 정보를 입력합니다.
- 4 **검증**을 클릭하여 통합을 확인합니다.
- 5 **추가**를 클릭합니다.

## 결과

Ansible을 클라우드 템플릿에서 사용할 수 있습니다.

## 다음에 수행할 작업

Ansible 구성 요소를 원하는 클라우드 템플릿에 추가합니다.

- 1 클라우드 템플릿 캔버스 페이지에서, 클라우드 템플릿 옵션 메뉴의 [구성 관리] 머리글 아래에서 [Ansible]을 선택하고 Ansible 구성 요소를 캔버스로 끕니다.
- 2 오른쪽 패넬을 사용하여 적절한 Ansible 속성을 구성합니다(예: 실행할 플레이북 지정).

Ansible에서 사용자는 변수를 단일 호스트에 할당한 다음 나중에 이를 플레이북에서 사용할 수 있습니다.

Ansible 오픈 소스 통합을 사용하면 이러한 호스트 변수를 클라우드 템플릿에 지정할 수 있습니다.

hostVariables 속성은 Ansible 제어 시스템이 예상하는 올바른 YAML 형식이어야 하며, 이 콘텐츠는 다음 위치에 배치됩니다.

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

Ansible 인벤토리 파일의 기본 위치는 Cloud Assembly의 [통합] 페이지에 추가된 Ansible 계정에서 정의됩니다. Ansible 통합은 클라우드 템플릿에서 hostVariable YAML 구문을 검증하지 않지만 형식 또는 구문이 잘못된 경우 플레이북을 실행할 때 Ansible 제어 시스템에서 예외가 발생합니다.

다음 클라우드 템플릿 YAML 코드 조각은 hostVariables 속성의 사용 예를 보여 줍니다.

```
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}
```

Ansible 통합은 다음 중 한 가지 방법으로 클라우드 템플릿에 인증 자격 증명을 제공해야 합니다.

- Ansible 리소스의 사용자 이름 및 암호.
- Ansible 리소스의 사용자 이름 및 `privateKeyFile`.
- `generatedPublicPrivateKey`에 `remoteAccess`를 지정하여 Ansible 리소스의 사용자 이름 및 계산 리소스의 `privatekey`.

클라우드 템플릿에서 통합 계정에 지정된 사용자가 Ansible 플레이 북 경로에 액세스할 수 있는지 확인합니다. 절대 경로를 사용하여 플레이 북 위치를 지정할 수 있지만 필수는 아닙니다. Ansible 통합 자격 증명 이 시간이 지나면서 변경되더라도 경로가 유효한 상태로 유지되려면 사용자의 홈 폴더에 절대 경로를 사용하는 것이 좋습니다.

## vRealize Automation Cloud Assembly에서 Ansible Tower 통합 구성

Ansible Tower를 vRealize Automation Cloud Assembly와 통합하여 배포된 리소스의 구성 관리를 지원할 수 있습니다. 통합을 구성한 후에는 클라우드 템플릿 편집기에서 Ansible 구성 요소를 새 배포 또는 기존 배포에 추가할 수 있습니다.

vRealize Automation Cloud Assembly는 Ansible Tower 버전 3.5, 3.6 및 3.7과의 통합을 지원합니다.

### 사전 요구 사항

- 관리자가 아닌 사용자에게 Ansible Tower에 액세스할 수 있는 적절한 권한을 부여합니다. 대부분의 구성에서 작동하는 옵션에는 두 가지가 있습니다. 구성에 가장 적합한 옵션을 선택합니다.
  - 조직 수준에서 인벤토리 관리자 및 작업 템플릿 관리자 역할을 사용자에게 부여합니다.
  - 특정 인벤토리에 대한 [관리자] 권한과 프로비저닝에 사용되는 모든 작업 템플릿에 대한 [실행] 역할을 사용자에게 부여합니다.
- Ansible Tower에서 배포에 사용할 적절한 자격 증명 및 템플릿을 구성해야 합니다. 템플릿은 배포에 사용할 인벤토리와 플레이 북을 정의합니다. 작업 템플릿과 플레이 북 간에는 1:1 매핑이 있습니다. 플레이 북은 YAML과 유사한 구문을 사용하여 템플릿과 연결된 작업을 정의합니다. 대부분의 일반적인 배포에서는 시스템 자격 증명을 사용하여 인증합니다.
  - a Ansible Tower에 로그인하고 [작업 템플릿] 섹션으로 이동합니다.
  - b [새 작업 템플릿 추가]를 선택합니다.
    - 이미 생성한 자격 증명을 선택합니다. Ansible Tower에서 관리할 시스템의 자격 증명입니다. 각 작업 템플릿에 대해 하나의 자격 증명 개체가 있을 수 있습니다.
    - [제한] 선택으로 [시작 메시지]를 선택합니다. 이렇게 하면 작업 템플릿이 vRealize Automation Cloud Assembly에서 프로비저닝 또는 프로비저닝 해제되는 노드에 대해 실행됩니다. 이 옵션을 선택하지 않으면 작업 템플릿이 포함된 Blueprint가 배포될 때 "제한이 설정되지 않음" 오류가 표시됩니다.
- [Ansible Tower 작업] 탭에서 vRealize Automation Cloud Assembly로부터 호출된 작업 템플릿의 실행을 볼 수 있습니다.

## 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 [Ansible Tower]를 클릭합니다.  
[Ansible 구성] 페이지가 나타납니다.
- 3 **호스트 이름**(IP 주소일 수 있음)을 입력하고 Ansible Tower 인스턴스에 대한 기타 필수 정보를 입력합니다.
- 4 해당하는 Ansible Tower 인스턴스에 대한 UI 기반 인증 **사용자 이름** 및 **암호**를 입력합니다.
- 5 **검증**을 클릭하여 통합을 확인합니다.
- 6 통합을 위한 적합한 **이름** 및 **설명**을 입력합니다.
- 7 **추가**를 클릭합니다.

## 결과

Ansible Tower를 클라우드 템플릿에서 사용할 수 있습니다.

### 다음에 수행할 작업

Ansible Tower 구성 요소를 원하는 클라우드 템플릿에 추가합니다. 통합 계정에 지정된 사용자에게 대한 실행 권한이 있는 적용 가능한 작업 템플릿을 지정해야 합니다.

- 1 클라우드 템플릿 캔버스 페이지에서, **Blueprint** 옵션 메뉴의 [구성 관리] 머리글 아래에서 [Ansible]을 선택하고 Ansible Tower 구성 요소를 캔버스로 끕니다.
- 2 오른쪽 패널을 사용하여 적절한 Ansible 속성을 구성합니다(예: 작업 템플릿).

## vRealize Automation에서 SaltStack Config 통합 구성

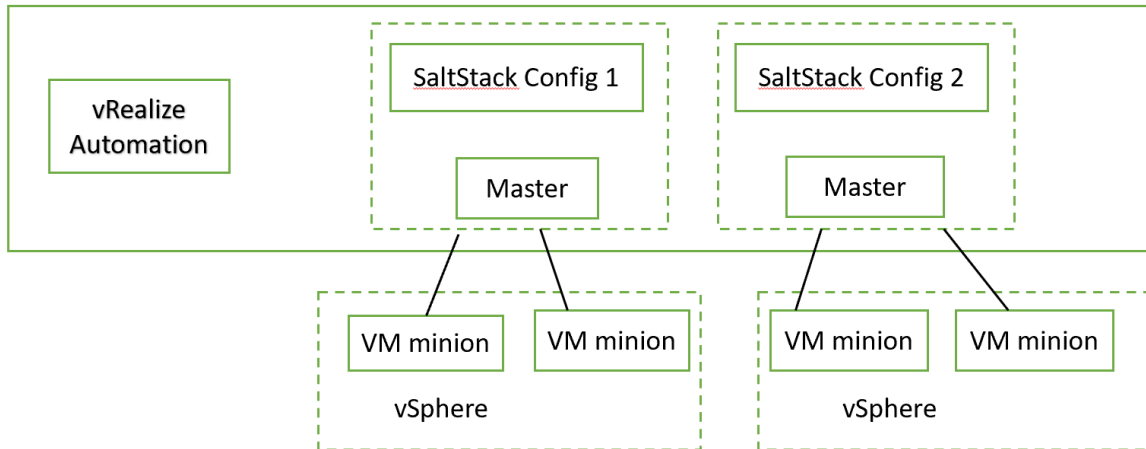
vRealize Automation에서 SaltStack Config 서비스에 액세스하고 SaltStack Config 개체 및 작업을 사용하여 하도록 SaltStack Config 통합을 구성할 수 있습니다.

vRealize Automation SaltStack Config를 사용하면 이벤트 기반 자동화를 사용하여 규모에 관계없이 가상 시스템에 소프트웨어를 프로비저닝, 구성 및 배포할 수 있습니다. SaltStack Config를 사용하여 전체 환경에 최적의 규정 준수 소프트웨어 상태를 정의하고 적용할 수도 있습니다.

통합 vRealize Automation SaltStack Config는 다음과 같은 조건에서 vRealize Automation에 사용할 수 있습니다.

- SaltStack Config를 사용하여 관리되는 시스템은 Salt 마스터에 연결할 수 있어야 하며 SaltStack Config 통합 지점 및 Salt 마스터와 동일한 네트워크에 상주해야 합니다.
- SaltStack Config 통합은 설치 중에 특정 호스트에 연결됩니다.
- 프라이빗 네트워크에 배포된 vSphere 시스템은 SaltStack Config 통합 및 Salt 마스터와의 연결을 시작할 수 있어야 합니다.
- Salt 미니언은 cloud-init 또는 Cloudbase-init 스크립트를 사용하여 설치됩니다.

- 구성 관리 방법, 특히 cloud-init 또는 Cloudbase-init 스크립팅을 사용하여 SaltStack Config 통합 정보 및 미니언을 vRealize Automation 클라우드 템플릿 및 배포에 추가합니다.
- 각 vRealize Automation 테넌트는 SaltStack Config 통합 하나와 Salt 마스터 하나를 지원할 수 있습니다. Salt 마스터는 여러 미니언을 지원할 수 있습니다.



- vRealize Automation에서 SaltStack Config 통합을 삭제해도 SaltStack Config 통합을 사용하는 기존 배포가 있는지 확인되지 않습니다.

## 사전 요구 사항

- vRealize Automation 관리자 자격 증명 및 SaltStack Config 관리자 자격 증명(루트 수준 액세스)이 있는지 확인합니다.

SaltStack Config 통합을 생성하려면 vRealize Automation 관리자 자격 증명 및 SaltStack Config 관리자 자격 증명(루트 수준 액세스)이 필요합니다.

또한, SaltStack Config 서비스 자체를 열고 작업하려면 SaltStack Config 관리자 자격 증명이 필요합니다.

vRealize Automation에 액세스하려면 vRealize Automation 자격 증명을, SaltStack Config에 액세스하려면 SaltStack Config 자격 증명을 사용합니다.

SaltStack Config 관리자 자격 증명에 대한 자세한 내용은 제품 설명서의 [SaltStack Config 설치 가이드](#)를 참조하십시오.

- SaltStack Config 서비스가 설치되었는지 확인합니다.

vRLCM(vRealize Suite Lifecycle Manager)을 사용하여 vRealize Automation과 함께 사용할 SaltStack Config 서비스를 설치합니다. 이 작업은 Salt 마스터 또는 호스트를 설치하고 필수 vRealize Automation 속성 그룹도 구성합니다. 필요한 SaltStack Config 설치 및 서비스 구성 지침은 [VMware vRealize Suite Lifecycle Manager 설명서](#)에서 다음 항목을 참조하십시오.

- [제품 세부 정보 구성](#)
- [vRealize Suite Lifecycle Manager에서 환경 생성](#)

- 설치 마법사를 사용하여 새 사설 클라우드 환경 생성
- SaltStack Config 통합에 사용할 Salt 마스터에 마스터 플러그인이 포함되어 있는지 확인합니다.  
마스터 플러그인 설치에 대한 자세한 내용은 [SaltStack Config 설치 및 업그레이드](#) 설명서를 참조하십시오.
- vRealize Automation에서 SaltStack Config 서비스 관리자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#)의 내용을 참조하십시오.
- vRealize Automation에서 Cloud Assembly 서비스 관리자 역할이 있는지 확인합니다. [vRealize Automation의 조직 및 서비스 사용자 역할](#)의 내용을 참조하십시오.

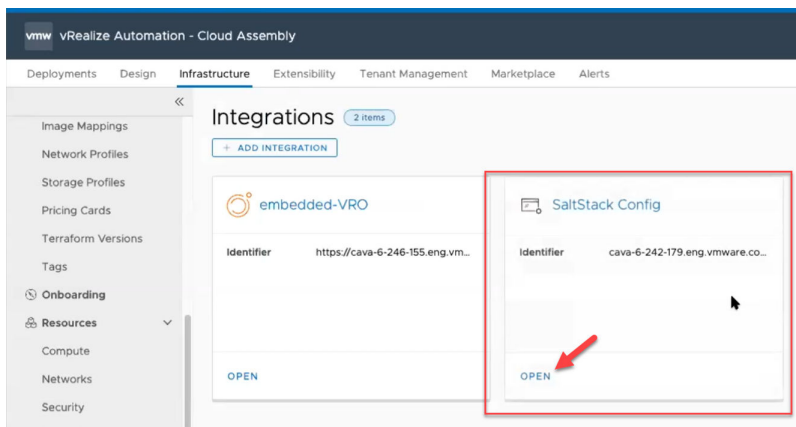
SaltStack Config 서비스 구성 중에 사용되는 API 호출에 대한 관련 정보는 [vRealize Automation 통합](#)을 위해 [SaltStack Config](#)를 설치 및 구성할 때 사용되는 [API 호출](#) 항목을 참조하십시오.

## vRealize Automation에서 SaltStack Config 통합 구성

vRealize Automation용 SaltStack Config를 설치한 후에는 Cloud Assembly의 **인프라 > 통합** 페이지에서 열거나 vRealize Automation 서비스 콘솔에서 서비스로 열 수 있습니다.

**참고** 다른 vRealize Automation 통합 유형과 달리 SaltStack Config 통합은 **인프라 > 연결 > 통합 > 통합 추가**를 선택하여 추가할 수 없습니다. SaltStack Config 통합은 vRealize Suite Lifecycle Manager를 사용하여 설치될 때까지 사용할 수 없습니다.

- 1 vRealize Automation Cloud Assembly에서 **인프라 > 연결 > 통합**을 선택하고 사용 가능한 SaltStack Config 통합 타일을 선택한 후 **열기**를 클릭합니다.



- 2 양식을 작성합니다.

The screenshot shows the 'SaltStack Config' configuration page in the vRealize Automation Cloud Assembly console. The left sidebar contains navigation options like Image Mappings, Network Profiles, Storage Profiles, Pricing Cards, Terraform Versions, Tags, Onboarding, Resources, Compute, Networks, Security, Storage, Machines, Volumes, Kubernetes, and Activity. The main content area has a 'Name' field set to 'SaltStack Config', a 'Description' field, and a 'SaltStack Integration' section with 'Hostname' (cava-6-242-179.eng.vmware.com), 'Username' (root), and 'Password' fields. A 'VALIDATE' button and a message 'Validate credentials before making changes.' are present. Below this is a 'Capabilities' section with a 'Capability tags' field. At the bottom are 'SAVE' and 'CANCEL' buttons.

a 설명을 입력하거나 편집합니다.

통합 이름은 SaltStack Config가 설치될 때 생성됩니다.

b 지정된 호스트에 액세스하는 데 사용되는 SaltStack Config 관리자 이름 및 암호를 입력합니다.

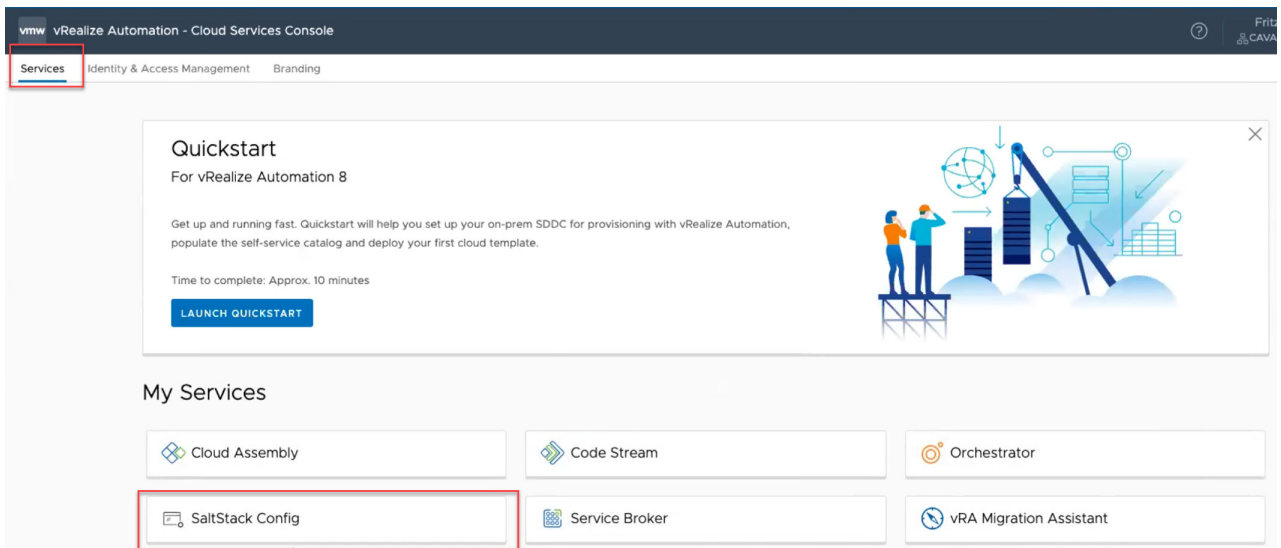
호스트 이름 값은 SaltStack Config 설치 중에 마스터 주소 값으로 지정됩니다. 설치 후에는 변경할 수 없습니다.

c 검증 버튼을 클릭하여 SaltStack Config 통합 호스트에 대한 관리자 액세스를 확인합니다.

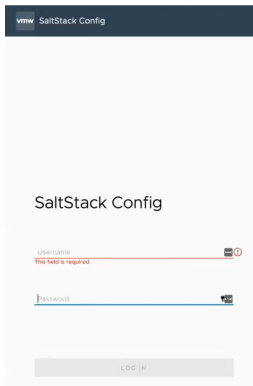
d (선택 사항) 기능 태그를 입력합니다. 태그 지정에 대한 자세한 내용은 [vRealize Automation Cloud Assembly](#)에서 [제약 조건 태그 사용 항목](#)을 참조하십시오.

e 저장 버튼을 클릭합니다.

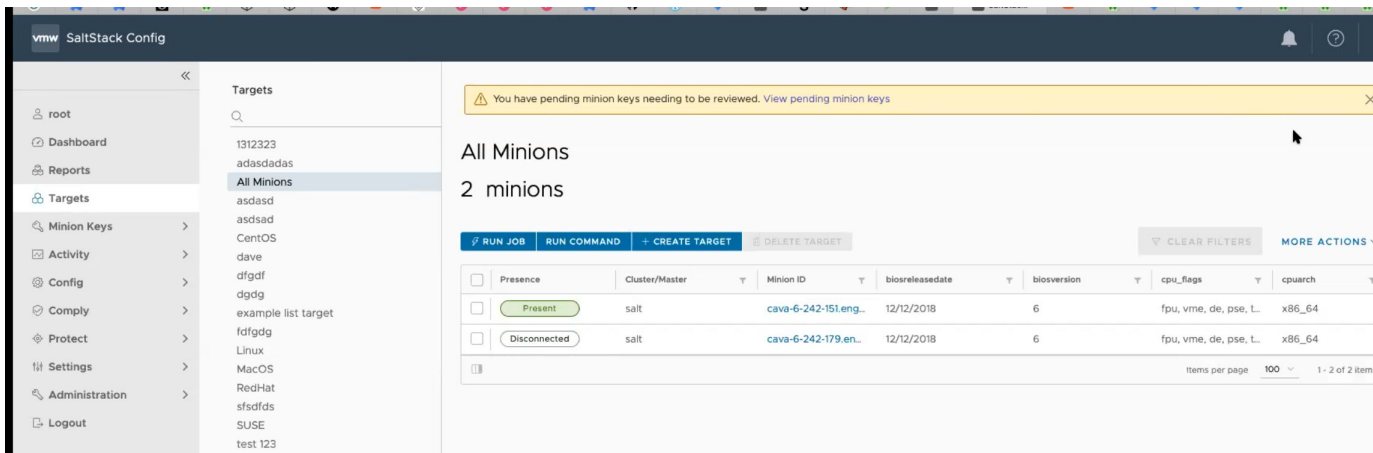
SaltStack Config 통합 지점을 저장한 후 통합 페이지에서 또는 vRealize Automation 서비스 콘솔에서 SaltStack Config 통합 서비스 자체를 열 수 있습니다.



- 3 저장된 통합 지점을 클릭하여 SaltStack Config 통합 서비스를 열고 호스트에 액세스합니다.
- 4 SaltStack Config에 로그인하라는 메시지가 표시되면 SaltStack Config 관리자 이름과 암호를 입력합니다.



샘플 SaltStack Config 페이지는 아래와 같습니다.



vRealize Automation 클라우드 템플릿 및 배포에서 SaltStack Config 개체와 작업을 사용하는 것에 대한 자세한 내용은 [vRealize Automation에서 SaltStack Config 미니언으로 작업하는 방법](#) 항목을 참조하십시오.

SaltStack Config 서비스 자체에서 SaltStack Config 개체를 구성하고 사용하는 방법에 대한 자세한 내용은 다음 "SaltStack Config 사용 방법 알아보기" 섹션에 설명된 리소스를 참조하십시오.

## SaltStack Config 사용 방법 알아보기

SaltStack Config는 vRealize Automation에서 통합하고 사용할 수 있는 독립형 제품입니다. SaltStack Config 제품 설명서는 별도의 PDF로 제공됩니다. SaltStack Config 제품 설명서를 사용하여 vRealize Automation 클라우드 템플릿 및 배포에서 SaltStack Config 기능을 활용하는 방법을 알아볼 수 있습니다.

SaltStack Config 개체를 생성하고 SaltStack Config 서비스 자체에서 작업하는 것에 대한 자세한 내용은 "vRealize Automation SaltStack Config 사용 및 관리"의 [vRealize Automation에서 SaltStack Config 사용](#)을 참조하십시오.

## vRealize Automation에서 SaltStack Config 미니언으로 작업하는 방법

vRealize Automation에서 구성된 SaltStack Config 통합에 대한 액세스 권한이 있는 클라우드 템플릿 개발자는 Salt 미니언을 클라우드 템플릿에서 대상 Salt 마스터에 설치하고 배포할 수 있습니다.

미니언은 Salt 미니언 서비스를 실행하는 노드입니다. 이 서비스는 Salt 마스터의 명령을 수신 대기하고 요청된 작업을 수행합니다. vRealize Automation 클라우드 템플릿에서 미니언을 배포할 있습니다.

### 사전 요구 사항

- vRealize Suite Lifecycle Manager를 사용하여 SaltStack Config 서비스를 설치하고 구성합니다.
- vRealize Automation에서 SaltStack Config 통합을 생성합니다.
- Salt 미니언 서비스를 설치합니다.

Salt 미니언 서비스를 설치하는 과정은 다면적입니다. RedHat Linux 및 CentOS에 Salt 미니언 서비스를 설치하는 방법에 대한 자세한 내용은 SaltStack Config 제품 설명서의 [SaltStack Config 설치 가이드](#)에서 "Salt 설치 또는 업그레이드"를 참조하십시오. 다른 운영 체제의 경우 <http://repo.saltstack.com/>에서 참조하십시오.

- cloud-init(Linux) 또는 Cloudbase-init(Windows)에 액세스할 수 있고 능숙하게 사용할 수 있어야 합니다. vRealize Automation SaltStack Config 통합에 대해 구성된 Salt 마스터에 Salt 미니언을 추가하려면 클라우드 템플릿의 가상 시스템이 cloud-init(Linux) 또는 Cloudbase-init(Windows)를 지원해야 합니다.

### SaltStackConfiguration 속성 그룹

vRealize Suite Lifecycle Manager에서 SaltStack Config 서비스를 설치하고 구성하는 과정에서 SaltStackConfiguration이라는 vRealize Automation 속성 그룹이 생성됩니다.

SaltStackConfiguration 속성 그룹 값은 vRealize Automation 클라우드 템플릿 또는 배포에서 SaltStack Config 통합을 구성할 때 사용합니다. 또한 vRealize Automation 템플릿(이전 명칭: Blueprint)의 cloud-init 또는 Cloudbase-init 기반 구성에서 미니언을 설치하는 데에도 사용합니다.

SaltStackConfiguration의 두 속성은 masterAddress(SaltStack Config 통합 지점의 **호스트 이름** 설정과 매칭) 및 masterFingerprint입니다. 샘플 SaltStackConfiguration 속성 그룹은 아래와 같습니다.

**SaltStackConfiguration** DELETE

Edit a property group to make it available for use in cloud templates

**Property Group Type** Constant Values ⓘ

**Name \*** SaltStackConfiguration

**Display Name** vRealize Automation SaltStack Config Server Properties

**Description** Property Group with vRealize Automation SaltStack Config server details such as primary node address and fingerprint.

**Scope** ☐ ⓘ Available for any project  
Property group will be available in any project

**Properties** 2 items  
Add at least one property in order to create a property group

[+ NEW PROPERTY](#) [X DELETE](#)

<input type="checkbox"/>	Name	Display Name	Type	Constant Value
<input type="checkbox"/>	masterAddress		string	cava-6-242-179.eng.vmware.com
<input type="checkbox"/>	masterFingerPrint		string	aa-94-9f1b-035e-dd39-3586-ce7e-2b4d-12a51c35f9c9acc655182d8058326fc2ace1

## vRealize Automation에 대해 구성된 Salt 마스터에 미니언 추가

SaltStack Config가 설치되면 Salt 마스터 IP 주소가 지정됩니다. 이 마스터 IP 주소는 vRealize Automation 클라우드 템플릿에서 미니언을 배포할 때 masterAddress 속성으로 사용됩니다.

vRealize Automation 클라우드 템플릿 또는 배포에서 cloud-init 또는 Cloudbase-init 스크립팅을 사용하여 미니언을 설치하고 배포합니다. 이러한 형식 중 하나를 사용하는 클라우드 구성 스크립트를 나타내는 이미지 매핑을 사용할 수도 있습니다. vRealize Automation SaltStack Config 통합에 연결된 Salt 마스터에 Salt 미니언을 추가하려면 대상 시스템이 cloud-init(Linux) 또는 Cloudbase-init(Windows)를 지원해야 합니다. vRealize Automation 클라우드 구성 스크립팅은 두 가지 형식을 모두 지원합니다.

클라우드 템플릿에서 minionId 값과 cloudConfig 값을 사용하여 시스템 리소스를 구성하고 속성 그룹 SaltStackConfiguration을 참조합니다. SaltStackConfiguration 속성 그룹은 vRealize Suite Lifecycle Manager에서 SaltStack Config 서비스를 설치하고 구성하는 동안 생성됩니다. 여기에는 masterAddress 및 masterFingerprint 속성이 포함됩니다.

minionId 값은 클라우드 템플릿 코드의 cloudConfig 섹션에서 시스템의 /salt/minion\_id에 대해 지정된 값과 일치해야 합니다.

Windows 기반 및 Linux 기반 vRealize Automation 클라우드 템플릿 코드의 예는 다음과 같습니다. 클라우드 구성 스크립팅은 다음 방법 중 하나를 사용하여 지정할 수 있습니다.

- 클라우드 템플릿 코드에서 호출되는 vRealize Automation 이미지
- 클라우드 템플릿 코드에서 호출되는 클라우드 구성 스크립트
- 클라우드 템플릿 코드에 직접 추가되는 클라우드 구성 스크립트 콘텐츠

참고: Salt 미니언이 포함된 클라우드 템플릿을 배포할 때 배포가 vRealize Automation Cloud Assembly에 보이지 않으면 vRealize Automation Service Broker 서비스를 사용하여 배포를 표시할 수 있습니다.

## 예 - Linux 기반 배포 및 cloud-init

cloud-init를 지원하는 Linux 기반 시스템용 미니언을 배포하기 위한 샘플 클라우드 템플릿 구성이 아래에 나와 있습니다.

```
inputs: {}
resources:
  LinuxVM-Minion:
    type: Cloud.vSphere.Machine
    properties:
      name: salt-minion-vra
      cpuCount: 1
      totalMemoryMB: 2048
      imageRef: 'https://build-artifactory.eng.vmware.com/symphony-infra-local/ubuntu/
releases/xenial/release-20190605/ubuntu-16.04-server-cloudimg-amd64.ova'
      minionId: '${resource.LinuxVM-Minion.resourceName}'
      cloudConfig: |
        #cloud-config
        runcmd:
          - sudo echo '${resource.LinuxVM-Minion.resourceName}' > /etc/salt/minion_id
        salt_minion:
          pkg_name: 'salt-minion'
          service_name: 'salt-minion'
          config_dir: '/etc/salt'
          conf:
            master: ${propgroup.SaltStackConfiguration.masterAddress}
            master_finger: ${propgroup.SaltStackConfiguration.masterFingerPrint}
        remoteAccess:
          authentication: publicPrivateKey
          sshKey: ssh-rsa your-public-key
        networks:
          - network: '${resource.vSphere_Network.id}'
  vSphere_Network:
    type: Cloud.Network
    properties:
      networkType: existing
```

## 예 - Windows 기반 배포 및 Cloudbase-init

Cloudbase-init를 지원하는 Windows 기반 시스템용 미니언을 배포하기 위한 샘플 클라우드 템플릿 구성이 아래에 나와 있습니다.

```
formatVersion: 1
inputs: {}
resources:
  WindowsVM-Minion:
    type: Cloud.vSphere.Machine
    properties:
      image: win2016
      flavor: medium
      customizationSpec: Windows
      minionId: '${resource.WindowsVM-Minion.resourceName}'
      networks:
        - network: '${resource.wpnet.id}'
```

```

    name: '${wpnet.name}'
    assignPublicIpAddress: true
  cloudConfig: |
    #ps1_sysnative
    [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12;
    Invoke-WebRequest -OutFile C:\Salt-Minion-3002.2-Py3-AMD64-Setup.exe -Uri https://
    repo.saltstack.com/windows/Salt-Minion-3002.2-Py3-AMD64-Setup.exe
    Start-Process -Wait -FilePath "C:\Salt-Minion-3002.2-Py3-AMD64-Setup.exe"
    -ArgumentList "/S" -PassThru
    ((Get-Content -path C:\salt\conf\minion -Raw) -replace "#master: salt", "master: $
    {propgroup.SaltStackConfiguration.masterAddress}") | Set-Content -Path C:\salt\conf\minion
    ((Get-Content -path C:\salt\conf\minion -Raw) -replace "#master_finger: '",
    "master_finger: '${propgroup.SaltStackConfiguration.masterFingerPrint}')" | Set-Content -Path
    C:\salt\conf\minion
    Set-Content -Path C:\salt\conf\minion_id -Value '${resource.WindowsVM-
    Minion.resourceName}'
    C:\salt\salt-call.bat service.restart salt-minion
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: existing

```

## 예 - 미리 정의된 CentOS 이미지 매핑을 기반으로 하는 배포

이 간단한 클라우드 템플릿 예에서는 vSphere 네트워크의 vSphere 시스템에 단일 미니언을 배포합니다. CentOS 이미지에는 필요한 모든 배포 매핑이 포함되어 있습니다.

```

inputs: {}
resources:
  SaltVM:
    type: Cloud.vSphere.Machine
    forceRecreate: true
    properties:
      name: salt-vm-vsphere
      image: centos72x64
      flavor: small
      saltConfiguration:
        masterId: saltstack_enterprise_installer
        minionId: minion1
        saltEnvironment: env1
        stateFiles:
          - state1
          - state2
      remoteAccess:
        authentication: usernamePassword
        username: root
        password: dynayo
      networks:
        - network: '${resource["SaltNetwork"].id}'
  SaltNetwork:

```

```

type: Cloud.Network
properties:
  name: salt-vsphere-network
  networkType: existing

```

## Powershell 명령을 구성하는 방법

다음 단계를 사용하여 cloudConfig 섹션에서 PowerShell 명령을 구성할 수 있습니다. 자세한 내용은 SaltStack Config [제품 설명서](#)를 참조하십시오.

- 1 SaltStack 웹 사이트에서 Salt 미니언 서비스 설치 관리자를 다운로드합니다.
- 2 Windows VM에 Salt 미니언 서비스를 설치합니다.
- 3 미니언 구성에서 Salt 마스터 호스트 이름 값을 업데이트합니다.
- 4 미니언 구성에서 Salt 마스터 지문 값을 업데이트합니다.
- 5 minion\_id를 Salt 마스터에서 미니언을 수락하는 데 사용될 VM 리소스 이름으로 설정합니다.
- 6 미니언을 다시 시작합니다.

## Salt 미니언 배포에 대한 추가 정보

통합 SaltStack Config를 사용하여 vRealize Automation 클라우드 템플릿에서 Salt 미니언을 배포하는 대체 방법은 VMware 블로그 문서 [Cloud Assembly 및 ABX 암호\(및 vRealize SaltStack 미니언 에이전트 설치에 사용하는 방법\)](#)를 참조하십시오.

## 스케줄링을 위한 SaltStack Config 사용 예

- 예: SaltStack Config API 호출을 수행하여 스케줄러를 설정합니다.

이 방법은 vRealize Automation 및 SaltStack Config를 구성할 때 사용합니다. Python 클라이언트 또는 HTTP-RPC 브리지를 사용하여 SaltStack Config에 연결하고 다음 스크립트를 실행합니다.

```

schedule.save(name="name_of_schedule_entry",
              schedule=<schedule description>,
              masters=<list of targeted masters or '*' for all of them>,
              tgt=<master+minion target description>,
              cmd=<one of "local", "runner", or "wheel">,
              fun="<name to schedule>",
              arg=<arguments to function>,
              tgt_uuid=<uuid of existing target, use this or tgt, not both>,
              job_uuid=<uuid of existing job, use this or fun/arg, not both>,
              enabled=<True or False to enable the schedule after saving>)

```

이 방법은 모든 스케줄을 동일한 위치에 유지합니다. 그렇지 않으면, 마스터/미니언 수준에서 생성된 스케줄을 보려는 경우 모든 마스터 및 이것과 연결된 미니언을 쿼리해야 합니다.

- 예: 미니언을 구성하여 스케줄러를 설정합니다.

이 방법은 미니언을 설치할 때 사용합니다.

`schedule.add` 함수를 사용하거나 미니언에 대해 `schedule.present` 상태를 실행하여 미니언을 지정합니다. 예를 들면 다음과 같습니다.

```
salt <minion_to_schedule> <name_of_job> function='cmd.run' job_args="[some command]"
seconds=3600
```

미니언에서 직접 **Salt** 호출을 실행하는 경우 마스터 없이도 스케줄러를 구성할 수 있습니다. 다음 예와 같이 `salt <minion_to_schedule> state.apply schedulestate` 및 파일 `schedulestate.sls`를 사용하여 미니언을 지정합니다.

```
job1:
  schedule.present:
    - function: state.sls
    - job_args:
      - httpd
    - job_kwargs:
      test: True
    - when:
      - Monday 5:00pm
      - Tuesday 3:00pm
      - Wednesday 5:00pm
      - Thursday 3:00pm
      - Friday 5:00pm
```

## vRealize Automation 통합을 위해 SaltStack Config를 설치 및 구성할 때 사용되는 API 호출

SaltStack Config 서버를 설치하는 것 외에도 vRealize Suite Lifecycle Manager는 vRealize Automation과의 통합을 지원하기 위해 추가 작업을 수행합니다.

vRealize Suite Lifecycle Manager가 SaltStack Config 서비스를 설치할 때, SaltStack Config 서버를 vRealize Automation의 통합 서비스 계정으로 추가하고 Salt 마스터 주소와 지문을 사용하여 새 vRealize Automation 속성 그룹을 생성합니다.

다음 코드 조각은 vRealize Suite Lifecycle Manager가 vRealize Automation에 대한 SaltStack Config 서비스를 설치 및 구성할 때 사용하는 API 호출에 대한 추가적인 정보를 제공합니다.

### SaltStack 통합 계정 생성

```
POST https://<vRA_hostname>/provisioning/uerp/provisioning/mgmt/endpoints?enhance Collapse
source
curl --location --request POST 'https://<vRA_hostname>/provisioning/uerp/provisioning/mgmt/
endpoints?enhance' \
--header 'Authorization: Bearer <token>' \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--data-raw '{
  "endpointProperties": {
    "privateKeyId": "<SaltStackConfigServer_Username>",
    "privateKey": "<SaltStackConfigServer_Password>",
    "hostName": "<SaltStackConfigServer_Hostname>"
  },
  }
```

```

    "customProperties": {
      "isExternal": "true"
    },
    "endpointType": "saltstack",
    "name": "SaltStack Configuration Server"
  }

```

## SaltStack 통합 계정 목록 가져오기

```

GET https://<vRA_hostname>/provisioning/uerp/provisioning/mgmt/endpoints?
expand&external&$filter=(endpointType eq 'saltstack' and customProperties.isExternal eq
'true')&$orderby=name asc&$top=20&$skip=0 Collapse source
curl --location --request GET 'https://<vRA_hostname>/provisioning/uerp/provisioning/mgmt/
endpoints?
expand&external&$filter=(endpointType%20eq%20%27saltstack%27%20and%20customProperties.isExtern
al%20eq%20%27true%27)&$orderby=name%20asc&$top=20&$skip=0' \
--header 'Authorization: <token>'

```

URL:

```

GET https://tenant1-200.lcm-34-209.sqa.local/provisioning/uerp/provisioning/mgmt/endpoints?
expand&external&$filter=(endpointType eq 'saltstack' and customProperties.isExternal eq
'true')&$orderby=name asc&$top=20&$skip=0

```

## SaltStack 통합 계정 삭제

```

DELETE https://<vRA_HostName>/provisioning/uerp/provisioning/mgmt/endpoints/resources/
endpoints/<EndpointID>?external Collapse source
curl --location --request DELETE 'https://<vRA_HostName>/provisioning/uerp/provisioning/mgmt/
endpoints/resources/endpoints/<EndpointID>?external' \
--header 'Authorization: Bearer <token>'

```

EndpointID should be the the id from documentSelfLink of the endpoint state

## Salt 마스터 주소 및 지문을 포함하는 속성 그룹 생성

```

POST https://<vRA_hostname>/properties/api/property-groups Collapse source
curl --location --request POST 'https://cava-6-243-085.eng.vmware.com/properties/api/property-
groups' \
--header 'Authorization: Bearer <token>' \
--header 'Content-Type: application/json' \
--data-raw '{
  "id": null,
  "name": "SaltStackConfiguration",
  "displayName": "SaltStack Config Server Properties",
  "description": "Property Group with SaltStack Config server details such as master
address, and fingerprint.",
  "projectId": null,
  "versions": null,
  "type": "CONSTANT",
  "properties": {
    "masterAddress": {
      "type": "string",

```

```

        "description": "Salt Master address associated with SaltStack Config Server.",
        "const": "cava-6-242-179.eng.vmware.com"
    },
    "masterFingerPrint": {
        "type": "string",
        "description": "Salt Master finger print",
        "const":
"aa:94:91:1b:03:5e:dd:39:35:86:ce:7e:2b:4d:12:a5:1c:35:f9:c9:ac:c6:55:18:2d:80:58:32:6f:c2:ac:
e1"
    }
}
}'

```

## SaltStack Config 서버에 vRealize Automation URL 추가

이 시퀀스를 통해 사용자가 SaltStack Config 통합에서 vRealize Automation으로 이동할 수 있습니다. vRealize Automation 환경에서 URL은 연결된 테넌트로 이동됩니다.

이 프로세스는 다른 API를 호출하기 전에 XSRF 토큰을 가져와야 하기 때문에 다단계 프로세스입니다.

1. /version API를 호출하고 응답 헤더에서 XSRF 토큰 헤더를 가져옵니다.

```

GET https://<SSC_server>/version Collapse source
curl --location -k -v --request GET 'https://<SaltStackConfigServer_Hostname>/account/info' \
--header 'Authorization: Basic cm9vdDpzYWx0'

```

2. /account/login API를 호출하여 후속 API의 전달자 토큰으로 사용할 JWT 토큰을 가져옵니다.

```

POST https://<SSC_server>/account/login Collapse source
curl --location --request POST 'https://<SaltStackConfigServer_Hostname>/account/login' \
--header 'X-Xsrftoken: <XSRF_Token_from_Response_Headers_of_Version_API>' \
--header 'Content-Type: application/json' \
--header 'Authorization: Basic cm9vdDpzYWx0' \
--header 'Cookie: _xsrf=<XSRF_Token_from_Response_Headers_of_Version_API>' \
--data-raw
'{"password":"<SaltStackConfigServer_Password>","username":"<SaltStackConfigServer_Hostname>","
"config_name":"internal","token_type":"jwt"}'

```

/rpc API를 호출하여 vRealize Automation 매개 변수를 SaltStack Config 서버에 저장합니다.

```

POST https://<SSC_server>/account/login Collapse source
curl --location --request POST 'https://<SaltStackConfigServer_Hostname>/account/rpc' \
--header 'X-Xsrftoken: <XSRF_Token_from_Response_Headers_of_Version_API>' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer <JWT_from_Response_of_Login_API>' \
--header 'Cookie: _xsrf=<XSRF_Token_from_Response_Headers_of_Version_API>' \
--data-raw '{"resource":"admin","method":"save_vra_params","kwarg":{"url":
"<vRA_Tenant_Specific_URL>"}}'

```

Example:

```

curl --location --request POST 'https://10.206.242.179/account/rpc' \
--header 'X-Xsrftoken: 2|4ca0b854|2dcdd4842fe32484d394740aa388b397|1604424534' \
--header 'Content-Type: application/json' \

```

```
--header 'Authorization: Basic cm9vdDpzYWx0' \
--header 'Cookie: _xsrf=2|4ca0b854|2dcdd4842fe32484d394740aa388b397|1604424534' \
--data-raw '{"resource":"admin","method":"save_vra_params","kwarg":{"url": "https://cava-n-81-037.eng.vmware.com/csp/gateway/portal"}}'
```

## vRealize Automation 메뉴에서 CSP 정의 업데이트

vRealize Automation 환경에 SaltStack Config 아이콘과 로그인 페이지를 표시하려면 vRealize Automation 호스트 정보를 업데이트해야 합니다.

서비스 토큰을 가져옵니다.

```
POST https://<vRA_hostname>/csp/gateway/am/api/auth/authorize Collapse source
curl --location --request POST 'https://<vRA_Hostname>/csp/gateway/am/api/auth/authorize' \
--header 'Authorization: Basic
cHJvdmlzaW9uaW5nLWlrbzVjb29sWWh0dHVFT286UldlRfNvYVl2UG05OHVlWm40Q3I3VXY3enZPQ3hIWVA=' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'orgId=<Org_Id>'
```

등록된 모든 정의를 가져옵니다.

```
GET https://<vRA_hostname>/csp/gateway/slc/api/definitions Collapse source
curl --location --request GET 'https://cava-n-81-037.eng.vmware.com/csp/gateway/slc/api/definitions' \
--header 'Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsImtpZCI6IjUxMDUyOTk3MDIzOTQ0MTQzMjg1fQ.eyJpc3MiOiJDTj1QcmVsdWR1IElkZW50aXR5IFNlcnZpY2UsTlU9Q01CVSxPPVZNd2FyZSxMPVNVZmlhLFNUPVNVZmlhLEM9QkciLCJpYXQiOiJlMjM0DUxNTQxNDQsImV4cCI6MTYwNTE4Mjk0NCwianRpIjoiaMTVmdG93NWItMWIwMi00MWU4LTlmOWUtNTRjZGIxMDM2ODhiIiwiaY29udGV4dCI6IjNhZjI0NGQ5LTU0ZWVtNDI3MS1hNDc0LTE3ZW5kZDQ2NTcxOSIsImF6cCI6InByb3Zpc2lvdmluZy1udmVTSTFQVXlmU0luSldhIiwic3ViIjoicHJvdmlzaW9uaW5nLW52ZVNjMVBVeWZTSW5KV2EiLCJwcm5fdHlwZSI6Ii1NFU1ZJQ0U1LCJwZXJ0cyI6IjEwMT0sImNvbRleHRfcmFtZSI6ImRlZmF1bHQifQ.fgnNx1VvM71GgPeVs7E4y8-OSQ53txR1EJt12_nmWbWbCXaZPu8EYlyYmIoGQbN1RzT-nlgm7V74HAPyp4BwBanCR6aim9KnZHX7LZdVleyQ1ehmMd2lQXPLxCQj7nzS10v3uaR_OdQWmsasFEMi_uU5qNn0QBxFGUPXzGB8OYxvEzFlTJKlPgVNNHdm2OVmn4a5i5g90bx_6IzzHkgyOd2OuNQPF7H3mc9PPoQ_jtpYVYSumPDeoRZ7w1_ogdpbTKrr_AddTCqbMztno81a-9bQeEEY8e8BYM56cGGGL-4wM3AX4FdcI4dWW0h0069MGLUts7_6GFACAOQQuDx8g' \
--data-raw ''
```

## vRealize Automation Cloud Assembly에서 Active Directory 통합을 생성하는 방법

vRealize Automation Cloud Assembly는 가상 시스템을 프로비저닝하기 전에 Active Directory 서버 내에 지정된 OU(조직 구성 단위)에서 바로 사용할 수 있는 컴퓨터 계정을 생성할 수 있도록 Active Directory 서버와의 통합을 지원합니다. Active Directory는 Active Directory 서버에 대한 LDAP 연결을 지원합니다.

프로젝트와 연결된 Active Directory 정책은 해당 프로젝트의 범위 내에서 프로비저닝된 모든 가상 시스템에 적용됩니다. 사용자는 일치하는 기능 태그가 있는 클라우드 영역에 프로비저닝된 가상 시스템에 정책을 선택적으로 적용하는 태그를 하나 이상 지정할 수 있습니다.

온-프레미스 배포의 경우, Active Directory 통합을 사용하면 필요한 확장성 클라우드 프록시를 포함하여 통합의 상태와 이에 의존하는 기본 ABX 통합을 표시하는 상태 점검 기능을 설정할 수 있습니다. Active Directory 정책을 적용하기 전에 vRealize Automation Cloud Assembly는 기본 통합의 상태를 확인합니다. 통합이 정상이면 vRealize Automation Cloud Assembly는 지정된 Active Directory에 배포된 컴퓨터 개체를 생성합니다. 통합이 비정상이면 배포 작업은 프로비저닝 중에 Active Directory 단계를 건너뛰니다.

### 사전 요구 사항

- Active Directory 통합에는 Active Directory 서버에 대한 LDAP 연결이 필요합니다.
- vCenter 온-프레미스와 Active Directory 통합을 구성하는 경우 확장성 클라우드 프록시를 사용하여 ABX 통합을 구성해야 합니다. **확장성 > 작업 > 통합**을 선택하고 **확장성 작업 온-프레미스**를 선택합니다.
- 클라우드에서 Active Directory와의 통합을 구성하는 경우에는 Microsoft Azure 또는 Amazon Web Services 계정이 있어야 합니다.
- Active Directory 통합에 사용할 수 있는 적절한 클라우드 영역과 이미지 및 버전 매핑으로 구성된 프로젝트가 있어야 합니다.
- Active Directory 통합을 프로젝트와 연결하기 전에 Active Directory에서 원하는 OU를 사전 생성해야 합니다.

### 절차

- 1 **인프라 > 연결 > 통합**을 선택한 다음, **새 통합**을 선택합니다.
- 2 **Active Directory**를 클릭합니다.
- 3 **요약** 탭에서 적절한 LDAP 호스트 및 환경 이름을 입력합니다.
- 4 LDAP 서버에 대한 이름과 암호를 입력합니다.
- 5 Active Directory에서 원하는 사용자 및 그룹에 대해 적절한 기본 DN을 입력합니다.

---

**참고** Active Directory 통합당 DN은 하나만 지정할 수 있습니다.

---

- 6 **유효성 검증**을 클릭하여 통합이 작동하는지 확인합니다.
- 7 이 통합의 이름과 설명을 입력합니다.
- 8 **저장**을 클릭합니다.
- 9 **프로젝트** 탭을 클릭하여 Active Directory 통합에 프로젝트를 추가합니다.

**프로젝트 추가** 대화 상자에서 프로젝트 이름과 상대 DN을 선택해야 합니다. 이 DN은 [요약] 탭에 지정된 기본 DN 내에 있는 DN입니다.

- 10 **저장**을 클릭합니다.

## 결과

이제 **Active Directory**와 통합된 프로젝트를 클라우드 템플릿에 연결할 수 있습니다. 이 클라우드 템플릿을 사용하여 시스템을 프로비저닝하면 지정된 **Active Directory**와 조직 구성 단위에서 시스템이 미리 준비됩니다.

다음과 같이 온-프레미스 **Active Directory** 통합에 대한 태그 기반 상태 점검을 구현할 수도 있습니다.

- 1 앞의 단계에 설명된 대로 **Active Directory** 통합을 생성합니다.
- 2 **프로젝트** 탭을 클릭하여 **Active Directory** 통합에 프로젝트를 추가합니다.
- 3 [프로젝트 추가] 대화 상자에서 프로젝트 이름과 상대 DN을 선택합니다. 상대 DN은 지정된 기본 DN 내에 있어야 합니다.

이 대화 상자에는 클라우드 템플릿에서 **Active Directory** 구성을 제어할 수 있는 두 개의 스위치가 있습니다. 이 두 스위치는 기본적으로 모두 꺼져 있습니다.

- 재정의 - 이 스위치를 사용하면 클라우드 템플릿의 **Active Directory** 속성, 특히 상대 DN을 재정의할 수 있습니다. 스위치를 켜면 클라우드 템플릿의 `relativeDN` 속성에 지정된 OU를 변경할 수 있습니다. 프로비저닝되면 시스템이 클라우드 템플릿의 `relativeDN` 속성에 지정된 OU에 추가됩니다. 다음 예는 이 속성이 표시되는 클라우드 템플릿 계층을 보여줍니다.

```
activeDirectory:
  relativeDN: OU=ad_integration_machine_override
```

- 무시 - 이 스위치를 사용하면 프로젝트에 대한 **Active Directory** 구성을 무시할 수 있습니다. 스위치를 켜면 연결된 가상 시스템에 대해 `ignoreActiveDirectory`라는 속성이 클라우드 템플릿에 추가됩니다. 이 속성을 `true`로 설정하면 배포 시 시스템이 **Active Directory**에 추가되지 않습니다.
- 4 적절한 태그를 추가합니다. 이러한 태그는 **Active Directory** 정책이 적용될 수 있는 클라우드 영역에 적용됩니다.
  - 5 [저장]을 클릭합니다.

**Active Directory** 통합의 상태는 vRealize Automation Cloud Assembly의 **인프라 > 연결 > 통합** 페이지에 있는 각 통합에 대해 표시됩니다.

클라우드 템플릿을 사용하여 프로젝트를 **Active Directory** 통합과 연결할 수 있습니다. 이 템플릿을 사용하여 시스템을 프로비저닝하면 지정된 **Active Directory**와 OU에서 시스템이 미리 준비됩니다.

## VMware SDDC Manager 통합 구성

VMware SDDC Manager 통합을 vRealize Automation에 추가하여 vRealize Automation 내에서 VCF(VMware Cloud Foundation) 클라우드 계정의 일부로 워크로드 도메인을 쉽게 사용할 수 있습니다.

### 사전 요구 사항

- vRealize Automation은 VMware SDDC Manager 4.1 이상과의 통합만 지원합니다.

## 절차

- 1 **인프라 > 연결 > 통합**을 선택하고 **통합 추가**를 클릭합니다.
- 2 SDDC Manager를 선택합니다.  
SDDC Manager 통합 구성 페이지가 나타납니다.
- 3 [요약] 섹션에서 통합의 **이름**과 **설명**을 입력합니다.
- 4 [SDDC Manager 자격 증명] 섹션에서 SDDC Manager 서버 시스템에 대한 **SDDC Manager IP 주소/FQDN**을 입력합니다.
- 5 SDDC Manager에 처음 연결하는 데 사용할 관리자 계정의 사용자 이름과 암호를 입력합니다. 가장 좋은 방법은 관리자 계정을 사용하여 연결하지 않는 것입니다. SDDC Manager에서 관리자 권한이 있는 다른 계정을 사용하여 서비스 역할을 생성합니다.  
이러한 자격 증명은 처음에 SDDC Manager에 대한 연결을 설정하는 데 사용되며, 이를 통해 VCF 클라우드 계정에서 연결할 때 사용할 서비스 자격 증명이 생성됩니다.
- 6 **검증**을 클릭하여 SDDC Manager에 대한 연결을 확인합니다.
- 7 **추가**를 클릭합니다.

## 결과

통합이 생성된 후 완료된 통합 페이지에 나타나는 [워크로드 도메인] 탭에서 SDDC와 연결된 워크로드를 볼 수 있습니다. 또한 통합과 연결된 워크로드를 보고 선택한 다음, **클라우드 계정 추가** 버튼을 클릭하여 선택한 워크로드를 사용할 VCF 클라우드 계정을 생성하기 위한 페이지를 열 수 있습니다.

## 다음에 수행할 작업

VCF 클라우드 계정을 구성하면 **클라우드 설정** 버튼이 페이지 맨 위에 나타납니다. 이 버튼을 클릭하여 VCF 클라우드 설정 마법사를 시작합니다.

## vRealize Operations Manager와 통합

vRealize Automation은 vRealize Operations Manager를 사용하여 고급 워크로드 배치를 수행하고, 배포 상태 및 가상 시스템 메트릭을 제공하고, 가격 책정을 표시할 수 있습니다.

## 통합 수 및 유형

두 제품 간의 통합은 온-프레미스와 온-프레미스 사이여야 하며 온-프레미스와 클라우드를 혼합하여 사용할 수 없습니다.

vRealize Automation 인스턴스 하나를 여러 vRealize Operations Manager 인스턴스와 통합할 수 있지만 vRealize Operations Manager 인스턴스는 vRealize Automation 인스턴스 하나에만 연결할 수 있습니다.

집계된 vRealize Operations Manager 클러스터는 vRealize Automation에 연결할 수 없습니다.

## 통합을 위한 기본 요구 사항

vRealize Operations Manager와 통합하려면 **인프라 > 연결 > 통합**으로 이동합니다. 통합을 추가하려면 vRealize Operations Manager URL 및 다음 섹션에 설명된 로그인 계정에 대한 자격 증명이 필요합니다. 또한 vRealize Automation 및 vRealize Operations Manager에서는 동일한 vSphere 끝점을 관리해야 합니다.

## 통합을 위한 로그인 계정

vRealize Operations Manager에서 통합에 사용할 로컬 또는 로컬이 아닌 vRealize Operations Manager 로그인 계정이 필요합니다. 이 계정에는 vSphere 끝점의 vCenter 어댑터 인스턴스에 대한 읽기 전용 권한이 필요합니다. 로컬이 아닌 계정은 vRealize Operations Manager에서 가져와서 읽기 전용 역할을 할당해야 할 수 있습니다. 통합의 경우 로컬이 아닌 계정 로그인을 위한 사용자 이름 형식은 `username@domain@authenticated-source`(예: `jdoe@company.com@workspaceone`)입니다. 인증된 소스는 vRealize Operations Manager 서버 초기 설정 중에 정의됩니다.

자세한 내용은 다음 섹션을 참조하십시오. 가격 책정 정보는 [vRealize Automation](#)에서 [가격 책정 카드](#)를 사용하는 방법의 내용을 참조하십시오.

## vRealize Operations Manager를 사용한 고급 워크로드 배치

vRealize Automation과 vRealize Operations Manager가 함께 작동하여 배포 워크로드를 최적으로 배치할 수 있습니다.

vSphere 기반 클라우드 영역 수준에서 워크로드 배치를 사용하도록 설정합니다. 클라우드 영역의 DRS(Distributed Resource Scheduler) 지원 클러스터만 vRealize Operations Manager를 사용한 고급 배치에 적합합니다.

- **vRealize Automation 배치** - vRealize Automation 배치 엔진은 애플리케이션 의도를 기반으로 합니다. 태그 기반 제약 조건, 프로젝트 멤버 자격 및 연결된 클라우드 영역을 비롯해 네트워크, 스토리지 및 계산 관련 선호도 필터를 고려합니다. 리소스 배치는 이러한 모든 요소와 동일한 배포에 다른 관련 대상 리소스가 있는지 여부에 따라 달라집니다.
- **vRealize Operations Manager 배치** - vRealize Operations Manager는 최적의 배치를 위해 작동 의도를 고려합니다. 작동 의도는 지난 워크로드 및 향후 가상 예측을 고려할 수 있습니다.  
고급 워크로드 배치를 사용하는 경우에는 vRealize Operations Manager 비즈니스 의도 옵션을 사용하는 대신 비즈니스 의도 의사 결정을 구현하기 위해 vRealize Automation 태그 지정을 적용해야 합니다.

vRealize Operations Manager와 통합하는 경우 vRealize Automation은 해당 애플리케이션 의도 모델과 관련 제약 조건을 계속 준수하여 대상 배치를 필터링합니다. 그런 다음, 이러한 결과 내에서 vRealize Operations Manager 권장 사항을 사용하여 배치를 더욱 구체화합니다.

### 권장 사항이 없는 경우

고급 워크로드 배치를 사용하도록 설정하고 vRealize Operations Manager 분석에서 권장 사항을 반환하지 않으면 기본 애플리케이션 의도 배치로 폴백하도록 vRealize Automation을 구성할 수 있습니다.

## 워크로드 배치에 대한 제한 사항

vRealize Operations Manager를 사용하여 워크로드를 배치할 때 특정 제한 사항이 적용됩니다.

- vRealize Operations Manager는 vCenter Server에서 리소스 풀에 대한 워크로드 배치를 지원하지 않습니다.
- vRealize Operations Manager가 다운된 경우 워크로드 배치 시간이 초과되어 vRealize Operations Manager 호출이 실패할 수 있습니다.
- 배치는 여러 클라우드 영역을 교차하지 않습니다. vRealize Automation은 단일 클라우드 영역 내의 배치 권장 사항을 위해 하나의 클라우드 영역을 vRealize Operations Manager로 전송합니다.

## 워크로드 배치를 사용하도록 설정하는 방법

워크로드 배치를 사용하도록 설정하려면 vSphere, vRealize Operations Manager 및 vRealize Automation에서 수행해야 하는 단계가 있습니다.

- 1 vRealize Automation Cloud Assembly에서 vCenter Server 클라우드 계정에 연결합니다.  
옵션은 **인프라 > 연결 > 클라우드 계정**에 있습니다.
- 2 vCenter Server에서 DRS 지원 클러스터가 있고 완전히 자동화됨으로 설정되었는지 확인합니다.
- 3 vRealize Operations Manager에서 동일한 vCenter Server가 관리되고 있는지 확인합니다.  
vRealize Operations Manager 8 이상이 필요합니다.
- 4 vRealize Automation Cloud Assembly에서 vRealize Operations Manager 통합을 추가합니다.  
옵션은 **인프라 > 연결 > 통합**에 있습니다.  
  
통합을 추가하려면 아래의 vRealize Operations Manager 기본 노드 URL과 로그인 사용자 이름 및 암호가 필요합니다.  
  
`https:// "operations-manager-IP-주소-또는-FQDN" /suite-api`  
  
값을 입력한 후 [검증]을 클릭합니다.
- 5 [동기화]를 클릭하여 vCenter Server에 대한 통합을 동기화합니다.  
  
또한 vRealize Automation Cloud Assembly 및 vRealize Operations Manager가 새 vCenter Server 관리를 시작하는 모든 시간을 동기화합니다.
- 6 vRealize Automation Cloud Assembly에서 vCenter Server 계정에 대한 클라우드 영역을 생성합니다.  
  
옵션은 **인프라 > 구성 > 클라우드 영역**에 있습니다.
- 7 클라우드 영역 [요약] 탭에서 [배치 정책]을 [고급]으로 설정합니다.
- 8 [배치 정책]에서 vRealize Operations Manager가 권장 사항을 반환하지 않을 경우 vRealize Automation을 기본 배치로 폴백할지 여부를 선택합니다.

## 워크로드 배치 문제 해결

귀하가 바라는 방식의 워크로드 배치를 vRealize Operations Manager가 권장하지 않는 경우 vRealize Automation Cloud Assembly 또는 vRealize Automation Service Broker에서 배포 요청 세부 정보를 검토하십시오.

- 1 **인프라 > 작업 > 요청**으로 이동하여 요청을 클릭합니다.
- 2 [요청 세부 정보]에서 할당 단계를 살펴봅니다.  
식별되거나 식별되지 못한 대상을 찾습니다.
- 3 [요청 세부 정보]의 오른쪽 위에 있는 [개발 모드]를 사용하도록 설정합니다.
- 4 요청 경로를 따라 필터 블록을 찾습니다.
- 5 필터 블록을 클릭하고 다음 섹션을 검토합니다.

```
filterName: ComputePlacementPolicyAffinityHostFilter
  V computeLinksBefore
  V computeLinksAfter
  V filteredOutHostsReasons
```

입력	설명
computeLinksBefore	vRealize Automation 알고리즘에 기반한 잠재적 배치 호스트 목록.
computeLinksAfter	선택한 배치 호스트.
filteredOutHostsReasons	호스트가 선택되거나 거부된 이유를 설명하는 메시지. vRealize Operations Manager가 호스트를 선택하면 다음 메시지가 나타납니다. advance policy filter: Filtered hosts based on recommendation from vROPS.

## vRealize Operations Manager를 사용한 지속적 최적화

vRealize Operations Manager에서 vRealize Automation 어댑터를 추가하면 vRealize Operations Manager가 vRealize Automation 기반 워크로드를 위한 새로운 CDC(사용자 지정 데이터 센터)를 자동으로 생성합니다.

지속적 최적화는 워크로드 재조정 및 재배포 기능을 제공하며 vRealize Automation에서 초기 워크로드 배치 외에도 vRealize Operations Manager를 사용할 수 있습니다. 가상화 리소스의 로드가 증가하거나 감소할 경우 vRealize Automation으로 프로비저닝한 워크로드를 필요에 따라 이동할 수 있습니다.

- 지속적 최적화는 vRealize Operations Manager에서 새로운 CDC를 자동으로 생성합니다.  
각 vRealize Automation vSphere 클라우드 영역에 대해 하나의 새로운 CDC가 생성됩니다.
- 새로 생성된 CDC에는 클라우드 영역에 연결된 모든 vRealize Automation 관리 클러스터가 포함됩니다.

**참고** vRealize Automation과 비 vRealize Automation 클러스터가 혼합된 CDC를 수동으로 생성하지 마십시오.

- vRealize Operations Manager를 사용하여 새로 생성된 vRealize Automation 기반 CDC에 대해 지속적인 최적화를 실행합니다.
- 워크로드는 동일한 클라우드 영역 또는 CDC 내에서만 재조정 또는 재배포될 수 있습니다.
- 최적화에서는 새로운 vRealize Automation 또는 vRealize Operations Manager 배치 위반이 생성되지 않습니다.
  - 기존 배치 위반이 있는 경우 최적화에서 vRealize Operations Manager 작동 의도 문제를 수정할 수 있습니다.
  - 기존 배치 위반이 있는 경우 vRealize Operations Manager 비즈니스 의도 문제는 최적화에서 수정할 수 없습니다.

예를 들어 vRealize Operations Manager를 사용하여 가상 시스템을 제약 조건이 지원되지 않는 클러스터로 수동으로 이동하면 vRealize Operations Manager에서 위반을 감지하지도 해결을 시도하지도 않습니다.

- 이 릴리스는 CDC 수준에서 작동 의도를 준수합니다. 모든 멤버 vRealize Automation 클러스터는 동일한 설정에 최적화됩니다.

클러스터에 서로 다른 작동 의도를 설정하려면 개별 vSphere 클라우드 영역에 연결된 별도의 vRealize Automation CDC에서 구성해야 합니다. 테스트 클러스터와 운영 클러스터가 다른 경우를 예로 들 수 있습니다.

- vRealize Automation 애플리케이션 의도 및 vRealize Automation에 정의된 제약 조건은 모든 최적화 재조정 또는 재배포 작업 중에 적용됩니다.
- vRealize Operations Manager 배치 태그는 vRealize Automation으로 프로비저닝된 워크로드에 적용할 수 없습니다.

또한 여러 시스템을 포함하는 최적화를 예약할 수 있습니다. 정기적으로 예약된 최적화는 양단간 프로세스가 아닙니다. 조건에 의해 시스템 이동이 중단되는 경우 성공적으로 재배포된 시스템은 재배포 상태로 유지되며 다음 vRealize Operations Manager 주기에서 나머지에 대한 재배포가 vRealize Operations Manager에서 일반적인 방식으로 수행됩니다. 이와 같이 최적화가 부분적으로 완료되더라도 vRealize Automation에서 부정적인 결과가 발생하지는 않습니다.

#### 지속적 최적화를 사용하도록 설정하는 방법

vRealize Operations Manager에서 vRealize Automation 어댑터를 추가하면 vRealize Operations Manager가 vRealize Automation 기반 워크로드를 위한 새로운 전용 데이터 센터를 생성합니다.

vRealize Automation Cloud Assembly 내에 통합을 추가하는 것 외에 지속적 최적화를 위해 수행해야 하는 개별 설치 단계는 없습니다. 새 데이터 센터에서 vRealize Operations Manager를 구성하고 사용하여 워크로드를 재배포할 수 있습니다. [지속적 최적화 예](#)의 내용을 참조하십시오.

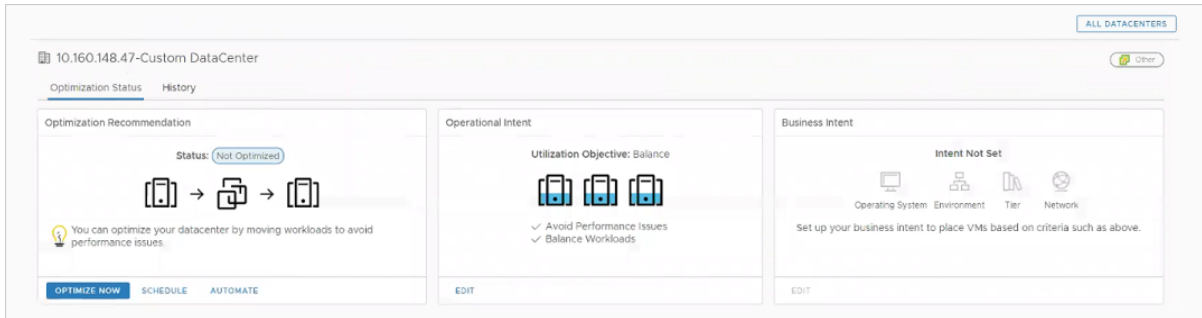
#### 지속적 최적화 예

다음 예는 vRealize Operations Manager를 사용한 vRealize Automation 지속적 최적화의 워크플로 재조정을 보여줍니다.

- 1 vRealize Operations Manager 홈 페이지에서 **워크로드 최적화**를 클릭합니다.
- 2 자동으로 생성된 vRealize Automation 데이터 센터를 선택합니다.

### 3 작동 의도에서 편집을 클릭하고 밸런스를 선택합니다.

vRealize Automation 최적화에 데이터 센터가 사용되는 경우 비즈니스 의도가 사용되지 않도록 설정되므로 선택하거나 편집할 수 없습니다.



### 4 최적화 권장 사항에서 지금 최적화를 클릭합니다.

vRealize Operations Manager에 제안된 작업의 전/후 다이어그램이 표시됩니다.

### 5 다음을 클릭합니다.

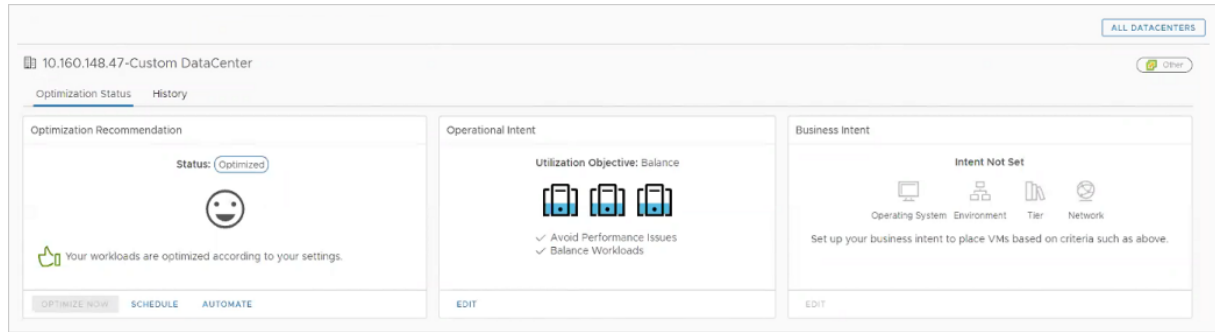
### 6 작업 시작을 클릭합니다.

### 7 vRealize Automation에서 배포를 클릭하고 이벤트 상태를 확인하여 진행 중인 작업을 모니터링합니다.

Events Request inputs			
#7 - Relocate RRD-WLP-003 <span>In Progress</span> Requested by: System User Requested for: Fritz Arbeiter Requested on: August 13, 2018 11:43 AM			
Tasks	Component	Status	Depends On
Submitted	Deployment	Successful	
Pre-approval	Deployment	Approved	
Relocate	Deployment	In Progress	
Post-approval	Deployment		
Completed	Deployment		

재조정이 완료되면 vRealize Automation이 새로 고쳐집니다. [계산 리소스] 페이지에서 시스템이 이동되었음을 알 수 있습니다.

vRealize Operations Manager에서 다음 데이터 수집 시 최적화 완료를 표시하도록 디스플레이가 새로 고쳐집니다.



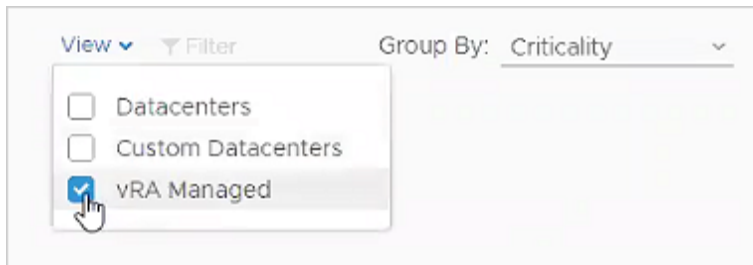
vRealize Operations Manager에서 **관리 > 기록 > 최근 작업**을 클릭하여 작업을 검토할 수 있습니다.

#### vRealize Automation 관리 데이터 센터 찾기

vRealize Operations Manager를 사용하여 vRealize Automation 관리 데이터 센터만 표시할 수 있습니다.

#### 절차

- 1 vRealize Operations Manager 홈 페이지에서 **워크로드 최적화**를 클릭합니다.
- 2 오른쪽 상단 근처에서 **보기** 드롭다운을 클릭합니다.
- 3 vRealize Automation 관리 데이터 센터만 선택합니다.



#### vRealize Operations Manager에 기반한 배포 모니터링

vRealize Automation에서 배포에 대한 vRealize Operations Manager 데이터를 확인할 수 있습니다.

필터링된 메트릭 집합을(vRealize Automation에서 바로) 검토하면 vRealize Operations Manager에 액세스하거나 검색하는 작업이 줄어듭니다. vRealize Operations Manager 컨텍스트에서 시작할 수는 없지만 vRealize Operations Manager에 로그인하고 사용하여 필요에 따라 추가 데이터를 확인할 수 있습니다.

#### vRealize Operations Manager 데이터 사용

vRealize Automation에서 vRealize Operations Manager 데이터를 표시하려면 vRealize Operations Manager 통합을 추가합니다.

#### 절차

- 1 vRealize Operations Manager에서 **관리 > 솔루션**으로 이동합니다.

- 2 구성된 어댑터 인스턴스에서 vRealize Automation가 프로비저닝하는 vSphere 클라우드 영역에 대한 vCenter 어댑터가 있고, 데이터를 수신 중인지 확인합니다.
- 3 vRealize Automation Cloud Assembly에서 **인프라 > 연결 > 통합**으로 이동합니다.
- 4 vRealize Operations Manager 기본 노드 URL과 vRealize Operations Manager 로그인 사용자 이름 및 암호를 입력합니다.

`https:// "operations-manager-IP-주소-또는-FQDN" /suite-api`

- 5 **배포**를 클릭하고, 배포를 선택한 다음 [모니터링] 탭이 표시되는지 확인합니다.

### vRealize Operations Manager에서 제공하는 상태 및 경고

모니터링을 사용하도록 설정하면 vRealize Automation에서 배포에 대한 vRealize Operations Manager 상태 및 연결된 경고를 검색합니다.

모니터링에 액세스하려면 배포를 클릭하고 **모니터링** 탭을 선택합니다. 탭이 없는 경우 [vRealize Operations Manager 데이터 사용](#) 항목을 참조하십시오.

경고를 보려면 왼쪽 패널의 구성 요소 트리 맨 위에서 배포 이름을 강조 표시합니다.

- 경고의 심각도 및 텍스트를 검토할 수 있습니다.
- 문제 영역을 집중적으로 보려면 열 데이터를 필터링하고 정렬합니다.
- 상태 배지 및 상태 경고만 나타납니다. 효율성 또는 위험 같은 다른 경고 유형은 지원되지 않습니다.

### vRealize Operations Manager가 제공하는 메트릭

모니터링을 사용하도록 설정하면 vRealize Automation이 배포에 대한 vRealize Operations Manager 메트릭을 검색합니다.

모니터링에 액세스하려면 배포를 클릭하고 **모니터링** 탭을 선택합니다. 탭이 없는 경우 [vRealize Operations Manager 데이터 사용](#) 항목을 참조하십시오.

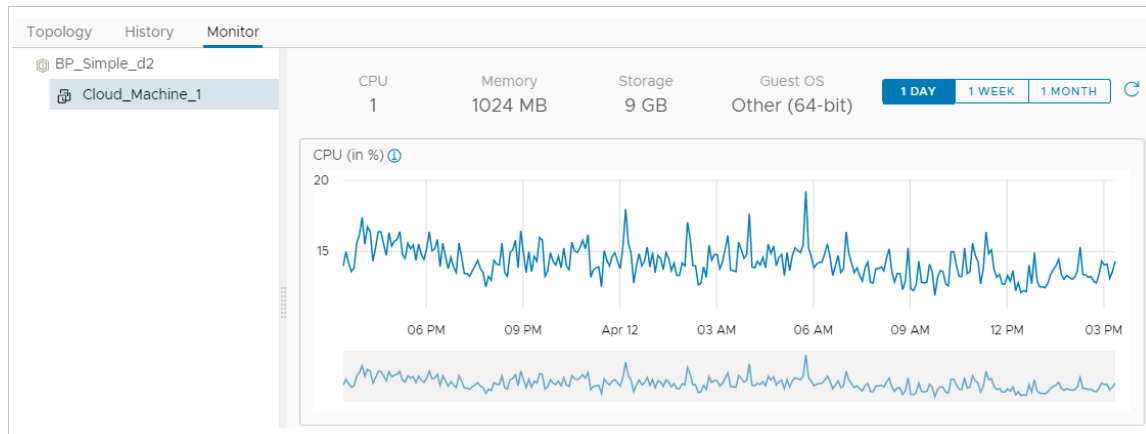
메트릭을 보려면 왼쪽에서 구성 요소 트리를 확장하고 가상 시스템을 강조 표시합니다.

- 메트릭은 캐시되지 않습니다. vRealize Operations Manager에서 직접 가져오며 로드하는 데 몇 분이 걸릴 수 있습니다.
- 가상 시스템 메트릭만 표시됩니다. vCloud Director, 소프트웨어 또는 XaaS와 같은 다른 구성 요소의 메트릭은 지원되지 않습니다.
- vSphere 가상 시스템 메트릭만 표시됩니다. AWS 또는 Azure 같은 다른 클라우드 제공자는 지원되지 않습니다.

메트릭은 다음 측정치에 대한 높고 낮음을 보여주는 타임라인 그래프로 표시됩니다.

- CPU
- 메모리
- 스토리지 IOPS
- 네트워크 MBPS

특정 메트릭 이름을 표시하려면 타임라인의 왼쪽 위에 있는 파란색 정보 아이콘을 클릭합니다.

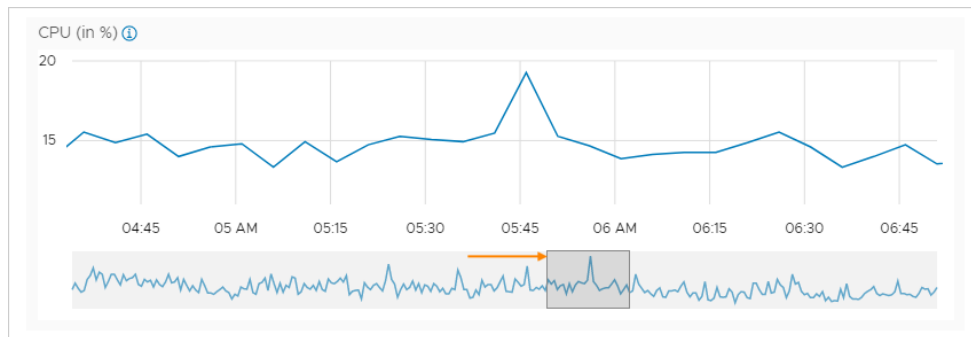


### vRealize Operations Manager가 제공하는 데이터에 대한 작업

vRealize Operations Manager가 제공한 메트릭이 문제를 노출하는 경우 다음을 수행할 수 있습니다. vRealize Automation에서 직접 문제 영역을 식별합니다.

vRealize Operations Manager가 제공하는 메트릭을 보려면 배포를 클릭하고 **모니터링** 탭을 선택합니다. 탭이 없는 경우 **vRealize Operations Manager 데이터 사용** 항목을 참조하십시오.

지난날, 지난주 또는 지난달의 메트릭이 제공됩니다. 문제 영역을 확대하려면 메트릭 타임라인에서 아래쪽의 음영 처리된 부분에 있는 작은 영역을 선택합니다.



### vRealize Automation에서 vRealize Operations Manager 메트릭을 사용하여 리소스 관리 및 배포 최적화

통합된 vRealize Automation 및 vRealize Operations Manager 환경에서는 vRealize Operations Manager에서 모니터링하는 vRealize Automation 개체에 대한 인사이트 및 경고에 액세스할 수 있습니다.

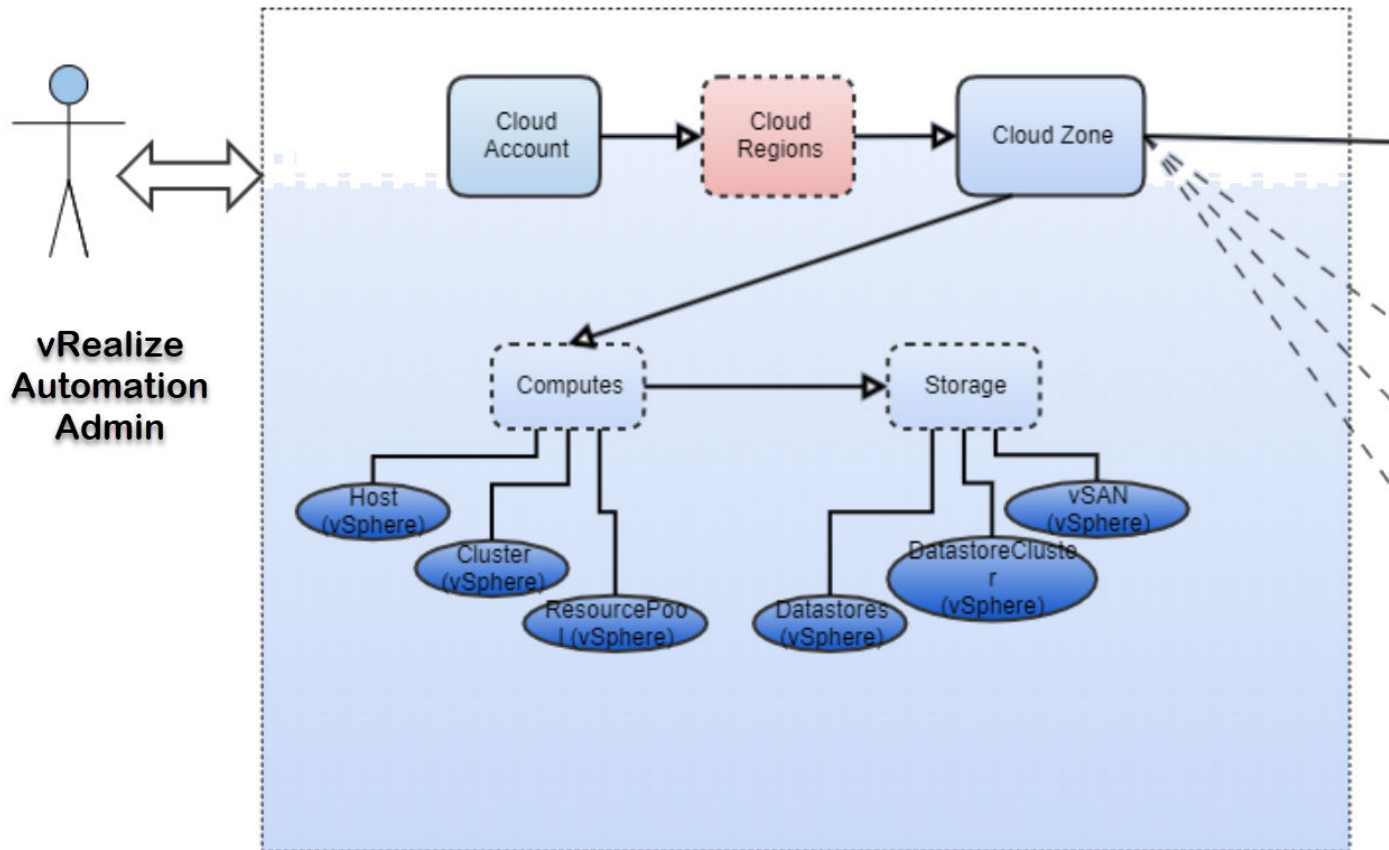
**인사이트** 대시보드 및 **경고** 탭 페이지는 vRealize Operations Manager를 열지 않고도 vRealize Automation에서 관리 결정을 내리는 데 필요한 실시간 용량 및 관련 인식 정보를 제공합니다. 정보는 연결된 vRealize Operations Manager 애플리케이션에서 제공됩니다.

## 인사이트 대시보드 및 리소스 경고 사용

**인사이트** 대시보드는 클라우드 영역 내의 모든 계산에서 용량 소비에 대한 정보(프로젝트별로 그룹화됨)를 전달합니다. 또한 최적화가 필요한 프로젝트 배포도 표시할 수 있습니다.

통합 **경고** 탭 페이지에는 클라우드 영역, 프로젝트, 배포 및 가상 시스템과 같은 개체에 대한 잠재적 용량 및 성능 문제나 이상 징후가 표시됩니다. 또한 배포를 최적화할 수 있는 프로젝트 소유자에 대한 알림 정보도 포함되어 있습니다. 언급된 각 배포 링크는 특정 지점이 제공되는 배포의 **최적화** 탭을 엽니다.

다음 흐름 다이어그램은 vRealize Automation 리소스와 배포의 관계 및 연결된 vRealize Operations Manager 애플리케이션이 vRealize Automation에서 제공하는 데이터를 보여줍니다.

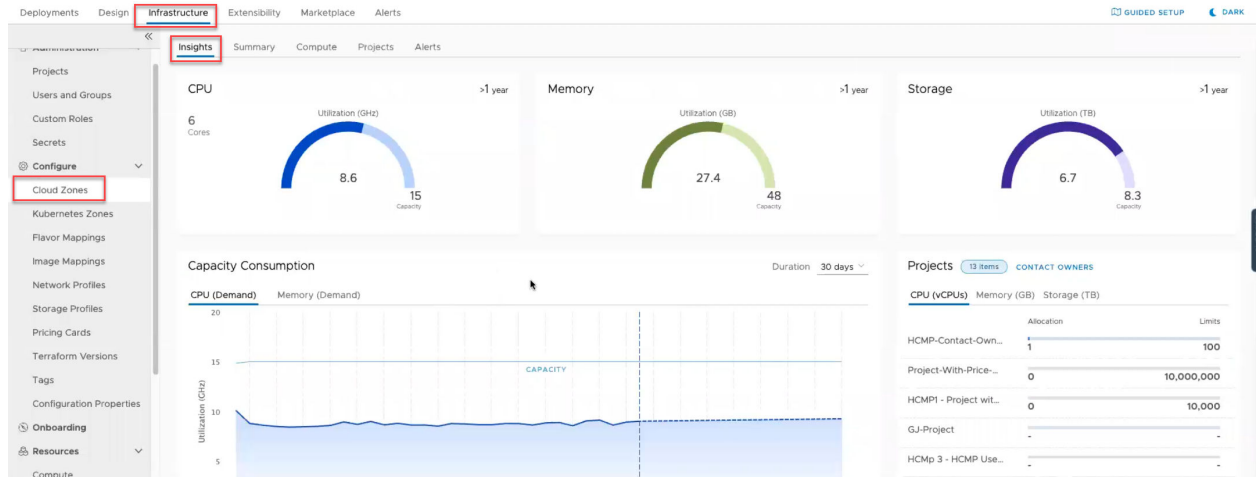


## 인사이트 대시보드 사용

각 클라우드 영역 페이지에서 사용할 수 있는 **인사이트** 대시보드는 다음과 같은 vRealize Operations Manager 메트릭을 제공합니다.

- CPU, 메모리 및 스토리지 활용률(용량의 백분율로 표시)
- 기능 사용 요약

- CPU 및 메모리 요구량 및 사용량 기록
- 프로젝트 전반의 사용량



또한 최적화할 수 있는 배포의 프로젝트 소유자에게 경고하는 옵션도 제공합니다.

**인사이트** 대시보드는 현재 vSphere 및 VMware Cloud on AWS 클라우드 영역에서 사용할 수 있습니다. 단, 클라우드 계정이 vRealize Automation 및 vRealize Operations Manager 둘 다에 구성되어 있고 vRealize Operations Manager에서 모니터링되는 경우에 한합니다.

자세한 내용은 [vRealize Automation](#)에서 인사이트 대시보드를 사용하여 리소스 용량을 모니터링하고 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알리는 방법 항목을 참조하십시오.

## 경고 사용

**경고** 페이지에서는 다음과 같은 필터링 범주를 제공합니다. 빠른 필터를 사용하여 각 필터를 더 구체화할 수 있습니다. 예를 들어 리소스 필터는 클라우드 영역, 가상 시스템, 배포 및 프로젝트 리소스의 빠른 필터 유형을 사용하여 더 구체화할 수 있습니다. 필터링 범주는 연결된 vRealize Operations Manager 애플리케이션에서 제공됩니다.

- 심각도
- 상태
- 영향
- 유형
- 하위 유형
- 리소스

필터 및 빠른 필터 조합을 사용하여 표시할 수 있는 경고를 제어합니다.

Deployments Design Infrastructure Extensibility Marketplace **Alerts**

Resource Type  Quick filters

Status: Active X Resource Type: Virtual Machine, Project, Cloud Zone X Impact: Health X

Today

- ☒ Cloud Zone
- ☒ Virtual Machine
- ☐ Deployment
- ☒ Project

Yesterday

**Virtual machine is powered off for more than 5 days** 4:40 PM

Virtual Machine » Cloud\_vSphere\_Machine\_1-mcm222450-155465769232

Virtual machine is powered off for more than 5 days

**Virtual machine is powered off for more than 5 days** 4:40 PM

Virtual Machine » Cloud\_vSphere\_Machine\_2-mcm222451-155465774235

Virtual machine is powered off for more than 5 days

**AlertDefinition\_20571bc0-a68c-477c-bb93-118da83...** 1:26 PM

Cloud Zone » sqa-vc65 / Datacenter

**AlertDefinition\_6b5667f5-eb02-4b2e-bcf9-40cbb2...** 1:26 PM

Cloud Zone » sqa-vc67.sqa.local / Datacenter

**AlertDefinition\_bf5e68e4-28f1-4992-af8d-94ea214ff...** 1:26 PM

Cloud Zone » sqa-vc67.sqa.local / Datacenter

1 / 7

**Virtual machine is powered off for more than 5 days**

Created: Dec 13, 2020, 4:40:46 PM | Updated: Dec 14, 2020, 7:04:47 PM

[Virtual Machine](#) » [Cloud\\_vSphere\\_Machine\\_1-mcm222450-155465769232](#)

Virtual machine is powered off for more than 5 days

Severity: Warning Status: Active Impact: Health Type: Infrastructure

Suggestions 2 [REVIEW DEPLOYMENT](#)

- Delete powered off machines
- Manually power on the virtual machine.

Notes

Leave a note...

ADD NOTE

일부 **경고**는 최적화할 수 있는 배포에 대한 정보와 링크를 제공합니다.

The screenshot displays the vRealize Automation Alerts interface. At the top, navigation tabs include Deployments, Design, Infrastructure, Extensibility, Marketplace, and Alerts (which is highlighted). Below the tabs, there are filters for Severity (dropdown), Quick filters, Status (Active), and Severity (Critical). The main content area is divided into 'Today' and 'Yesterday' sections, each listing alerts. A red box highlights an alert titled 'The project has some deployments t...' with a timestamp of 6:17 PM. Red arrows point from this alert to a detailed view on the right. This detailed view shows the alert's creation and update times, project name ('vc65 project'), and a description: 'The project has some deployments that contain optimizable resources.' It also includes tags for Severity (Critical), Status (Active), Impact (Efficiency), Type (Application), and Subtype (Performance). Below this, a 'Suggestions' section recommends reviewing the project. A red box highlights a 'Deployments to review' table with one entry: 'contact-owner-test-dep-2'. At the bottom, there is a 'Notes' section with the text 'Investigating' and an 'ADD NOTE' button.

경고는 현재 vSphere 및 VMware Cloud on AWS 리소스 개체에만 사용할 수 있습니다.

통합 경고를 구성하고 사용하는 방법에 대한 자세한 내용은 vRealize Automation에서 경고를 사용하여 리소스 용량, 성능 및 가용성을 관리하는 방법 및 vRealize Automation에서 인사이트 대시보드로부터 수집된 데이터를 기반으로 배포를 최적화하는 방법 항목을 참조하십시오.

## vRealize Automation Cloud Assembly의 온보딩 계획이란?

워크로드 온보딩 계획을 사용하여 대상 지역이나 데이터 센터의 클라우드 계정 유형에서 데이터가 수집되지만 아직 vRealize Automation Cloud Assembly 프로젝트를 통해 관리되지 않는 시스템을 식별합니다.

vRealize Automation Cloud Assembly 외부에 배포된 시스템이 포함된 클라우드 계정을 추가하는 경우 해당 시스템을 온보딩할 때까지 Cloud Assembly에서 관리되지 않습니다. 온보딩 계획을 사용하여 관리되지 않는 시스템을 vRealize Automation Cloud Assembly 관리 대상으로 가져옵니다. 계획을 생성하고 시스템으로 채운 다음 계획을 실행하여 시스템을 가져옵니다. 온보딩 계획을 사용하면 클라우드 템플릿을 생성하고 하나 이상의 배포를 생성할 수도 있습니다.

하나 이상의 관리되지 않는 시스템을 단일 계획으로 온보딩할 수 있습니다. 수동으로 또는 필터링 규칙을 사용하여 시스템을 선택할 수 있습니다. 필터링 규칙은 시스템 이름, 상태, IP 주소 및 태그와 같은 조건을 기반으로 온보딩을 위한 시스템을 선택합니다.

- 시간당 단일 온보딩 계획 내에서 최대 3,500개의 관리되지 않는 시스템을 온보딩할 수 있습니다.
- 시간당 여러 온보딩 계획 내에서 최대 17,000개의 관리되지 않는 시스템을 동시에 온보딩할 수 있습니다.

워크로드 온보딩에 사용할 수 있는 시스템은 특정 클라우드 계정 유형 및 지역과 관련된 **리소스 > 시스템** 페이지에서 [원본] 열에 Discovered 레이블로 표시됩니다. 데이터 수집된 시스템만 나열됩니다. 시스템을 온보딩한 후에는 [원본] 열에 Deployed로 표시됩니다.

워크로드 온보딩 계획을 실행하는 사용자는 시스템 소유자로 자동 할당됩니다.

또한 온보딩은 연결된 디스크와 사용자 지정 속성 온보딩을 지원하고 배포 소유자 변경을 지원합니다.

- 사용자 지정 속성 - 계획 및 개별 시스템 수준에서 사용자 지정 속성을 설정할 수 있습니다. 시스템 수준에서 설정된 사용자 지정 속성은 계획 수준에서 동일한 속성을 재정의합니다.
- 연결된 디스크 - 시스템에 부팅 불가능한 디스크가 있는 경우 해당 디스크는 자동으로 상위 시스템에 온보딩됩니다. 부팅 불가능한 디스크를 보려면 계획에서 시스템 이름을 클릭한 후 **스토리지** 탭으로 이동합니다.
- 배포 소유권 - 온보딩을 통해 기본 배포 소유자를 변경할 수 있습니다. 소유자를 변경하려면 **배포** 탭에서 배포를 선택하고 **소유자 편집**을 클릭한 다음 프로젝트와 연결된 원하는 사용자를 선택합니다.

#### 온보딩 예시

온보딩 기술의 예시는 예: 선택한 시스템을 vRealize Automation Cloud Assembly의 단일 배포로 온보딩 및 예: vRealize Automation Cloud Assembly에서 개별 배포로서 온보딩 규칙으로 필터링된 시스템 항목을 참조하십시오.

#### 온보딩 이벤트 구독

계획을 실행하면 Deployment Onboarded 이벤트가 생성됩니다. 확장성 탭 옵션을 사용하면 이러한 배포 이벤트를 구독하고, 이벤트에 대해 작업을 수행할 수 있습니다.

## 예: 선택한 시스템을 vRealize Automation Cloud Assembly의 단일 배포로 온보딩

이 예에서는 관리되지 않는 두 대의 시스템을 단일 vRealize Automation Cloud Assembly 배포로 온보딩하고 계획의 모든 시스템에 대한 단일 클라우드 템플릿을 생성합니다.

클라우드 계정을 생성할 경우, 이 클라우드 계정에 연결된 모든 시스템의 데이터가 수집되어 **인프라 > 리소스 > 시스템** 페이지에 표시됩니다. 클라우드 계정에 vRealize Automation Cloud Assembly 외부에 배포된 시스템이 있는 경우에는 온보딩 계획을 사용하여 vRealize Automation Cloud Assembly가 해당 시스템을 관리하도록 할 수 있습니다.

**참고** 배포가 온보딩되기 전에만 배포 이름을 변경할 수 있습니다. 온보딩 후에는 **이름 변경** 옵션이 사용되지 않도록 설정됩니다.

### 사전 요구 사항

- 필요한 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- [vRealize Automation Cloud Assembly의 온보딩 계획이란?](#) 항목을 검토합니다.
- vRealize Automation Cloud Assembly 프로젝트를 생성하고 준비합니다.

이 절차에는 기본 Wordpress 사용 사례의 일부 단계가 포함됩니다. [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트](#) 항목을 참조하십시오.

- 프로젝트를 생성하고, 사용자를 추가하고, 프로젝트에 사용자 역할을 할당합니다. [2부: 예제 vRealize Automation Cloud Assembly 프로젝트 생성](#) 항목을 참조하십시오.
- 프로젝트에 대한 Amazon Web Services 클라우드 계정을 생성합니다. [1. 클라우드 계정 추가](#) 항목을 참조하십시오.

이 절차의 Amazon Web Services 클라우드 계정은 클라우드 계정이 vRealize Automation Cloud Assembly에 추가되기 전에 배포되고 vRealize Automation Cloud Assembly 이외의 애플리케이션이 배포한 시스템을 포함합니다.

- 온보딩할 시스템이 **시스템** 페이지에 포함되어 있는지 확인합니다. [vRealize Automation의 시스템 리소스](#) 항목을 참조하십시오.

### 절차

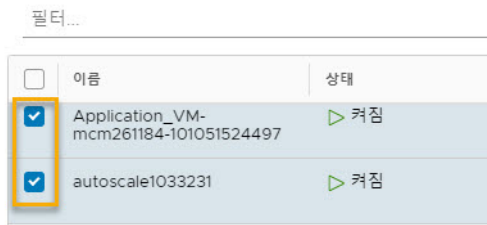
- 인프라 > 온보딩**으로 이동합니다.
- 새 온보딩 계획**을 클릭하고 샘플 값을 입력합니다.

설정	샘플 값
계획 이름	VC-sqa-deployments
설명	OurCo-AWS 클라우드 계정의 AWS 시스템 온보딩 계획 샘플
클라우드 계정	OurCo-AWS
기본 프로젝트	WordPress

- 생성**을 클릭합니다.

- 4 계획의 **배포** 탭에서 **시스템 선택**을 클릭하고 시스템을 하나 이상 선택한 후 **확인**을 클릭합니다.

### 시스템 선택



- 5 모든 시스템이 포함된 단일 배포 생성을 선택하고 **생성**을 클릭합니다.
- 6 새 배포 이름 옆의 확인란을 클릭하고 **클라우드 템플릿...**을 클릭합니다.
- 7 **Cloud Assembly 형식으로 클라우드 템플릿 생성**을 클릭합니다.
- 8 클라우드 템플릿 이름을 입력하고 **저장**을 클릭합니다.

**참고** 온보딩 계획이 vSphere 시스템을 사용하는 경우에는 온보딩 프로세스가 완료된 후 클라우드 템플릿을 편집해야 합니다. 온보딩 프로세스는 소스 vSphere 시스템과 해당 시스템 템플릿을 연결할 수 없으며 결과로 생성된 클라우드 템플릿에서는 클라우드 템플릿 코드에 `imageRef: "no image available"` 항목이 포함됩니다. `imageRef`: 필드에 올바른 템플릿 이름을 지정해야 클라우드 템플릿을 배포할 수 있습니다. 온보딩 프로세스가 완료된 후 클라우드 템플릿을 쉽게 찾고 업데이트하려면 배포의 **클라우드 템플릿 구성** 페이지에서 **클라우드 템플릿 이름** 옵션을 사용합니다. 자동 생성된 클라우드 템플릿 이름을 기록하거나 원하는 클라우드 템플릿 이름을 입력하고 기록합니다. 온보딩이 완료되면 클라우드 템플릿을 찾아서 열고 `imageRef`: 필드의 "no image available" 항목을 올바른 템플릿 이름으로 바꿉니다.

- 9 배포 이름 확인란을 클릭하고 **실행**을 클릭한 다음, **계획 실행** 페이지에서 **실행**을 다시 클릭합니다.  
선택한 Amazon Web Services 시스템이 해당하는 클라우드 템플릿과 함께 단일 배포로 온보딩됩니다.
- 10 **클라우드 템플릿** 탭을 클릭한 다음, 클라우드 템플릿 이름을 클릭하여 클라우드 템플릿을 열고 검토합니다.
- 11 **배포** 탭을 클릭한 다음, 배포 이름을 클릭하여 배포를 열고 검토합니다.

## 예: vRealize Automation Cloud Assembly에서 개별 배포로서 온보딩 규칙으로 필터링된 시스템

이 예시에서는 필터링 규칙을 사용하여 상태가 [On]이고 이름이 BG로 시작하는 시스템을 온보딩합니다. 또한 계획의 각 시스템에 대해 별도의 vRealize Automation Cloud Assembly 클라우드 템플릿 및 배포를 생성합니다.

클라우드 계정을 생성할 경우, 이 클라우드 계정에 연결된 모든 시스템의 데이터가 수집되어 **인프라 > 리소스 > 시스템** 페이지에 표시됩니다. 클라우드 계정에 vRealize Automation Cloud Assembly 외부에 배포된 시스템이 있는 경우에는 온보딩 계획을 사용하여 vRealize Automation Cloud Assembly가 해당 시스템을 관리하도록 할 수 있습니다.

### 사전 요구 사항

- 필요한 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- [vRealize Automation Cloud Assembly의 온보딩 계획이란?](#) 항목을 검토합니다.
- vRealize Automation Cloud Assembly 프로젝트를 생성하여 준비하고 하나 이상의 클라우드 계정으로 채웁니다.

이를 위해서는 단계별 설정 절차의 몇 가지 기본적인 단계를 수행해야 합니다.

- 프로젝트를 생성하고, 사용자를 추가하고, 프로젝트에 사용자 역할을 할당합니다. [2부: 예제 vRealize Automation Cloud Assembly 프로젝트 생성](#) 항목을 참조하십시오.
- 프로젝트의 지정된 지역에 하나 이상의 클라우드 계정을 생성합니다.
- 온보딩할 시스템이 **시스템** 페이지에 포함되어 있는지 확인합니다. [vRealize Automation의 시스템 리소스](#) 항목을 참조하십시오.

### 절차

- 1 **인프라 > 온보딩**으로 이동합니다.
- 2 **새 온보딩 계획**을 클릭하고 값을 입력합니다.

설정	샘플 값
계획 이름	ob_rules_1
설명	Machine onboarding with rules1
클라우드 계정	rs-aws
기본 프로젝트	rs-project

## 새 온보딩 계획



계획 이름 ob\_rules\_1

설명

Machine onboarding with rules1

## 사전 요구 사항

클라우드 계정을 추가하고 온보딩할 시스템이 있는 계산 리소스에 대한 클라우드 영역을 생성합니다.

사용자가 한 명 이상인 프로젝트를 생성하고 클라우드 영역에 대한 프로젝트 액세스 권한을 부여합니다.

클라우드 계정 Q rs-aws



기본 프로젝트 Q rs-project



취소

생성

## 3 생성을 클릭합니다.

ob\_rules\_1

요약 규칙 시스템 배포

계획 이름 ob\_rules\_1

설명 Machine onboarding with rules1

계획 상태

마지막 실행 없음

소스 정보

클라우드 계정 346test\_vc\_account

배포 태그 키

적용

대상 구성

기본 프로젝트 Q 123

저장 실행 취소

## 4 규칙 탭을 클릭한 후 규칙 추가를 클릭합니다.

하나 이상의 규칙을 생성하여 구체적인 시스템 특징을 기반으로 온보딩 대상 시스템 그룹을 선택할 수 있습니다.

ob\_rules\_1

요약 **규칙** 시스템 배포

규칙을 사용하여 이 계획에 시스템을 추가합니다. ⓘ

**규칙 추가** 편집 삭제

☐ 이름

## 5 규칙 이름을 입력합니다(예: ob\_rules\_1).

추가 규칙

생성 이 계획의 시스템을 채우는 데 사용되는 필터 기반 규칙입니다.

규칙 이름 \* ob\_rules\_1

## 6 필터를 추가하여 규칙을 구축합니다.

이 예시에서는 **필터** 드롭다운 메뉴에서 **상태** 및 **이름** 필터를 사용하여 이름에 BG\*가 포함되어 있고 상태가 On인 모든 시스템을 지정합니다.

규칙 이름 \*

ob\_rules\_1

필터...

속성	상태	주소	생성 시간	태그
임의:				
이름:	켜짐	10.184.68.223	2020년 1월 09일	
상태:				
주소:				
태그:				

4 시스템

규칙 이름 \*

ob\_rules\_1

이름: BG\* × 상태: 켜짐

## 7 저장을 클릭합니다.

규칙을 더 만들 수도 있지만 이 예시에서는 하나의 규칙을 사용합니다.

ob\_rules\_1

요약 **규칙** 시스템 배포

규칙을 사용하여 이 계획에 시스템을 추가합니다. ⓘ

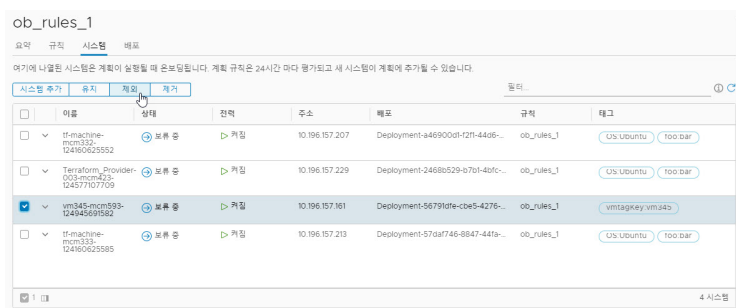
**규칙 추가** 편집 삭제

<input type="checkbox"/> 이름	상태	필터
<input type="checkbox"/> ob_rules_1	✓ 확인	Name: BG*

- 8 시스템 탭을 클릭합니다. 이 예시에서는 시스템 4개가 선택되어 있는데, 그 중 3개는 이름이 BG로 시작하고 나머지 1개는 이름에 BG가 포함되어 있습니다.

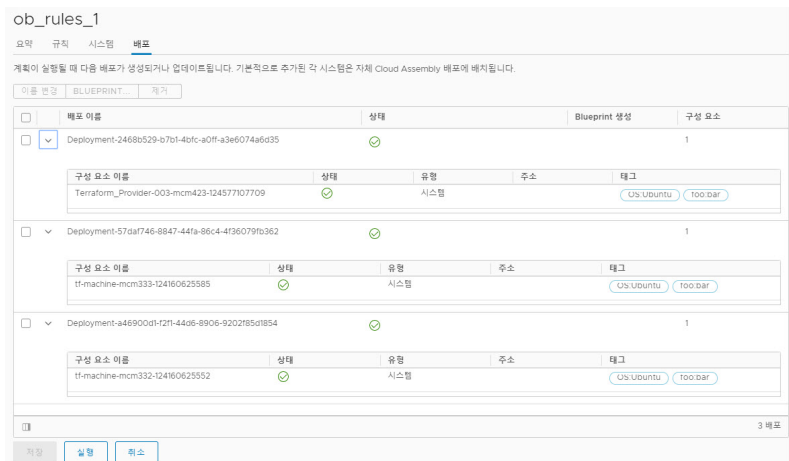


- 9 이름이 BG로 시작하지 않는 시스템의 확인란을 선택한 후 **제외**를 클릭하여 해당 시스템을 제거합니다.



- 10 배포 탭을 클릭합니다.

이름이 BG로 시작하고 전원이 On 상태에 있는 시스템 3개에 대한 배포 준비가 완료되었습니다. 기본적으로 각 시스템에 대해 별도의 클라우드 템플릿과 배포가 생성됩니다.



- 11 배포 이름 3개 옆의 확인란을 클릭한 후 **클라우드 템플릿, Cloud Assembly 형식으로 클라우드 템플릿 생성**을 차례로 클릭하고 **저장**을 클릭합니다.

**참고** 온보딩 계획이 vSphere 시스템을 사용하는 경우에는 온보딩 프로세스가 완료된 후 클라우드 템플릿을 편집해야 합니다. 온보딩 프로세스는 소스 vSphere 시스템과 해당 시스템 템플릿을 연결할 수 없으며 결과로 생성된 클라우드 템플릿에서는 클라우드 템플릿 코드에 imageRef: "no image available" 항목이 포함됩니다. imageRef: 필드에 올바른 템플릿 이름을 지정해야 클라우드 템플릿을 배포할 수 있습니다. 온보딩 프로세스가 완료된 후 클라우드 템플릿을 쉽게 찾고 업데이트하려면 배포의 **클라우드 템플릿 구성** 페이지에서 **클라우드 템플릿 이름** 옵션을 사용합니다. 자동 생성된 클라우드 템플릿 이름을 기록하거나 원하는 클라우드 템플릿 이름을 입력하고 기록합니다. 온보딩이 완료되면 클라우드 템플릿을 찾아서 열고 imageRef: 필드의 "no image available" 항목을 올바른 템플릿 이름으로 바꿉니다.

- 12 배포 페이지에서 배포 이름 3개 옆의 확인란을 클릭한 후 **실행**을 클릭합니다.



- 13 확인 메시지가 표시되면 **실행**을 클릭하여 시스템을 온보딩합니다.

계획 실행

계획 이름

ob\_rules\_1

설명

Machine onboarding with rules1

클라우드 계정

346test\_vc\_account

기본 프로젝트

123

배포

3

마지막 실행

없음

취소

실행

계획이 실행되고 시스템이 vRealize Automation Cloud Assembly 관리 대상으로 온보딩됩니다. 각 시스템에 대해 별도의 클라우드 템플릿과 배포가 생성됩니다.

## vRealize Automation Cloud Assembly 환경에 대한 고급 구성

프로젝트 구성, 통합 및 배포를 추가로 지원하도록 vRealize Automation Cloud Assembly 환경을 구성할 수 있습니다.

사용자 및 로그 사용, 고객 환경 향상 프로그램 참여 또는 탈퇴와 같은 관리 방법에 대한 관련 정보와 추가 정보는 [vRealize Automation 관리](#) 도움말을 참조하십시오.

### vRealize Automation에 대한 인터넷 프록시 서버를 구성하는 방법

인터넷에 직접 액세스할 수 없는 격리된 네트워크에 설치된 vRealize Automation의 경우, 인터넷 프록시 서버를 사용하여 프록시 기능을 통해 인터넷을 허용할 수 있습니다. 인터넷 프록시 서버는 HTTP 및 HTTPS를 지원합니다.

vRealize Automation을 사용하여 AWS(Amazon Web Services), Microsoft Azure, GCP(Google Cloud Platform)와 같은 공용 클라우드 제공자 및 IPAM, Ansible, Puppet과 같은 외부 통합 지점을 구성하고 사용하려면 내부 vRealize Automation 인터넷 프록시 서버에 액세스하도록 인터넷 프록시 서버를 구성해야 합니다.

vRealize Automation에는 인터넷 프록시 서버와 통신하는 내부 프록시 서버가 포함되어 있습니다. 이 서버는 `vracli proxy set ...` 명령으로 구성된 경우 프록시 서버와 통신합니다. 조직에 대해 인터넷 프록시 서버를 구성하지 않은 경우 vRealize Automation 내부 프록시 서버가 인터넷에 직접 연결하려고 시도합니다.

제공된 `vracli` 명령줄 유틸리티를 사용하여 인터넷 프록시 서버를 사용하도록 vRealize Automation을 설정할 수 있습니다. `vracli` API를 사용하는 방법에 대한 정보를 보려면 `vracli` 명령줄에서 `--help` 인수를 사용하면 됩니다(예: `vracli proxy --help`).

인터넷 프록시 서버에 액세스하려면 vRealize Automation에 내장된 ABX(작업 기반 확장성) 온-프레미스 내장형 제어를 사용해야 합니다.

---

**참고** 인터넷 프록시를 통한 Workspace ONE Access(이전 명칭 VMware Identity Manager)에 대한 액세스는 지원되지 않습니다. `vracli set vidm` 명령을 사용하여 인터넷 프록시 서버를 통해 Workspace ONE Access에 액세스할 수 없습니다.

---

인터넷 프록시 서버는 기본 IP 형식으로 IPv4가 필요합니다. TLS(HTTPS) 인증서 트래픽에는 인터넷 프로토콜 제한, 인증 또는 중간자(man-in-the-middle) 작업이 필요하지 않습니다.

#### 사전 요구 사항

- 송신 트래픽을 외부 사이트로 전달할 수 있는 vRealize Automation 네트워크에서 인터넷 프록시 서버로 사용할 수 있는 기존의 HTTP 또는 HTTPS 서버가 있는지 확인합니다. IPv4에 대해 연결을 구성해야 합니다.
- 대상 인터넷 프록시 서버가 기본 IP 형식으로 IPv6이 아닌 IPv4를 지원하도록 구성되어 있는지 확인합니다.

- 인터넷 프록시 서버가 TLS를 사용 중이고 해당 클라이언트에 대해 HTTPS 연결이 필요한 경우에는 프록시 구성을 설정하기 전에 다음 명령 중 하나를 사용하여 서버 인증서를 가져와야 합니다.

- `vracli certificate proxy --set path_to_proxy_certificate.pem`
- `vracli certificate proxy --set stdin`

대화형 입력에 대해 `stdin` 매개 변수를 사용합니다.

## 절차

- 1 Kubernetes에서 사용되는 포트 또는 컨테이너에 대한 프록시 구성을 생성합니다. 이 예에서는 HTTP 체계를 사용하여 프록시 서버에 액세스합니다.

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

- 2 프록시 구성을 표시합니다.

```
vracli proxy show
```

결과는 다음과 유사합니다.

```
{
  "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.\n\nacl proxy-exclude dstdomain 192.168.\n\nacl proxy-exclude dstdomain 172.16.\n\nacl proxy-exclude dstdomain kubernetes\n\nacl proxy-exclude dstdomain 10.192.204.9\n\nacl proxy-exclude dstdomain .eng.vmware.com\n\nacl proxy-exclude dstdomain 10.192.213.146\n\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"
}
```

**참고** 조직에 대해 인터넷 프록시 서버를 구성한 경우 위의 예에서 'default' 대신 "internal.proxy.config.type": "non-default"가 표시됩니다. 암호는 보안을 위해 표시되지 않습니다.

**참고** -proxy-exclude 매개 변수를 사용하는 경우 기본값을 편집해야 합니다. 예를 들어 인터넷 프록시 서버를 사용하여 액세스할 수 없는 도메인으로 acme.com을 추가하려는 경우 다음 단계를 사용합니다.

- a vracli proxy default-no-proxy를 입력하여 기본 proxy-exclude 설정을 가져옵니다. 이것은 자동으로 생성되는 도메인 및 네트워크의 목록입니다.
- b 값을 편집하여 .acme.com을 추가합니다.
- c vracli proxy set .... --proxy-exclude ...를 입력하여 구성 설정을 업데이트합니다.
- d /opt/scripts/deploy.sh 명령을 실행하여 환경을 다시 배포합니다.

### 3 (선택 사항) 인터넷 프록시 서버에서 DNS 도메인, FQDN 및 IP 주소에 액세스하지 못하도록 합니다.

항상 parameter --proxy-exclude를 사용하여 proxy-exclude 변수의 기본값을 수정합니다. exclude.vmware.com 도메인을 추가하려면 우선 vrali proxy show 명령을 사용하고, proxy-exclude 변수를 복사한 다음, 아래와 같이 vracli proxy set ... 명령을 사용하여 도메인 값을 추가합니다.

```
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"
```

**참고** 값을 바꾸는 대신 요소를 proxy-exclude에 추가합니다. proxy-exclude 기본값을 삭제하면 vRealize Automation이 제대로 작동하지 않습니다. 이러한 상황이 발생하면 프록시 구성을 삭제하고 다시 시작합니다.

### 4 vracli proxy set ... 명령으로 인터넷 프록시 서버를 설정한 후에는 vracli proxy apply 명령을 사용하여 인터넷 프록시 서버 구성을 업데이트하고 최신 프록시 설정을 활성화할 수 있습니다.

### 5 아직 수행하지 않은 경우 다음 명령을 실행하여 스크립트 변경 내용을 활성화합니다.

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

### 6 (선택 사항) 필요하다면 포트 22에서 외부 액세스를 지원하도록 프록시 서버를 구성합니다.

Puppet 및 Ansible과 같은 통합을 지원하려면 프록시 서버에서 포트 22가 관련 호스트에 액세스할 수 있도록 허용해야 합니다.

## 예제: 샘플 Squid 구성

1단계를 기준으로, Squid 프록시를 설정하는 경우 /etc/squid/squid.conf에서 다음 샘플에 맞추어 구성을 조정할 수 있습니다.

```
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
```

## vRealize Automation에서 여러 vCenter에 대한 NSX-T 매핑으로 수행할 수 있는 작업

NSX-T 클라우드 계정을 하나 이상의 vCenter 클라우드 계정에 연결하여 다양한 배포 목표를 지원할 수 있습니다.

동일한 기존 NSX-T 네트워크를 다른 vCenter의 네트워크 프로파일에 연결하고 제약 조건에 따라 vCenter에 배포를 프로비저닝할 수 있습니다. 몇 가지 예시가 아래에 나열되어 있습니다.

- 해당 네트워크 프로파일에 여러 vCenter에 걸쳐 있는 NSX-T 네트워크가 포함된 동일한 네트워크 프로파일을 사용하는 여러 NIC가 있는 단일 시스템이 포함된 클라우드 템플릿입니다.
- 서버넷 기반 격리가 포함된 네트워크 프로파일을 사용하고 여러 vCenter에 걸쳐 있는 NSX-T "기존" 네트워크를 사용하는 "전용" 네트워크에 있는 시스템이 포함된 클라우드 템플릿입니다.

- 보안 그룹 기반 격리가 포함된 네트워크 프로파일을 사용하고 vCenter에 걸쳐 있는 NSX-T 네트워크를 사용하는 "전용" 네트워크에 있는 단일 시스템이 포함된 클라우드 템플릿입니다.
- 여러 vCenter에 걸쳐 있는 NSX-T 네트워크를 포함한 네트워크 프로파일을 사용하는 "라우팅된" 네트워크에 있는 단일 시스템이 포함된 클라우드 템플릿입니다.
- 로드 밸런서가 네트워크의 모든 vCenter 시스템에 적용되는 네트워크 프로파일에 정의된 주문형 로드 밸런서가 포함된 클라우드 템플릿입니다.
- 주문형 네트워크가 네트워크 프로파일을 사용하는 모든 vCenter에서 사용되는 네트워크 프로파일에 정의된 주문형 네트워크가 포함된 클라우드 템플릿입니다.
- 선택적으로 방화벽 규칙을 포함하며 보안 그룹이 네트워크의 모든 vCenter에 연결된 주문형 보안 그룹이 포함된 클라우드 템플릿입니다.

NSX-T 네트워크에서 vRealize Automation 내부 또는 외부 IPAM을 구성하고 다른 vCenter에 프로비저닝된 시스템에 대해 동일한 IP 주소를 공유할 수 있습니다.

시스템에 네트워크 프로파일이 정의되어 있지 않은 경우 단일 "기본" NSX-T 네트워크를 공유하는 여러 vCenter에 여러 시스템이 포함된 클라우드 템플릿을 프로비저닝할 수 있습니다.

## vRealize Automation에서 NSX 클라우드 계정 연결을 제거하면 어떻게 됩니까?

NSX 클라우드 계정과 vCenter 클라우드 계정 간의 연결을 제거하는 경우 관련된 네트워크 프로파일을 업데이트하여 연결된 NSX 개체를 제거해야 합니다.

NSX 클라우드 계정과 vCenter 클라우드 계정 간의 연결을 제거하는 경우 인프라 요소가 vRealize Automation에 의해 자동으로 업데이트되지 않습니다. 연결된 NSX 개체를 제거하려면 기존 네트워크 프로파일을 업데이트해야 합니다.

사용자 인터페이스는 영향을 받는 네트워크 프로파일 요소를 다음과 같이 강조 표시하는 데 도움이 되는 정보를 제공합니다.

- 네트워크 프로파일에 NSX 기존 네트워크가 선택되어 있는 경우:
  - 개체가 "잘못됨" 으로 표시되고 메시지 "일부 네트워크 개체가 누락되었거나 잘못되었습니다."가 표시됩니다.
  - 네트워크 프로파일을 저장하면 개체가 제거됩니다.
- 네트워크 프로파일에 App 분리가 구성되어 있는 경우 네트워크 프로파일을 저장하기 전에 분리 정책 설정을 업데이트해야 합니다.
- 네트워크 프로파일에 보안 그룹 또는 로드 밸런서가 선택된 경우 네트워크 프로파일을 저장할 때 해당 개체가 제거됩니다.

기존 배포는 기존 구성 요소에 대해 설계된 대로 계속 작동하지만 새 구성 요소를 생성할 때(예: 확장 작업시)에는 실패합니다.

연결을 다시 설정하면 네트워크 프로파일이 다시 채워지고 기존 배포는 설계된 대로 작동합니다.

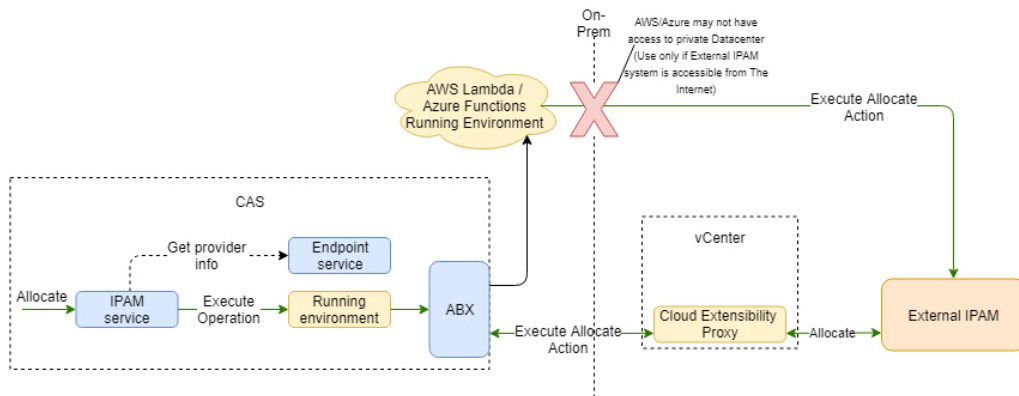
NSX 클라우드 계정을 제거하는 경우 위의 동작이 동일하지만 네트워크 개체는 "잘못됨" 이 아닌 "누락됨" 으로 표시됩니다.

## IPAM SDK를 사용하여 vRealize Automation에 대한 제공자별 외부 IPAM 통합 패키지를 생성하는 방법

외부 IPAM 벤더 및 파트너는 IPAM SDK를 다운로드하고 사용하여 vRealize Automation에서 제공자별 IPAM 솔루션을 지원할 수 있도록 하는 IPAM 통합 패키지를 생성할 수 있습니다.

제공된 IPAM SDK를 사용하여 vRealize Automation에 대한 사용자 지정 IPAM 통합 패키지를 구축하고 배포하는 프로세스는 [VMware Cloud Assembly](#)에 대한 제공자별 IPAM 통합 패키지 생성 및 배포에 설명되어 있습니다. 문서의 설명대로 [VMware code](#) 사이트에서 최신 "VMware vRealize Automation 타사 IPAM SDK" 를 다운로드할 수 있습니다. 다음과 같은 IPAM SDK 패키지를 사용할 수 있습니다.

- [VMware vRealize Automation 타사 IPAM SDK 1.1.0](#)
- [VMware vRealize Automation 타사 IPAM SDK 1.0.0](#)



IPAM SDK를 사용하여 벤더별 IPAM 통합 패키지를 생성하기 전에 vRealize Automation에 대한 패키지가 이미 있는지 확인합니다. IPAM 제공자 웹 사이트, [VMware Marketplace](#) 및 vRealize Automation 마켓플레이스 탭에서 제공자별 IPAM 통합 패키지가 있는지 확인할 수 있습니다.

**Tutorial:** [vRealize Automation에 대한 제공자별 외부 IPAM 통합 구성 예제](#)는 벤더에 따라 다르며 유용한 참조 정보도 포함되어 있습니다.

## Azure VMware Solution 또는 Google Cloud VMware Engine과 함께 vRealize Automation 사용

이 절차에서는 Microsoft Azure 또는 Google Cloud 셀프 서비스 하이브리드 클라우드 환경 내에서 vRealize Automation 워크로드를 사용할 수 있도록 이러한 환경과 작동할 수 있게 vRealize Automation을 설정하는 방법에 대해 설명합니다.

vRealize Automation는 AVS(Azure VMware Solution) 및 GCVE(Google Cloud VMware Engine)와의 연결을 지원하여 Azure 또는 Google Cloud에서 VMware 워크로드를 이동하고 실행합니다. AVS 및 GCVE는 VMware 환경과의 상호 작용을 지원하도록 Microsoft 및 Google에서 만든 제품입니다. 각 제품에는 VMware 배포를 실행하기 위한 고유한 방법과 절차가 있습니다.

AVS 및 GCVE 사용은 해당 벤더에 의해 잘 문서화되어 있습니다. 다음 위치에서 설명서를 찾을 수 있습니다.

- Azure VMware Solution - <https://docs.microsoft.com/en-us/azure/azure-vmware/>
- Google Cloud VMware Engine - <https://cloud.google.com/vmware-engine/docs>

AVS 또는 GCVE를 사용하려면 vCenter 및 NSX-T 클라우드 계정을 모두 설정해야 합니다. 이러한 클라우드 계정을 설정하려면 다음 설명서를 참조하십시오.

- vCenter 클라우드 계정 설정 - [vRealize Automation에서 vCenter 클라우드 계정 생성](#)
- NSX-T 클라우드 계정 생성 - [vRealize Automation에서 NSX-T 클라우드 계정 생성](#)

다음 절차는 AVS 또는 GCVE에 vRealize Automation 워크로드를 배포할 수 있도록 환경을 구성하는 개략적인 단계를 설명합니다.

- 1 벤더 지침에 따라 Azure VMware Solution 또는 Google Cloud VMware Engine을 환경에 맞게 설치하고 구성합니다.
- 2 vRealize Automation 배포 내에서 vCenter와 NSX-T 클라우드 계정을 생성합니다.

# vRealize Automation Cloud Assembly 리소스 인프라 구축

## 4

vRealize Automation Cloud Assembly 리소스 인프라는 클라우드 계정 영역을 클라우드 템플릿 및 해당 워크로드를 배포할 수 있는 영역으로 정의하는 곳입니다.

또한 리소스 인프라에는 이미지 및 시스템 크기와 클라우드 계정 지역 또는 데이터 센터 간에 네트워크 및 스토리지 기능을 정의하는 프로파일에 대한 일반적인 매핑이 생성됩니다.

본 장은 다음 항목을 포함합니다.

- vRealize Automation Cloud Assembly 대상 배치 지역 또는 데이터 센터를 정의하는 클라우드 영역을 추가하는 방법
- 공통 시스템 크기를 지정하기 위해 vRealize Automation에서 버전 매핑을 추가하는 방법
- 공통 운영 체제에 액세스하도록 vRealize Automation에서 이미지 매핑을 추가하는 방법
- vRealize Automation에서 네트워크 프로파일을 추가하는 방법
- 다른 요구 사항을 설명하는 vRealize Automation Cloud Assembly 스토리지 프로파일을 추가하는 방법
- vRealize Automation에서 가격 책정 카드를 사용하는 방법
- 태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법
- vRealize Automation에서 리소스로 작업하는 방법
- vRealize Automation으로 다중 제공자 테넌트 리소스 구성

## vRealize Automation Cloud Assembly 대상 배치 지역 또는 데이터 센터를 정의하는 클라우드 영역을 추가하는 방법

vRealize Automation Cloud Assembly 클라우드 영역은 클라우드 계정 유형(예: AWS 또는 vSphere) 내의 리소스 집합입니다.

특정 계정 지역의 클라우드 영역은 클라우드 템플릿이 워크로드를 배포하는 위치입니다. 각 클라우드 영역은 vRealize Automation Cloud Assembly 프로젝트와 연결됩니다.

**인프라 > 구성 > 클라우드 영역**을 선택하고 **새 영역 추가**를 클릭합니다.

## vRealize Automation Cloud Assembly 클라우드 영역에 대해 알아보기

vRealize Automation Cloud Assembly 클라우드 영역은 클라우드 계정 유형(예: AWS 또는 vSphere)에만 해당하는 계산 리소스 섹션입니다.

클라우드 영역은 지역에 따라 달라지며 프로젝트에 할당해야 합니다. 클라우드 영역과 프로젝트 간에는 다대다 관계가 있습니다. vRealize Automation Cloud Assembly는 vSphere 뿐만 아니라 Azure, AWS 및 GCP를 비롯하여 가장 널리 사용되는 공용 클라우드에 대한 배포를 지원합니다. [vRealize Automation Cloud Assembly에 클라우드 계정 추가](#)의 내용을 참조하십시오.

추가적인 배치 컨트롤로는 배치 정책 옵션, 기능 태그 및 계산 태그가 포함됩니다.

### ■ 배치 정책

배치 정책은 지정된 클라우드 영역 내에서 배포할 호스트를 선택하는 데 사용됩니다.

- **default** - 클러스터와 호스트 전체에 계산 리소스를 임의로 배포합니다. 이 옵션은 개별 시스템 수준에서 작동합니다. 예를 들어, 특정 배포의 모든 시스템이 요구 사항을 충족하는 사용 가능한 클러스터와 호스트 전체에 무작위로 배포됩니다.
- **binpack** - 지정된 계산을 실행하기에 충분한 가용 리소스가 있는, 가장 많이 로드된 호스트에 계산 리소스를 배치합니다.
- **spread** - 가상 시스템 수가 가장 적은 호스트 또는 클러스터에, 배포 수준에서, 계산 리소스를 프로비저닝합니다. vSphere의 경우 DRS(Distributed Resource Scheduler)는 호스트 전체에 가상 시스템을 분산합니다. 예를 들어 배포에서 요청된 모든 시스템은 동일한 클러스터에 배치되지만 다음 배포에서는 현재 로드 여부에 따라 다른 vSphere 클러스터를 선택할 수 있습니다.

예를 들어 다음과 같은 구성이 있다고 가정해 보겠습니다.

- 가상 시스템이 5개인 DRS 클러스터 1
- 가상 시스템이 9개인 DRS 클러스터 2
- 가상 시스템이 6개인 DRS 클러스터 3

가상 시스템이 3개인 클러스터를 요청하고 **Spread** 정책을 선택하면 모두 클러스터 1에 배치해야 합니다. 업데이트된 로드는 클러스터 1에 대해 가상 시스템이 8개가 되고, 클러스터 2와 3에서는 9개와 6개가 유지됩니다.

그런 다음 가상 시스템 2개를 추가로 요청하면 DRS 클러스터 3에 배치되어 가상 시스템이 8개가 됩니다. 클러스터 1과 3에서는 8개와 9개가 유지됩니다.

두 클라우드 영역이 둘 다 프로비저닝에 필요한 모든 조건과 일치하는 경우 배치 논리는 우선 순위가 더 높은 영역을 선택합니다.

### ■ 기능 태그

Blueprint에는 배포 배치를 결정하는 데 도움이 되는 제약 조건 태그가 포함됩니다. 배포 시 Blueprint 제약 조건 태그가 클라우드 영역 및 계산 리소스의 일치하는 기능 태그에 매핑되어, 가상 시스템 계산 리소스 배치에 사용할 수 있는 클라우드 영역이 결정됩니다.

### ■ 계산

이 클라우드 영역에 워크로드를 프로비저닝하는 데 사용할 수 있는 계산 리소스(예: AWS 가용성 영역 및 vCenter 클러스터)를 보고 관리할 수 있습니다.

**참고** vRealize Automation 8.3 릴리스부터는 클라우드 영역이 계산 리소스를 더 이상 공유할 수 없습니다. 공유 계산 리소스를 사용하는 레거시 클라우드 영역은 계속 지원되지만 현재 표준을 준수하도록 업데이트하라는 메시지가 사용자에게 표시됩니다.

클라우드 계정 생성 중 자동 생성된 클라우드 영역은 데이터 수집 후 기본 계산 리소스와 연결됩니다.

vCenter 계산 클러스터가 DRS를 지원하는 경우 클라우드 영역은 계산 목록에 클러스터만 표시하고 하위 호스트는 표시하지 않습니다. vCenter 계산 클러스터가 DRS를 지원하지 않은 경우 클라우드 영역은 독립형 ESXi 호스트(있는 경우)만 표시합니다.

클라우드 영역에 적합하게 계산 리소스를 추가합니다. [계산] 탭에는 계산 리소스가 클라우드 영역에 포함되는 방식을 제어할 수 있는 필터 메커니즘이 있습니다. 초기에 필터 선택은 [모든 계산 포함]이며, 아래 목록에는 사용 가능한 계산 리소스가 표시되고, 모두 배포에 사용할 수 있습니다. 계산 리소스를 클라우드 영역에 추가하는 두 가지 추가 옵션이 있습니다.

- 수동으로 계산 선택 - 아래 목록에서 계산 리소스를 수동으로 선택하려면 이 옵션을 선택합니다. 선택한 후에 [계산 추가]를 클릭하여 리소스를 영역에 추가합니다. 선택한 리소스는 배포에서 사용할 수 있습니다.
- 동적으로 태그별 계산 포함 - 태그를 기반으로 영역에 대한 계산 리소스를 포함하거나 제외하려면 이 옵션을 선택합니다. 계산 리소스의 기존 태그와 일치하는 적절한 태그를 추가할 때까지 모든 계산 리소스가 표시됩니다. 하나 이상의 태그를 추가한 후 필터와 일치하는 태그가 있는 계산 리소스는 영역에 포함되고 배포에 사용할 수 있지만, 일치하지 않는 리소스 제외됩니다.

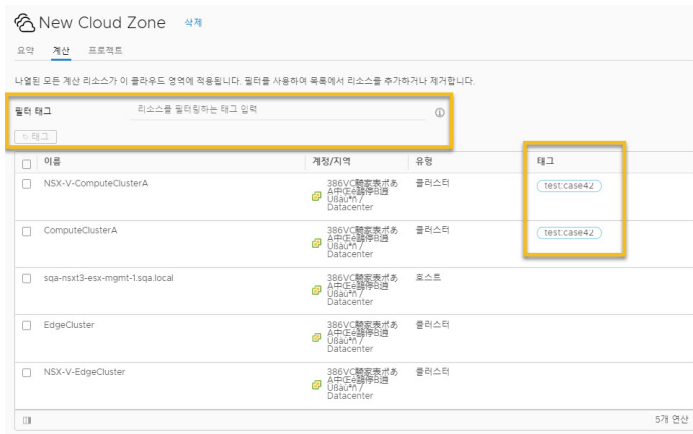
어느 계산 옵션이든, 오른쪽 상자를 선택하고 [제거]를 클릭하면 페이지에 보이는 계산 리소스를 하나 이상 제거할 수 있습니다.

계산 태그는 배치를 추가적으로 제어하는 데 도움이 됩니다. 다음의 예시와 같이 하나 이상의 태그와 일치하는 계산 리소스만 필터링할 수 있습니다.

- 계산 리소스에 태그가 없고 필터링을 사용하지 않습니다.



- 계산 리소스 2개에 동일한 태그가 있고 필터링을 사용하지 않습니다.



- 계산 리소스 2개에 동일한 태그가 있고, 태그가 계산 리소스 2개에 사용된 태그와 일치합니다.



## ■ 프로젝트

이 클라우드 영역으로의 워크로드 프로비저닝을 지원하도록 구성된 프로젝트를 볼 수 있습니다.

클라우드 영역을 생성한 후에는 해당 구성을 검증할 수 있습니다.

## 인사이트 대시보드

vRealize Automation과 함께 작동하도록 구성된 연결된 vRealize Operations Manager 애플리케이션이 있으면 클라우드 영역의 **인사이트** 대시보드에 액세스할 수 있습니다. 이 대시보드에는 vSphere 또는 VMware Cloud on AWS 클라우드 영역의 리소스 및 배포에 대한 용량 관련 정보가 표시됩니다. 단, 클라우드 계정이 vRealize Automation 및 vRealize Operations Manager 둘 다에 구성되어 있고 vRealize Operations Manager에서 모니터링되는 경우에만 해당됩니다. **인사이트** 대시보드에 대해 자세히 알아보려면 [vRealize Automation](#)에서 [vRealize Operations Manager](#) 메트릭을 사용하여 리소스 관리 및 배포 [최적화](#) 항목을 참조하십시오.

## 공통 시스템 크기를 지정하기 위해 vRealize Automation에서 버전 매핑을 추가하는 방법

vRealize Automation 버전 맵은 자연어를 사용하여 특정 클라우드 계정/지역에 대한 대상 배포 크기를 정의하는 곳입니다.

버전 맵은 환경에 적합한 배포 크기를 표현합니다. 한 가지 예는 명명된 데이터 센터의 vCenter 계정에 대한 CPU 1개 메모리 2GB의 경우 "소규모", CPU 2개 메모리 8GB의 경우 "대규모" 및 명명된 지역의 Amazon Web Services 계정에 대한 t2.nano입니다.

**인프라 > 구성 > 버전 매핑**을 선택하고 **새 버전 매핑**을 클릭합니다.

## vRealize Automation의 버전 매핑에 대해 알아보기

버전 매핑은 자연어 이름 지정 방법을 사용하여 vRealize Automation에서 특정 클라우드 계정/지역에 대해 대상 배포 크기 집합을 그룹화합니다.

버전 매핑을 사용하면 계정 지역 전체에서 유사한 버전 크기를 포함하는 이름이 지정된 매핑을 생성할 수 있습니다. 예를 들어 standard\_small이라는 버전 맵에는 프로젝트의 사용 가능한 계정/지역 전체 또는 일부에 사용할 수 있는 유사한 버전 크기(예: CPU 1개, 2GB RAM)가 포함될 수 있습니다. 클라우드 템플릿을 구축할 때는 필요에 맞는 사용 가능한 버전을 선택합니다.

배포 의도에 따라 프로젝트에 대한 버전 매핑을 구성합니다.

클라우드 템플릿 생성을 간소화하기 위해, 새 클라우드 계정을 추가할 때 사전 구성 옵션을 선택할 수 있습니다. 사전 구성 옵션을 선택하면 지정된 지역에 대해 조직에서 가장 널리 사용되는 버전 매핑 및 이미지 매핑이 선택됩니다.

vSphere 리소스가 포함된 클라우드 템플릿의 이미지 매핑과 관련하여, vSphere 클라우드 영역에 대해 정의된 버전 매핑이 없으면 클라우드 템플릿에서 vSphere 관련 설정을 사용하여 무제한 메모리 및 CPU를 구성할 수 있습니다. vSphere 클라우드 영역에 대해 정의된 버전 매핑이 있으면 버전 매핑은 클라우드 템플릿에서 vSphere 관련 구성에 대한 제한으로 작동합니다.

## 공통 운영 체제에 액세스하도록 vRealize Automation에서 이미지 매핑을 추가하는 방법

vRealize Automation 이미지 맵은 자연어를 사용하여 특정 클라우드 계정/지역에 대한 대상 배포 운영 체제를 정의하는 곳입니다.

**인프라 > 구성 > 이미지 매핑**을 선택하고 **새 이미지 매핑**을 클릭합니다.

## vRealize Automation의 이미지 매핑에 대해 알아보기

이미지 매핑은 자연어 이름 지정을 사용하여 vRealize Automation에서 특정 클라우드 계정/지역에 대해 미리 정의된 대상 운영 체제 규격 집합을 그룹화합니다.

Microsoft Azure 및 Amazon Web Services 같은 클라우드 벤더 계정은 이미지를 사용하여 OS 및 관련 구성 설정을 포함한 대상 배포 조건 집합을 그룹화합니다. VMware Cloud on AWS를 포함한 vCenter 및 NSX 기반 환경에서도 유사한 그룹화 메커니즘을 사용하여 OS 배포 조건 집합을 정의합니다. 클라우드 템플릿을 구축한 후 최종적으로 배포하고 반복할 때 필요에 가장 적합한 이미지를 선택합니다.

프로젝트에 대한 이미지 매핑은 유사한 운영 체제, 태그 지정 전략 및 기능적인 배포 의도에 맞게 구성해야 합니다.

클라우드 템플릿 생성을 간소화하기 위해, 새 클라우드 계정을 추가할 때 사전 구성 옵션을 선택할 수 있습니다. 사전 구성 옵션을 선택하면 지정된 지역에 대해 조직에서 가장 널리 사용되는 버전 매핑 및 이미지 매핑이 선택됩니다.

클라우드 템플릿에 이미지 정보를 추가하는 경우에는 시스템 구성 요소의 properties 섹션에서 image 또는 imageRef 항목 중 하나를 사용합니다. 예를 들어 스냅샷에서 복제하려는 경우에는 imageRef 속성을 사용합니다.

클라우드 템플릿 코드의 image 및 imageRef 항목에 대한 예는 [장 6 vRealize Automation Cloud Assembly 배포 설계](#) 항목을 참조하십시오.

컨텐츠 라이브러리에 대한 사용 권한을 할당하려면 관리자가 사용 권한을 사용자에게 글로벌 사용 권한으로 부여해야 합니다. 관련 정보는 [VMware vSphere 설명서](#)의 "vSphere 가상 시스템 관리"에서 "컨텐츠 라이브러리에 대한 사용 권한의 계층적 상속"을 참조하십시오.

## 클라우드 계정/지역에 대한 이미지 동기화

이미지 동기화를 실행하여 **인프라 > 구성 > 이미지 매핑** 페이지에서 지정된 클라우드 계정/지역에 대해 추가하거나 제거할 이미지가 최신 상태인지 확인할 수 있습니다.

- 1 **인프라 > 연결 > 클라우드 계정**을 선택하여 연결된 **클라우드 계정/지역**을 엽니다. 기존 클라우드 계정/지역을 선택합니다.
- 2 **이미지 동기화** 버튼을 클릭하고 작업을 완료합니다.



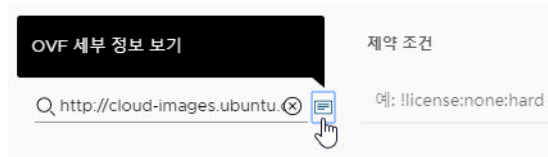
- 3 작업이 완료되면 **인프라 > 구성 > 이미지 매핑**을 클릭합니다. 기존 이미지 매핑을 편집하거나 새로 정의하고 1단계의 클라우드 계정/지역을 선택합니다.
- 4 **이미지 매핑** 페이지에서 이미지 동기화 아이콘을 클릭합니다.



- 5 **이미지 매핑** 페이지에서 지정된 클라우드 계정/지역에 대한 이미지 매핑 설정을 구성합니다.

## OVF 세부 정보 보기

vRealize Automation Cloud Assembly 클라우드 템플릿 개체(예: vCenter 시스템 구성 요소 및 이미지 맵)에 OVF 규격을 포함할 수 있습니다. 이미지에 OVF 파일이 포함되면 파일을 열지 않고도 해당 콘텐츠를 검색할 수 있습니다. OVF 위로 마우스를 이동하면 이름과 위치를 비롯한 OVF 세부 정보가 표시됩니다. OVF 파일 형식에 대한 자세한 내용은 [vcenter ovf: 속성](#)을 참조하십시오.



## Microsoft Azure 이미지 갤러리에서 공유된 최신 이미지 사용

Microsoft Azure용 이미지 매핑을 생성할 때 구독의 공유 Azure 이미지 갤러리에서 이미지를 선택할 수 있습니다. 드롭다운 메뉴의 이미지는 데이터가 수집되고 선택한 지역에 따라 제공됩니다.

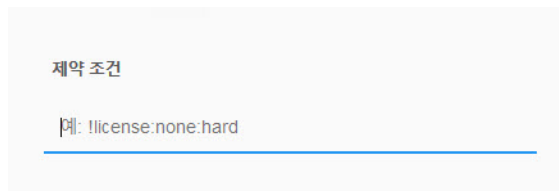
공유 이미지 갤러리를 여러 구독에서 사용할 수 있지만, 여러 구독의 이미지 매핑 드롭다운 메뉴에 나열할 수는 없습니다. 특정 구독의 이미지만 데이터가 수집되고 이미지 매핑 목록에 나열됩니다. 다른 구독에 있는 이미지 갤러리의 이미지를 사용하려면 이미지 매핑에 이미지 ID를 제공하고 클라우드 템플릿에서 해당 이미지 매핑을 사용합니다.

## 제약 조건 및 태그를 사용하여 이미지 선택 구체화

클라우드 템플릿에서 이미지 선택을 더욱 구체화하려면, 제약 조건을 하나 이상 추가하여, 배포할 수 있는 이미지 유형에 대해 태그 기반 제약을 지정하면 됩니다. 이미지 매핑 구성을 생성하거나 편집할 때 표시되는 제공된 **제약 조건**의 예는 `!license:none:hard`입니다. 이 예시는 클라우드 템플릿에

`license:none` 태그가 "없는" 경우에만 이미지를 사용할 수 있는 태그 기반 제약을 보여줍니다.

`license:88` 및 `license:92`와 같은 태그를 추가하면 클라우드 템플릿에 `license:88` 및 `license:92` 태그가 "있는" 경우에만 지정된 이미지를 사용할 수 있습니다.



## 클라우드 구성 스크립트를 사용하여 배포 제어

이미지 맵, 클라우드 템플릿 또는 둘 다에 클라우드 구성 스크립트를 사용하여 vRealize Automation Cloud Assembly 배포에 사용할 사용자 지정 OS 특성을 정의할 수 있습니다. 예를 들어 클라우드 템플릿을 공용 또는 사설 클라우드 중 어디에 배포하는지에 따라 특정 사용자 사용 권한, OS 사용 권한 또는 기타 조건을 이미지에 적용할 수 있습니다. 클라우드 구성 스크립트는 Linux 기반 이미지에 대해서는 `cloud-init` 형식을 따르고 Windows 기반 이미지에 대해서는 `cloudbase-init` 형식을 따릅니다. vRealize Automation Cloud Assembly는 Linux 시스템에 대해서는 `cloud-init` 도구를 지원하고 Windows에 대해서는 `cloudbase-init` 도구를 지원합니다.

Windows 시스템의 경우에는 `cloudbase-init`에서 지원하는 모든 클라우드 구성 스크립트 형식을 사용할 수 있습니다.

아래의 샘플 클라우드 템플릿 코드에 나오는 시스템 리소스는 클라우드 구성 스크립트가 포함된 이미지를 사용하며, 해당 내용은 image 항목 아래에 나와 있습니다.

```
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}
```

## 이미지 매핑과 클라우드 템플릿에 클라우드 구성 스크립트가 포함된 경우 발생하는 결과

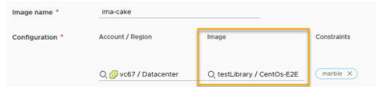
클라우드 구성 스크립트가 포함된 클라우드 템플릿이 클라우드 구성 스크립트가 포함된 이미지 매핑을 사용할 경우, 두 스크립트가 결합됩니다. 병합 작업은 스크립트가 #cloud-config 형식인지 여부를 고려하면서 이미지 매핑 스크립트의 콘텐츠를 우선 처리하고 다음으로 클라우드 템플릿 스크립트의 콘텐츠를 처리합니다.

- #cloud-config 형식의 스크립트에 대해, 병합은 각 모듈(예: runcmd, users 및 write\_files)의 콘텐츠를 다음과 같이 결합합니다.
  - 콘텐츠가 목록인 모듈의 경우 두 목록 모두에서 동일한 명령은 제외하고 이미지 매핑의 명령 목록과 클라우드 템플릿의 명령 목록이 병합됩니다.
  - 콘텐츠가 사전인 모듈의 경우 명령이 병합되고 그 결과 사전이 결합됩니다. 두 사전에 동일한 키가 있는 경우 이미지 매핑 스크립트 사전의 키는 유지되고 클라우드 템플릿 스크립트 사전의 키는 무시됩니다.
  - 콘텐츠가 문자열인 모듈의 경우 이미지 매핑 스크립트의 콘텐츠 값이 유지되고 클라우드 템플릿 스크립트의 콘텐츠 값은 무시됩니다.
- 두 스크립트 모두 #cloud-config 이외의 형식이거나 한 스크립트는 #cloud-config 형식이지만 다른 스크립트가 다른 형식인 경우, 두 스크립트는 이미지 매핑 스크립트가 먼저 실행되고 클라우드 템플릿 스크립트는 이미지 매핑 스크립트가 완료될 때 실행되는 방식으로 결합됩니다.

관련 정보는 [사용자 데이터 섹션 병합](#)을 참조하십시오.

## vCenter 콘텐츠 라이브러리에서 이미지 추가

로컬 또는 게시자 콘텐츠 라이브러리가 vRealize Automation 조직에서 관리하는 vCenter에 있으면 이미지 드롭다운 메뉴에 콘텐츠 라이브러리 템플릿 이미지가 나타납니다. 나열되는 이미지에는 로컬 또는 게시자 vCenter 콘텐츠 라이브러리의 OVF 및 VM 템플릿 이미지가 포함됩니다. 구독자 콘텐츠 라이브러리의 이미지는 드롭다운 메뉴에 나타나지 않습니다. VM이 복제된 템플릿은 시스템 배포 사용자 인터페이스의 시스템 세부 정보 섹션에 표시됩니다.



**참고** vRealize Automation에서 게시자 콘텐츠 라이브러리 vCenter를 관리하는 경우 게시자 정보가 다음과 같은 형식으로 이미지 매핑 선택 그리드에 표시됩니다. *`publisher_content_library_name / content_item_name`*

콘텐츠 라이브러리에 대한 사용 권한을 할당하려면 관리자가 사용 권한을 사용자에게 글로벌 사용 권한으로 부여해야 합니다. 관련 정보는 [VMware vSphere 설명서](#)의 "vSphere 가상 시스템 관리"에서 "콘텐츠 라이브러리에 대한 사용 권한의 계층적 상속"을 참조하십시오.

vRealize Automation에서 게시자 콘텐츠 라이브러리 vCenter를 관리하지 않는 경우 구독자 정보가 다음과 같은 형식으로 이미지 매핑 선택 그리드에 표시됩니다. *`subscriber_content_library_name / content_item_name`*

VM 템플릿 이미지 매핑이 포함된 클라우드 템플릿을 배포하는 경우 vRealize Automation는 데이터스토어에 가장 가까운 콘텐츠 라이브러리의 매핑된 이미지에 액세스한 다음 프로비저닝할 시스템 중에서 호스트에 가장 가까운 이미지에 액세스하려고 시도합니다. 여기에는 로컬 콘텐츠 라이브러리뿐만 아니라 게시자 또는 구독자 콘텐츠 라이브러리도 포함됩니다.

OVF 템플릿 이미지 매핑이 포함된 클라우드 템플릿을 배포할 때 이미지가 로컬 콘텐츠 라이브러리에 있거나 지정된 원격 게시자 콘텐츠 라이브러리의 로컬 구독자에 있으면 OVF 이미지는 이미지 매핑 행에 지정된 대로 액세스됩니다.

## 클라우드 구성 스크립트 구성 및 사용에 대한 자세한 정보

클라우드 템플릿에서 클라우드 구성 스크립트의 사용에 대한 자세한 내용은 [vRealize Automation Cloud Assembly](#) 템플릿에서 시스템을 자동으로 초기화하는 방법 항목을 참조하십시오.

VMware 블로그 문서 [vSphere Customization with Cloud-init While Using vRealize Automation 8 or Cloud](#) 및 [Customizing Cloud Assembly Deployments with Cloud-Init](#)도 참조하십시오.

## vRealize Automation에서 네트워크 프로파일을 추가하는 방법

vRealize Automation 네트워크 프로파일은 배포할 네트워크의 동작을 설명합니다.

예를 들어 네트워크는 내부 전용이 아니라 인터넷에 연결되어야 할 수 있습니다.

네트워크 및 네트워크 프로파일은 클라우드별로 다릅니다.

**인프라 > 구성 > 네트워크 프로파일**을 선택하고 **새 네트워크 프로파일**을 클릭합니다.

## vRealize Automation의 네트워크 프로파일에 대해 알아보기

네트워크 프로파일은 vRealize Automation의 특정 지역 또는 데이터 센터의 클라우드 계정에서 사용할 수 있는 네트워크 설정 및 네트워크 그룹을 정의합니다.

일반적으로 네트워크 프로파일은 예를 들어 기존 네트워크에 아웃바운드 액세스만 있는 소규모 테스트 환경이나 일련의 보안 정책이 필요한 대규모의 로드 밸런싱된 운영 환경과 같은 대상 배포 환경을 지원하기 위해 정의합니다. 네트워크 프로파일을 워크로드별 네트워크 특성의 모음으로 생각할 수 있습니다.

### 네트워크 프로파일의 내용

네트워크 프로파일에는 다음 설정을 포함하여 vRealize Automation의 명명된 클라우드 계정 유형 및 지역에 대한 특정 정보가 포함되어 있습니다.

- 네트워크 프로파일에 대한 명명된 클라우드 계정/지역 및 선택적 기능 태그.
- 명명된 기존 네트워크 및 해당 설정.
- 네트워크 프로파일의 주문형 네트워크 및 기타 측면을 정의하는 네트워크 정책.
- 기존 로드 밸런서가 선택적으로 포함됩니다.
- 기존 보안 그룹이 선택적으로 포함됩니다.

네트워크 프로파일을 기반으로 네트워크 IP 관리 기능을 결정합니다.

네트워크 프로파일 기능 태그는 클라우드 템플릿의 제약 조건 태그와 일치하기 때문에 네트워크 선택을 제어하는 데 도움을 줍니다. 네트워크 프로파일별로 수집되는 네트워크에 할당된 모든 태그도 클라우드 템플릿의 태그와 일치하기 때문에 클라우드 템플릿을 배포할 때 네트워크 선택을 제어하는 데 유용합니다.

기능 태그는 선택 사항입니다. 기능 태그는 네트워크 프로파일의 모든 네트워크에 적용되지만 이것은 네트워크가 해당 네트워크 프로파일의 일부로 사용되는 경우로 국한됩니다. 기능 태그가 없는 네트워크 프로파일의 경우 네트워크 태그에서만 태그 일치가 발생합니다. 매칭된 네트워크 프로파일에 정의된 네트워크 및 보안 설정은 클라우드 템플릿이 배포될 때 적용됩니다.

고정 IP를 사용하는 경우 주소 범위는 vRealize Automation에서 관리합니다. DHCP의 경우 IP 시작 및 끝 주소는 vRealize Automation이 아닌 독립 DHCP 서버에서 관리합니다. DHCP 또는 혼합 네트워크 주소 할당을 사용하는 경우 네트워크 활용도 값은 0으로 설정됩니다. 주문형 네트워크 할당 범위는 네트워크 프로파일에 지정된 CIDR 및 서브넷 크기를 기반으로 합니다. 배포에서 정적 및 동적 할당을 모두 지원하기 위해 할당된 범위는 정적 할당을 위한 범위 하나와 동적 할당을 위한 범위 하나의 두 개 범위로 나뉩니다.

### 네트워크

서브넷이라고도 하는 네트워크는 IP 네트워크의 논리적인 한 구획입니다. 네트워크는 클라우드 계정, IP 주소 또는 범위, 네트워크 태그를 그룹화하여 클라우드 템플릿 배포를 프로비저닝하는 방법과 위치를 제어합니다. 프로파일의 네트워크 매개 변수는 배포의 시스템이 IP 레이어 3을 통해 서로 통신할 수 있는 방법을 정의합니다. 네트워크에는 태그가 있을 수 있습니다.

네트워크를 네트워크 프로파일에 추가하고, 네트워크 프로파일에 사용되는 네트워크 측면을 편집하고, 네트워크 프로파일에서 네트워크를 제거할 수 있습니다.

**참고** VCF(VMware Cloud Foundation) 클라우드 계정 유형의 경우 네트워크 프로파일에 NSX 네트워크만 추가할 수 있으며 vSphere 네트워크는 추가할 수 없습니다. NSX 네트워크 세그먼트는 NSX-T 네트워크에 로컬로 생성되고 글로벌 네트워크로 생성되지 않습니다.

## ■ 네트워크 도메인 또는 전송 영역

네트워크 도메인이나 전송 영역은 vSphere vNetwork Distributed 포트 그룹(dvPortGroup)에 대한 분산 가상 스위치(dvSwitch)입니다. 전송 영역은 *dvSwitch* 또는 *dvPortGroup*과 같은 용어와 유사한 기존 NSX 개념입니다.

NSX 클라우드 계정을 사용하는 경우 페이지의 요소 이름은 **전송 영역**이고, 그러한 경우가 아니면 **네트워크 도메인**입니다.

표준 스위치의 경우 네트워크 도메인이나 전송 영역은 스위치 자체와 동일합니다. 네트워크 도메인이나 전송 영역은 vCenter 내 서브넷의 경계를 정의합니다.

전송 영역은 NSX 논리적 스위치가 도달할 수 있는 호스트를 제어합니다. 하나 이상의 vSphere 클러스터에 걸쳐있을 수 있습니다. 전송 영역에서는 특정 네트워크 사용에 참여할 수 있는 클러스터 및 가상 시스템을 제어합니다. 동일한 NSX 전송 영역에 속하는 서브넷은 동일한 시스템 호스트에 사용할 수 있습니다.

## ■ 도메인

시스템의 도메인 이름을 나타냅니다. 도메인 이름이 vSphere 시스템 사용자 지정 규격으로 전달됩니다.

## ■ IPv4 CIDR 및 IPv4 기본 게이트웨이

클라우드 템플릿의 vSphere 클라우드 계정 및 vSphere 시스템 구성 요소는 이중 IPv6 및 IPv4 인터넷 프로토콜 메서드를 지원합니다. 예: 192.168.100.14/24는 IPv4 주소 192.168.100.14 및 연결된 라우팅 접두사 192.168.100.0, 또는 해당 서브넷 마스크 255.255.255.0을 나타냅니다. 여기에는 24개의 선행 1비트가 있습니다. IPv4 블록 192.168.100.0/22는 192.168.100.0에서 192.168.103.255까지의 IP 주소 1024개를 나타냅니다.

## ■ IPv6 CIDR 및 IPv6 기본 게이트웨이

클라우드 템플릿의 vSphere 클라우드 계정 및 vSphere 시스템 구성 요소는 이중 IPv6 및 IPv4 인터넷 프로토콜 메서드를 지원합니다. 예를 들어 2001:db8::/48은 2001:db8:0:0:0:0:0:0에서 2001:db8:0:ffff:ffff:ffff:ffff:ffff까지의 IPv6 주소 블록을 나타냅니다.

IPv6 형식은 주문형 네트워크에 대해 지원되지 않습니다.

## ■ DNS 서버 및 DNS 검색 도메인

## ■ 공용 IP 지원

네트워크에 공용으로 플래그 지정하려면 이 옵션을 선택합니다. `network type: public` 속성이 있는 클라우드 템플릿의 네트워크 구성 요소는 공용으로 플래그가 지정된 네트워크와 매칭됩니다. 네트워크 선택을 결정하기 위해 클라우드 템플릿 배포 중에 추가 매칭이 발생합니다.

## ■ 영역에 대한 기본값

네트워크에 클라우드 영역의 기본 네트워크로 플래그를 지정하려면 이 옵션을 선택합니다. 클라우드 템플릿 배포 중 다른 네트워크보다 기본 네트워크가 선호됩니다.

## ■ 원본

네트워크 소스를 식별합니다.

## ■ 태그

네트워크에 할당된 하나 이상의 태그를 지정합니다. 태그는 선택 사항입니다. 태그 매칭은 클라우드 템플릿 배포에 사용할 수 있는 네트워크에 영향을 미칩니다.

네트워크 태그는 네트워크 프로파일과 관계없이 네트워크 항목 자체에 존재합니다. 네트워크 태그는 태그가 추가된 네트워크의 모든 발생 및 해당 네트워크를 포함하는 모든 네트워크 프로파일에 적용됩니다. 네트워크는 임의의 수의 네트워크 프로파일에 인스턴스될 수 있습니다. 네트워크 프로파일 상주 여부와 관계없이, 네트워크 태그는 네트워크가 사용되는 모든 위치에서 네트워크와 연결됩니다.

클라우드 템플릿을 배포할 때 클라우드 템플릿의 네트워크 구성 요소에 있는 제약 조건 태그가 네트워크 프로파일 기능 태그를 포함한 네트워크 태그에 매칭됩니다. 기능 태그를 포함하는 네트워크 프로파일의 경우 기능 태그가 해당 네트워크 프로파일이 사용할 수 있는 모든 네트워크에 적용됩니다. 매칭된 네트워크 프로파일에 정의된 네트워크 및 보안 설정은 클라우드 템플릿이 배포될 때 적용됩니다.

## 네트워크 정책

네트워크 프로파일을 사용하여 정적, DHCP 또는 정적 및 DHCP IP 주소 설정이 혼합된 기존 네트워크 도메인의 서브넷을 정의할 수 있습니다. **네트워크 정책** 탭을 사용하여 서브넷을 정의하고 IP 주소 설정을 지정할 수 있습니다.

NSX-V, NSX-T 또는 VMware Cloud on AWS를 사용하는 경우, 클라우드 템플릿에 `networkType: outbound` 또는 `networkType: private`이 필요하거나 NSX 네트워크에 `networkType: routed`가 필요하다면 네트워크 정책 설정이 사용됩니다.

연결된 클라우드 계정에 따라 네트워크 정책을 사용하여 `outbound`, `private` 및 `routed` 네트워크 유형 및 주문형 보안 그룹의 설정을 정의할 수 있습니다. 또한 `existing` 네트워크와 연결된 로드 밸런서가 있을 때 네트워크 정책을 사용하여 해당 네트워크를 제어할 수도 있습니다.

아웃바운드 네트워크는 업스트림 네트워크에 대한 단방향 액세스를 허용합니다. 전용 네트워크는 외부 액세스를 허용하지 않습니다. 라우팅된 네트워크는 라우팅된 네트워크 간에 **East/West** 트래픽을 허용합니다. 이 프로파일의 기존 및 공용 네트워크는 기본 또는 업스트림 네트워크로 사용됩니다.

다음과 같은 주문형 선택 사항에 대한 옵션은 **네트워크 프로파일** 화면 도움말에 설명되어 있고 아래에 요약되어 있습니다.

## ■ 주문형 네트워크 또는 주문형 보안 그룹 생성 안 함

이 옵션은 existing 또는 public 네트워크 유형을 지정할 때 사용할 수 있습니다. outbound, private 또는 routed 네트워크가 필요한 클라우드 템플릿은 이 프로파일과 매칭되지 않습니다.

#### ■ 주문형 네트워크 생성

이 옵션은 outbound, private 또는 routed 네트워크 유형을 지정할 때 사용할 수 있습니다.

Amazon Web Services, Microsoft Azure, NSX, vSphere 및 VMware Cloud on AWS는 이 옵션을 지원합니다.

#### ■ 주문형 보안 그룹 생성

이 옵션은 outbound 또는 private 네트워크 유형을 지정할 때 사용할 수 있습니다.

네트워크 유형이 outbound 또는 private인 경우 일치하는 클라우드 템플릿에 대해 새로운 보안 그룹이 생성됩니다.

Amazon Web Services, Microsoft Azure, NSX 및 VMware Cloud on AWS는 이 옵션을 지원합니다.

네트워크 정책 설정은 클라우드 계정 유형에 따라 다를 수 있습니다. 이러한 설정은 화면 포지판 도움말에 설명되어 있고 아래에 요약되어 있습니다.

#### ■ 네트워크 도메인 또는 전송 영역

네트워크 도메인이나 전송 영역은 vSphere vNetwork Distributed 포트 그룹(dvPortGroup)에 대한 분산 가상 스위치(dvSwitch)입니다. 전송 영역은 dvSwitch 또는 dvPortGroup과 같은 용어와 유사한 기존 NSX 개념입니다.

NSX 클라우드 계정을 사용하는 경우 페이지의 요소 이름은 **전송 영역**이고, 그러한 경우가 아니면 **네트워크 도메인**입니다.

표준 스위치의 경우 네트워크 도메인이나 전송 영역은 스위치 자체와 동일합니다. 네트워크 도메인이나 전송 영역은 vCenter 내 서브넷의 경계를 정의합니다.

전송 영역은 NSX 논리적 스위치가 도달할 수 있는 호스트를 제어합니다. 하나 이상의 vSphere 클러스터에 걸쳐있을 수 있습니다. 전송 영역에서는 특정 네트워크 사용에 참여할 수 있는 클러스터 및 가상 시스템을 제어합니다. 동일한 NSX 전송 영역에 속하는 서브넷은 동일한 시스템 호스트에 사용할 수 있습니다.

#### ■ 외부 서브넷

아웃바운드 액세스 권한이 있는 주문형 네트워크에는 아웃바운드 액세스 권한이 있는 외부 서브넷이 필요합니다. 외부 서브넷은 클라우드 템플릿에서 요청된 경우 아웃바운드 액세스를 제공하는 데 사용되며 네트워크 배치를 제어하지 않습니다. 예를 들어 외부 서브넷은 전용 네트워크를 배치하는 데 영향을 주지 않습니다.

#### ■ CIDR

CIDR 표기법은 IP 주소 및 연결된 라우팅 접두사를 간단하게 표현한 것입니다. CIDR 값은 프로비저닝 중에 서브넷을 생성하는 데 사용할 네트워크 주소 범위를 지정합니다. **네트워크 정책** 탭의 이 CIDR 설정은 /nn으로 끝나고 0 - 32 사이의 값을 포함하는 IPv4 표기법을 수락합니다.

## ■ 서브넷 크기

이 옵션은 이 네트워크 프로파일을 사용하는 배포에서 격리된 각 네트워크에 대해 IPv4 표기법을 사용하여 주문형 네트워크 크기를 지정합니다. 서브넷 크기 설정은 내부 또는 외부 IP 주소 관리에 사용할 수 있습니다.

IPv6 형식은 주문형 네트워크에 대해 지원되지 않습니다.

## ■ 논리적 분산 라우터

NSX-V 클라우드 계정을 사용하는 경우 주문형 라우팅된 네트워크에 대해 논리적 분산 라우터를 지정해야 합니다.

DLR(논리적 분산 라우터)은 NSX-V의 주문형 라우팅된 네트워크 간에 East/West 트래픽을 라우팅하는 데 사용됩니다. 이 옵션은 네트워크 프로파일의 계정/지역 값이 NSX-V 클라우드 계정에 연결된 경우에만 표시됩니다.

## ■ IP 범위 할당

이 옵션은 vSphere를 포함하여 NSX 또는 VMware Cloud on AWS를 지원하는 클라우드 계정에 사용할 수 있습니다.

외부 IPAM 통합 지점이 있는 기존 네트워크를 사용하는 경우 IP 범위 설정을 사용할 수 있습니다.

다음 세 옵션 중 하나를 선택하여 배포 네트워크에 대한 IP 범위 할당 유형을 지정할 수 있습니다.

### ■ 정적 및 DHCP

기본값이고 권장됩니다. 이 혼합 옵션은 할당된 CIDR 및 서브넷 범위 설정을 사용하여, DHCP 서버 풀이 DHCP(동적) 메서드를 사용하여 주소 공간 할당의 절반을 지원하고 정적 메서드를 사용하여 IP 주소 공간 할당의 절반을 지원하도록 구성합니다. 주문형 네트워크에 연결된 일부 시스템에는 할당된 고정 IP 주소가 필요하고 일부는 동적 IP 주소가 필요한 경우 이 옵션을 사용합니다. 두 개의 IP 범위가 생성됩니다.

이 옵션은 주문형 네트워크에 연결된 시스템을 사용하는 배포(일부 시스템에는 고정 IP가 할당되고 다른 시스템에는 NSX DHCP 서버에서 동적으로 할당한 IP가 있는 배포) 및 로드 밸런서 VIP가 정적인 배포에 가장 효과적입니다.

### ■ DHCP(동적)

이 옵션은 할당된 CIDR을 사용하여 DHCP 서버에서 IP 풀을 구성합니다. 이 네트워크의 모든 IP 주소는 동적으로 할당됩니다. 할당된 각 CIDR에 대해 단일 IP 범위가 생성됩니다.

### ■ 정적

이 옵션은 할당된 CIDR을 사용하여 IP 주소를 정적으로 할당합니다. 이 네트워크에 대해 DHCP 서버를 구성할 필요가 없는 경우 이 옵션을 사용합니다. 할당된 각 CIDR에 대해 단일 IP 범위가 생성됩니다.

## ■ IP 블록

외부 IPAM 통합 지점이 있는 주문형 네트워크를 사용하는 경우 IP 블록 설정을 사용할 수 있습니다.

IP 블록 설정을 사용하여 통합된 외부 IPAM 제공자의 네트워크 프로파일에 명명된 IP 블록 또는 범위를 추가할 수 있습니다. 네트워크 프로파일에서 추가된 IP 블록을 제거할 수도 있습니다. 외부 IPAM 통합 생성 방법에 대한 자세한 내용은 [vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가 항목](#)을 참조하십시오.

다음 클라우드 계정/지역 유형에 대해 외부 IPAM을 사용할 수 있습니다.

- vSphere
- NSX-T를 사용한 vSphere
- NSX-V를 사용한 vSphere
- **네트워크 리소스 - 외부 네트워크**

외부 네트워크를 기존 네트워크라고도 합니다. 이러한 네트워크는 데이터가 수집되며 선택이 가능하도록 설정됩니다.

- **네트워크 리소스 - Tier-O 논리적 라우터**

NSX-T는 NSX 배포 외부의 네트워크에 대한 게이트웨이로 Tier-O 논리적 라우터를 사용합니다. Tier-O 논리적 라우터는 주문형 네트워크에 대한 아웃바운드 액세스를 구성합니다.

- **네트워크 리소스 - Edge 클러스터**

지정된 Edge 클러스터는 라우팅 서비스를 제공합니다. Edge 클러스터는 주문형 네트워크 및 로드 밸런서에 대한 아웃바운드 액세스를 구성하는 데 사용됩니다. Edge 장치를 배포할 리소스 풀 또는 Edge 클러스터를 식별합니다.

- **네트워크 리소스 - Edge 데이터스토어**

지정된 Edge 데이터스토어는 Edge 장치를 프로비저닝하는 데 사용됩니다. 이 설정은 NSX-V에만 적용됩니다.

태그를 사용하여 클라우드 템플릿에서 사용할 수 있는 네트워크를 지정할 수 있습니다.

## 로드 밸런서

네트워크 프로파일에 로드 밸런서를 추가할 수 있습니다. 나열된 로드 밸런서는 소스 클라우드 계정에서 데이터를 수집한 정보를 기반으로 사용할 수 있습니다.

네트워크 프로파일의 로드 밸런서에 있는 태그가 클라우드 템플릿의 로드 밸런서 구성 요소에 있는 태그와 일치하는 경우 해당 로드 밸런서가 배포 중에 고려됩니다. 일치하는 네트워크 프로파일의 로드 밸런서가 클라우드 템플릿을 배포할 때 사용됩니다.

## 보안 그룹

클라우드 템플릿을 배포할 때 해당 네트워크 프로파일의 보안 그룹이 프로비저닝된 시스템 NIC에 적용됩니다. Amazon Web Services 특정 네트워크 프로파일의 경우 네트워크 프로파일의 보안 그룹은 [네트워크] 탭에 나열된 네트워크와 동일한 네트워크 도메인(VPC)에서 사용할 수 있습니다. 네트워크 프로파일의 [네트워크] 탭에 네트워크가 나열되어 있지 않으면 사용 가능한 모든 보안 그룹이 표시됩니다.

보안 그룹을 사용하여 주문형 private 또는 outbound 네트워크에 대한 격리 설정을 추가로 정의할 수 있습니다. 보안 그룹은 existing 네트워크에도 적용됩니다.

나열된 보안 그룹은 소스 클라우드 계정에서 데이터 수집된 정보 또는 프로젝트 클라우드 템플릿에 주문형 보안 그룹으로 추가된 정보를 기반으로 사용할 수 있습니다.

보안 그룹은 네트워크 프로파일과 일치하는 네트워크에 연결된 배포의 모든 시스템에 적용됩니다. 클라우드 템플릿에 여러 네트워크가 있을 수 있고 각각 서로 다른 네트워크 프로파일과 일치하므로 네트워크별로 서로 다른 보안 그룹을 사용할 수 있습니다.

기존 보안 그룹에 태그를 추가하면 클라우드 템플릿 Cloud.SecurityGroup 구성 요소에서 보안 그룹을 사용할 수 있습니다. 보안 그룹에는 하나 이상의 태그가 있어야 합니다. 그렇지 않으면 클라우드 템플릿에서 사용할 수 없습니다.

## 네트워크 프로파일, 네트워크, 클라우드 템플릿 및 태그에 대한 추가 정보

네트워크에 대한 추가 정보는 [vRealize Automation의 네트워크 리소스](#) 항목을 참조하십시오.

샘플 네트워크 자동화 워크플로는 [Cloud Assembly](#) 및 [NSX](#)를 사용하여 [네트워크 자동화](#)를 참조하십시오.

태그 및 태그 전략에 대한 추가 정보는 [태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법](#) 항목을 참조하십시오.

## vRealize Automation에서 클라우드 템플릿 및 네트워크 프로파일의 네트워크 설정 사용

vRealize Automation에서 네트워크 및 네트워크 프로파일을 사용하면 배포에 대한 네트워크 프로비저닝 동작을 정의하는 데 도움이 됩니다.

vRealize Automation에서는 클라우드별 네트워크 프로파일을 정의할 수 있습니다. [vRealize Automation의 네트워크 프로파일에 대해 알아보기](#)의 내용을 참조하십시오.

네트워크 및 네트워크 프로파일 설정을 사용하면 vRealize Automation 클라우드 템플릿 및 배포에서 네트워크 IP 주소를 사용하는 방법을 제어할 수 있습니다.

## vRealize Automation 네트워크의 IPv4 및 IPv6 지원

vRealize Automation 네트워크는 순수 IPv4 또는 이중 스택 IPv4 및 IPv6을 지원합니다. 순수 IPv6은 현재 지원되지 않습니다.

모든 클라우드 계정 및 통합 유형에 대해 순수 IPv4가 지원되지만 이중 스택 IPv4 및 IPv6은 vSphere 클라우드 계정과 해당 끝점에 대해서만 지원됩니다.

IPv6은 현재 로드 밸런서, NSX 주문형 네트워크 또는 외부 타사 IPAM 제공자에 사용할 수 없습니다.

## 외부 IPAM 제공자 지원

제공된 내부 IPAM 지원 외에도, 외부 IPAM 제공자를 사용하여 네트워크에 IP 주소를 동적 또는 정적으로 할당할 수 있습니다. 예를 들어, 클라우드 템플릿 설계 및 배포의 기존 네트워크에 IP 범위를 할당하고 클라우드 템플릿 설계 및 배포의 주문형 네트워크에 IP 블록을 할당합니다.

Infoblox와 같은 외부 IPAM 제공자에 대한 지원은 벤더별 IPAM 통합 지점에서 가능하며, 이러한 통합 지점은 **인프라 > 연결 > 통합 추가 > IPAM** 메뉴 시퀀스를 사용하여 생성할 수 있습니다.

외부 IPAM 제공자 주소 정보를 정의하는 옵션은 **네트워크 정책 > IPAM IP 범위 추가** 페이지에서 **IPAM IP 범위 추가** 옵션을 통해 사용할 수 있습니다.

외부 IPAM 통합 지점 생성 방법에 대한 자세한 내용은 **vRealize Automation**에서 **외부 IPAM 통합을 구성하는 방법** 항목을 참조하십시오. 특정 IPAM 벤더에 대한 IPAM 통합 지점을 생성하는 방법에 대한 예는 **Tutorial: vRealize Automation에 대한 제공자별 외부 IPAM 통합 구성** 항목을 참조하십시오.

## 네트워크 유형

클라우드 템플릿의 네트워크 구성 요소는 다음 networkType 유형 중 하나로 정의됩니다.

네트워크 유형	정의
existing	기본 클라우드 제공자(예: vCenter, Amazon Web Services 및 Microsoft Azure)에 구성된 기존 네트워크를 선택합니다. 기존 네트워크는 outbound 주문형 네트워크에 필요합니다. 기존 네트워크에서 정적 IP 주소의 범위를 정의할 수 있습니다.
public	공용 네트워크의 시스템은 인터넷에서 액세스할 수 있습니다. IT 관리자는 이러한 네트워크를 정의합니다. public 네트워크에 대한 정의는 공용 네트워크를 따라 네트워크 트래픽이 발생하도록 허용하는 네트워크에 대한 existing 네트워크의 정의와 동일합니다.
private	주문형 네트워크 유형입니다. 배포된 네트워크의 리소스 간에만 네트워크 트래픽이 발생하도록 제한합니다. 인바운드 및 아웃바운드 트래픽을 방지합니다. NSX에서 주문형 NAT 일대다와 같을 수 있습니다.
outbound	주문형 네트워크 유형입니다. 배포의 계산 리소스 간에만 네트워크 트래픽이 발생하도록 제한하되, 단방향 아웃바운드 네트워크 트래픽도 허용합니다. NSX에서 외부 IP를 사용한 주문형 NAT와 같을 수 있습니다.
routed	주문형 네트워크 유형입니다. 라우팅된 네트워크에는 서로 연결된 사용 가능한 서브넷에 걸쳐 나누어진 라우팅 가능 IP 공간이 포함됩니다. 라우팅된 네트워크 프로파일이 동일한 라우팅된 네트워크로 프로비저닝된 가상 시스템은 서로 통신할 수 있으며 기존 네트워크와도 통신할 수 있습니다. 라우팅된 네트워크는 NSX-V 및 NSX-T 네트워크에서 사용할 수 있는 주문형 네트워크 유형입니다. 이 연결은 기본적으로 Microsoft Azure 및 Amazon Web Services에서 제공됩니다. routed 네트워크는 Cloud.NSX.Network 네트워크 구성 요소의 클라우드 템플릿 규격에만 사용할 수 있습니다.

자세한 내용은 **vRealize Automation** 클라우드 템플릿에서 **네트워크 리소스 사용** 항목을 참조하십시오.

네트워크 구성 요소 데이터를 포함하는 채워진 클라우드 템플릿의 예는 **vRealize Automation** 클라우드 템플릿의 **네트워크, 보안 및 로드 밸런서** 예 항목을 참조하십시오.

## 네트워킹 시나리오

다음과 같은 네트워크 프로파일 구성을 사용하는 클라우드 템플릿을 배포할 경우 다음과 같은 동작을 예상할 수 있습니다.

네트워크 유형 또는 시나리오	클라우드 영역에 대해 네트워크 프로파일을 사용할 수 없음	클라우드 영역에 대해 네트워크 프로파일을 사용할 수 있음
네트워크 없음	클라우드 템플릿에 네트워크가 지정되지 않은 경우, 계산과 동일한 프로비저닝 지역에서 임의의 네트워크가 선택됩니다. 기본값으로 레이블이 지정된 네트워크가 기본으로 사용됩니다. 사용 가능한 프로비저닝 지역에 네트워크가 없으면 프로비저닝이 실패합니다.	일치하는 네트워크 프로파일에서 네트워크가 선택됩니다. 기본값으로 레이블이 지정된 네트워크가 기본으로 사용됩니다. 조건을 충족하는 네트워크 프로파일이 없으면 프로비저닝이 실패합니다.
기존 네트워크	클라우드 템플릿의 네트워크 구성 요소에 제약 조건 태그가 포함된 경우, 해당 제약 조건은 사용 가능한 네트워크 목록을 필터링하는 데 사용됩니다. 클라우드 템플릿의 네트워크 구성 요소에 포함된 제약 조건 태그는 네트워크 태그 및 네트워크 프로파일 제약 조건 태그(사용 가능한 경우)와 일치 여부가 확인됩니다. 필터링된 네트워크 목록에서 계산과 동일한 프로비저닝 지역에 있는 단일 네트워크가 선택됩니다. 기본값으로 레이블이 지정된 네트워크가 기본으로 사용됩니다. 제약 조건에 기반하여 필터링한 이후 프로비저닝 지역에 네트워크가 없으면 프로비저닝이 실패합니다.	일치하는 네트워크 프로파일에서 네트워크가 선택됩니다. 기본값으로 레이블이 지정된 네트워크가 기본으로 사용됩니다. 조건을 충족하는 네트워크 프로파일이 없으면 프로비저닝이 실패합니다. 네트워크 제약 조건을 사용하여, 미리 할당된 태그를 기준으로 프로파일에 있는 기존 네트워크를 필터링할 수 있습니다.
공용 네트워크	네트워크에 제약 조건이 있는 경우, supports public IP 특성이 설정된 사용 가능한 네트워크 목록을 필터링하는 데 해당 제약 조건이 사용됩니다. 필터링된 네트워크 목록에서 계산과 동일한 프로비저닝 지역에 있는 임의의 네트워크가 선택됩니다. 기본값으로 레이블이 지정된 네트워크가 기본으로 사용됩니다. 제약 조건에 기반하여 필터링한 이후 프로비저닝 지역에 공용 네트워크가 없으면 프로비저닝이 실패합니다.	일치하는 네트워크 프로파일에서 supports public IP 특성을 가진 네트워크가 선택됩니다. 기본값으로 레이블이 지정된 네트워크가 기본으로 사용됩니다. 네트워크 제약 조건을 사용하여, 미리 할당된 태그를 기준으로 프로파일에 있는 기존 공용 네트워크를 필터링할 수 있습니다.
전용 네트워크	전용 네트워크에 네트워크 프로파일의 정보가 필요하기 때문에 프로비저닝이 실패합니다.	새 네트워크 또는 새 보안 그룹은 일치하는 네트워크 프로파일의 설정을 기반으로 생성됩니다. 네트워크 제약 조건을 사용하여 네트워크 프로파일 및 네트워크를 필터링할 수 있습니다.

네트워크 유형 또는 시나리오	클라우드 영역에 대해 네트워크 프로파일을 사용할 수 없음	클라우드 영역에 대해 네트워크 프로파일을 사용할 수 있음
아웃바운드 네트워크	아웃바운드 네트워크에 네트워크 프로파일의 정보가 필요하기 때문에 프로비저닝이 실패합니다.	새 네트워크 또는 새 보안 그룹은 일치하는 네트워크 프로파일의 설정을 기반으로 생성됩니다.  네트워크 제약 조건을 사용하여 네트워크 프로파일 및 네트워크를 필터링할 수 있습니다.
주문형 라우팅된 네트워크	라우팅된 네트워크에 네트워크 프로파일의 정보가 필요하기 때문에 프로비저닝이 실패합니다.	NSX-V의 경우 DLR(논리적 분산 라우터)를 선택해야 합니다.  NSX-T 및 VMware Cloud on AWS의 경우 전용 및 아웃바운드와 유사한 주문형 설정이 필요합니다.
기존 네트워크 또는 공용 네트워크를 사용하는 Wordpress 사용 사례 예시	기존 네트워크 또는 공용 네트워크에 대한 설명 대로 프로비저닝이 수행됩니다.	기존 네트워크 및 공용 네트워크 동작에 대한 위의 설명을 참조하십시오.  자습서: <a href="#">vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트 항목을 참조하십시오.</a>
기존 또는 공용 네트워크 및 전용 또는 아웃바운드 네트워크를 사용하는 Wordpress 사용 사례 예시	네트워크에 네트워크 프로파일의 정보가 필요하기 때문에 프로비저닝이 실패합니다.	전용 네트워크 및 아웃바운드 네트워크에 대한 위의 설명을 참조하십시오.  자습서: <a href="#">vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트 항목을 참조하십시오.</a>
로드 밸런서를 사용하는 Wordpress 사용 사례 예시	로드 밸런서에 네트워크 프로파일의 정보가 필요하기 때문에 프로비저닝이 실패합니다.  기존 로드 밸런서가 있으면 프로비저닝을 수행할 수 있습니다.	네트워크 프로파일 구성에 기반하여 새 로드 밸런서가 생성됩니다.  네트워크 프로파일에 사용하도록 설정된 기존 로드 밸런서를 지정할 수 있습니다.  기존 로드 밸런서를 요청했지만 네트워크 프로파일의 제약 조건을 충족하는 로드 밸런서가 없으면 프로비저닝이 실패합니다.  자습서: <a href="#">vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트 항목을 참조하십시오.</a>

## vRealize Automation Cloud Assembly에서 네트워크 프로파일 및 클라우드 템플릿 설계의 보안 그룹 설정 사용

네트워크 프로파일 및 클라우드 템플릿 설계의 보안 그룹 설정을 정의하고 변경할 수 있습니다.

보안 그룹 기능을 사용하는 방법은 다음과 같이 다양합니다.

- 네트워크 프로파일에 지정된 기존 보안 그룹

기존 보안 그룹을 네트워크 프로파일에 추가할 수 있습니다. 클라우드 템플릿 설계에서 네트워크 프로파일을 사용하는 경우 해당 시스템은 보안 그룹의 멤버로 함께 그룹화됩니다. 이 방법을 사용하는 경우 클라우드 템플릿 설계에 보안 그룹 리소스를 추가할 필요가 없습니다. 또한 이 구성에서 로드 밸런서를 사용할 수 있습니다. 관련 정보는 [vRealize Automation 클라우드 템플릿에서 로드 밸런서 리소스 사용의 내용](#)을 참조하십시오.

#### ■ 클라우드 템플릿 설계의 시스템 리소스에 연결된 보안 그룹 구성 요소

클라우드 템플릿 설계에 보안 그룹 리소스를 끌어다 놓고 클라우드 템플릿 설계의 기존 보안 그룹 리소스 및 데이터 수집된 리소스의 기존 보안 그룹에서 제약 조건 태그를 사용하여 시스템 NIC에 보안 그룹 리소스를 바인딩할 수 있습니다. 또한 설계 캔버스의 시스템에 네트워크를 연결하는 방법과 유사하게 클라우드 템플릿 설계 캔버스의 연결 줄과 함께 개체를 연결하여 이 연결을 설정할 수 있습니다.

보안 그룹 리소스를 클라우드 템플릿 설계 캔버스에 끌어다 놓을 때 existing 또는 new 유형일 수 있습니다. existing 보안 그룹 유형인 경우 메시지가 표시되면 태그 제약 조건 값을 추가해야 합니다. new 보안 그룹 유형인 경우 방화벽 규칙을 구성할 수 있습니다.

#### ■ 태그 제약 조건으로 할당되고 클라우드 템플릿의 시스템 NIC와 연결된 기존 보안 그룹

예를 들어 두 리소스 간에 태그를 일치시켜 보안 그룹 리소스를 클라우드 템플릿 설계에 있는 시스템 리소스의 시스템 NIC와 연결할 수 있습니다.

소스 끝점에 태그가 지정된 경우 NSX-T에 대한 예로, NSX-T 애플리케이션에 지정된 NSX-T 태그를 사용할 수 있습니다. 그런 다음, 네트워크 리소스가 클라우드 템플릿 설계의 시스템 NIC에 연결되어 있는 클라우드 템플릿 설계의 네트워크 리소스에서 제약 조건으로 지정된 NSX-T 태그를 사용할 수 있습니다. NSX-T 태그를 사용하면 NSX-T 소스 끝점에서 데이터 수집된 미리 정의된 NSX-T 태그를 사용하여 시스템을 동적으로 그룹화할 수 있습니다. NSX-T에서 NSX-T 태그를 생성할 때 논리적 포트를 사용합니다.

#### ■ 클라우드 템플릿 설계의 주문형 보안 그룹 리소스의 방화벽 규칙

클라우드 템플릿 설계의 주문형 보안 그룹에 방화벽 규칙을 추가할 수 있습니다.

사용 가능한 방화벽 규칙에 대한 자세한 내용은 [vRealize Automation 클라우드 템플릿에서 보안 그룹 리소스 사용의 내용](#)을 참조하십시오.

## 자세히 알아보기

네트워크 프로파일의 보안 그룹 정의에 대한 자세한 내용은 [vRealize Automation의 네트워크 프로파일](#)에 대해 [알아보기](#)의 내용을 참조하십시오.

인프라 리소스 페이지에서 보안 그룹 설정 보기 및 변경에 대한 자세한 내용은 [vRealize Automation의 보안 리소스](#)의 내용을 참조하십시오.

클라우드 템플릿 설계의 보안 그룹 정의에 대한 자세한 내용은 [vRealize Automation 클라우드 템플릿에서 보안 그룹 리소스 사용의 내용](#)을 참조하십시오.

클라우드 템플릿 설계의 보안 그룹 리소스 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예의 내용](#)을 참조하십시오.

## vRealize Automation Cloud Assembly에서 네트워크 프로파일의 로드 밸런서 설정 사용

네트워크 프로파일 구성에서 로드 밸런서 설정을 구성할 수 있습니다.

**로드 밸런서** 탭을 사용하여 네트워크 프로파일에 기존 로드 밸런서를 추가할 수 있습니다.

로드 밸런서를 하나 이상의 로드 밸런서가 포함된 네트워크 프로파일에 연결하거나 클라우드 템플릿 설계 캔버스 또는 코드에서 로드 밸런서 리소스를 사용하여 직접 해당 로드 밸런서를 클라우드 템플릿 설계에 추가할 수 있습니다.

### 네트워크 프로파일에서 보안 그룹을 사용하는 방식으로 로드 밸런서 VIP를 포함하는 예

네트워크 프로파일에서 사용할 수 있는 보안 그룹에는 두 가지 유형이 있습니다. 즉, **보안 그룹** 탭에서 선택하는 기존 보안 그룹과 **네트워크 정책** 탭에서 격리 정책을 사용하여 생성하는 주문형 보안 그룹이 있습니다.

네트워크 프로파일 설정을 통해 로드 밸런서 VIP가 보안 그룹에 연결되면 네트워크 프로파일이 해당 보안 그룹 구성을 제공합니다.

다음 표에서는 몇 가지 샘플 시나리오를 보여 줍니다.

클라우드 템플릿 설계 토폴로지 - 연결된 리소스	네트워크 프로파일 구성	보안 그룹 멤버 자격
VIP가 전용 네트워크에 있고 시스템이 동일한 전용 네트워크에 있는 1개 암 로드 밸런서	선택한 네트워크 프로파일은 주문형 보안 그룹으로 정의된 격리 정책을 사용합니다.	시스템 NIC 및 로드 밸런서 VIP가 격리 보안 그룹에 추가됩니다.
VIP가 전용 네트워크에 있고 시스템이 동일한 전용 네트워크에 있는 1개 암 로드 밸런서	선택한 네트워크 프로파일은 기존 보안 그룹을 사용하고 주문형 보안 그룹으로 정의된 격리 정책을 사용합니다.	시스템 NIC 및 로드 밸런서 VIP가 격리 보안 그룹 및 기존 보안 그룹에 추가됩니다.
VIP가 공용 네트워크에 있고 시스템이 전용 네트워크에 있는 2개 암 로드 밸런서	선택한 네트워크 프로파일은 기존 보안 그룹을 사용하고 주문형 보안 그룹으로 정의된 격리 정책을 사용합니다.	시스템 NIC 및 로드 밸런서 VIP가 격리 보안 그룹 및 기존 보안 그룹에 추가됩니다.
공용 네트워크의 VIP 및 전용 네트워크의 시스템이 포함된 2개 암 로드 밸런서.	선택한 네트워크 프로파일은 기존 보안 그룹을 사용합니다.	시스템 NIC 및 로드 밸런서 VIP가 기존 보안 그룹에 추가됩니다.
VIP가 네트워크 1에 있고 시스템이 네트워크 2에 있는 2개 암 로드 밸런서	2개의 네트워크 프로파일: <ul style="list-style-type: none"> <li>■ 네트워크 프로파일 1: 기존 보안 그룹 1을 사용합니다.</li> <li>■ 네트워크 프로파일 2: 기존 보안 그룹 2를 사용합니다.</li> </ul>	로드 밸런서가 네트워크 프로파일 1에 있고 시스템이 네트워크 프로파일 2에 있습니다. 로드 밸런서 VIP가 보안 그룹 1에 추가되고 시스템 NIC가 보안 그룹 2에 추가됩니다.

### 자세히 알아보기

클라우드 템플릿 설계에 로드 밸런서 리소스를 추가하는 방법에 대한 자세한 내용은 [vRealize Automation 클라우드 템플릿에서 로드 밸런서 리소스 사용](#)의 내용을 참조하십시오.

로드 밸런서를 포함하는 클라우드 템플릿 설계에 대한 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)의 내용을 참조하십시오.

## vRealize Automation에서 외부 IPAM 통합에 대해 주문형 네트워크를 지원하도록 네트워크 프로파일을 구성하는 방법

외부 IPAM 통합을 사용하는 vRealize Automation 클라우드 템플릿에서 네트워크 프로파일을 사용하는 경우 주문형 네트워크에 대해 IP 주소 블록을 지원하도록 네트워크 프로파일을 구성할 수 있습니다.

특정 외부 IPAM 제공자에 대한 기존 통합을 사용하면 주문형 네트워크를 프로비저닝하여 외부 IPAM 시스템에 새 네트워크를 생성할 수 있습니다.

이 프로세스를 사용하여, vRealize Automation의 내부 IPAM을 사용할 때와 마찬가지로 상위 CIDR을 제공하는 대신 IP 주소 블록을 구성합니다. IP 주소 블록은 주문형 네트워크 프로비저닝 중에 새 네트워크를 세그먼트화하는 데 사용됩니다. IP 블록은 통합이 주문형 네트워킹을 지원하는 경우 외부 IPAM 제공자에서 데이터 수집됩니다. 예를 들어 Infoblox IPAM 통합을 사용하는 경우 IP 블록은 Infoblox 네트워크 컨테이너를 나타냅니다.

클라우드 템플릿에서 주문형 네트워크 프로파일 및 외부 IPAM 통합을 사용하는 경우, 클라우드 템플릿을 배포할 때 다음 이벤트가 발생합니다.

- 외부 IPAM 제공자에 네트워크가 생성됩니다.
- vRealize Automation에도 네트워크가 생성되며, CIDR 및 게이트웨이 속성과 같은 설정을 포함하여 IPAM 제공자의 새 네트워크 구성을 반영합니다.
- 배포된 가상 시스템의 IP 주소는 새로 생성된 네트워크에서 가져옵니다.

이 주문형 네트워킹 예시에서는 클라우드 템플릿 배포에서 Infoblox를 외부 IPAM 제공자로 사용하여 vSphere의 주문형 네트워크에 시스템을 프로비저닝하도록 허용하는 네트워크 프로파일을 구성합니다.

관련 정보는 [vRealize Automation에서 외부 IPAM 통합에 대해 기존 네트워크를 지원하도록 네트워크 프로파일을 구성하는 방법](#) 항목을 참조하십시오. 두 네트워크 구성 예시 모두 [자습서: vRealize Automation에 대해 VMware Cloud on AWS 구성의 외부 IPAM 통합에 대한 전체 벤더별 워크플로에 적합합니다.](#)

### 사전 요구 사항

다음 사전 요구 사항은 네트워크 프로파일을 생성하거나 편집하는 사용자에게 적용되지만 네트워크 프로파일 자체는 IPAM 통합이 포함된 클라우드 템플릿 배포에서 사용될 때 적용할 수 있습니다. 벤더별 IPAM 통합 지점에 대한 자세한 내용은 [vRealize Automation에서 외부 IPAM 통합을 구성하는 방법](#) 항목을 참조하십시오.

다음 일련의 단계는 IPAM 제공자 통합 워크플로의 컨텍스트에서 표시됩니다. [Tutorial: vRealize Automation에 대한 제공자별 외부 IPAM 통합 구성](#)의 내용을 참조하십시오.

- 클라우드 관리자 자격 증명이 있는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명](#) 항목을 참조하십시오.
- 클라우드 관리자 사용자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.

- 외부 IPAM 제공자(예: [Infoblox](#) 또는 [Bluecat](#))의 계정이 있고, IPAM 제공자를 통해 조직의 계정에 대한 올바른 액세스 자격 증명이 있는지 확인합니다. 이 예시 워크플로에서 IPAM 제공자는 Infoblox입니다.
- IPAM 제공자에 대한 IPAM 통합 지점이 있고 IPAM 통합을 생성하는 데 사용된 IPAM 패키지가 주문형 네트워크를 지원하는지 확인합니다. vRealize Automation에서 [Infoblox에 대한 외부 IPAM 통합 추가](#)의 내용을 참조하십시오.

Infoblox IPAM 패키지는 주문형 네트워크를 지원하지만 다른 제공자에 대한 외부 IPAM 통합을 사용하는 경우에는 해당 IPAM 통합 패키지가 주문형 네트워크를 지원하는지 확인합니다.

## 절차

- 1 네트워크 프로파일을 구성하려면 **인프라 > 구성 > 네트워크 프로파일**을 클릭합니다.
- 2 **새 네트워크 프로파일**을 클릭합니다.
- 3 **요약** 탭을 클릭하고 다음 샘플 설정을 지정합니다.
  - vSphere 클라우드 계정/지역을 지정합니다(예: **vSphere-IPAM-OnDemandA/Datacenter**).  
이 예시에서는 NSX 클라우드 계정에 연결되지 않은 vSphere 클라우드 계정을 사용한다고 가정합니다.
  - 네트워크 프로파일의 이름을 지정합니다(예: **Infoblox-OnDemandNP**).
  - 네트워크 프로파일에 대한 기능 태그를 추가합니다(예: **infoblox\_ondemandA**).  
**기능 태그 값을 기록해 둡니다. 클라우드 템플릿을 프로비저닝할 때 네트워크 프로파일 연결을 사용하기 위해 기능 태그를 클라우드 템플릿 제약 조건 태그로 사용해야 하기 때문입니다.**
- 4 **네트워크 정책** 탭을 클릭하고 다음 샘플 설정을 지정합니다.
  - **분리 정책** 드롭다운 메뉴에서 **주문형 네트워크**를 선택합니다.  
이 옵션을 사용하면 외부 IPAM IP 블록을 사용할 수 있습니다. 클라우드 계정에 따라 새 옵션이 표시됩니다. 예를 들어 NSX 클라우드 계정에 연결된 vSphere 클라우드 계정을 사용하는 경우 다음 옵션이 표시됩니다.
    - 전송 영역
    - Tier-0 논리적 라우터
    - Edge 클러스터
 이 예시에서는 vSphere 클라우드 계정이 NSX에 연결되어 있지 않으므로 **네트워크 도메인** 메뉴 옵션이 나타납니다.
    - **네트워크 도메인** 옵션을 비워 둡니다.
- 5 주소 관리 **소스**로 **외부**를 클릭합니다.
- 6 **IP 블록 추가**를 클릭하여 **IPAM IP 블록 추가** 페이지를 엽니다.

**7 IPAM IP 블록 추가** 페이지의 **제공자** 메뉴에서 기존 외부 IPAM 통합을 선택합니다. 예를 들어, 예시 워크플로의 vRealize Automation에서 Infoblox에 대한 외부 IPAM 통합 추가 에서 생성한 *Infoblox\_Integration* 통합 지점을 선택합니다.

**8 주소 공간** 메뉴에서 사용 가능 및 나열된 IP 블록 중 하나를 선택하고(예: **10.23.118.0/24**) 추가합니다.

IPAM 제공자가 주소 공간을 지원하는 경우 **주소 공간** 메뉴가 표시됩니다. Infoblox 통합의 경우 주소 공간은 Infoblox 네트워크 보기로 표시됩니다.

**9 서브넷 크기**(예: **/29(-6개 IP 주소)**)를 선택합니다.

**10 생성**을 클릭합니다.

## 결과

지정된 외부 IPAM 통합을 사용하여 주문형 네트워크를 프로비저닝하는 데 사용할 수 있는 네트워크 프로파일이 생성됩니다. 다음 샘플 클라우드 템플릿은 이 새 네트워크 프로파일에 의해 정의된 네트워크에 배포되는 단일 시스템을 보여줍니다.

```
formatVersion: 1
inputs: {}
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: ubuntu
      flavor: small
      networks:
        - network: '${resource.Cloud_Network_1.id}'
          assignment: static
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: private
      constraints: - tag: infoblox_ondemandA
```

**참고** 클라우드 템플릿을 배포할 때 지정된 IP 블록에서 사용 가능한 첫 번째 네트워크를 가져와 네트워크 CIDR로 간주합니다. 클라우드 템플릿에서 NSX 네트워크를 사용하는 경우에는 아래와 같이 네트워크 속성 `networkCidr`를 사용하여 네트워크의 CIDR을 수동으로 설정할 수 있습니다. 그러면 CIDR을 수동으로 설정하고 연결된 네트워크 프로파일에 지정된 IP 블록 및 서브넷 크기 설정을 재정의할 수 있습니다.

```
Cloud_Network_1:
  type: Cloud.Network
  properties:
    networkCidr: 10.10.0.0/16
```

## vRealize Automation에서 외부 IPAM 통합에 대해 기존 네트워크를 지원하도록 네트워크 프로파일을 구성하는 방법

외부 IPAM 통합을 사용하는 vRealize Automation Blueprint에서 네트워크 프로파일을 사용하는 경우 기존 네트워크에 대해 IP 주소 범위를 지원하도록 해당 네트워크 프로파일을 구성할 수 있습니다.

vRealize Automation에서 기존 네트워크에 대해 외부 IPAM을 사용하도록 네트워크 및 네트워크 프로파일 구성에서 벤더별 샘플 워크플로와 관련된 예시를 제공합니다. 외부 IPAM 통합에 대한 전체 벤더별 워크플로는 [자습서: vRealize Automation에 대해 VMware Cloud on AWS 구성에서 제공됩니다.](#)

관련 정보는 [vRealize Automation에서 외부 IPAM 통합에 대해 주문형 네트워크를 지원하도록 네트워크 프로파일을 구성하는 방법](#) 항목을 참조하십시오.

## 다른 요구 사항을 설명하는 vRealize Automation Cloud Assembly 스토리지 프로파일을 추가하는 방법

vRealize Automation Cloud Assembly 스토리지 프로파일은 배포할 스토리지의 종류를 설명합니다.

스토리지는 일반적으로 서비스 수준 또는 비용, 성능 또는 용도(예: 백업)와 같은 특징에 따라 프로파일링됩니다.

**인프라 > 구성 > 스토리지 프로파일**을 선택하고 **새 스토리지 프로파일**을 클릭합니다.

### vRealize Automation의 스토리지 프로파일에 대해 알아보기

클라우드 계정 지역에는 스토리지 프로파일이 포함됩니다. 이를 통해 클라우드 관리자가 vRealize Automation에서 해당 지역의 스토리지를 정의할 수 있습니다.

스토리지 프로파일에는 디스크 사용자 지정 정보 및 기능 태그를 통해 스토리지 유형을 식별할 수 있는 방법이 포함됩니다. 이러한 태그는 배포 시 원하는 스토리지를 생성하기 위해 프로비저닝 서비스 요청 조건과 일치됩니다.

스토리지 프로파일은 클라우드별 지역에 구성됩니다. 클라우드 계정 하나에 여러 지역이 포함될 수 있으며, 각 지역에 스토리지 프로파일이 여러 개 포함될 수 있습니다.

벤더에 관계없이 스토리지 프로파일을 배치할 수도 있습니다. 예를 들어 서로 다른 벤더 계정 3개가 있고 각 계정에 지역이 1개 있다고 가정해 보겠습니다. 각 지역에는 "fast" 라는 기능 태그가 지정된 스토리지 프로파일이 포함되어 있습니다. 프로비저닝 시 하드 "fast" 조건 태그가 포함된 요청은 어느 벤더 클라우드가 리소스를 제공하는지에 관계 없이, 일치하는 "fast" 기능을 찾습니다. 이 일치 항목은 배포된 스토리지 항목을 생성하는 동안 연결된 스토리지 프로파일의 설정을 적용합니다.

---

**참고** 클라우드 스토리지마다 성능 특성이 다를 수 있지만 그래도 태그를 지정한 관리자가 제공하는 "fast" 오퍼링으로 간주됩니다.

---

스토리지 프로파일에 추가하는 기능 태그는 실제 리소스 대상을 식별하는 게 아니라 스토리지 유형을 설명합니다. 실제 리소스에 대한 자세한 내용은 [vRealize Automation의 스토리지 리소스](#) 항목을 참조하십시오.

vRealize Automation API를 사용하거나 스토리지 프로파일 페이지에서 **디스크 유형** 옵션을 사용하여 FCD(First Class Disk) 스토리지 또는 표준 디스크 스토리지를 지원하도록 스토리지 프로파일을 생성할 수 있습니다. FCD(First Class Disk) 옵션을 선택하면 vSphere 스토리지 프로파일을 효과적으로 생성할 수 있습니다.

#### ■ FCD(First Class Disk)

FCD(First Class Disk)는 vSphere VM과 독립적으로 생성하고 관리할 수 있습니다. FCD에는 수명 주기 관리 기능이 있으며 이 역시 VM과 독립적으로 작동합니다. FCD는 vSphere 버전 6.7 업데이트 2 이상에서 사용할 수 있으며 현재 vRealize Automation에 API 전용 기능으로 구현되어 있습니다.

vRealize Automation API를 사용하여 사용할 수 있는 기능 및 API 설명서 자체에 대한 링크를 포함한 FCD(First Class Disk) 스토리지에 대한 자세한 내용은 [vRealize Automation에서 FCD\(First Class Disk\) 스토리지로 수행할 수 있는 작업](#)에서 참조하십시오.

#### ■ 표준 디스크

표준 디스크 스토리지는 VM의 통합 구성 요소로 생성 및 관리됩니다.

표준 디스크 스토리지에 대한 자세한 내용은 [vRealize Automation에서 표준 디스크 스토리지로 수행할 수 있는 작업](#) 및 [vRealize Automation에서 영구 디스크 스토리지로 수행할 수 있는 작업](#)에서 참조하십시오.

## vRealize Automation에서 가격 책정 카드를 사용하는 방법

vRealize Automation Cloud Assembly 가격 책정 카드는 클라우드 관리자가 리소스를 관리할 수 있도록 개별 배포의 금전적 영향에 대한 가격 책정 정책을 정의하고 할당하는 데 도움을 줍니다.

---

**참고** 다중 테넌트 환경에서 가격 책정이 작동하려면 각 vRealize Automation 테넌트에 대해 별도의 vRealize Operations Manager 인스턴스가 있어야 합니다.

---

가격 책정 카드는 가격 책정 정책에 대한 요금을 정의합니다. 그런 다음 가격 책정 정책을 특정 프로젝트에 할당하여 총 가격을 정의할 수 있습니다. vRealize Operations Manager 또는 CloudHealth 끝점을 생성한 후 미리 정의된 기본 요금 카드를 **인프라 > 가격 책정 카드** 탭의 가격 구성과 동일한 비용으로 사용할 수 있습니다. 프로젝트에만 또는 클라우드 영역에 적용되는 가격 책정 카드를 생성할 수 있습니다. 기본적으로 모든 새 가격 책정 카드는 프로젝트에 적용됩니다.

---

**참고** 모든 가격 책정 카드의 적용 대상 설정을 변경하는 경우 기존의 모든 가격 책정 카드 할당이 삭제됩니다. 또한 vRealize Operations Manager 끝점이 vRealize Automation Cloud Assembly에서 삭제되면 모든 가격 책정 카드 및 할당도 삭제됩니다.

---

시간 경과에 따른 배포 가격은 배포 카드 및 프로젝트에 월간 누계 가격으로 표시되며 매월 초에 0으로 재설정됩니다. 구성 요소 비용 명세는 배포 세부 정보에서 사용할 수 있습니다. 배포 수준에서 이 정보를 제공하면 클라우드 관리자가 이 내용을 알 수 있으며 멤버 역시 자신들의 작업이 예산 및 장기적인 개발에 미칠 수 있는 영향을 이해할 수 있습니다.

**가격 책정 정보 표시** 버튼을 선택하여 vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker 사용자의 가격 책정 정보를 표시하도록 선택할 수 있습니다. 사용하지 않도록 설정한 상태로 두면 가격 책정 정보가 vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker 사용자에게 표시되지 않습니다.

## 가격 계산 방법

계산 및 스토리지 리소스에 대한 배포 수준에서 표시되는 초기 가격은 업계 표준 벤치마크 요금에 기반하며 이후 시간 경과에 따라 계산됩니다. 요금은 호스트에 적용되고 서비스는 CPU 및 메모리 요금을 계산합니다. 서버는 6시간마다 가격을 다시 계산합니다.

새 정책, 할당 및 선행 가격 책정은 다음에 발생하는 데이터 수집 주기 동안 가격이 책정됩니다. 기본적으로 데이터 수집 주기는 5분마다 실행됩니다. 새 정책 또는 변경 내용이 프로젝트 및 배포에서 업데이트되려면 최대 6시간이 걸릴 수 있습니다.

## 내 배포 및 프로젝트의 가격을 예상하는 방법

카탈로그 항목을 배포하기 전에 선행 가격을 배포에 대한 가격 예상으로 사용할 수 있습니다. vRealize Automation Cloud Assembly에서 가격을 보려면 가격 책정이 가능하도록 구성되고 통화가 미리 설정된 vRealize Operations Manager 통합 끝점이 있어야 합니다.

### Daily Price Estimate



Guest OS and one time prices are excluded in this estimate.



price-service-f309c00

\$0.54



Cloud\_vSphere\_Machine\_1

\$0.53

Compute

\$0.39

Storage

\$0.03

Additional charges

\$0.11



Cloud\_vSphere\_Disk\_1

\$0.01

Storage

\$0.01

CLOSE

선행 가격 예상의 경우, VM당 부팅 디스크 크기는 항상 8GB입니다.

배포에 대한 선행 가격은 지정된 카탈로그 항목(배포되기 전)에 대한 리소스 할당을 기반으로 하는 일별 가격 예상입니다. 카탈로그 항목이 배포된 후에는 **배포 및 인프라 > 프로젝트** 탭에서 선행 가격의 집계로 월간 누계 가격을 볼 수 있습니다. 선행 가격 책정은 vSphere 시스템 및 vSphere 디스크와 같은 사설 클라우드 리소스, vRealize Automation Cloud Assembly 카탈로그 항목 및 사설 클라우드에 대해 구성된 vCenter를 사용하는 클라우드를 애그노스틱 항목에 대해 지원됩니다.

**참고** 공용 클라우드 리소스 또는 비 vSphere 시스템 또는 디스크 사설 클라우드 리소스에는 선행 가격 책정이 지원되지 않습니다.

배포 비용을 예측하려면 [카탈로그]에서 카탈로그 항목을 선택하고 **요청 > 계산**을 클릭합니다. 가격이 적절한 경우 **제출**을 클릭합니다.

프로젝트 가격 책정 카드를 사용하여 모든 프로젝트의 총 가격을 예상할 수 있습니다.

프로젝트의 비용을 예상하려면 인프라 가격 책정 카드 페이지의 **모든 가격 책정 카드의 적용 대상** 옆에서 **편집**을 클릭하고 **프로젝트**를 선택합니다.

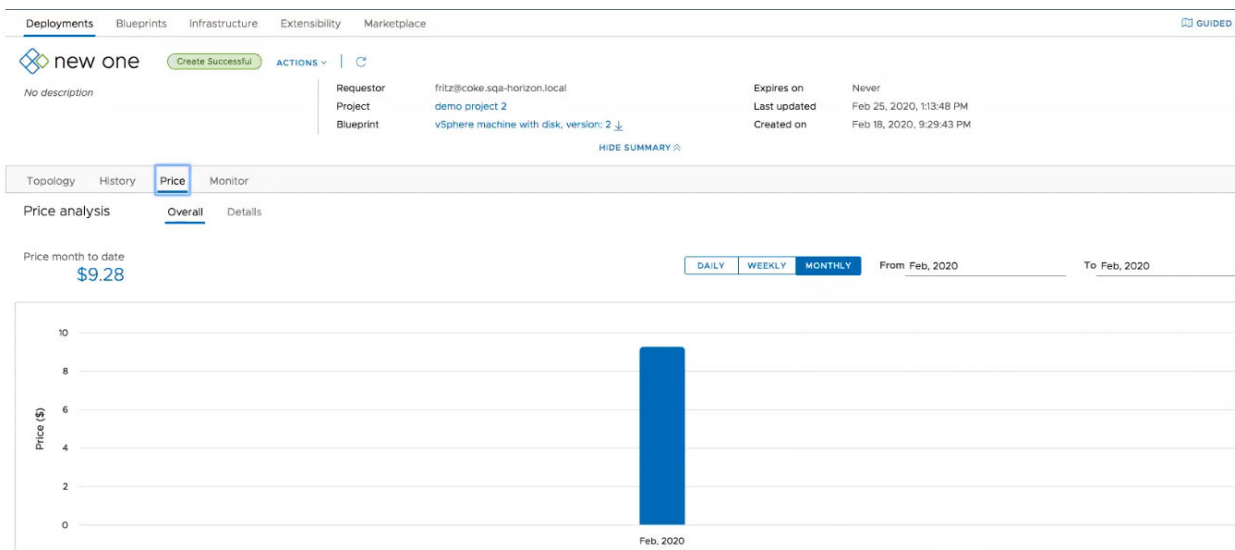
**모든 가격 책정 카드의 적용 대상** 설정을 변경하는 경우 기존의 모든 가격 책정 카드 할당이 삭제됩니다. 비용 기반 접근 방식을 사용하여 가격 책정 카드 및 할당을 생성합니다.

## vSphere 및 VMC에 대한 가격 책정 카드를 생성하는 방법

사설 클라우드 배포를 위해 클라우드 관리자가 정한 가격 책정 전략에 따라 가격 책정 카드를 프로젝트 또는 클라우드 영역에 생성하고 할당할 수 있습니다.

가격 책정 카드는 사용자가 선택한 매개 변수를 기반으로 사용자 지정할 수 있습니다. 가격 책정 카드를 구상한 후에는 가격 책정 전략에 따라 결정된 하나 이상의 프로젝트 및 클라우드 영역에 할당할 수 있습니다.

[vROps 끝점] 페이지, **인프라 > 통합 > vROps 끝점 >**에서 언제든지 가격 서버를 수동으로 새로 고칠 수도 있습니다. vCenter Servers 섹션에서 **동기화**를 클릭합니다. **동기화** 옵션을 사용하여 가격 서버를 수동으로 새로 고치면 조직의 모든 프로젝트에 대해 가격이 다시 계산됩니다. 조직의 프로젝트 수에 따라 이 프로세스가 많고 시간이 오래 걸릴 수 있습니다.



가격 책정 카드를 생성하고 할당한 후에는 배포 및 프로젝트의 가격 기록을 볼 수 있습니다. 가격 기록을 보려면 배포로 이동한 후 **가격**을 클릭합니다. 가격 분석은 배포 가격에 대한 개요 및 세부 보기를 가격 월별 누계 값과 함께 제공합니다. 그래픽 표현을 변경하여 배포 가격을 일별, 주별 또는 월별 값으로 표시할 수 있습니다. 또한 가격 기록에 대해 정확한 날짜 범위 또는 월을 지정할 수 있습니다.

비용 구성 요소별 가격 분석을 보려면 **세부 정보**를 클릭합니다.

가격은 비용이 계산된 구성 요소 유형에 따라 결정됩니다.

**표 4-1. 비용 계산된 구성 요소 유형**

Blueprint 구성 요소 유형	서비스 이름/개체 유형	Blueprint 리소스 유형	설명
클라우드 애그노스틱	시스템	Cloud.Machine	애그노스틱 시스템이 vSphere로 구성된 경우 배포 비용을 볼 수 있습니다.
	디스크	Cloud.Volume	애그노스틱 디스크가 vSphere로 구성된 가상 시스템에 연결되어 있는 경우 배포 비용을 볼 수 있습니다.
vSphere	vSphere 시스템	Cloud.vSphere.Machine	클라우드별 Blueprint를 사용하여 배포됩니다.
	vSphere 디스크	Cloud.vSphere.Disk	가상 시스템에 연결된 클라우드별 Blueprint를 사용하여 배포됩니다.
VMC(VMware 관리 클라우드)	vSphere 시스템	Cloud.vSphere.Machine	VMC는 요금 기반 가격 책정 카드만 지원합니다(비용 기반 가격 책정 카드는 지원되지 않음).
	vSphere 디스크	Cloud.vSphere.Disk	

## 사전 요구 사항

가격 책정 카드를 생성하거나 할당하려면 먼저 가격 책정을 구성하고 사용하도록 설정하고 vRealize Operations에서 vRealize Automation 과 함께 작동하도록 통화를 구성해야 합니다. vRealize Automation 으로 vRealize Operations을 구성하는 경우에는 두 애플리케이션을 동일한 표준 시간대로 설정해야 합니다. vRealize Operations에서 표준 시간대를 구성하려면 SSH를 사용하도록 설정하고 각 vRealize Operations 노드에 로그인한 후 `$ALIVE_Base/user/conf/analytics/advanced.properties` 파일을 편집하고 `timeZoneUseInMeteringCalculation = <time zone>`을 추가합니다.

다중 테넌트 환경에서 가격이 책정을 수행하려면 각 vRA 테넌트에 대해 별도의 vROps 인스턴스가 있어야 합니다.

가격 책정 카드를 구성하려면 먼저 vRealize Operations 끝점을 구성해야 합니다. vRealize Operations 끝점을 구성하려면 **인프라 > 연결 > 통합 > 통합 추가**로 이동합니다.

**참고** 여러 vRealize Operations 끝점이 추가되면 동일한 vCenter를 모니터링하지 않아야 합니다.

## 절차

- 1 **인프라 > 가격 책정 카드 > 새 가격 책정 카드**로 이동합니다.
- 2 [요약] 탭에서 가격 책정 카드에 대한 이름과 설명을 입력합니다. 정책이 가격 책정 탭에 정의되면 개요 테이블이 가격 책정 카드 요금으로 채워집니다.

**참고** 통화 단위는 vRealize Operations에서 선택한 값에 따라 결정됩니다.

- 3 선택 사항. **할당되지 않은 프로젝트에 대한 기본값으로 지정** 확인란을 선택하여 기본적으로 이 가격 책정 카드를 모든 할당되지 않은 프로젝트에 할당합니다.
- 4 **가격 책정**을 클릭하고 가격 책정 정책의 세부 정보를 구성합니다.

표 4-2. 가격 책정 정책 구성

매개 변수	설명
기본 청구	<p>정책의 이름과 설명을 입력합니다. 비용 또는 요금 기반을 선택합니다.</p> <ul style="list-style-type: none"> <li>■ <b>비용</b> - 비용은 vRealize Operations에 정의되어 있습니다. 선택하는 경우 증배 계수가 필요합니다. 예를 들어 1.1을 계수로 선택하면 비용에 1.1을 곱하여 계산된 비용에 10%가 증가합니다. 비용을 사용하는 가격 등식은 &lt;비용&gt; x &lt;증배 계수&gt; = 가격입니다.</li> <li>■ <b>요금</b> - 선택하는 경우에는 절대 값을 사용하여 비용을 결정해야 합니다. 요금을 사용하는 가격 등식은 &lt;요금&gt; = 가격입니다. 드롭다운 목록에서 요금 간격을 선택하여 이 요금을 청구할 방법을 지정합니다.</li> </ul> <p>기본 청구 섹션에서는 CPU, 메모리, 스토리지 및 추가 기타 비용에 대한 비용 또는 요금을 정의합니다.</p>
게스트 운영 체제	<p><b>요금 추가</b>를 클릭하여 게스트 OS 요금을 정의할 수 있습니다.</p> <p>게스트 OS 이름을 입력하고 청구 방법 및 기본 요금을 정의합니다.</p> <ul style="list-style-type: none"> <li>■ <b>되풀이</b> - 기본 요금을 입력하고 되풀이 간격을 청구 기간으로 정의합니다. 절대 요금 값은 필수이며 전체 가격에 추가됩니다.</li> <li>■ <b>일회성</b> - 일회성 기본 요금 청구를 정의합니다. 절대값은 필수이며 일회성 가격으로 추가됩니다.</li> <li>■ <b>요금 계수</b> - 특정 청구 범주에 적용되는 증배 계수가 필요합니다. CPU 요금 및 요금 계수 2를 선택하는 경우를 예로 들 수 있습니다. 게스트 OS CPU는 표준 비용 값의 2배만큼 청구됩니다.</li> </ul> <p><b>요금 추가</b>를 클릭하고 추가 요금 정책을 구성하여 다른 요금으로 여러 게스트 운영 체제를 추가할 수 있습니다.</p> <p><b>참고</b> 게스트 운영 체제에 대한 선행 요금은 정책에 포함된 경우에도 요약 페이지에 표시되지 않습니다.</p>

표 4-2. 가격 책정 정책 구성 (계속)

매개 변수	설명
태그	<p><b>요금 추가</b>를 클릭하여 태그 요금을 정의할 수 있습니다.</p> <p>태그 이름을 선택하고 청구 방법 및 기본 요금을 정의합니다.</p> <ul style="list-style-type: none"> <li>■ <b>되풀이</b> - 기본 요금을 입력하고 되풀이 간격을 청구 기간으로 정의합니다. 절대 요금 값은 필수이며 전체 가격에 추가됩니다.</li> <li>■ <b>일회성</b> - 일회성 기본 요금 청구를 정의합니다. 절대값은 필수이며 일회성 가격으로 추가됩니다.</li> <li>■ <b>요금 계수</b> - 특정 청구 범주에 적용되는 증배 계수가 필요합니다.</li> </ul> <p>전원이 켜진 상태에 기반하여 태그를 청구하는 방법을 선택합니다.</p> <p><b>요금 추가</b>를 클릭하고 추가 요금 정책을 구성하여 다른 요금으로 여러 태그를 추가할 수 있습니다.</p> <hr/> <p><b>참고</b> 계산된 최종 가격의 추가 요금은 VM의 태그에 포함되며 디스크 및 네트워크의 태그는 포함되지 않습니다.</p>
사용자 지정 속성	<p><b>요금 추가</b>를 클릭하여 사용자 지정 속성 요금을 정의할 수 있습니다.</p> <p>속성 이름과 값을 입력하고 청구 방법 및 기본 요금을 정의합니다.</p> <ul style="list-style-type: none"> <li>■ <b>되풀이</b> - 기본 요금을 입력하고 되풀이 간격을 청구 기간으로 정의합니다. 절대 요금 값은 필수이며 전체 가격에 추가됩니다.</li> <li>■ <b>일회성</b> - 일회성 기본 요금 청구를 정의합니다. 절대값은 필수이며 일회성 가격으로 추가됩니다.</li> <li>■ <b>요금 계수</b> - 특정 청구 범주에 적용되는 증배 계수가 필요합니다.</li> </ul> <p>전원이 켜진 상태를 기반으로 사용자 지정 속성을 청구하는 방법을 선택합니다.</p> <p><b>요금 추가</b>를 클릭하고 추가 요금 정책을 구성하여 다른 요금으로 여러 사용자 지정 속성을 추가할 수 있습니다.</p>
전체 요금	<p>가격 책정 정책에 추가하고자 하는 추가 요금을 정의합니다.</p> <p>일회성 및 되풀이 요금을 모두 추가할 수 있습니다.</p>

일회성 요금은 카탈로그 항목 또는 요약 탭의 가격 예상에 표시되지 않습니다. 지정된 카탈로그 항목에 대한 일별 가격 예상만 표시됩니다.

- 할당** 탭을 클릭하고 **프로젝트 할당**을 클릭합니다. 가격 책정 카드를 할당할 프로젝트를 하나 이상 선택합니다.

**참고** 기본적으로 가격 책정 카드는 프로젝트에 적용됩니다. **인프라 > 가격 책정 카드** 탭에서 가격 책정 카드를 클라우드 영역에 추가하도록 선택할 수 있습니다. 클라우드 영역을 선택한 경우 [할당] 탭에서 **클라우드 영역 할당**을 클릭합니다.

- 생성**을 클릭하여 가격 책정 정책을 저장하고 생성합니다.

## 결과

새 가격 책정 정책이 [가격 책정 카드] 페이지에 표시됩니다. 정책 세부 정보 및 구성을 보거나 편집하려면 열기를 클릭합니다.

## 태그를 사용하여 vRealize Automation Cloud Assembly 리소스 및 배포를 관리하는 방법

태그는 기능 및 제약 조건의 일치를 통해 배포의 배치를 적용하는 vRealize Automation Cloud Assembly의 주요 구성 요소입니다. vRealize Automation Cloud Assembly를 최적화된 방식으로 사용하려면 태그를 이해하고 효과적으로 구현해야 합니다.

기본적으로 태그는 vRealize Automation Cloud Assembly 항목에 추가하는 레이블입니다. 조직과 구현 내용에 적절한 모든 태그를 생성할 수 있습니다. 태그는 vRealize Automation Cloud Assembly가 리소스와 인프라를 사용하여 배포 가능한 서비스를 구축하는 방식과 위치를 제어하므로 레이블 이상의 기능을 합니다. 또한 태그는 Cloud Assembly 내의 거버넌스를 지원합니다.

### 태그 구조

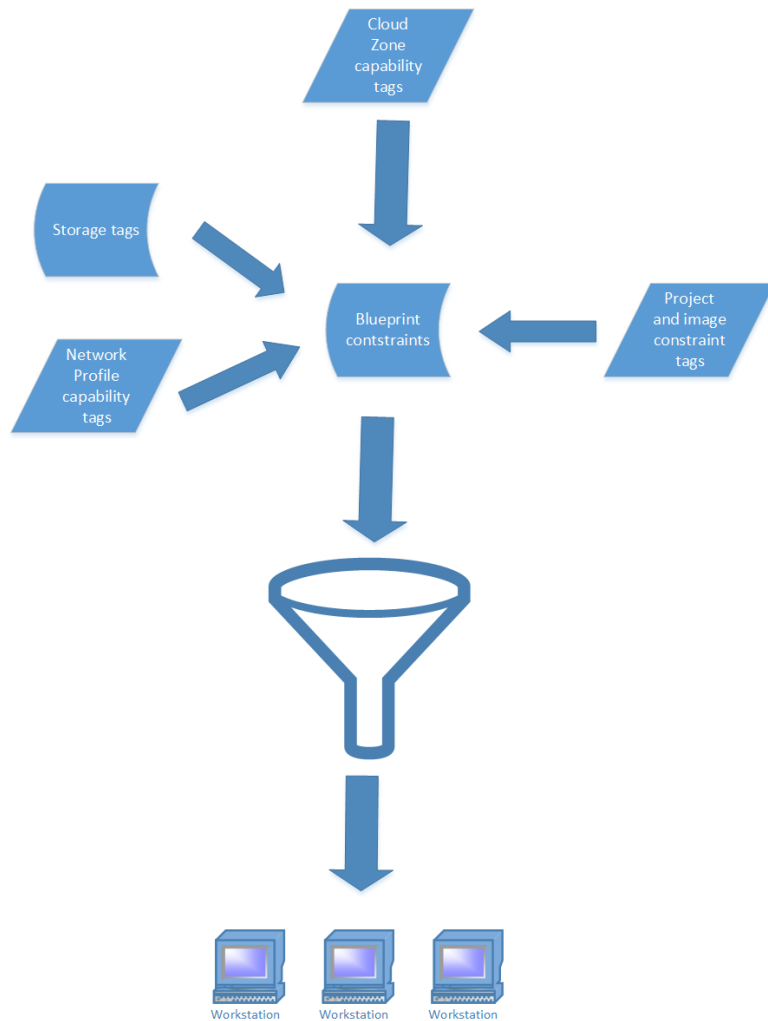
구조적으로 태그는 name:value 쌍 규칙을 따라야 하지만 그렇지 않은 경우 태그의 구조는 대부분 자유 형식입니다. vRealize Automation Cloud Assembly 전체에서 모든 태그는 동일하게 나타나고 태그 기능은 컨텍스트에 의해 결정됩니다.

예를 들어 인프라 리소스의 태그는 주로 기능 태그로 작동합니다. 그 이유는 vRealize Automation Cloud Assembly가 이러한 태그를 사용하여 리소스와 배포를 매칭하기 때문입니다. 부수적으로는 리소스도 식별합니다.

### 태그 기능

태그의 기본 기능은 vRealize Automation Cloud Assembly가 배포를 정의하는 데 사용하는 기능 및 제약 조건을 표현하는 것입니다. 컨텍스트는 태그의 기능을 결정합니다. 클라우드 영역, 네트워크 및 스토리지 프로파일, 개별 인프라 리소스에 배치된 태그는 기능 태그로 작동하며 배포에 사용된 인프라에 필요한 기능을 정의합니다. 클라우드 템플릿에 배치된 태그는 배포를 위한 리소스를 정의하는 제약 조건으로 작동합니다. 또한 클라우드 관리자는 프로젝트에 제약 조건 태그를 배치하여 해당 프로젝트에 대한 거버넌스의 한 형태를 이행할 수 있습니다. 이러한 제약 조건 태그는 클라우드 템플릿에 표시된 다른 제약 조건에 추가됩니다.

프로비저닝 중에, vRealize Automation Cloud Assembly는 이러한 기능을 클라우드 템플릿에서 태그로 표시된 제약 조건과 일치시켜 배포 구성을 정의합니다. 이 태그 기반 기능 및 제약 조건 기능은 vRealize Automation Cloud Assembly에서 배포 구성을 위한 기초 역할을 합니다. 예를 들어 태그를 사용하여 특정 지역의 PCI 리소스에서만 인프라를 사용할 수 있게 만들 수 있습니다.



또한 다른 차원에서 태그는 스토리지, 네트워크 항목 및 다른 인프라 리소스의 검색 및 식별을 용이하게 합니다.

예를 들어 클라우드 영역을 설정 중이고 여러 계산 리소스를 사용할 수 있다고 가정합니다. 이러한 계산 리소스에 대해 적절하게 태그를 지정했다면 [클라우드 영역] 페이지의 [계산] 탭에서 검색 기능을 사용하여 특정 클라우드 영역과 연결된 리소스를 필터링할 수 있습니다.

또한 Cloud Assembly [태그 관리] 페이지 및 리소스 구성 페이지에는 태그 이름으로 항목을 찾을 수 있는 검색 기능이 포함되어 있습니다. 이와 같은 검색 및 식별 기능을 용이하게 하려면 이러한 항목에 대해 논리적이고 사람이 읽을 수 있는 태그를 사용하는 것이 매우 중요합니다.

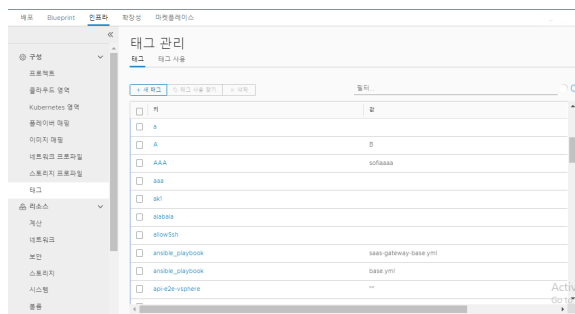
다음 Youtube 비디오에서 태그 사용에 대한 자세한 내용 및 예제를 살펴보십시오. <https://youtu.be/4zNQ33RyQio>

## 외부 태그

vRealize Automation Cloud Assembly는 외부 태그를 포함할 수도 있습니다. 이러한 태그는 vRealize Automation Cloud Assembly 인스턴스와 연결한 클라우드 계정에서 자동으로 가져옵니다. 이러한 태그를 vSphere, AWS, Azure 또는 기타 외부 소프트웨어 제품으로부터 가져올 수도 있습니다. 가져온 태그는 사용자가 생성한 태그와 동일한 방식으로 사용할 수 있습니다.

## 태그 관리

vRealize Automation Cloud Assembly에서 [태그 관리] 페이지를 사용하여 태그 라이브러리를 모니터링하고 관리할 수 있습니다. 이 페이지에서 태그를 생성할 수도 있습니다. 또한 [태그 관리] 페이지는 외부 태그를 보고 식별할 수 있는 유일한 페이지입니다.



## 태그 전략

혼란을 최소화하려면 vRealize Automation Cloud Assembly에서 태그를 생성하기 전에 적절한 태그 전략 및 태그 지정 규칙을 고안하여 태그를 생성하고 사용하는 모든 사용자가 태그의 의미와 태그가 사용되는 방식을 이해할 수 있도록 해야 합니다. **태그 지정 전략 생성** 항목을 참조하십시오.

## 태그 지정 전략 생성

Cloud Assembly 기능을 최대한 효과적으로 활용하고 잠재적인 혼란을 최소화하기 위해 조직의 IT 구조와 목표에 따라 적절한 태그 지정 전략을 신중하게 계획하고 구현해야 합니다.

태그 지정에는 몇 가지 공통된 목표가 있지만 배포 요구 사항, 구조 및 목표에 맞추어 적절한 태그 지정 전략을 생성해야 합니다.

## 태그 지정 모범 사례

효과적인 태그 전략의 몇 가지 일반적인 특성은 다음과 같습니다.

- 조직의 비즈니스 구조와 연관지어 일관된 태그 지정 전략을 설계 및 구현하고 모든 관련 사용자에게 이 계획을 전달합니다. 전략은 배포 요구 사항을 지원하며 명확하고 가독성 있는 언어를 사용하고 모든 관련 사용자가 이해할 수 있어야 합니다.
- 간단명료하고 알기 쉬운 이름 및 값을 태그에 사용합니다. 예를 들어 스토리지 및 네트워크 항목의 태그 이름이 명확하고 일관되어야 사용자가 배포된 리소스를 선택하거나 배포된 리소스에 대한 태그 할당을 검토할 때 바로 파악할 수 있습니다.

- 값 없이 이름만 사용하여 태그를 생성할 수도 있지만 모범 사례로는 다른 사용자가 태그 용도를 명확히 파악할 수 있도록 각 태그 이름에 관련 값을 지정하는 것이 더 좋습니다.
- 중복되거나 관련 없는 태그를 생성하지 마십시오. 예를 들어 스토리지 문제와 관련된 스토리지 항목에만 태그를 생성합니다.

## 태그 지정 구현

기본적인 태그 지정 전략을 위한 주요 고려 사항을 확인하십시오. 다음 목록은 전략을 계획할 때 고려해야 할 일반적인 고려 사항입니다. 이러한 고려 사항은 절대적인 기준이 아니라 대표적인 예입니다. 사용 사례와 관련된 다른 고려 사항이 있을 수 있으며, 구체적인 사용 사례에 맞는 전략을 계획해야 합니다.

- 배포할 환경 수. 일반적으로 각 환경을 나타내는 태그를 생성합니다.
- 배포를 지원하기 위해 계산 리소스를 구성하고 사용하는 방식
- 배포할 영역 또는 위치 수. 일반적으로 각 영역 또는 위치를 나타내는 프로파일 수준에서 태그를 생성합니다.
- 배포에 사용할 수 있는 스토리지 옵션 수 및 이러한 옵션을 구분하는 방법. 이러한 옵션은 태그로 나타내야 합니다.
- 네트워킹 옵션을 범주화하고 모든 해당 옵션에 대한 태그를 생성합니다.
- 일반적인 배포 변수. 예를 들어 배포할 환경 수를 고려해야 합니다. 일반적으로 많은 조직이 최소한 테스트, 개발 및 운영 환경을 두고 있습니다. 하나 이상의 이러한 환경에 배포를 쉽게 구성할 수 있도록 제약 조건 태그와 클라우드 영역 기능 태그를 생성하고 조정하는 것이 좋습니다.
- 사용되는 네트워크 및 스토리지 프로파일의 컨텍스트에 맞게 네트워크 및 스토리지 리소스에 대한 태그를 조정합니다. 리소스 태그를 사용하면 리소스 배포를 더 세부적으로 제어할 수 있습니다.
- 클라우드 영역 및 네트워크 프로파일 기능 태그와 기타 기능 태그와 제약 조건 태그를 상호 조정합니다. 대개는 관리자가 먼저 클라우드 영역 및 네트워크 프로파일에 대한 기능 태그를 생성하면 다른 사용자가 이러한 기능 태그와 일치하는 제약 조건을 사용하여 클라우드 템플릿을 설계할 수 있습니다.

조직에 중요한 고려 사항을 파악한 후에 이러한 고려 사항을 해결하는 적절한 태그 이름을 체계적으로 계획할 수 있습니다. 그런 다음 전략의 개요를 작성하여 태그를 생성하고 편집할 수 있는 권한을 가진 모든 사용자에게 제공합니다.

유용한 구현 방식으로, 일단 모든 계산 인프라 리소스에 개별적으로 태그를 지정할 수 있습니다. 앞서 언급한 대로 특정 리소스와 연관된 논리적 범주를 태그 이름에 사용합니다. 예를 들어 스토리지 리소스에 대한 태그를 tier1, tier2 등으로 지정할 수 있습니다. 또한 계산 리소스에 대한 태그는 Windows, Linux 등과 같은 해당 운영 체제를 기반으로 지정할 수 있습니다.

리소스에 태그를 지정한 후에는 요구 사항에 가장 잘 맞는 방식으로 클라우드 영역과 스토리지 및 네트워크 프로파일에 대한 태그를 생성할 수 있습니다.

## vRealize Automation Cloud Assembly에서 기능 태그 사용

vRealize Automation Cloud Assembly에서 기능 태그를 사용하면 인프라 구성 요소의 배포 기능을 정의할 수 있습니다. 기능 태그는 제약 조건과 함께 vRealize Automation에서 배치 논리의 기초로 작동합니다.

계산 리소스, 클라우드 영역, 이미지와 이미지 맵, 네트워크와 네트워크 프로파일에 대한 기능 태그를 생성할 수 있습니다. 이러한 리소스를 생성하기 위한 페이지에는 기능 태그 생성을 위한 옵션이 포함되어 있습니다. vRealize Automation Cloud Assembly의 [태그 관리] 페이지를 사용하여 기능 태그를 생성할 수도 있습니다. 클라우드 영역과 네트워크 프로파일의 기능 태그는 해당 영역 또는 프로파일 내의 모든 리소스에 영향을 줍니다. 스토리지 또는 네트워크 구성 요소의 기능 태그는 해당 태그가 적용되는 구성 요소에만 영향을 줍니다.

일반적으로 기능 태그는 계산 리소스의 위치, 네트워크의 어댑터 유형 또는 스토리지 리소스의 계층 수준과 같은 특징을 정의할 수 있습니다. 이 밖에 환경 위치 또는 유형과 기타 비즈니스 고려 사항도 정의할 수 있습니다. 전반적인 태그 지정 전략과 마찬가지로 기능 태그는 비즈니스 요구에 맞게 논리적 방식으로 구성해야 합니다.

vRealize Automation Cloud Assembly는 배포 시 클라우드 템플릿의 제약 조건과 클라우드 영역의 기능 태그를 일치시킵니다. 따라서 기능 태그를 생성하고 사용할 때에는 일치가 예상대로 수행되도록 적절한 클라우드 템플릿 제약 조건을 이해하고 생성하도록 계획해야 합니다.

예를 들어, 설명서에 포함된 **1부: 예제 vRealize Automation Cloud Assembly 인프라 구성**의 클라우드 영역 섹션은 OurCo-AWS-US-East 및 OurCo AWS-US-West 클라우드 영역에 대한 개발 및 테스트 태그를 생성하는 방법을 설명합니다. 이 자습서에서 이러한 태그는 OurCo-AWS-US-East 영역이 개발 환경이고 OurCo-AWS-US-West 영역이 테스트 환경임을 나타냅니다. 클라우드 템플릿에서 유사한 제약 조건 태그를 생성하는 경우 이러한 기능 태그를 사용하여 원하는 환경으로 배포를 지시할 수 있습니다.

## 태그 상속

Cloud Assembly는 태그 상속을 사용하여 클라우드 계정의 태그를 다른 관련 리소스에 선택적으로 전파합니다. 특히 클라우드 계정에 태그를 생성하면 이 클라우드 계정에 해당하는 모든 스토리지 프로파일 및 계산 리소스에도 적용됩니다.

**참고** 태그 전파 동작은 스토리지 프로파일에 적용되지 않습니다. vRealize Automation는 스토리지 프로파일에 대한 제약 조건을 자동으로 선택하지 않으므로 선택 및 스토리지 프로파일에 적용하는 데 필요한 제약 조건 태그를 사용자가 수동으로 추가해야 합니다.

다음 예는 태그 상속이 작동하는 방식을 보여줍니다.

계산 리소스

- cluster-1 태그가 있는 Cluster1
- cluster-2 태그가 있는 Cluster2
- cluster-3 태그가 있는 Cluster3

```
Vm resource:
  properties:
    constraints:
      - tag: 'cluster-01'
```

스토리지 프로파일

- storage-01 태그가 있는 Datastorecluster1의 프로파일 1

- storage-02 태그가 있는 Datastorecluster2의 프로파일 2
- storage-03 태그가 있는 Datastorecluster3의 프로파일 3

```
vm-resource:
  properties:
    storage:
      constraints:
        - tag: 'storage-01'
```

## 클라우드 계정

세 가지 태그(cluster-1, cluster-2 및 cluster-3)가 모두 있는 vSphere 클라우드 계정

스토리지 프로파일과 계산 리소스에 대한 태그를 통합하는 동안 Cloud Assembly는 클라우드 계정 수준 태그도 고려합니다. 모든 스토리지 프로파일 및 계산에 대한 유효 태그는 cluster-1, cluster-2, cluster-3이며, 따라서 앞의 예제에 표시된 대로 이러한 태그를 제공하면 모든 스토리지 프로파일과 계산이 배치에 적합한 상태가 되고 시스템이 임의의 계산 호스트에 배치될 수 있습니다.

예기치 않은 결과와 태그 혼란을 최소화하려면, 모든 하위 계산 및 스토리지 리소스에 대해 적합한 기능이 있는 태그는 클라우드 계정 수준에서만 적용하는 것이 가장 좋습니다.

## vRealize Automation Cloud Assembly에서 제약 조건 태그 사용

프로젝트 및 클라우드 템플릿에 추가된 태그는 인프라 리소스, 프로 파일 및 클라우드 영역의 기능 태그를 일치시키는 데 사용될 때 제약 조건 태그로 작동합니다. 클라우드 템플릿의 경우 vRealize Automation Cloud Assembly는 이 일치 기능을 사용하여 배포를 위해 리소스를 할당합니다.

vRealize Automation Cloud Assembly를 사용하면 두 가지의 기본적인 방법으로 제약 조건 태그를 사용할 수 있습니다. 첫 번째는 프로젝트 및 이미지를 구성할 때입니다. 태그를 제약 조건으로 사용하여 리소스를 프로젝트 또는 이미지와 연결할 수 있습니다. 두 번째는 클라우드 템플릿에서 제약 조건으로 지정된 태그를 사용하여 배포를 위한 리소스를 선택하는 것입니다. 이러한 방법에 모두 적용되는 제약 조건은 클라우드 템플릿에 병합되어 배포에 사용 가능한 리소스를 정의하는 배포 요구 사항 집합을 구성합니다.

## 프로젝트에서 제약 조건 태그가 작동하는 방식

클라우드 관리자는 vRealize Automation Cloud Assembly 리소스를 구성할 때 프로젝트에 제약 조건 태그를 적용할 수 있습니다. 이러한 방식으로 관리자는 프로젝트 수준에서 거버넌스 제약 조건을 직접 적용할 수 있습니다. 이 수준에서 추가된 모든 제약 조건은 적용 가능한 프로젝트에 대해 요청된 모든 클라우드 템플릿에 적용되며 이러한 제약 조건 태그는 다른 태그보다 우선합니다.

프로젝트의 제약 조건 태그와 클라우드 템플릿의 제약 조건 태그가 충돌하는 경우에는 프로젝트 태그가 우선하므로 클라우드 관리자는 거버넌스 규칙을 적용할 수 있습니다. 예를 들어 클라우드 관리자가 프로젝트에서 location:london 태그를 생성했지만 개발자가 클라우드 템플릿에 location:boston 태그를 배치한 경우 전자가 후자에 우선하기 때문에 리소스는 location:london 태그를 포함하는 인프라에 배포됩니다.

프로젝트에 최대 3개의 제약 조건을 적용할 수 있습니다. 프로젝트 제약 조건은 경성 또는 연성입니다. 기본적으로 제약 조건은 경성입니다. 경성 제약 조건을 사용하면 배포 제한을 엄격하게 적용할 수 있습니다. 경성 제약 조건이 하나 이상 충족되지 않으면 배포가 실패합니다. 연성 제약 조건은 사용 가능한 경우에 선택되는 기본 설정을 나타내는 방식으로 제시되지만 연성 제약 조건이 충족되지 않아도 배포는 실패하지 않습니다.

## 클라우드 템플릿에서 제약 조건 태그가 작동하는 방식

클라우드 템플릿에서는 제약 조건 태그를 리소스에 YAML 코드로 추가하여 클라우드 관리자가 리소스, 클라우드 영역 및 스토리지, 네트워크 프로파일에서 생성한 적절한 기능 태그를 일치시킵니다. 또한 제약 조건 태그를 구현하기 위한 더 복잡한 다른 옵션이 있습니다. 예를 들어 요청 시 변수를 사용하여 하나 이상의 태그를 채울 수 있습니다. 이를 통해 요청 시간에 하나 이상의 태그를 지정할 수 있습니다.

클라우드 템플릿 YAML 코드에서 제약 조건 머리글 아래의 tag 레이블을 사용하여 제약 조건 태그를 생성합니다. 프로젝트의 제약 조건 태그가 클라우드 템플릿에서 생성된 제약 조건 태그에 추가됩니다.

vRealize Automation Cloud Assembly는 YAML 파일에서 더 쉽게 제약 조건을 사용하도록 간단한 문자열 형식을 지원합니다.

```
[!]tag_key[:tag_value][:hard|:soft]
```

기본적으로 vRealize Automation Cloud Assembly는 경성 적용이 포함된 엄격한 제약 조건을 생성합니다. 태그 값은 애플리케이션의 나머지 부분에서와 마찬가지로 권장되지만 선택적입니다.

다음 "WordPress with MySQL" 예제는 계산 리소스에 대한 특정 위치 정보를 나타내는 YAML 제약 조건 태그를 보여 줍니다.

```
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 ...
```

클라우드 템플릿 사용 방법에 대한 자세한 내용은 [3부: 예제 vRealize Automation Cloud Assembly 템플릿 설계 및 배포 항목](#)을 참조하십시오.

## 프로젝트와 클라우드 템플릿에서 경성 및 연성 제약 조건이 작동하는 방식

프로젝트와 클라우드 템플릿의 제약 조건은 모두 경성이거나 연성일 수 있습니다. 앞의 코드 조각은 경성 및 연성 제약 조건의 예를 보여 줍니다. 기본적으로 모든 제약 조건은 경성입니다. 경성 제약 조건을 사용하면 배포 제한을 엄격하게 적용할 수 있습니다. 경성 제약 조건이 하나 이상 충족되지 않으면 배포가 실패합니다. 연성 제약 조건은 사용 가능한 경우에 적용되는 기본 설정을 나타내지만 조건이 충족되지 않아도 배포가 실패하지 않습니다.

특정 리소스 유형에 대해 일련의 경성 및 연성 제약 조건이 있는 경우 연성 제약 조건이 '결정자' 역할을 할 수도 있습니다. 즉, 여러 리소스가 하나의 경성 제약 조건을 충족하는 경우 배포에 사용될 실제 리소스를 선택하는 데 연성 제약 조건이 사용됩니다.

예를 들어 네트워크, 스토리지 및 확장성 항목의 조합으로 프로젝트에 최대 3개의 제약 조건을 지정할 수 있습니다. 또한 각 제약 조건이 경성인지 아니면 연성인지 선택할 수 있습니다. `location:boston` 태그를 사용하여 경성인 스토리지 제약 조건을 생성했다고 가정합니다. 프로젝트에 이 제약 조건에 일치하는 스토리지가 없다면 관련된 모든 배포가 실패합니다.

## 표준 태그

vRealize Automation Cloud Assembly는 배포된 리소스에 대해 분석, 모니터링 및 그룹화를 지원하기 위해 일부 배포에 표준 태그를 적용합니다.

표준 태그는 vRealize Automation Cloud Assembly 내에서 고유합니다. 다른 태그와 달리 표준 태그는 사용자가 배포를 구성하는 동안 사용하지 않으며, 제약 조건도 적용되지 않습니다. 이러한 태그는 AWS, Azure 및 vSphere 배포를 프로비저닝하는 동안 자동으로 적용됩니다. 표준 태그는 시스템 사용자 지정 속성으로 저장되며 프로비저닝 이후에 배포에 추가됩니다.

다음은 표준 태그 목록입니다.

**표 4-3. 표준 태그**

설명	태그
조직	<code>org:orgID</code>
프로젝트	<code>project:projectID</code>
요청자	<code>requester:username</code>
배포	<code>deployment:deploymentID</code>
클라우드 템플릿 참조(해당되는 경우)	<code>Blueprint:blueprintID</code>
Blueprint의 구성 요소 이름	<code>blueprintResourceName:CloudMachine_1</code>
배치 제약 조건: Blueprint 및 요청 매개 변수에 적용 또는 IT 정책을 통해 적용	<code>constraints:key:value:soft</code>
클라우드 계정	<code>cloudAccount:accountID</code>
영역 또는 프로파일(해당하는 경우)	<code>zone:zoneID, networkProfile:profileID, storageProfile:profileID</code>

## vRealize Automation Cloud Assembly가 태그를 처리하는 방법

vRealize Automation Cloud Assembly에서 태그는 프로비저닝 과정에서 리소스가 프로비저닝된 배포에 할당되는 방법과 위치를 결정하는 기능 및 제약 조건을 표시합니다.

vRealize Automation Cloud Assembly는 프로비저닝된 배포를 생성할 때 특정 순서와 작업의 계층을 사용하여 태그를 확인합니다. 이 프로세스의 기본적인 사항을 이해하면 효율적으로 태그를 구현하여 예측 가능한 배포를 생성할 수 있습니다.

다음 목록에서는 Cloud Assembly에서 태그를 확인하고 배포를 정의하는 데 사용하는 개략적인 작업 및 시퀀스를 요약합니다.

- 클라우드 영역은 가용성 및 프로파일을 비롯한 여러 기준으로 필터링됩니다. 이때 클라우드 영역이 속한 영역의 프로파일에 포함된 태그를 일치시킵니다.
- 영역 및 계산 기능 태그는 고정 제약 조건으로 나머지 클라우드 영역을 필터링하는 데 사용됩니다.
- 필터링된 영역 중에서는 우선 순위에 따라 클라우드 영역이 선택됩니다. 우선 순위가 동일한 여러 클라우드 영역이 있는 경우 클라우드 영역과 계산 기능을 조합한 유동 제약 조건을 일치시켜 이러한 영역이 정렬됩니다.
- 클라우드 영역을 선택한 후에는 클라우드 템플릿에 표시된 고정 및 유동 제약 조건을 포함한 일련의 필터와 일치시켜 호스트가 선택됩니다.

## 간단한 태그 지정 구조를 설정하는 방법

이 항목에서는 논리적인 vRealize Automation Cloud Assembly 태그 지정 전략을 위한 기본적인 접근 방식과 옵션을 설명합니다. 이러한 예를 실제 배포의 시작점으로 활용할 수도 있고 요구 사항에 가장 잘 맞는 다른 전략을 세울 수도 있습니다.

대개 클라우드 관리자가 태그 생성 및 유지 관리를 주로 담당합니다.

이 항목에서는 vRealize Automation Cloud Assembly 설명서의 다른 곳에서 설명한 WordPress 사용 사례를 제시하여 일부 주요 항목에 태그를 추가하는 방법을 설명합니다. 또한 WordPress 사용 사례에 나온 태그 지정 예시에 대한 가능한 대안과 확대 사례를 설명합니다.

WordPress 사용 사례에 대한 자세한 내용은 [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트](#) 항목을 참조하십시오.

WordPress 사용 사례에서는 클라우드 영역과 스토리지 및 네트워크 프로파일에 태그를 배치하는 방법을 설명합니다. 이러한 프로파일은 구성된 리소스 패키지로 생각할 수 있습니다. 프로파일에 배치된 태그는 프로파일 내 모든 항목에 적용됩니다. 또한 스토리지 리소스와 개별 네트워크 항목 및 계산 리소스에도 태그를 생성하고 배치할 수 있지만 이러한 태그는 태그가 배치된 특정 리소스에만 적용됩니다. 태그를 설정할 때는 먼저 계산 리소스에 태그를 지정하고 이후에 프로파일과 클라우드 영역에 태그를 추가하는 것이 좋습니다. 또한 이러한 태그를 사용하여 클라우드 영역의 계산 리소스 목록을 필터링할 수 있습니다.

예를 들어, 이 사용 사례에 나온 대로 스토리지 프로파일에 태그를 배치할 수 있지만 개별 스토리지 정책, 데이터스토어 및 스토리지 계정도 태그를 배치할 수 있습니다. 이러한 리소스에 태그를 배치하면 스토리지 리소스의 배포 방식을 보다 세부적으로 제어할 수 있습니다. 배포를 준비하기 위해 처리되는 과정에서 이러한 태그는 프로파일 태그 다음에 처리되는 수준으로 확인됩니다.

일반적인 고객 시나리오를 구성하는 방법의 예로 `region: eastern` 태그를 네트워크 프로파일에 배치할 수 있습니다. 이 태그는 해당 프로파일 내 모든 리소스에 적용됩니다. 그런 다음 해당 프로파일 내 PCI 네트워크 리소스에 `networktype:pci` 태그를 배치할 수 있습니다. `eastern` 및 `pci`의 제약 조건을 가진 클라우드 템플릿은 Eastern 영역의 이 PCI 네트워크를 사용하는 배포를 생성합니다.

## 절차

### 1 논리적이고 적절한 방식으로 계산 인프라 리소스에 태그를 지정하십시오.

[클라우드 영역 생성] 페이지의 [계산] 탭에서 검색 기능을 사용하여 찾을 수 있도록 논리적인 방식으로 계산 리소스에 태그를 지정하는 것이 특히 중요합니다. 이 검색 기능을 사용하여 특정 클라우드 영역에 연결된 계산 리소스를 빠르게 필터링할 수 있습니다. 프로파일 수준에서 스토리지와 네트워크에 태그를 지정하는 경우 개별 스토리지 및 네트워크 리소스에는 태그를 지정할 필요가 없습니다.

- a **리소스 > 계산**을 선택하여 vRealize Automation Cloud Assembly 인스턴스에 가져온 계산 리소스를 봅니다.
- b 필요한 대로 각 계산 리소스를 선택하고 **태그**를 클릭하여 해당 리소스에 태그를 추가합니다. 필요한 경우 각 리소스에 둘 이상의 태그를 추가할 수 있습니다.
- c 필요에 따라 스토리지 및 네트워크 리소스에 대해 위의 단계를 반복합니다.

### 2 클라우드 영역 및 네트워크 프로파일 기능 태그를 생성합니다.

동일한 태그를 클라우드 영역과 네트워크 프로파일에 모두 사용하거나, 구현하기에 더 적합한 경우 각 항목에 고유한 태그를 생성할 수 있습니다.

네트워크 프로파일에서는 전체 프로파일과 프로파일 내 서브넷에 태그를 배치할 수 있습니다. 프로파일 수준에서 적용된 태그는 서브넷을 포함하여 해당 프로파일 내 모든 구성 요소에 적용되는 한편, 서브넷에 배치된 태그는 해당 태그가 배치된 특정 서브넷에만 적용됩니다. 태그 처리 시 프로파일 수준의 태그가 서브넷 수준의 태그보다 우선합니다.

이 예에서는 vRealize Automation Cloud Assembly 클라우드 영역 및 네트워크 프로파일 태그에 대한 사용 사례 설명서 전반에 나오는 간단한 태그 3개를 생성합니다. 이러한 태그는 프로파일 구성 요소의 환경을 식별합니다.

- `zone:test`
- `zone:dev`
- `zone:prod`

### 3 스토리지 구성 요소에 대한 스토리지 프로파일 태그를 생성합니다.

일반적으로 스토리지 태그는 스토리지 항목의 성능 수준(예: Tier1 또는 Tier2)을 식별하거나 PCI 같은 스토리지 항목의 특성을 식별합니다.

스토리지 프로파일에 태그를 추가하는 방법에 대한 자세한 내용은 [6. 스토리지 프로파일 추가 항목을 참조하십시오.](#)

- `usage:general`
- `usage:fast`

## 결과

기본적인 태그 지정 구조를 생성한 후에는 이 구조를 사용하여 필요에 따라 태그를 추가하거나 편집하면서 태그 지정 기능을 세분화하고 확장할 수 있습니다.

## vRealize Automation에서 리소스로 작업하는 방법

클라우드 관리자는 데이터 수집을 통해 노출되는 vRealize Automation 리소스를 검토할 수 있습니다.

클라우드 관리자는 리소스에 기능 태그를 지정하여 vRealize Automation 클라우드 템플릿을 배포할 위치에 영향을 줄 수 있습니다.

## vRealize Automation의 계산 리소스

클라우드 관리자는 데이터 수집을 통해 노출되는 계산 리소스를 검토할 수 있습니다.

클라우드 관리자는 리소스에 태그를 직접 적용하도록 선택하여 vRealize Automation 프로비저닝의 목적에 맞게 기능에 레이블을 지정할 수 있습니다.

## vRealize Automation의 네트워크 리소스

vRealize Automation에서 클라우드 관리자는 프로젝트에 매핑된 클라우드 계정 및 통합에서 데이터가 수집된 네트워크 리소스를 보고 편집할 수 있습니다.

**인프라 > 연결 > 클라우드 계정** 메뉴 시퀀스 등을 사용하여 vRealize Automation Cloud Assembly 인프라에 클라우드 계정을 추가한 후 데이터 수집은 클라우드 계정의 네트워크 및 보안 정보를 검색합니다. 그런 다음 네트워크, 네트워크 프로파일 및 기타 정의에서 해당 정보를 사용할 수 있습니다.

네트워크는 사용 가능한 네트워크 도메인 또는 전송 영역의 IP별 구성 요소입니다. Amazon Web Services 또는 Microsoft Azure 사용자인 경우 네트워크를 서브넷으로 간주합니다.

**인프라 > 리소스 > 네트워크** 페이지를 사용하여 프로젝트의 네트워크에 대한 정보를 표시할 수 있습니다.

vRealize Automation Cloud Assembly **네트워크** 페이지에는 다음과 같은 정보가 포함되어 있습니다.

- 클라우드 계정의 네트워크 도메인(예: vCenter, NSX-V 또는 Amazon Web Services)에서 외부적으로 정의된 네트워크 및 로드 밸런서.
- 클라우드 관리자가 배포한 네트워크 및 로드 밸런서.
- 클라우드 관리자가 정의 또는 수정한 IP 범위 및 기타 네트워크 특성.
- 제공자별 외부 IPAM 통합의 특정 주소 공간에 대한 외부 IPAM 제공자 IP 범위.

네트워크에 대한 자세한 내용은 다음 정보, **네트워크** 페이지의 다양한 설정에 대한 표지판 도움말 및 **vRealize Automation의 네트워크 프로파일에 대해 알아보기** 항목을 참조하십시오.

## 네트워크

네트워크와 해당 특성을 보고 편집할 수 있습니다. 예를 들어 태그를 추가하거나 공용 IP 액세스에 대한 지원을 제거할 수 있습니다. DNS, CIDR, 게이트웨이 및 태그 값과 같은 네트워크 설정도 관리할 수 있습니다. 또한 네트워크 내에서 새로운 IP 범위를 정의하거나 기존 IP 범위를 관리할 수 있습니다.

기존 네트워크의 경우, 네트워크의 확인란을 선택하고 **IP 범위 관리** 또는 **태그**를 선택하여 IP 범위와 태그 설정을 변경할 수 있습니다. 또는 네트워크 자체를 선택하여 정보를 편집할 수 있습니다.

태그는 적절한 네트워크와 네트워크 프로파일(선택 사항)을 클라우드 템플릿의 네트워크 구성 요소에 일치시키는 방법을 제공합니다. 네트워크 태그는 네트워크가 상주할 수 있는 네트워크 프로파일에 관계없이 해당 네트워크의 모든 인스턴스에 적용됩니다. 네트워크는 임의의 수의 네트워크 프로파일에 인스턴스될 수 있습니다. 네트워크 프로파일 상주 여부와 관계없이, 네트워크 태그는 네트워크가 사용되는 모든 위치에서 네트워크와 연결됩니다. 클라우드 템플릿이 하나 이상의 네트워크 프로파일과 매칭된 후 클라우드 템플릿의 다른 구성 요소와 네트워크 태그 매칭이 발생합니다.

## IP 범위

IP 범위를 사용하여 조직의 특정 네트워크에 대한 시작 및 끝 IP 주소를 정의하거나 변경할 수 있습니다. 나열된 네트워크의 IP 범위를 표시하고 관리할 수 있습니다. 네트워크가 외부 IPAM 제공자에 의해 관리되는 경우 연결된 IPAM 통합 지점과 관련하여 IP 범위를 관리할 수 있습니다.

**새 IP 범위**를 클릭하여 네트워크에 추가 IP 범위를 추가합니다. **내부 IP 범위**를 지정하거나 사용 가능한 유효한 IPAM 통합이 있는 경우 **외부 IP 범위**를 지정할 수 있습니다.

기본 게이트웨이는 IP 범위에 포함할 수 없습니다. 서브넷 IP 범위에는 서브넷 게이트웨이 값이 포함될 수 없습니다.

특정 IPAM 제공자에 대해 외부 IPAM 통합을 사용하는 경우 **외부 IP 범위**를 사용하여 사용 가능한 외부 IPAM 통합 지점에서 IP 범위를 선택할 수 있습니다. 이 프로세스는 [vRealize Automation에서 기존 네트워크에 대해 외부 IPAM을 사용하도록 네트워크 및 네트워크 프로파일 구성](#)에서 전체 외부 IPAM 통합 워크플로와 관련하여 설명되어 있습니다.

## IP 주소

조직에서 현재 사용 중인 IP 주소를 보고 상태(예: available 또는 allocated)를 표시할 수 있습니다. 표시되는 IP 주소는 외부 IPAM 제공자 통합을 포함하는 배포에 지정된 vRealize Automation 또는 IP 주소를 통해 내부적으로 관리되는 IP 주소입니다. 외부 IPAM 제공자는 자체 IP 주소 할당을 관리합니다.

네트워크가 외부 IPAM 제공자가 아닌 vRealize Automation에 의해 내부적으로 관리되는 경우 IP 주소를 해제할 수도 있습니다.

내부 IPAM을 사용하고 IP 주소를 해제하는 경우(예: IP 주소를 사용하던 시스템을 삭제한 후) 주소가 해제되고 그 주소를 다시 사용할 수 있게 될 때까지 30분의 대기 시간이 필요합니다. 대기 기간 동안 DNS 캐시를 지울 수 있습니다. 그런 다음, 새 시스템에 IP 주소를 할당할 수 있습니다. 이제 이전에 삭제한 시스템과 동일한 IP 주소를 사용하여 시스템을 프로비저닝할 수 있습니다.

## 로드 밸런서

조직의 계정/지역 클라우드 계정에 사용 가능한 로드 밸런서에 대한 정보를 관리할 수 있습니다. 사용 가능한 각 로드 밸런서에 대해 구성된 설정을 열고 표시할 수 있습니다. 로드 밸런서에 대한 태그를 추가 및 제거할 수도 있습니다.

## 네트워크 도메인

네트워크 도메인 목록에는 관련 네트워크와 겹치지 않는 네트워크가 포함됩니다.

## vRealize Automation의 보안 리소스

vRealize Automation Cloud Assembly에서 클라우드 계정을 추가하면 데이터 수집에서 클라우드 계정의 네트워크 및 보안 정보를 검색하고 해당 정보를 네트워크 프로파일 및 기타 옵션에서 사용 가능한 상태로 설정합니다.

보안 그룹 및 방화벽 규칙은 네트워크 격리를 지원합니다. 보안 그룹은 데이터 수집됩니다. 방화벽 규칙은 데이터를 수집하지 않습니다.

### 보안 그룹

**인프라 > 리소스 > 보안** 메뉴 시퀀스를 사용하여 vRealize Automation Cloud Assembly 클라우드 템플릿 설계에서 생성된 주문형 보안 그룹과 NSX-T 및 Amazon Web Services와 같은 소스 애플리케이션에서 생성되었던 기존 보안 그룹을 볼 수 있습니다. 사용 가능한 보안 그룹은 데이터 수집 프로세스에 의해 노출됩니다.

사용 가능한 보안 그룹을 보고 선택한 보안 그룹에 대해 태그를 추가하거나 제거할 수 있습니다. 클라우드 템플릿 작성자는 시스템 NIC에 하나 이상의 보안 그룹을 할당하여 배포를 위한 보안을 제어할 수 있습니다.

클라우드 템플릿 설계에서 보안 그룹 리소스의 securityGroupType 매개 변수는 기존 보안 그룹에 대해 existing으로 지정되거나 주문형 보안 그룹에 대해 new로 지정됩니다.

기본 클라우드 계정 끝점(예: NSX-V, NSX-T 또는 Amazon Web Services 애플리케이션)의 기존 보안 그룹을 사용할 수 있습니다. 조직의 클라우드 템플릿 설계에서 생성된 주문형 보안 그룹도 데이터가 수집됩니다. 현재 주문형 보안 그룹은 NSX-V 및 NSX-T에만 사용할 수 있습니다.

기존 보안 그룹은 **원본** 열에 Discovered로 표시되고 분류됩니다. vRealize Automation Cloud Assembly에서 생성하고 클라우드 템플릿 또는 네트워크 프로파일에 위치하는 주문형 보안 그룹은 **원본** 열에 Managed by Cloud Assembly로 표시되고 분류됩니다. 네트워크 프로파일의 일부로 생성하는 주문형 보안 그룹은 내부적으로 미리 구성된 방화벽 규칙을 사용하여 격리 보안 그룹으로 분류되며, 보안 그룹 리소스로 클라우드 템플릿 설계에 추가되지 않습니다. 클라우드 템플릿 설계에서 생성하고 express 방화벽 규칙을 포함할 수 있는 주문형 보안 그룹은 new로 분류된 보안 그룹 리소스의 일부로 추가됩니다.

vRealize Automation Cloud Assembly가 아닌 소스 애플리케이션(예: 소스 NSX 애플리케이션)에서 직접 기존 보안 그룹을 편집하는 경우 vRealize Automation Cloud Assembly 내에서 데이터 수집이 실행되고 데이터가 연결된 클라우드 계정 또는 통합 지점을 수집할 때까지 vRealize Automation Cloud Assembly에 업데이트가 표시되지 않습니다. 데이터 수집은 10분마다 실행됩니다.

클라우드 관리자는 클라우드 템플릿에서 사용할 수 있도록 기존 보안 그룹에 태그를 하나 이상 할당할 수 있습니다. 클라우드 템플릿 작성자는 클라우드 템플릿 설계의 Cloud.SecurityGroup 리소스를 통해 태그 제약 조건을 사용하여 기존 보안 그룹을 할당할 수 있습니다. 기존 보안 그룹을 사용하려면 클라우드 템플릿 설계에서 보안 리소스에 하나 이상의 제약 조건 태그를 지정해야 합니다.

## 보안 그룹에서 방화벽 규칙 사용

클라우드 템플릿 설계 코드의 보안 그룹 리소스에서 직접 NSX-V 및 NSX-T에 대해 주문형 보안 그룹에 대한 방화벽 규칙을 생성할 수 있습니다.

**적용 대상** 열에는 NSX DFW(분산 방화벽)에 의해 분류되거나 관리되는 보안 그룹이 포함되지 않습니다. 애플리케이션에 적용되는 방화벽 규칙은 East/West DFW 트래픽에 대한 것입니다.

일부 방화벽 규칙은 소스 애플리케이션에서만 관리할 수 있으며 vRealize Automation Cloud Assembly에서는 편집할 수 없습니다. 예를 들어 이더넷, 긴급, 인프라 및 환경 규칙은 NSX-T에서 관리됩니다.

## 자세히 알아보기

네트워크 프로파일의 보안 그룹 사용에 대한 자세한 내용은 [vRealize Automation의 네트워크 프로파일](#)에 대해 [알아보기](#) 항목을 참조하십시오.

방화벽 규칙 정의에 대한 자세한 내용은 [vRealize Automation Cloud Assembly](#)에서 [네트워크 프로파일](#) 및 [클라우드 템플릿 설계의 보안 그룹 설정 사용](#) 및 [vRealize Automation 클라우드 템플릿에서 보안 그룹 리소스 사용](#)의 내용을 참조하십시오.

보안 그룹이 포함된 클라우드 템플릿 설계 코드 샘플은 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서](#) 예의 내용을 참조하십시오.

## vRealize Automation의 스토리지 리소스

클라우드 관리자는 연결된 클라우드 계정의 vRealize Automation 데이터 수집을 통해 검색된 스토리지 리소스와 해당 기능을 사용할 수 있습니다.

스토리지 리소스 기능은 일반적으로 소스 클라우드 계정에서 시작되는 태그를 통해 노출됩니다. 클라우드 관리자는 vRealize Automation Cloud Assembly를 사용하여 스토리지 리소스에 추가 태그를 직접 적용하도록 선택할 수 있습니다. 추가 태그는 프로비저닝 시간에 목적 일치를 위한 특정 기능에 레이블을 지정할 수 있습니다.

vRealize Automation은 표준 디스크 및 FCD(First Class Disk) 기능을 지원합니다. FCD(First Class Disk)는 vSphere에만 사용할 수 있습니다.

- [vRealize Automation](#)에서 [표준 디스크 스토리지](#)로 수행할 수 있는 작업
- [vRealize Automation](#)에서 [FCD\(First Class Disk\) 스토리지](#)로 수행할 수 있는 작업

스토리지 리소스의 기능은 vRealize Automation Cloud Assembly 스토리지 프로파일 정의의 일부로 표시됩니다. [vRealize Automation의 스토리지 프로파일](#)에 대해 [알아보기](#) 항목을 참조하십시오.

데이터 수집된 FCD(First Class Disk)는 [불륨](#) 리소스 페이지에 표시됩니다. [vRealize Automation의 불륨 리소스](#)의 내용을 참조하십시오.

## vRealize Automation의 시스템 리소스

vRealize Automation에서 모든 사용자는 데이터 수집을 통해 노출되는 시스템 리소스를 검토할 수 있습니다.

프로젝트의 모든 시스템에 대한 정보를 사용할 수 있습니다. 시스템만 나열하거나 필터를 지정하여 나열된 시스템의 표시를 제어할 수 있습니다.

관리되는 시스템과 마찬가지로, 프로젝트에서 클라우드 계정에 연결되어 있는 관리되지 않는 시스템이 이 목록에 나타납니다. [원본] 열은 시스템 상태를 나타냅니다.

- 검색됨 - 아직 온보딩되지 않은 시스템.
- 배포됨 - vRealize Automation에서 온보딩되거나 프로비저닝된 시스템이며 관리되는 시스템으로 간주됩니다.

사용자 지정 속성 및 메모리 같은 다른 시스템 정보도 수집 및 표시됩니다.

워크로드 온보딩 계획을 사용하여 관리되지 않는 시스템을 vRealize Automation 관리로 가져올 수 있습니다.

vRealize Automation는 네트워크 스위치 또는 서브넷 정보가 있어야 이더넷 카드를 열거할 수 있기 때문에 연결이 끊긴 시스템 NIC가 나열되지 않습니다. 예를 들어 배포에서 시스템 NIC를 제거한 경우 NIC가 나열되지 않습니다.

온보딩 계획을 사용하여 관리되지 않는 시스템을 vRealize Automation 관리로 가져오는 방법에 대한 자세한 내용은 [vRealize Automation Cloud Assembly의 온보딩 계획이란?](#) 항목을 참조하십시오.

## vRealize Automation의 볼륨 리소스

vRealize Automation에서 모든 사용자는 볼륨 리소스를 검토할 수 있습니다.

vRealize Automation Cloud Assembly는 다음 두 가지 소스에서 만들어진 볼륨 또는 논리적 드라이브를 표시합니다.

- 소스 클라우드 계정의 데이터 수집을 통해 검색된 볼륨
- vRealize Automation Cloud Assembly에서 프로비저닝한 워크로드와 연결된 볼륨

볼륨 또는 논리적 드라이브에 따라 용량 및 기능을 검토할 수 있습니다. 또한 소스 클라우드 계정에서 시작되었거나 vRealize Automation Cloud Assembly 자체에 추가된 기능 태그가 목록에 표시됩니다.

FCD(First Class Disk)로서 볼륨의 상태도 명시됩니다. FCD(First Class Disk) 스토리지 볼륨에 대한 자세한 내용은 [vRealize Automation에서 FCD\(First Class Disk\) 스토리지로 수행할 수 있는 작업](#) 항목을 참조하십시오.

## vRealize Automation Cloud Assembly의 리소스에 대해 알아보기

vRealize Automation Cloud Assembly는 데이터 수집 리소스(예: 가격 책정 카드)에 대한 추가 정보를 노출할 수 있습니다.

### vRealize Automation에서 데이터 수집의 작동 방식

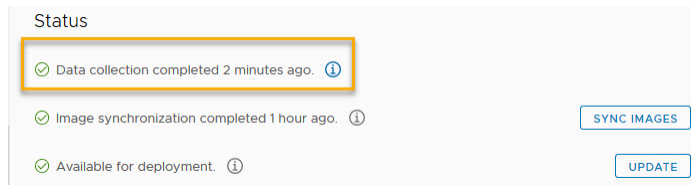
초기 데이터 수집 후에는 리소스 데이터 수집이 10분마다 자동으로 발생합니다. 데이터 수집 간격을 구성할 수 없으며 데이터 수집을 수동으로 시작할 수 없습니다.

해당 페이지의 [상태] 섹션에서 기존 클라우드 계정에 대한 리소스 데이터 수집 및 이미지 동기화 관련 정보를 검색할 수 있습니다. **인프라 > 연결 > 클라우드 계정**을 선택한 다음 선택한 기존 클라우드 계정에서 **열기**를 클릭하면 됩니다.

기존 클라우드 계정을 열고 해당 페이지의 **상태** 섹션에서 연결된 끝점 버전을 볼 수 있습니다. 연결된 끝점이 업그레이드된 경우 새 끝점 버전이 데이터 수집 중에 검색되고 클라우드 계정 페이지의 **상태** 섹션에 반영됩니다.

## 리소스 데이터 수집

데이터 수집은 10분마다 발생합니다. 각 클라우드 계정은 해당 데이터 수집이 마지막으로 완료된 시간을 표시합니다.

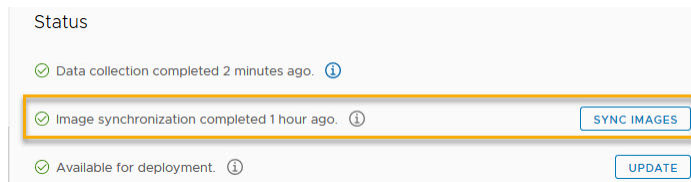


## 이미지 데이터 수집

이미지 동기화는 24시간마다 발생합니다. 일부 클라우드 계정 유형에 대해 이미지 동기화를 시작할 수 있습니다. 이미지 동기화를 시작하려면 클라우드 계정을 열고(**인프라 > 클라우드 계정**으로 이동한 후 기존 클라우드 계정을 선택하여 열기) **이미지 동기화** 버튼을 클릭합니다. NSX 클라우드 계정에 대한 이미지 동기화 옵션은 없습니다.

**참고** 이미지는 내부적으로 공용 또는 전용으로 분류됩니다. 공용 이미지는 공유되고 특정 클라우드 구독 또는 조직에 국한되지 않습니다. 전용 이미지는 공유되지 않으며 특정 구독에 국한됩니다. 공용 및 전용 이미지는 24시간마다 자동으로 동기화됩니다. 클라우드 계정 페이지의 옵션을 사용하여 전용 이미지에 대한 동기화를 트리거할 수 있습니다.

클라우드 계정 페이지는 이미지 동기화가 마지막으로 완료된 시간을 표시합니다.



배포에서 Fault Tolerance 및 고가용성을 용이하게 하기 위해 각 NSX-T 데이터 센터 끝점은 NSX Manager 3개의 클러스터를 나타냅니다. 관련 정보는 vRealize Automation에서 **NSX-T 클라우드 계정 생성** 항목을 참조하십시오.

## 클라우드 계정 및 온보딩 계획

클라우드 계정을 생성할 경우, 이 클라우드 계정에 연결된 모든 시스템의 데이터가 수집되어 **인프라 > 리소스 > 시스템** 페이지에 표시됩니다. 클라우드 계정에 vRealize Automation Cloud Assembly 외부에 배포된 시스템이 있는 경우에는 온보딩 계획을 사용하여 vRealize Automation Cloud Assembly가 해당 시스템을 관리하도록 할 수 있습니다.

클라우드 계정 추가에 대한 자세한 내용은 [vRealize Automation Cloud Assembly](#)에 클라우드 계정 추가의 내용을 참조하십시오.

관리되지 않는 시스템의 온보딩에 대한 자세한 내용은 [vRealize Automation Cloud Assembly의 온보딩 계획이란?](#)의 내용을 참조하십시오.

## vRealize Automation에서 인사이트 대시보드를 사용하여 리소스 용량을 모니터링하고 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알리는 방법

클라우드 관리자는 각 클라우드 영역 내에서 인프라 리소스 및 배포 최적화를 모니터링하여 지원되는 프로젝트에 대한 리소스 용량과 배포 최적화를 사전 예방적으로 관리해야 합니다.

### 해결 방법

인사이트 대시보드를 사용하여 관리하는 프로젝트 내에서 클라우드 영역의 리소스 및 배포에 대한 메트릭 데이터를 살펴볼 수 있습니다. vRealize Automation 및 통합 vRealize Operations Manager 애플리케이션의 조합에서 제공되는 정보를 사용하여 메모리, CPU 등에 필요한 조정을 수행하거나 해당 정보를 팀과 공유하여 정보를 잘 파악하고 필요한 조정을 수행할 수 있도록 합니다.

최적화할 수 있는 배포의 프로젝트 소유자가 **경고** 페이지에서 알림을 받도록 지정할 수도 있습니다. 알림에는 해당 이름 및 최적화할 수 있는 각 배포의 이름(및 링크)이 포함됩니다.

인사이트 대시보드는 현재 vSphere 및 VMware Cloud on AWS 클라우드 영역에서 사용할 수 있습니다. 단, 클라우드 계정이 vRealize Automation 및 vRealize Operations Manager 둘 다에 구성되어 있고 vRealize Operations Manager에서 모니터링되는 경우에 한합니다.

### 사전 요구 사항

- [vRealize Automation](#)에서 [vRealize Operations Manager](#) 메트릭을 사용하여 리소스 관리 및 배포 최적화 항목을 검토합니다.
- vRealize Automation 클라우드 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation](#)에서 클라우드 계정 작업에 필요한 자격 증명 항목을 참조하십시오.
- vRealize Automation 클라우드 관리자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- vRealize Operations Manager와 vRealize Automation 통합을 구성합니다.
- vRealize Operations Manager에서 vRealize Automation 어댑터를 구성합니다.

### vRealize Operations Manager 및 수집된 리소스 용량 메트릭 정보

vRealize Operations Manager는 사용자 및 사용자가 지원하는 팀이 vRealize Automation에서 사용하는 것과 동일한 인프라 리소스에 대한 용량 메트릭을 수집합니다. vRealize Automation을 vRealize Operations Manager와 통합하면 vRealize Operations Manager 메트릭 데이터를 사용할 수 있고 각 클라우드 영역 내의 **인사이트** 대시보드에 있는 각 관리 프로젝트에 대해 표시됩니다.

프로젝트 데이터는 통합된 vRealize Operations Manager 애플리케이션에서 vRealize Automation 대시보드에 대해 구분 분석됩니다. 대시보드에는 다음과 같은 정보가 표시됩니다.

- 용량 대비 CPU 활용률 백분율
- 용량 대비 메모리 활용률 백분율
- 용량 대비 스토리지 활용률 백분율
- 계산된 CPU 및 메모리 요구량 기록 및 예상 요구량
- 최적화할 수 있는 배포의 소유자에게 알리는 경고 옵션

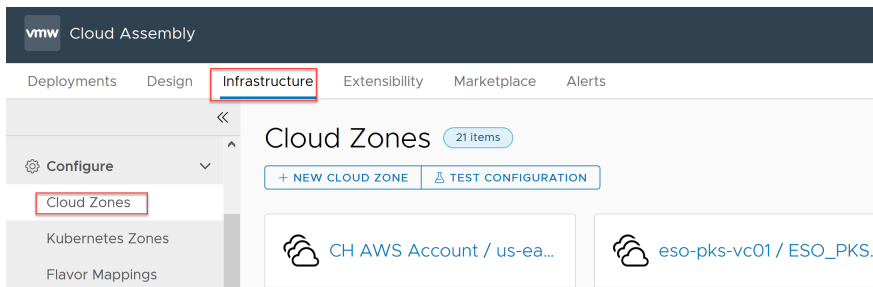
인사이트 대시보드는 현재 vSphere 리소스에 대해서만 사용할 수 있습니다.

관련 정보는 [vRealize Automation](#)에서 [vRealize Operations Manager](#) 메트릭을 사용하여 리소스 관리 및 배포 최적화 항목을 참조하십시오.

## 절차

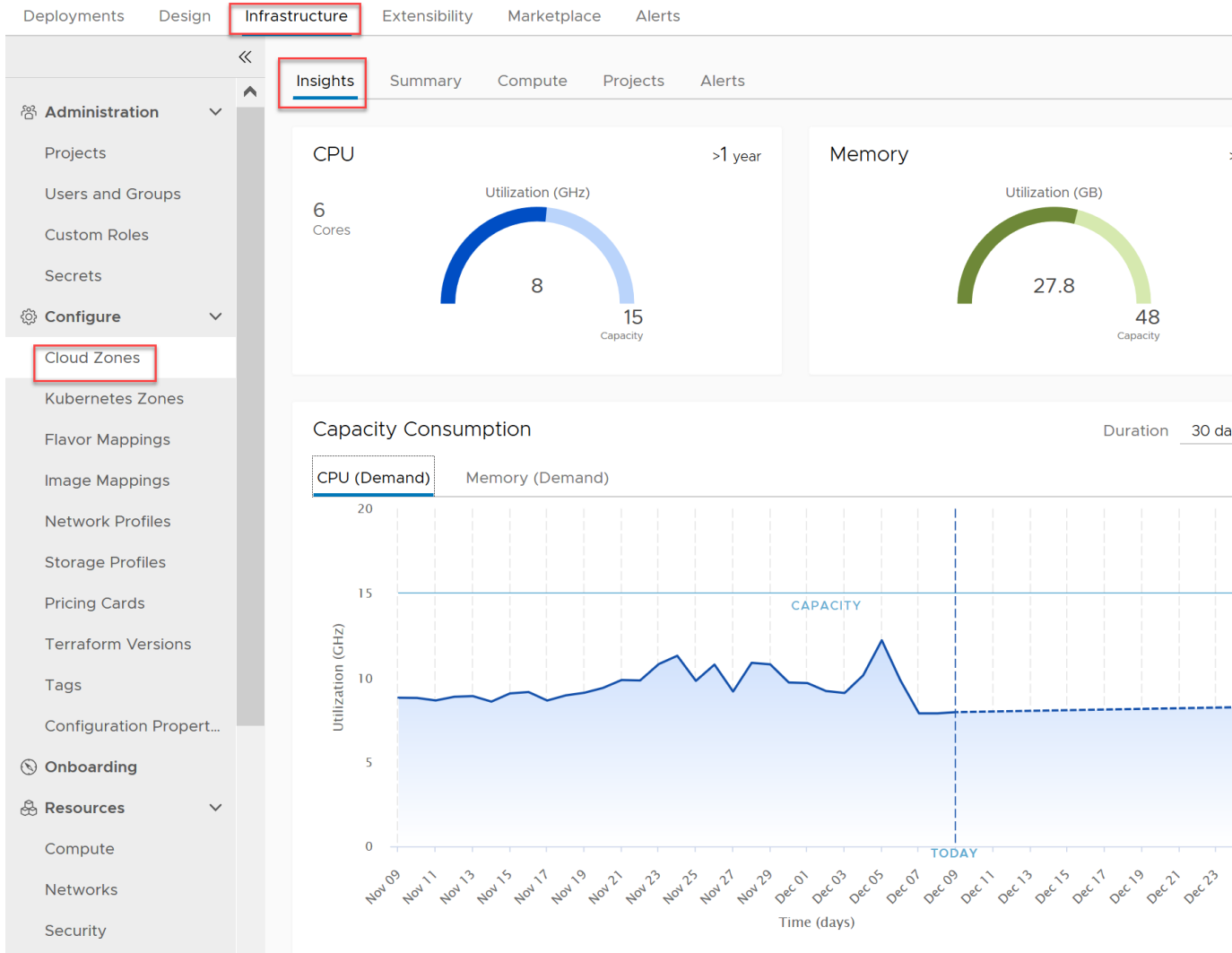
클라우드 영역을 열어 용량 메트릭을 검색하고, 필요한 경우 최적화할 수 있는 프로젝트 배포에 대한 정보를 가져옵니다. 연결된 vRealize Operations Manager 애플리케이션에서 데이터가 수집되고 제공됩니다.

- 1 vRealize Automation Cloud Assembly에서 **인프라 > 구성 > 클라우드 영역**을 클릭하고 클라우드 영역을 선택합니다.

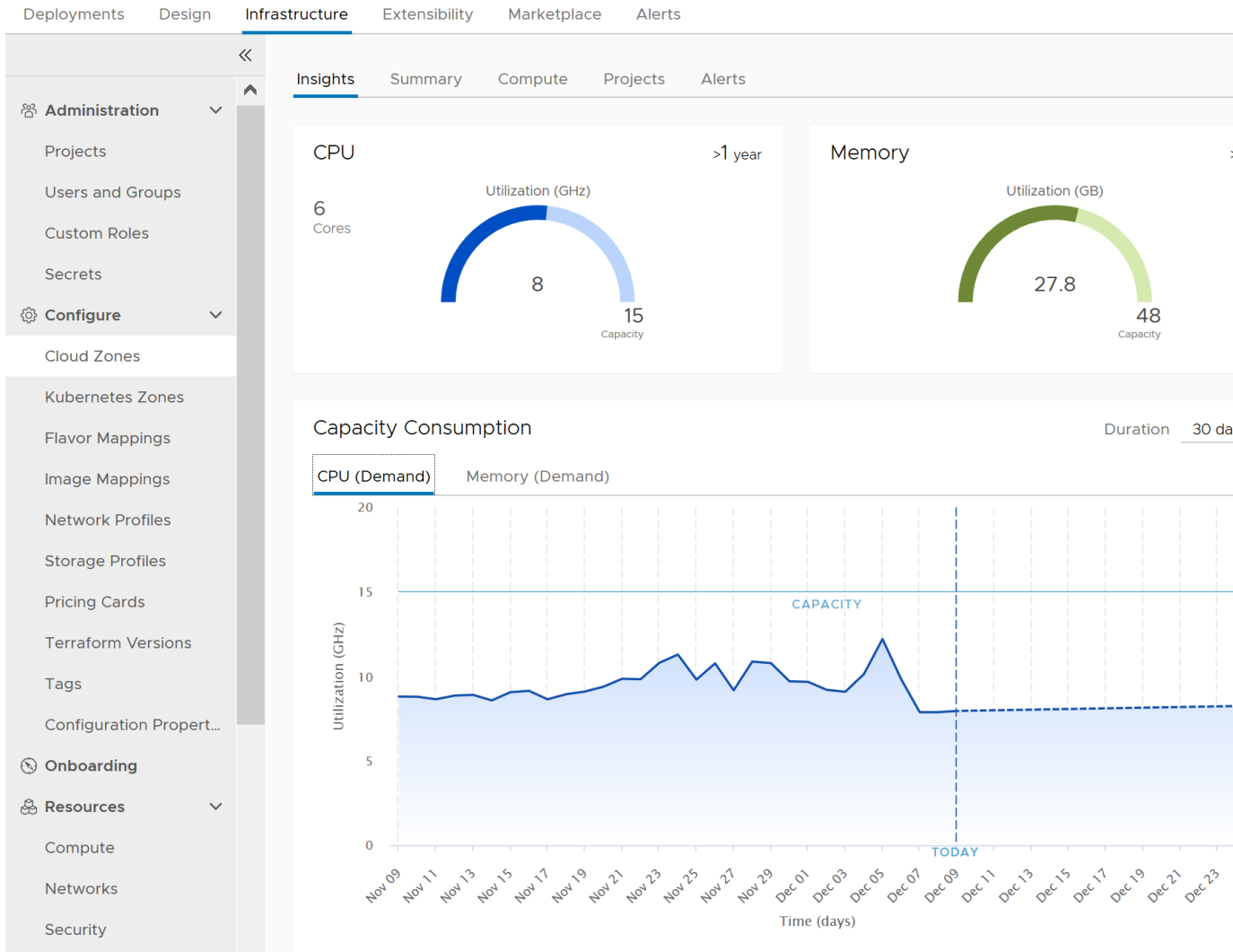


- 2 **인사이트** 탭을 클릭하고 인사이트 대시보드를 검토합니다.

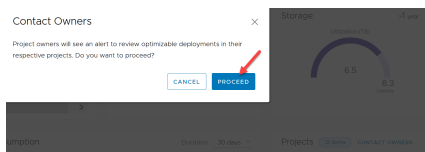
다음 예에서는 클라우드 영역의 프로젝트에서 사용하는 리소스에 대한 CPU, 메모리 및 스토리지 용량 정보를 표시합니다.



- 3 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알려려면 **프로젝트** 섹션에서 **소유자에게 연락**을 클릭합니다. 알림은 **경고** 탭 페이지에 표시됩니다.

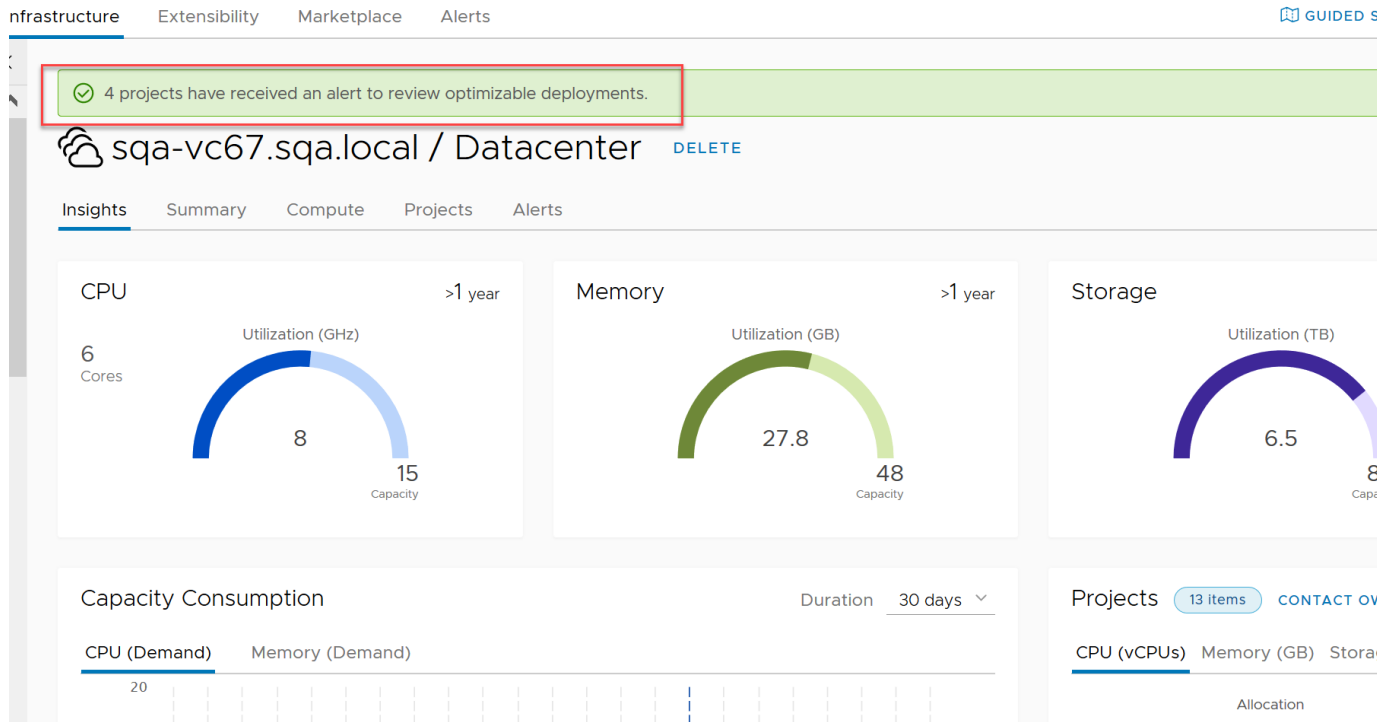


- 4 프로젝트의 각 배포에 대한 최적화 정보를 가져오려면 **계속**을 클릭합니다.



최적화할 수 있는 배포가 프로젝트에 포함되어 있는 경우 해당 정보는 vRealize Automation Cloud Assembly **경고** 탭을 통해 프로젝트 소유자에게 전달됩니다.

- 5 최적화할 수 있는 배포의 수를 나타내는 메시지가 표시됩니다.



이러한 리소스 및 배포에 대한 알림 정보는 vRealize Automation Cloud Assembly **경고** 탭을 통해 프로젝트 소유자에게 전달됩니다. 이 예에서는 알림 정보에 다음 예에 표시된 것처럼 최적화할 수 있는 각 배포의 이름 및 링크가 포함됩니다.

vmw Cloud Assembly

Deployments Design Infrastructure Extensibility Marketplace **Alerts**

Severity Quick filters Status: Active X

Older

**The project has some deployments that contain optimizable resources** 2:39 AM  
Virtual Machine » Cap\_Idle

**Virtual machine disk I/O read/write latency is high** 2:09 AM  
Virtual Machine » OversizedIO  
The following deployments under this project can be optimized <ul><li><a href=>

**The project has some deployments that contain optimizable resources** 2:00 AM  
Project » All Clouds Project  
The project has some deployments that contain optimizable resources.

**HCMP\_Definition\_Deployment\_VM** 8:50 AM  
Deployment » vSphere Deployment

**HCMP\_AlertDefinition\_15a83e3d-fdae-44ba-8b87-99b7251968i9** 8:50 AM  
Virtual Machine » pricing-ft-don't-delete-mcm22578-134135715041

**HCMP\_Definition\_Deployment\_VM** 8:50 AM  
Deployment » vSphere dep with price card

**HCMP\_AlertDefinition\_15a83e3d-fdae-44ba-8b87-99b7251968i9** 8:46 AM  
Virtual Machine » pricing-ft-don't-delete-mcm21321-13404381

**HCMP\_AlertDefinition\_project\_has\_zero\_machine** 2:09 AM  
Project » KB Recommendation Test 1  
HCMP\_AlertDefinition\_project\_has\_zero\_machine

**HCMP\_AlertDefinition\_Deployment\_has\_cost** 7:04 PM

1 / 13

**The project has some deployments that contain optimizable resources**  
Created: Nov 27, 2020, 2:00:00 AM | Updated: Nov 30, 2020, 11:27:19 AM  
Project » All Clouds Project

The project has some deployments that contain optimizable resources.

Severity: Critical Status: Active Impact: Efficiency Type: Application

Suggestions **REVIEW PROJECT**

- If the project is experiencing increased provisioning, you can review the project longer in use.

**Deployments to review**

Name
DND-Deployment-Gerin

**Notes**

Leave a note...

ADD NOTE

has requested reviewing optimizable resources at Nov 30, 2020, 11:27:19 AM

has requested reviewing optimizable resources at Nov 30, 2020, 2:42:24 AM

## 다음 단계

인사이드 대시보드에서 얻은 정보를 사용하여 자신이 관리하는 리소스에 필요한 조정을 수행합니다. 경고 페이지를 열어 추가 정보, 제안된 작업 및 최적화할 수 있는 배포에 대한 링크를 가져옵니다. vRealize Automation에서 경고를 사용하여 리소스 용량, 성능 및 가용성을 관리하는 방법의 내용을 참조하십시오.

## vRealize Automation에서 경고를 사용하여 리소스 용량, 성능 및 가용성을 관리하는 방법

클라우드 관리자는 사용자의 리소스가 부족해지기 전에 사전 예방적으로 대응할 수 있도록 vRealize Automation 용량, 성능 또는 가용성이 문제가 되는 시기를 알아야 합니다.

## 해결 방법

연결된 vRealize Operations Manager 애플리케이션에서 제공되는 vSphere 및 VMware Cloud on AWS 리소스 개체에 대한 다양한 경고 정보를 표시할 수 있습니다. 이 정보를 사용하여 자신이 관리하는 리소스 및 배포에 대해 조정을 수행하거나, 해당 정보를 팀과 공유하여 팀에서 리소스를 조정할 수 있도록 합니다.

**참고** 최적화를 고려해야 하는 프로젝트 배포를 검토하고 작업하려면 [vRealize Automation에서 인사이트 대시보드](#)로부터 수집된 데이터를 기반으로 배포를 최적화하는 방법의 내용을 참조하십시오.

경고는 현재 vSphere 및 VMware Cloud on AWS 리소스 개체에만 사용할 수 있습니다. **경고** 탭은 vRealize Operations Manager에 대한 액세스가 구성된 경우에만 사용할 수 있습니다.

vRealize Automation 경고 임계값은 vRealize Operations Manager에서 설정됩니다. 일부 vRealize Automation 경고는 현재 미리 정의되어 있습니다. 경고 알림은 vRealize Operations Manager에서도 설정됩니다. 경고 정의 설정 및 알림 구성에 대한 자세한 내용은 vRealize Operations Manager [제품 설명서](#)를 참조하십시오.

## 사전 요구 사항

- [vRealize Automation에서 vRealize Operations Manager 메트릭](#)을 사용하여 리소스 관리 및 배포 최적화 항목을 검토합니다.
- vRealize Automation 클라우드 관리자 자격 증명이 있고 포트 443에서 HTTPS 액세스를 사용하도록 설정했는지 확인합니다. [vRealize Automation에서 클라우드 계정 작업에 필요한 자격 증명 항목](#)을 참조하십시오.
- vRealize Automation 클라우드 관리자 역할이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- vRealize Operations Manager와 vRealize Automation 통합을 구성합니다.
- vRealize Operations Manager에서 vRealize Automation 어댑터를 구성합니다.

## vRealize Operations Manager 및 리소스 경고 정보

vRealize Operations Manager는 vRealize Automation에서 관리하는 동일한 인프라 리소스 및 배포에 대한 상태, 사용량 및 기타 메트릭을 수집합니다. vRealize Automation을 vRealize Operations Manager와 통합하면, 모니터링되는 데이터를 vRealize Automation Cloud Assembly 주 메뉴의 **경고** 탭을 사용하여 vRealize Automation에서 사용할 수 있게 됩니다.

vRealize Operations Manager에서 제공하는 경고 데이터에는 클라우드 템플릿, 배포, 조직 및 프로젝트에 대한 상태 및 위험 임계값 관련 문제가 포함됩니다. 또한 클라우드 영역 **인사이트** 탭에서 수행된 작업으로 연락된 소유자를 기반으로 최적화할 수 있는 배포에 대한 정보도 포함되어 있습니다. [vRealize Automation에서 인사이트 대시보드](#)를 사용하여 리소스 용량을 모니터링하고 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알리는 방법의 내용을 참조하십시오.

각 배포에 대한 경고 세부 정보는 다음과 같습니다.

- 프로젝트 이름
- 최적화할 수 있는 리소스를 포함하는 배포의 이름(및 배포에 대한 링크)

- 제안된 작업
- 회수 및 최적화에 따른 잠재적 비용 절감
- 배포에 사용된 총 가상 CPU 수
- 배포에 사용된 총 RAM 메모리 양
- 배포에 사용된 총 스토리지 양
- 리소스 이름, 유휴 시스템, 전원이 꺼진 시스템, 크기가 초과된 시스템과 크기가 부족한 시스템, 과소 사용된 시스템 및 시스템 스냅샷을 비롯하여 회수 및 최적화에 권장되는 배포의 가상 시스템

## 절차

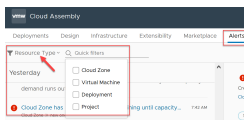
**경고** 페이지의 필터링 옵션을 사용하여 관리하는 리소스에 대한 경고 임계값 정보를 표시할 수 있습니다. 경고 데이터는 연결된 vRealize Operations Manager 애플리케이션에서 제공됩니다. 각 경고에 대해 제안된 작업이 제공됩니다.

또한 **검토할 배포** 섹션에서 배포를 선택하여 해당 배포를 열고 최적화할 수 있습니다. vRealize Automation에서 인사이트 대시보드로부터 수집된 데이터를 기반으로 배포를 최적화하는 방법의 내용을 참조하십시오.

- 1 vRealize Automation Cloud Assembly 서비스 내 주 메뉴에서 **경고** 탭을 클릭합니다.



- 2 경고가 표시되는 방식을 제어하려면 사용 가능한 필터를 테스트합니다. 예를 들어 필터 드롭다운 메뉴에서 **리소스** 옵션을 선택합니다.



- 3 경고와 이러한 경고에 대한 제안 작업을 표시하려면 선택기 패널에서 빠른 필터 옵션을 사용합니다.
  - 클라우드 영역 리소스에 대한 경고를 표시합니다.

vmw Cloud Assembly

Deployments Design Infrastructure Extensibility Marketplace Alerts

Resource Type ▾ Quick filters Resource Type: Cloud Zone X

Yesterday

Cloud Zone has less than 60 days remaining until capacity r... 7:42 AM

The time remaining on cloud zone is less than 60 days until capacity demand runs out.

Cloud Zone » sq-a-vc6

Cloud Zone has less than 60 days remaining until capacity r... 7:42 AM

Cloud Zone » new one

The time remaining on cloud zone is less than 60 days until capacity demand runs out.

HCMP\_AlertDefinition\_5ec2e9e9-3ead-43a8-a418-57a72c6b... 7:32 AM

Cloud Zone » sq-a-vc65 / Datacenter

HCMP\_AlertDefinition\_bf5e68e4-28f1-4992-af8d-94ea214ff8... 7:32 AM

Cloud Zone » sq-a-vc65 / Datacenter

HCMP\_AlertDefinition\_a317b9df-1c23-48c4-88a6-dc580859... 7:32 AM

Cloud Zone » eso-pks-vc01 / ESO\_PKS\_VC01\_DC01

HCMP\_AlertDefinition\_15a83e3d-fdae-44ba-8b87-99b72519... 7:32 AM

Cloud Zone » new one

HCMP\_AlertDefinition\_c749bb0c-df38-4152-a952-8aae7cfd... 7:32 AM

Cloud Zone » sq-a-vc67.sqa.local / Datacenter

1 / 29

Cloud Zone

Virtual Machine

Deployment

Project

Cloud

Created: Nov 6

Cloud Zone » s

The time rem

Severity: Crit

Suggestion

- Remove ca
- cloud zone
- more work
- Select an a
- Add comp
- Analyze as
- cloud zone

Notes

Leave a not

ADD NOTE

vRealize Operations Manager는 남은 시간, 남은 용량, 회수 가능 용량 등을 모니터링할 수 있습니다.

- 가상 시스템 리소스에 대한 경고를 표시합니다.

vmw Cloud Assembly

Deployments

Design

Infrastructure

Extensibility

Marketplace

Alerts

Resource Type ▾

Quick filters

Older

☐ Cloud Zone

☒ Virtual Machine

☐ Deployment

☐ Project

Resource Type: Virtual Machine X

<div><div>⚠</div><div>Virtual machine is powered off for more than 5 days</div><div>Virtual Machine &gt;&gt; 15-01-2025 08:06 AM</div><div>Virtual machine is powered off for more than 5 days</div></div>	
<div><div>⚠</div><div>Virtual machine is powered off for more than 5 days</div><div>Virtual Machine &gt;&gt; Cloud_vSphere_Machine_1-mcm151531-151568977758</div><div>Virtual machine is powered off for more than 5 days</div></div>	
<div><div>⚠</div><div>Virtual machine is powered off for more than 5 days</div><div>Virtual Machine &gt;&gt; pricing-ft-don't-delete-mcm21321-134043817143</div><div>Virtual machine is powered off for more than 5 days</div></div>	
<div><div>❗</div><div>Virtual machine disk I/O read/write latency is high</div><div>Virtual Machine &gt;&gt; pricing-ft-don't-delete-mcm22578-134135715041</div><div>Virtual machine disk I/O read latency is high</div></div>	
<div><div>❗</div><div>Virtual machine disk I/O read/write latency is high</div><div>Virtual Machine &gt;&gt; Cloud_vSphere_Machine_1-mcm158902-152191639667</div><div>Virtual machine disk I/O read latency is high</div></div>	

⚠ Virtual machine is powered off for more than 5 days

Created: Nov 4 2024 08:06 AM

Virtual Machine

Virtual machine is powered off for more than 5 days

Severity: Warning

Suggestions

- Delete powered off virtual machines
- Manually power on virtual machines
- If the virtual machine is not needed, delete it

Notes

Leave a note

ADD NOTE

대다수의 가상 시스템 경고는 켜짐/꺼짐 상태, 지연 시간 등과 관련이 있습니다.

- 배포 리소스에 대한 경고를 표시합니다.

vmw Cloud Assembly

Deployments Design Infrastructure Extensibility Marketplace Alerts

Resource Type ▾ Quick filters

Today

- ☐ Cloud Zone
- ☐ Virtual Machine
- ☒ Deployment
- ☐ Project

Resource Type: Deployment X

Yesterday

HCMP\_Definition\_Deployment\_VM 3:52 AM  
Deployment » Deployment-f11212bd-ee7f-48ad-b6c3-6bc2d88765cb

HCMP\_Definition\_Deployment\_VM 3:52 AM  
Deployment » Deployment-1d3e88a2-d9d2-4180-a9d6-aa477b04401b

HCMP\_Definition\_DeploymentType 3:42 AM  
Deployment » Deployment-1d3e88a2-d9d2-4180-a9d6-aa477b04401b

HCMP\_Definition\_DeploymentType 3:42 AM  
Deployment » Deployment-f11212bd-ee7f-48ad-b6c3-6bc2d88765cb

HCMP\_AlertDefinition\_Deployment\_alert\_test 3:37 AM  
Deployment » Deployment-1d3e88a2-d9d2-4180-a9d6-aa477b04401b

HCMP\_...  
Created: Nov 6  
Deployment » D  
Severity: Info  
Suggestions  
• Review the  
Notes  
Leave a note  
ADD NOTE

배포 경고는 회수 가능 리소스 및 적정 크기 조정과 관련이 있습니다.

- 프로젝트 리소스에 대한 경고를 표시합니다.

vmw Cloud Assembly

Deployments Design Infrastructure Extensibility Marketplace **Alerts**

Resource Type ▾ Quick filters Resource Type: Project X

Older

Project » GJ-Project

HCMP\_AlertDefinition\_Project\_has... 1:34 PM

Project » HCMP 3 - HC

HCMP\_AlertDefinition\_project\_has... 1:34 PM

Project » test1

HCMP\_AlertDefinition\_project\_has\_zero\_machin

HCMP\_AlertDefinition\_project\_has... 1:34 PM

Project » 测试g11n-vmc project

HCMP\_AlertDefinition\_project\_has\_zero\_machin

HCMP\_AlertDefinition\_Project\_has... 1:34 PM

Project » vc65 project

HCMP\_AlertDefinition\_project\_has... 4:14 PM

HCMP\_AlertDefinition\_project\_has...

Severity: Information Status: Canceled

Suggestions 1 REVIEW PROJECT

- If the project is experiencing increas deployments and poweroff /delete t

Notes

Leave a note...

ADD NOTE

4 / 8

프로젝트 경고는 회수 가능 리소스 및 할당 제한과 관련이 있습니다.

- 다른 필터 유형과 빠른 필터링 옵션을 탐색하여 경고 목록을 추가로 제어합니다.
  - 상태, 위험 및 효율성의 **영향** 빠른 필터를 사용합니다.
  - 위험, 즉시, 주의 및 정보의 **심각도** 빠른 필터를 사용합니다.
  - 활성, 취소됨 및 해제됨의 **상태** 빠른 필터를 사용합니다.
  - 가용성, 성능 및 용량의 **하위 유형** 필터를 사용합니다.
  - 애플리케이션, 하드웨어, 인프라, 스토리지 및 네트워크의 **유형** 빠른 필터를 사용합니다.
- 경고 데이터 및 제안을 기반으로 필요한 모든 작업을 수행합니다.

## 다음 단계

사용할 수 있는 다른 작업에 대해 알아보려면 [vRealize Automation](#)에서 [인사이트](#) 대시보드로부터 수집된 데이터를 기반으로 배포를 최적화하는 방법의 내용을 참조하십시오.

또한 관리하는 프로젝트의 클라우드 영역 기반 리소스에 대한 용량 [인사이트](#)를 표시할 수 있습니다.

vRealize Automation에서 vRealize Operations Manager 제공 [인사이트](#) 데이터를 사용하는 방법에 대한 자세한 내용은 [vRealize Automation](#)에서 [인사이트](#) 대시보드를 사용하여 리소스 용량을 모니터링하고 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알리는 방법 항목을 참조하십시오.

## vRealize Automation에서 인사이트 대시보드로부터 수집된 데이터를 기반으로 배포를 최적화하는 방법

클라우드 관리자 또는 프로젝트 소유자는 배포의 시스템 리소스를 최적화할 수 있는 시기와 방법을 알아야 하며 이를 vRealize Automation 내에서 수행할 수 있어야 합니다.

### 해결 방법

vRealize Automation 설치를 vRealize Operations Manager에 연결하면 관리하는 프로젝트의 리소스에 대한 데이터 수집 정보에 액세스할 수 있습니다. 경고 데이터는 관리하는 프로젝트에 대한 다양한 우려 사항을 알려주기 위해 제공됩니다.

배포 최적화 경고는 인사이트 대시보드에서 생성될 수 있습니다. [vRealize Automation](#)에서 [인사이트](#) 대시보드를 사용하여 리소스 용량을 모니터링하고 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알리는 방법의 내용을 참조하십시오. **경고** 페이지에 제공된 링크를 통해 최적화할 명명된 배포를 열 수 있습니다. 또한 프로젝트 소유자는 배포를 직접 열고 **최적화** 탭을 사용하여 사용 가능한 최적화 작업을 수행할 수 있습니다.

---

**참고** 수행할 수 있는 다른 리소스 업데이트 적용 작업에 대해 알아보려면 [vRealize Automation](#)에서 [경고](#)를 사용하여 리소스 용량, 성능 및 가용성을 관리하는 방법 항목을 참조하십시오.

---

### 사전 요구 사항

vRealize Automation에서 vRealize Operations Manager 데이터에 액세스하는 데 필요한 자격 증명 및 구성 정보는 [vRealize Automation](#)에서 [경고](#)를 사용하여 리소스 용량, 성능 및 가용성을 관리하는 방법 항목을 참조하십시오.

프로젝트 소유자에게 최적화할 수 있는 배포에 대한 알림이 제공되도록 요청하려면 [vRealize Automation](#)에서 [인사이트](#) 대시보드를 사용하여 리소스 용량을 모니터링하고 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알리는 방법의 내용을 참조하십시오.

### 정보

각 배포에는 **최적화** 탭이 포함되어 있습니다. 다음과 같은 최적화 매개 변수를 사용할 수 있습니다.

- **적정 크기 조정**이 가능한 시스템 - 배포에 있는 크기가 초과된 시스템 및 크기가 부족한 시스템에 대한 정보와 작업을 최적화 비용 절감액과 함께 표시합니다.
- **과소 사용된 시스템** - 배포에 있는 유휴 시스템 및 전원이 꺼진 시스템에 대한 정보와 작업을 최적화 비용 절감액과 함께 표시합니다.

- 스냅샷 - 배포에 있는 시스템에 스냅샷이 포함된 경우 시스템 스냅샷에 대한 정보와 작업을 최적화 비용 절감액과 함께 표시합니다.

관리자는 프로젝트 소유자에게 최적화할 배포가 있음을 알릴 수 있습니다. 알림은 vRealize Automation Cloud Assembly의 **경고** 탭에 표시됩니다. **경고** 탭은 vRealize Operations Manager에 대한 액세스가 구성된 경우에만 사용할 수 있습니다. 프로젝트 소유자는 배포를 열고 최적화하여 경고에 대응할 수 있습니다.

## 절차

**경고** 페이지의 필터링 옵션을 사용하여 관리하는 리소스에 대한 경고 임계값 정보를 표시할 수 있습니다. 경고 데이터는 연결된 vRealize Operations Manager 애플리케이션에서 제공됩니다. 각 경고에 대해 제안된 작업이 제공됩니다. 이 예에서 프로젝트 소유자는 경고 알림에 제공된 링크를 통해 배포를 엽니다. 배포의 **최적화** 탭에는 최적화할 수 있는 시스템 매개 변수가 표시됩니다.

- 1 프로젝트 소유자 또는 관리자는 기본 메뉴에서 **경고** 탭을 클릭합니다.



- 2 최적화할 수 있는 배포에 대한 정보를 포함하는 경고를 찾고 **검토할 배포**에서 배포 이름을 클릭하여 배포를 열고 해당 **최적화** 탭을 표시합니다.

Deployments Design Infrastructure Extensibility Marketplace **Alerts**

Severity ▾ Quick filters Status: Active × Severity: Critical ×

**Today**

- The project has some deployments t...** 6:17 PM  
Project » vc65 project  
The project has some deployments that contain optimizable resources.
- Cloud Zone has less than 60 days r...** 11:46 AM  
Cloud Zone » sq-a-vc67.sqa.local / Datacenter  
The time remaining on cloud zone is less than 60 days until capacity demand runs out.

**Yesterday**

- Cloud Zone has less than 60 days re...** 1:35 PM  
Cloud Zone » 测试Zone-g11n  
The time remaining on cloud zone is less than 60 days until capacity demand runs out.
- Cloud Zone has less than 60 days re...** 1:35 PM  
Cloud Zone » vmc staging-vsphere / SDDC-Datacenter  
The time remaining on cloud zone is less than 60 days until capacity demand runs out.
- AlertDefinition\_ff4d3d96-fa4f-4022-...** 1:26 PM

**The project has some deployments that contain optimizable resources.**  
Created: Dec 14, 2020, 6:17:44 PM | Updated: Dec 14, 2020, 6:17:44 PM  
Project » vc65 project

The project has some deployments that contain optimizable resources.

Severity: Critical Status: Active Impact: Efficiency Type: Application Subtype:

**Suggestions** **REVIEW PROJECT**

- If the project is experiencing increased provisioning, you can review the project to deployments and poweroff/delete the ones that are no longer in use.

**Deployments to review**

Name	Owner
contact-owner-test-dep-2	

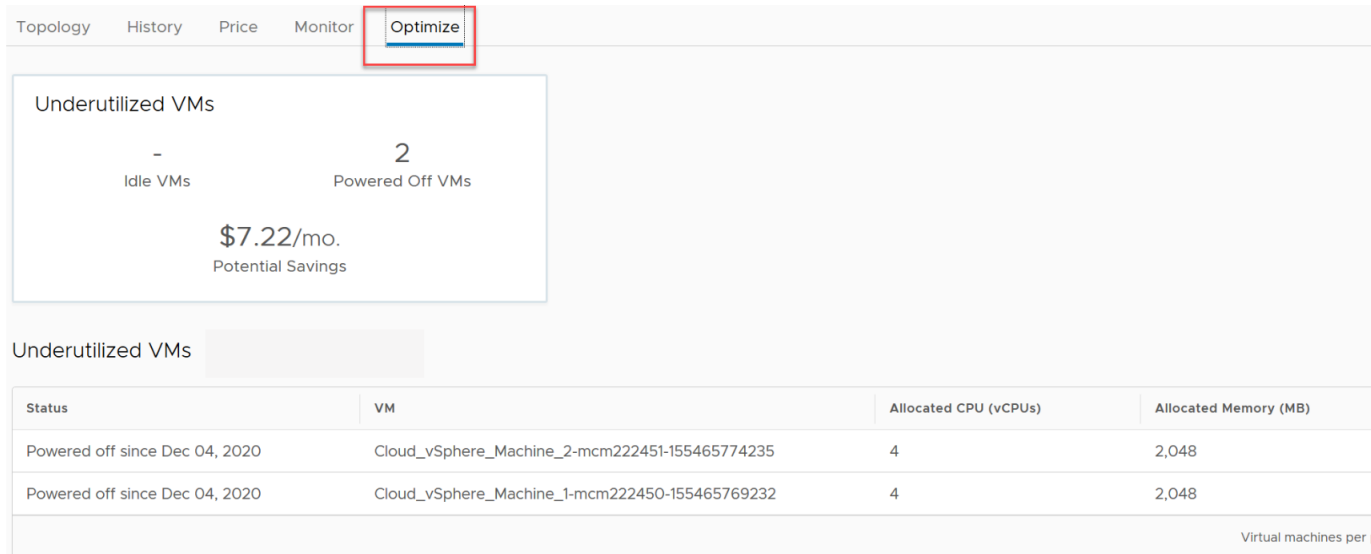
Items per page 10

**Notes**

Investigating

ADD NOTE

- 3 배포가 열리면 **최적화** 탭을 클릭합니다.



- 과소 사용된 시스템이 있는 경우 유휴 시스템과 전원이 꺼진 시스템을 검사하고 작업을 수행합니다. 크기가 부족한 배포의 전원을 끄거나 삭제할 수 있습니다.
- 적정 크기 조정이 가능한 시스템이 있는 경우 배포에서 크기가 초과된 시스템과 크기가 부족한 시스템을 모두 검사하고 작업을 수행합니다.
- 배포에 있는 하나 이상의 시스템에 스냅샷이 포함된 경우에는 각 스냅샷을 삭제하거나 내보낼 수 있습니다.
- 완료되면 배포가 만족스럽게 최적화되었는지 확인하고 배포를 닫습니다.

## 다음 단계

사용할 수 있는 다른 작업에 대해 알아보려면 [vRealize Automation](#)에서 경고를 사용하여 리소스 용량, 성능 및 가용성을 관리하는 방법의 내용을 참조하십시오.

또한 관리하는 프로젝트의 클라우드 영역 기반 리소스에 대한 용량 [인사이트](#)를 표시할 수 있습니다.

vRealize Automation에서 vRealize Operations Manager 제공 [인사이트](#) 데이터를 사용하는 방법에 대한 자세한 내용은 [vRealize Automation](#)에서 인사이트 대시보드를 사용하여 리소스 용량을 모니터링하고 프로젝트 소유자에게 최적화할 수 있는 배포에 대해 알리는 방법 항목을 참조하십시오.

## vRealize Automation에서 표준 디스크 스토리지로 수행할 수 있는 작업

표준 디스크는 영구 디스크이거나 비영구 디스크일 수 있습니다.

vRealize Automation는 표준 디스크 및 FCD(First Class Disk)라는 두 가지 범주의 스토리지를 지원합니다. FCD(First Class Disk)는 vSphere에만 사용할 수 있습니다.

### ■ vSphere

vSphere는 종속(기본값), 독립형 영구, 독립형 비영구 표준 디스크를 지원합니다. 관련 정보는 [vRealize Automation](#)에서 [영구 디스크 스토리지로 수행할 수 있는 작업](#) 항목을 참조하십시오.

가상 시스템을 삭제하면 종속 및 독립형 비영구 디스크도 삭제됩니다.

가상 시스템을 삭제해도 독립형 영구 디스크는 삭제되지 않습니다.

중속 및 독립형 비영구 디스크의 스냅샷을 생성할 수 있습니다. 독립형 영구 디스크의 스냅샷은 생성할 수 없습니다.

#### ■ AWS(Amazon Web Services) EBS

EBS 볼륨을 AWS 계산 인스턴스에 연결하거나 AWS 계산 인스턴스에서 EBS 볼륨을 분리할 수 있습니다.

가상 시스템을 삭제하면 연결된 EBS 볼륨이 분리되지만 삭제되지는 않습니다.

#### ■ Microsoft Azure VHD

연결된 디스크는 항상 영구적입니다.

가상 시스템을 삭제할 때 연결된 스토리지 디스크를 제거할지 여부를 지정합니다.

#### ■ GCP(Google Cloud Platform)

연결된 디스크는 항상 영구적입니다.

영구 디스크는 가상 시스템 인스턴스와 독립적으로 배치되므로 인스턴스를 삭제한 후에도 영구 디스크를 분리하거나 이동하여 데이터를 보존할 수 있습니다.

가상 시스템을 삭제하면 연결된 디스크가 분리되지만 삭제되지는 않습니다.

관련 정보는 [vRealize Automation의 스토리지 프로파일에 대해 알아보기](#) 항목을 참조하십시오.

### vRealize Automation에서 영구 디스크 스토리지로 수행할 수 있는 작업

영구 디스크는 중요한 데이터가 실수로 삭제되지 않도록 보존합니다.

클라우드 템플릿에서 볼륨 아래에 `persistent: true` 속성을 추가하여 디스크가 vRealize Automation Cloud Assembly 또는 vRealize Automation Service Broker 삭제 후에도 유지되도록 할 수 있습니다. 영구 디스크는 배포 삭제, 2일차 삭제 또는 디스크 제거 작업 동안 제거되지 않습니다.

따라서 영구 디스크는 배포 삭제 또는 디스크 삭제 후에도 인프라에 남아 있을 수 있습니다. 이러한 항목을 제거하려면 다음 기술을 사용할 수 있습니다.

■ DELETE API를 사용하여 지우기 플래그를 쿼리 매개 변수로 명시적으로 전달합니다.

■ 클라우드 끝점에서 직접 삭제합니다.

영구 디스크 제거를 위한 vRealize Automation Cloud Assembly 또는 vRealize Automation Service Broker 사용자 인터페이스는 없습니다.

### vRealize Automation에서 FCD(First Class Disk) 스토리지로 수행할 수 있는 작업

FCD(First Class Disk)는 가상 디스크에 대한 스토리지 수명 주기 관리를 서비스로 제공되는 디스크 또는 EBS와 유사한 디스크 스토리지로 제공하기 때문에 vSphere 가상 시스템에 관계 없이 디스크를 생성하고 관리할 수 있습니다.

vRealize Automation는 표준 디스크 및 FCD(First Class Disk)라는 두 가지 범주의 스토리지 디스크를 지원합니다. FCD(First Class Disk) 기능은 vSphere에만 지원됩니다. 현재 vRealize Automation는 FCD(First Class Disk) 기능을 API 전용 기능으로 제공합니다.

FCD(First Class Disk)에는 VM과 독립적으로 작동하는 자체 수명 주기 관리 기능이 있습니다. FCD(First Class Disk)가 독립형 영구 디스크와 다른 한 가지 방식은 FCD(First Class Disk)를 사용하면 VM과 별도로 스냅샷을 생성하고 관리할 수 있다는 점입니다.

새 vRealize Automation 스토리지 프로파일을 생성하여 FCD(First Class Disk) 기능 또는 표준 디스크 기능을 지원할 수 있습니다. [vRealize Automation의 스토리지 프로파일에 대해 알아보기](#) 및 [vRealize Automation의 스토리지 리소스 항목을 참조하십시오](#).

vRealize Automation 클라우드 템플릿 및 배포에 Cloud.vSphere.Disk FCD(First Class Disk) 요소를 추가하여 vSphere FCD(First Class Disk)를 지원할 수도 있습니다. 데이터 수집된 FCD(First Class Disk)는 [볼륨 리소스](#) 페이지에 표시됩니다. [vRealize Automation의 볼륨 리소스](#)의 내용을 참조하십시오.

vCenter에서는 FCD(First Class Disk)를 "IVD(향상된 가상 디스크)" 또는 "관리 가상 디스크" 라고도 합니다.

## 기능

vRealize Automation API 기능을 사용하여 수행할 수 있는 작업:

- FCD(First Class Disk)를 생성, 나열 및 삭제합니다.
- FCD(First Class Disk) 크기를 조정합니다.
- FCD(First Class Disk)를 연결하고 분리합니다.
- FCD(First Class Disk) 스냅샷을 생성하고 관리합니다.
- 기존 표준 디스크를 FCD(First Class Disk)로 변환합니다.

FCD(First Class Disk) 기능을 사용하도록 스토리지 프로파일을 정의하는 방법을 비롯한, vRealize Automation API를 사용하여 FCD(First Class Disk) 스토리지 생성 및 관리에 대한 관련 API 정보는 [code.vmware.com](#)에 있는 [vRealize Automation Cloud API 소개 및 사용하는 방법](#)을 참조하거나 다음 위치에서 탐색하여 참조할 수 있습니다.

- FCD에 대한 API 설명서는 [Virtual Disk Development Kit 프로그래밍 가이드](#)의 [FCD\(First Class Disk\)](#) 섹션에서 참조할 수 있습니다.
- vRealize Automation의 FCD에 대한 API 사용 사례 설명서 링크는 [vRealize Automation 릴리스의 vRealize Automation API 설명서 페이지](#)에서 참조할 수 있습니다.

## 고려 사항 및 제한 사항

FCD(First Class Disk) 고려 사항 및 제한 사항은 현재 다음과 같습니다.

- FCD(First Class Disk)는 vSphere VM에만 사용할 수 있습니다.
- FCD(First Class Disk)를 사용하려면 vSphere 6.7 업데이트 2 이상이 필요합니다.
- 데이터스토어 클러스터에서 FCD(First Class Disk) 프로비저닝은 지원되지 않습니다.

- FCD(First Class Disk)에는 볼륨 다중 연결이 지원되지 않습니다.
- 스냅샷이 있는 FCD(First Class Disk)는 크기를 조정할 수 없습니다.
- 스냅샷이 있는 FCD(First Class Disk)는 삭제할 수 없습니다.
- FCD(First Class Disk) 스냅샷 계층은 `createdAt` API 옵션을 사용해서만 구성할 수 있습니다.
- FCD(First Class Disk)를 연결하는 데 필요한 최소 VM 하드웨어 버전은 `vmx-13`(ESX 6.5 호환)입니다.

## vRealize Automation으로 다중 제공자 테넌트 리소스 구성

다중 테넌시 환경에서 고객은 VPZ(가상 개인 영역)를 사용하여 테넌트별로 리소스 할당을 관리할 수 있습니다.

vRealize Automation 8.x에서 고객은 VMware Lifecycle Manager 및 Workspace ONE Access를 사용하여 다중 테넌시 환경을 구성할 수 있습니다. 이러한 도구를 통해 사용자는 다중 테넌시를 설정하고 테넌트를 생성 및 구성할 수 있습니다. 테넌트가 구성되면 제공자 관리자는 vRealize Automation Cloud Assembly에서 가상 개인 영역을 생성한 후 vRealize Automation Cloud Assembly 테넌트 관리 기능을 사용하여 영역을 테넌트에 할당할 수 있습니다.

다중 테넌시는 아래에 설명된 것과 같이 3개의 다른 VMware 제품의 조정 및 구성에 의존합니다.

- **Workspace ONE Access** - 이 제품은 다중 테넌시에 대한 인프라 지원 및 테넌트 조직 내에서 사용자 및 그룹 관리를 제공하는 Active Directory 도메인 연결을 제공합니다.
- **vRealize Suite Lifecycle Manager** - 이 제품은 vRealize Automation과 같은 지원되는 제품에 대한 테넌트의 생성 및 구성을 지원합니다. 또한 몇 가지 인증서 관리 기능을 제공합니다.
- **vRealize Automation** - 제공자 및 사용자는 vRealize Automation에 로그인하여 배포를 생성하고 관리하는 테넌트에 액세스합니다.

다중 테넌시를 구성하는 경우 사용자는 3개의 모든 제품과 관련 설명서를 숙지해야 합니다.

Lifecycle Manager 및 Workspace ONE Access 작업에 대한 자세한 내용은 [VMware Identity Manager](#)를 사용한 사용자 관리 및 사용자 및 그룹 관리의 내용을 참조하십시오.

## vRealize Automation용 가상 개인 영역을 생성하는 방법

제공자 관리자는 VPZ(가상 개인 영역)를 생성하여 다중 조직 vRealize Automation 환경의 테넌트에 인프라 리소스를 할당할 수 있습니다. 관리자가 VPZ를 사용하여 단일 테넌트 배포에서 리소스 할당을 제어할 수도 있습니다.

가상 개인 영역을 사용하여 이미지, 네트워크, 스토리지 리소스와 같은 리소스를 할당할 수 있습니다. VPZ는 테넌트 단위로 클라우드 영역처럼 작동하지만 다중 테넌트 배포에 사용하도록 특별히 설계되었습니다. 특정 프로젝트에 클라우드 영역 또는 VPZ 중 하나만 사용할 수 있고 둘 다 사용할 수는 없습니다. 또한 VPZ와 테넌트 사이에는 일대일 관계가 있습니다. 즉, VPZ는 한 번에 하나의 테넌트에만 할당될 수 있습니다.

---

**참고** [테넌트 관리] 페이지에서 VPZ에 대한 이미지 및 버전 매핑을 구성합니다.

---

NSX를 사용하거나 사용하지 않고 VPZ를 생성할 수 있습니다. NSX 없이 영역을 생성하면 vSphere 끝점의 NSX 관련 기능에 대한 제한이 있습니다.

- 보안(그룹, 방화벽)
- 네트워크 구성 요소(NAT)

#### 사전 요구 사항

- VMware Lifecycle Manager 및 VMware Workspace ONE Access를 사용하여 vRealize Automation 배포에서 멀티 테넌시를 사용하도록 설정하고 구성합니다.
- 테넌트 구성에 적절하게 테넌트 관리자를 생성합니다.
- NSX를 사용하려면 제공자 조직에 적절한 NSX 클라우드 계정을 생성해야 합니다.

#### 절차

##### 1 인프라 > 구성 > 가상 개인 영역을 선택합니다.

VPZ 페이지에는 기존 영역이 모두 표시되고 영역을 생성할 수 있습니다.

## 2 새 가상 개인 영역을 클릭합니다.

### New Virtual Private Zone

영역에 대한 인프라 구성 요소 및 요약 정보를 구성하는 데 사용할 수 있는 4가지 선택 항목이 페이지 왼쪽에 있습니다.

## 3 새 영역에 대한 요약 정보를 입력합니다.

- a 이름과 설명을 추가합니다.
- b 영역을 적용할 계정을 선택합니다.
- c 배치 정책을 선택합니다.

배치 정책은 지정된 클라우드 영역 내에서 배포할 호스트를 선택하는 데 사용됩니다.

- **Default** - 클러스터와 호스트 전체에 계산 리소스를 임의로 배포합니다. 이 선택은 개별 시스템 수준에서 작동합니다. 예를 들어, 특정 배포의 모든 시스템이 요구 사항을 충족하는 사용 가능한 클러스터와 호스트 전체에 무작위로 배포됩니다.
- **binpack** - 지정된 계산을 실행하기에 충분한 가용 리소스가 있는, 가장 많이 로드된 호스트에 계산 리소스를 배치합니다.
- **spread** - 가상 시스템 수가 가장 적은 호스트 또는 클러스터에 배포 계산 리소스를 프로비저닝합니다. vSphere의 경우 DRS(Distributed Resource Scheduler)는 호스트 전체에 가상 시스템을 분산합니다. 예를 들어 배포에서 요청된 모든 시스템은 동일한 클러스터에 배치되지만 다음 배포에서는 현재 로드율에 따라 다른 vSphere 클러스터를 선택할 수 있습니다.

#### 4 영역에 대한 계산 리소스를 선택합니다.

클라우드 영역에 적합하게 계산 리소스를 추가합니다. 초기에 필터 선택은 [모든 계산 포함]이며, 다음 목록에는 사용 가능한 모든 계산 리소스가 표시되고 해당 영역에 할당됩니다. 계산 리소스를 클라우드 영역에 추가하는 두 가지 추가 옵션이 있습니다.

- 수동으로 계산 선택 - 아래 목록에서 계산 리소스를 수동으로 선택하려면 이 메뉴 항목을 선택합니다. 선택한 후에 [계산 추가]를 클릭하여 리소스를 영역에 추가합니다.
- 동적으로 태그별 계산 포함 - 태그를 기반으로 영역에 추가할 계산 리소스를 선택하려면 이 메뉴 항목을 선택합니다. 적절한 태그를 추가할 때까지 모든 계산 리소스가 표시됩니다. 이러한 태그를 통해 계산을 포함하는 옵션에서 하나 이상의 태그를 선택하거나 입력할 수 있습니다.

어느 계산 선택이든, 오른쪽에 있는 상자를 선택하고 [제거]를 클릭하면 페이지에 보이는 계산 리소스를 하나 이상 제거할 수 있습니다.

#### 5 적절한 태그를 입력하거나 선택합니다.

#### 6 왼쪽 메뉴에서 [스토리지]를 선택하고 영역에 대한 [스토리지 정책] 및 기타 스토리지 구성을 선택합니다.

#### 7 왼쪽 메뉴에서 [네트워크]를 선택하여 네트워크를 정의하고 이 영역에 사용할 네트워크 정책(선택 사항)을 정의합니다. 선택한 네트워크 정책에 대한 로드 밸런서 및 보안 그룹을 구성할 수도 있습니다.

네트워크	<ul style="list-style-type: none"> <li>■ 이 VPZ와 연결된 모든 기존 네트워크는 [네트워크] 탭의 표에 표시됩니다.</li> <li>■ <b>네트워크 추가</b>를 클릭하면 선택한 지역과 연결된 모든 네트워크를 볼 수 있습니다. 이 영역에서 사용할 네트워크를 추가합니다.</li> <li>■ 네트워크를 선택하고 <b>태그</b>를 클릭하여 지정된 네트워크에 하나 이상의 태그를 추가합니다.</li> <li>■ <b>IP 범위 관리</b>를 선택하여 사용자가 이 네트워크에 액세스할 수 있는 IP 범위를 지정합니다.</li> <li>■ 해당되는 경우 [네트워크 정책] 탭을 클릭하고 격리 정책을 선택합니다.</li> </ul>
네트워크 정책	<p>구성되어 있으면, 이 영역에서 사용할 네트워크 정책을 선택하여 아웃바운드 및 프라이빗 네트워크에 대한 격리 정책을 적용합니다.</p> <ul style="list-style-type: none"> <li>■ 필요한 경우 격리 정책을 선택합니다.</li> <li>■ 필요한 경우 Tier-O 논리적 라우터 및 Edge 클러스터를 선택합니다.</li> </ul>
로드 밸런서	<b>로드 밸런서 추가</b> 를 클릭하면 계정/지역 클라우드 계정에 대한 로드 밸런서를 구성할 수 있습니다.
보안 그룹	<b>보안 그룹 추가</b> 를 클릭하면 보안 그룹을 사용하여 프로비저닝된 시스템에 방화벽 규칙을 적용할 수 있습니다.

## 결과

지정된 리소스 할당을 사용하여 가상 개인 영역이 생성됩니다.

## 다음에 수행할 작업

클라우드 관리자는 VPZ를 프로젝트와 연결할 수 있습니다.

- 1 Cloud Assembly에서 **관리 > 프로젝트**를 선택합니다.
- 2 [프로비저닝] 탭을 선택합니다.
- 3 **영역 추가**를 클릭하고 [가상 개인 영역 추가]를 선택합니다.
- 4 목록에서 원하는 VPZ를 선택합니다.
- 5 프로비저닝 우선 순위와 인스턴스 수, 사용 가능한 메모리 양, 사용 가능한 CPU 수에 대한 제한을 설정할 수 있습니다.
- 6 **추가**를 클릭합니다.

## vRealize Automation 테넌트에 대한 가상 개인 영역 구성 관리

제공자 관리자는 vRealize Automation Cloud Assembly 내에서 VPZ(가상 개인 영역)를 관리하여 테넌트별로 인프라 리소스 할당을 제어할 수 있습니다. 관리자는 [테넌트 관리] 페이지를 사용하여 테넌트 및 VPZ 영역을 보고 테넌트에 VPZ를 사용하거나 사용하지 않도록 설정할 수 있습니다.

기본적으로 가상 개인 영역은 어떤 테넌트에도 할당되지 않습니다. 테넌트가 사용할 수 있도록 이 페이지에 VPZ를 할당해야 합니다.

처음 생성 시 VPZ는 기본적으로 사용되도록 설정됩니다. 사용되도록 설정된 VPZ는 지정된 테넌트와 함께 할당되고 사용될 준비가 된 것입니다. VPZ를 사용하지 않도록 설정하면 프로비저닝에 사용하거나 테넌트에 할당할 수 없습니다. VPZ를 사용하지 않도록 설정할 수 있지만 테넌트에 대해 계속 할당할 수 있습니다.

제공자 관리자가 [테넌트 관리] 페이지로 이동하면 페이지에 사용 가능한 모든 테넌트가 표시되고 관리자가 하나를 선택할 수 있습니다. 테넌트를 선택하면 페이지에 해당 테넌트에 대해 현재 할당된 VPZ가 표시됩니다(있는 경우). 관리자는 이 페이지를 사용하여 VPZ를 선택한 테넌트에 할당할 수 있습니다.

VPZ가 할당되면 테넌트 관리자가 해당 프로젝트에 추가할 수 있으며 테넌트 사용자가 프로비저닝할 수 있게 됩니다. VPZ가 하나의 테넌트에 할당된 후에 다른 테넌트에 할당할 수 있습니다.

VPZ가 사용되도록 설정된 후에는 지정된 테넌트 내에서 사용할 준비가 된 것입니다. 제공자 관리자는 VPZ를 사용하지 않도록 설정하여 유지 보수 또는 테넌트 재구성을 용이하게 하고 사용자에게 사용하지 않도록 설정 알림을 제공할 수 있습니다. 테넌트가 VPZ를 보다 영구적으로 사용할 수 없게 하려면 할당을 취소하면 됩니다. 어떤 이유로 테넌트에서 기존 VPZ가 할당 취소되면 해당 테넌트에서 배포를 생성하는 데 사용할 수 없습니다.

### 사전 요구 사항

- 다중 테넌시를 설정하고 배포에 적합한 가상 개인 영역을 생성합니다.
- Cloud Assembly의 [테넌트 관리] 페이지 왼쪽에 있는 이미지 매핑 및 버전 매핑 메뉴 선택을 사용하여 VPZ 구성과 테넌트 구성에 대한 글로벌 이미지 및 버전 매핑을 구성합니다. [vRealize Automation 테넌트에 대한 글로벌 이미지 및 버전 매핑 생성](#)의 내용을 참조하십시오.

[테넌트 관리] 페이지의 맨 위에 있는 테넌트 특정 이미지 및 버전 매핑 선택을 사용하여 지금 또는 나중에 이러한 글로벌 할당을 재정의할 수 있습니다. **vRealize Automation에 대한 테넌트 특정 이미지 및 버전 매핑 구성**의 내용을 참조하십시오.

## 절차

- 1 vRealize Automation Cloud Assembly에서 [테넌트 관리]를 선택합니다.

[테넌트 관리] 페이지에는 카드 보기에서 관리자 조직에 대해 구성된 모든 테넌트가 표시됩니다.

- 2 테넌트를 클릭하여 선택합니다.
- 3 [인프라 관리] 탭을 클릭하여 테넌트에 대해 할당된 모든 VPZ를 확인합니다.
- 4 **개인 가상 영역 할당**을 선택하여 테넌트에 현재 할당되지 않은 모든 영역을 표시하는 대화 상자를 엽니다. 영역을 테넌트에 할당합니다.
- 5 대화 상자에서 하나 이상의 영역을 선택하고 **테넌트에 할당**을 클릭합니다.

## 다음에 수행할 작업

VPZ가 할당되면 테넌트 관리자가 프로젝트에 할당할 수 있습니다.

제공자 관리자는 테넌트의 카드 보기를 사용하여 VPZ의 상태를 모니터링하고 관리할 수 있습니다.

- 테넌트를 사용하지 않도록 설정하려면 테넌트에 대한 카드에서 **사용 안 함**을 클릭합니다.
- 테넌트를 사용하도록 설정하려면 테넌트에 대한 카드에서 **사용**을 클릭합니다.
- 테넌트의 할당을 취소하려면 해당 테넌트에 대한 카드에서 **할당 취소**를 클릭합니다.

## vRealize Automation 테넌트에 대한 글로벌 이미지 및 버전 매핑 생성

제공자 관리자는 vRealize Automation 테넌트에 할당할 수 있는 글로벌 이미지 및 버전 매핑을 선택하거나 생성할 수 있습니다.

글로벌 이미지 및 버전 매핑을 사용하면 여러 테넌트에 적용되는 매핑을 신속하게 설정할 수 있습니다. 또한 이러한 매핑을 빠르게 업데이트할 수도 있습니다. [테넌트 관리] 페이지에서는 기본 구성을 재정의할 수 있는 테넌트 특정 이미지 및 버전 매핑을 생성할 수도 있습니다.

---

**참고** [테넌트 관리] 페이지에 구성된 이미지 및 버전 매핑은 구성된 테넌트에만 적용되고 더 광범위한 제공자 조직에는 적용되지 않습니다.

---

## 사전 요구 사항

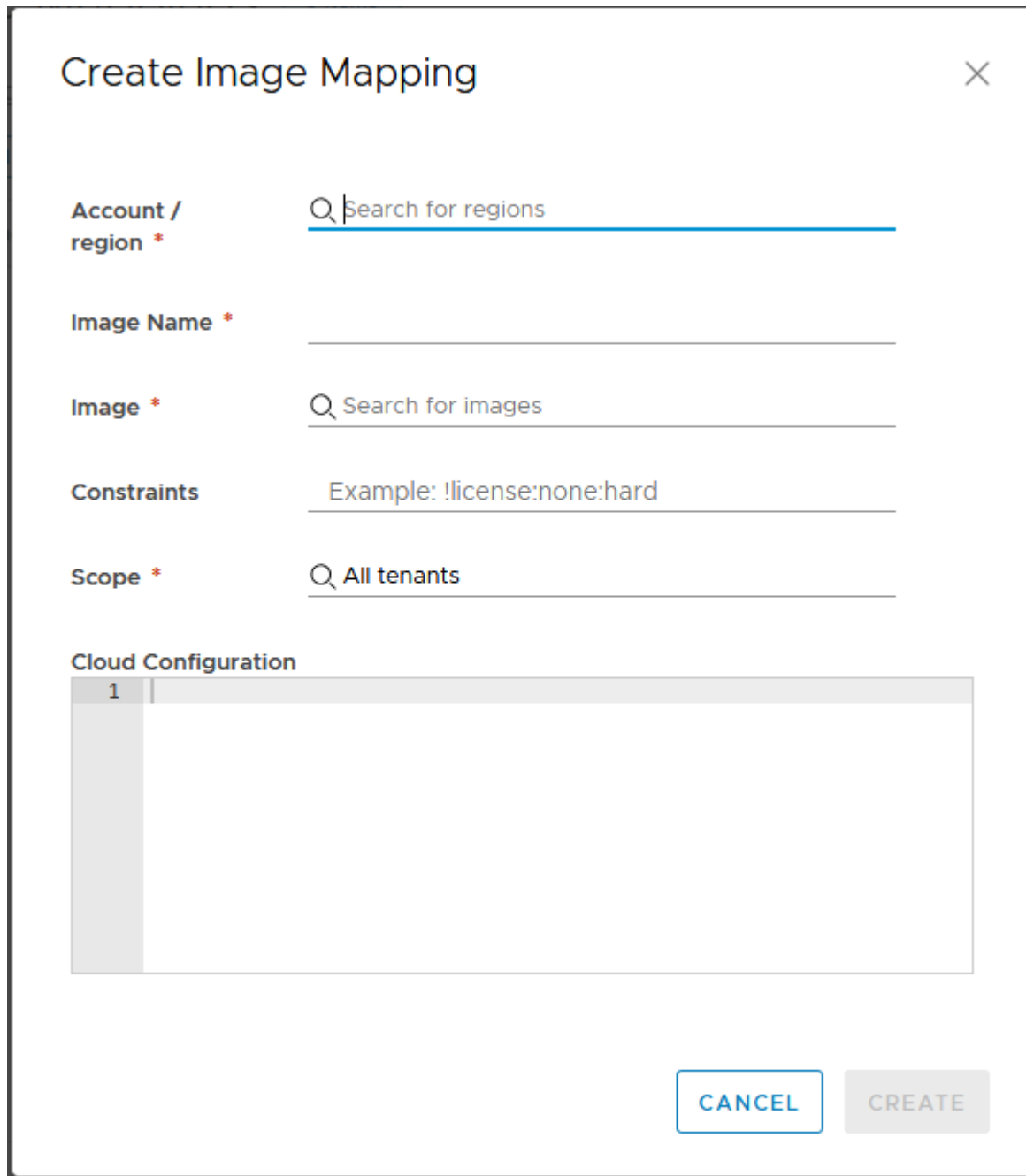
## 절차

- 1 vRealize Automation Cloud Assembly에서 [테넌트 관리]를 선택합니다.

[테넌트 관리] 페이지에는 카드 보기에서 관리자 조직에 대해 구성된 모든 테넌트가 표시됩니다.

2 [테넌트 관리] 페이지 왼쪽 메뉴에서 [이미지 매핑]을 선택합니다.

[이미지 매핑] 페이지에는 현재 Cloud Assembly에서 테넌트에 대해 구성된 모든 이미지가 표시되고, 매핑이 글로벌인지 아니면 특정 테넌트에 연결되었는지 나타냅니다.



The dialog box titled "Create Image Mapping" contains the following fields and sections:

- Account / region \***: A search field with the placeholder text "Search for regions".
- Image Name \***: A text input field.
- Image \***: A search field with the placeholder text "Search for images".
- Constraints**: A text input field with the placeholder text "Example: !license:none:hard".
- Scope \***: A search field with the placeholder text "All tenants".
- Cloud Configuration**: A table with one column and one row. The row is labeled "1" in the header.

At the bottom right of the dialog are two buttons: "CANCEL" and "CREATE".

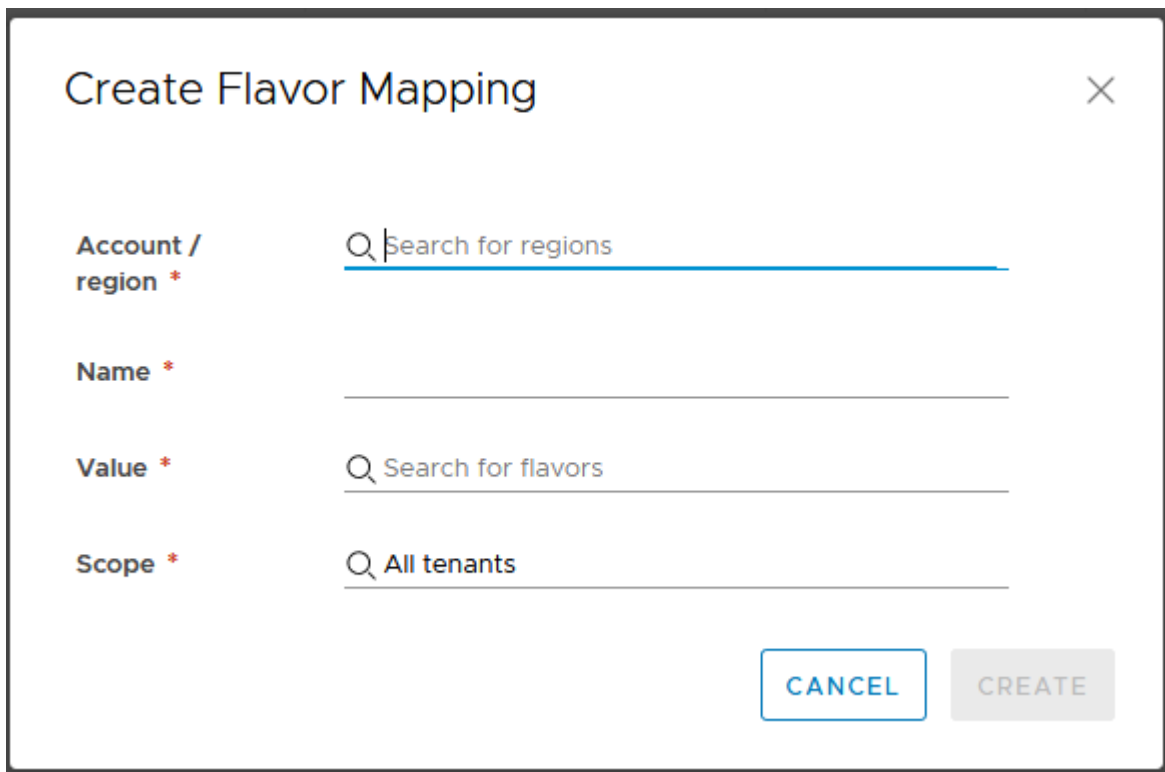
3 **이미지 매핑 추가**를 선택하여 테넌트에 사용할 이미지 매핑을 추가합니다.

- a 이미지 매핑을 적용할 계정/지역을 선택합니다.
- b 이미지 매핑의 이름을 입력하고 관련된 특정 이미지 인스턴스 또는 버전을 선택합니다.

- c 원하는 제약 조건 태그를 입력합니다.
  - d 이미지 매핑의 범위를 선택합니다. 범위로 모든 테넌트 또는 글로벌을 선택하거나 이미지 매핑을 적용할 특정 테넌트를 선택할 수도 있습니다.
- 4 필요한 경우 클라우드 구성 스크립트를 사용하여 배포에 대한 사용자 지정 OS 특성을 정의할 수 있습니다.

예를 들어 클라우드 템플릿을 공용 또는 사설 클라우드 중 어디에 배포하는지에 따라 특정 사용자 사용 권한, OS 사용 권한 또는 기타 조건을 이미지에 적용할 수 있습니다. 클라우드 구성 스크립트는 Linux 기반 이미지에 대해서는 cloud-init 형식을 따르고 Windows 기반 이미지에 대해서는 cloudbase-init 형식을 따릅니다. 자세한 내용은 [vRealize Automation의 이미지 매핑에 대해 알아보기](#) 항목을 참조하십시오.

- 5 **생성**을 클릭하여 이미지 매핑을 생성합니다.
- 6 **버전 매핑 추가**를 선택하여 테넌트에 사용할 버전 매핑을 추가합니다.



The image shows a 'Create Flavor Mapping' dialog box with a close button (X) in the top right corner. It contains four input fields, each with a magnifying glass icon and a placeholder text:

- Account / region \***: Search for regions
- Name \***: (empty text field)
- Value \***: Search for flavors
- Scope \***: All tenants

At the bottom right, there are two buttons: 'CANCEL' (outlined) and 'CREATE' (solid gray).

- a 버전 매핑을 적용할 계정/지역을 선택합니다.
  - b 생성 중인 버전 매핑의 이름을 입력합니다.
  - c 생성 중인 버전 매핑의 크기 매개 변수를 선택합니다.
- 이 버전에 대한 프로세서 수와 메모리 양을 지정할 수 있습니다.
- d 버전 매핑의 범위를 선택합니다. 범위로 모든 테넌트 또는 글로벌을 선택하거나 버전 매핑을 적용할 특정 테넌트를 선택할 수도 있습니다. 모든 테넌트가 제공자 관리자 조직의 모든 테넌트에 적용됩니다.

7 **생성**을 클릭하여 버전 매핑을 생성합니다.

## 결과

글로벌 매핑을 생성한 후에는 해당하는 테넌트에 대한 [테넌트 관리] 페이지의 [버전 매핑] 또는 [테넌트 매핑] 탭에 이러한 매핑이 표시됩니다.

## 다음에 수행할 작업

이 페이지에서 글로벌 이미지 및 버전 매핑을 편집하거나 삭제할 수 있습니다. 매핑을 편집하려면 매핑을 선택하고 원하는 대로 변경합니다.

## vRealize Automation에 대한 테넌트 특정 이미지 및 버전 매핑 구성

vRealize Automation Cloud Assembly를 사용하면 조직 내의 모든 VPZ(가상 개인 영역)에서 사용할 수 있는 글로벌 이미지 및 버전 매핑을 구성할 수 있습니다. 또는 글로벌 설정을 재정의하고 배포에 적절하게 테넌트 특정 이미지 및 버전 매핑을 구성할 수 있습니다.

일반적으로 클라우드 관리자는 [테넌트 관리] 페이지의 왼쪽 탐색 링크를 사용하여 글로벌 이미지 및 버전 매핑을 구성하며 이러한 매핑은 모든 테넌트에 대해 보드 전체에 적용됩니다. 경우에 따라 특정 테넌트에 대해 사용자 지정 및 테넌트 특정 이미지 매핑과 버전 매핑을 생성할 수 있습니다. [테넌트 관리] 페이지는 이 옵션을 지원합니다.

이미지 및 버전 매핑은 [테넌트 관리] 페이지의 해당 탭에 표시됩니다. 기존 이미지 및 버전 매핑을 편집하려면 해당 매핑을 클릭합니다. 이미지 또는 버전 매핑을 삭제하려면 매핑을 선택하고 **삭제**를 클릭합니다.

## 사전 요구 사항

- 다중 테넌시를 사용하도록 설정하고 배포에 대해 테넌트를 구성합니다.
- 적절한 VPZ를 생성합니다.

## 절차

- 1 Cloud Assembly 기본 메뉴에서 [테넌트 관리]를 선택합니다.
- 2 사용자 지정 이미지 또는 버전 매핑을 구성할 테넌트를 선택합니다.
- 3 페이지 맨 위에서 [이미지 매핑] 링크를 선택하고 **이미지 매핑 추가**를 클릭합니다.  
[이미지 매핑 생성] 대화 상자가 나타납니다.
- 4 지정된 계정/지역이 올바른지 확인하고 **이미지 이름** 텍스트 상자에 매핑의 이름을 추가합니다.
- 5 **이미지** 드롭다운에서 사용할 기본 시스템 이미지를 선택합니다.
- 6 해당하는 경우 이미지 사용에 대한 제약 조건 태그를 추가합니다.
- 7 이미지에 대해 적절한 **범위**를 선택합니다.
  - 이 이미지 매핑을 선택한 테넌트에서만 사용할 수 있게 하려면 [이 테넌트에서만 사용 가능] 라디오 버튼을 클릭합니다.

- 이 이미지 매핑을 다른 테넌트에서 사용할 수 있게 하려면 [테넌트 간에 공유] 라디오 버튼을 클릭합니다.
- 8 이미지 매핑을 구성된 상태로 저장하려면 **생성**을 클릭합니다.
  - 9 페이지 맨 위에서 [버전 매핑] 링크를 선택하고 **버전 매핑 추가**를 클릭하여 버전 매핑을 생성합니다.  
[버전 매핑 생성] 대화 상자가 나타납니다.
  - 10 지정된 계정/지역이 올바른지 확인하고 **이름** 텍스트 상자에 매핑의 이름을 추가합니다.
  - 11 **값** 필드에서 버전 CPU 및 메모리 설정을 지정합니다.
  - 12 이미지에 대해 적절한 **범위**를 선택합니다.
    - 이 이미지 매핑을 선택한 테넌트에서만 사용할 수 있게 하려면 [이 테넌트에서만 사용 가능] 라디오 버튼을 클릭합니다.
    - 이 이미지 매핑을 다른 테넌트에서 사용할 수 있게 하려면 [테넌트 간에 공유] 라디오 버튼을 클릭합니다.
  - 13 버전 매핑을 구성된 상태로 저장하려면 **생성**을 클릭합니다.

## 결과

테넌트 특정 이미지 및 버전 매핑은 지정된 대로 구성됩니다.

## 최신 버전의 vRealize Automation에서 기존 가상 개인 영역 사용

VPZ에 대한 구성 옵션이 vRealize Automation Cloud Assembly에서 변경되었습니다. 현재 버전의 vRealize Automation에서 기존 가상 개인 영역을 업데이트하거나 사용할 수 있습니다.

vRealize Automation 8.2에서는 사용자가 VPZ 내에서 이미지 및 버전 매핑을 구성했습니다. 최신 버전의 vRealize Automation에서는, 사용자가 테넌트별로 이미지 및 버전 매핑을 생성하므로, 특히 테넌트의 수가 많은 배포에서 효율성과 구성 유연성이 향상됩니다. vRealize Automation 8.2에서 생성된 기존 VPZ를 마이그레이션할 수 있는 방법은 없지만 최신 버전의 vRealize Automation에서 사용할 수 있는 몇 가지 옵션이 있습니다.

가장 유연한 첫 번째 옵션은 기존 VPZ에서 레거시 이미지 및 버전 매핑을 삭제하고 [테넌트 관리] 페이지에서 생성된 새 매핑으로 재구성하는 것입니다.

- 1 **인프라 > 구성 > 가상 개인 영역**을 선택하여 VPZ 페이지를 엽니다.
- 2 [이미지 매핑]을 선택하여 기존 매핑을 봅니다.
- 3 매핑을 선택하고 클릭하여 삭제합니다.
- 4 [이미지 매핑]을 선택하여 기존 매핑을 봅니다.
- 5 매핑을 선택하고 클릭하여 삭제합니다.
- 6 VPZ 페이지를 닫습니다.

7 [테넌트 매핑] 및 생성을 선택하고 적용 가능한 테넌트에 대한 전역 매핑을 선택하거나 테넌트별 매핑을 생성합니다.

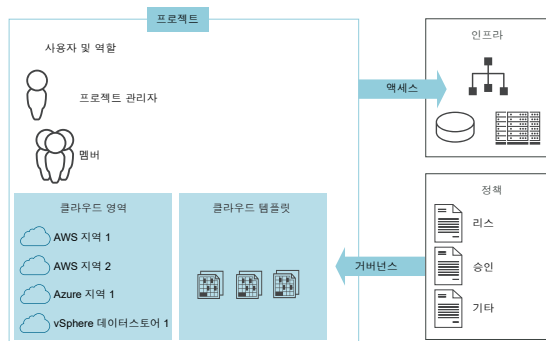
또는 기존 구성으로 최신 버전의 **vRA**와 함께 기존 **VPZ**를 사용할 수 있습니다. 레거시 이미지 및 버전 매핑은 구성된 대로 계속 작동하지만 해당 구성 옵션은 **VPZ** 페이지에서 읽기 전용입니다. 이 옵션은 첫 번째 옵션보다 유연성이 떨어집니다.

# vRealize Automation Cloud Assembly 프로젝트 추가 및 관리

## 5

프로젝트는 vRealize Automation Cloud Assembly 클라우드 템플릿에 액세스할 수 있는 사용자 및 템플릿이 배포되는 위치를 제어합니다. 프로젝트를 사용하면 사용자가 수행할 수 있는 작업과 클라우드 인프라에서 클라우드 템플릿을 배포할 수 있는 클라우드 영역을 구성하고 제어할 수 있습니다.

클라우드 관리자는 사용자 및 클라우드 영역을 추가할 수 있는 프로젝트를 설정합니다. 클라우드 템플릿을 생성하고 배포하는 사용자는 프로젝트 하나 이상의 멤버여야 합니다.



본 장은 다음 항목을 포함합니다.

- [vRealize Automation Cloud Assembly 개발 팀용 프로젝트를 추가하는 방법](#)
- [vRealize Automation Cloud Assembly 프로젝트에 대해 알아보기](#)

## vRealize Automation Cloud Assembly 개발 팀용 프로젝트를 추가하는 방법

프로젝트를 생성하여 멤버와 클라우드 영역을 추가하면 프로젝트 멤버가 클라우드 템플릿을 연결된 영역에 배포할 수 있습니다. vRealize Automation Cloud Assembly 관리자 권한으로 개발 팀을 위한 프로젝트를 생성합니다. 그런 다음 프로젝트 관리자를 할당하거나 프로젝트 관리자 권한으로 작업할 수 있습니다.

클라우드 템플릿 생성 시, 클라우드 템플릿을 연결할 프로젝트를 먼저 선택합니다. 프로젝트는 클라우드 템플릿을 생성하기 전에 존재해야 합니다.

프로젝트가 개발 팀의 비즈니스 요구 사항을 지원하는지 확인합니다.

- 프로젝트에서 팀의 목표를 지원하는 리소스를 제공합니까? 인프라 리소스와 프로젝트가 클라우드 템플릿을 지원하는 방식에 대한 예는 [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트](#) 항목을 참조하십시오.
- 프로젝트 멤버가 배포를 공유하거나 비공개로 처리할 것을 요구 또는 기대합니까? 공유 배포는 배포 중인 멤버뿐만 아니라 [배포] 탭에 있는 모든 프로젝트 멤버가 사용할 수 있습니다. 배포 공유 상태는 언제든지 변경할 수 있습니다.

프로젝트 멤버와 배포를 공유하는 경우 멤버는 동일한 2일차 작업을 실행할 수 있습니다. 2일차 작업을 실행하는 멤버의 기능을 관리하기 위해 vRealize Automation Service Broker에서 2일차 정책을 생성할 수 있습니다. 정책은 vRealize Automation Cloud Assembly 및 vRealize Automation Service Broker 배포에 적용됩니다.

2일차 정책에 대해 자세히 알아보려면 [정책을 사용하여 배포 사용자에게 2일차 작업에 대한 권한을 부여하는 방법](#)을 참조하십시오.

이 절차는 기본 구성만 포함된 초기 프로젝트 생성을 기반으로 합니다. 개발 팀에서 클라우드 템플릿을 생성하고 배포하면 프로젝트를 수정할 수 있습니다. 제약 조건, 사용자 지정 속성 및 기타 옵션을 추가하여 배포 효율성을 향상시킬 수 있습니다. [vRealize Automation Cloud Assembly 프로젝트에 대해 알아보기](#)에 있는 문서를 참조하십시오.

### 사전 요구 사항

- 클라우드 영역을 구성했는지 확인합니다. [장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축](#) 항목을 참조하십시오.
- 이 프로젝트의 클라우드 영역으로 포함된 영역에 대한 매핑 및 프로파일을 구성했는지 확인합니다. [장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축](#) 항목을 참조하십시오.
- 이 작업을 수행하는 데 필요한 사용 권한이 있는지 확인합니다. [vRealize Automation 사용자 역할이란?](#)의 내용을 참조하십시오.
- 프로젝트 관리자로 지정할 사용자를 결정합니다. 프로젝트 관리자가 vRealize Automation Cloud Assembly에서 수행할 수 있는 작업을 이해하려면 [vRealize Automation 사용자 역할이란?](#) 항목을 참조하십시오.
- 프로젝트에 Active Directory 그룹을 추가하는 경우 조직에 대한 Active Directory 그룹을 구성했는지 확인합니다. "vRealize Automation 관리"에 있는 [vRealize Automation에서 그룹 역할 할당 편집](#)을 참조하십시오. 그룹이 동기화되지 않은 경우 프로젝트에 추가하려고 할 때 사용할 수 없습니다.

### 절차

- 1 **인프라 > 관리 > 프로젝트**를 선택하고 **새 프로젝트**를 클릭합니다.
- 2 프로젝트 이름을 입력합니다.

### 3 사용자 탭을 클릭합니다.

- a 프로젝트 멤버의 배포에 요청하는 사용자만 액세스할 수 있게 하려면 **배포 공유**를 해제합니다. 배포의 소유권을 프로젝트의 다른 멤버에게 할당할 수 있도록 하려면 **배포 공유**가 설정되어 있는지 확인합니다.
- b 할당된 역할이 있는 사용자를 추가합니다.

### 4 프로비저닝 탭을 클릭하고 하나 이상의 클라우드 영역을 추가합니다.

프로젝트 사용자가 배포한 클라우드 템플릿을 지원하는 리소스가 포함된 클라우드 영역과 가상 개인 영역을 추가합니다.

각 영역에 대해 영역 우선 순위를 설정하고 프로젝트에 사용할 수 있는 리소스 양을 제한할 수 있습니다. 가능한 제한에는 인스턴스, 메모리 및 CPU의 수가 포함됩니다. vSphere 클라우드 영역의 경우에만 vSphere VM 템플릿을 기반으로 배포된 리소스에 대한 스토리지 제한을 구성할 수 있습니다. 스토리지 제한은 배포 요청 시 평가되며 디스크 크기 조정, 부팅 디스크 크기 조정, 디스크 제거 및 업데이트 수 작업을 사용하여 변경할 때 평가됩니다. 이러한 스토리지 제한은 AWS, Microsoft Azure 또는 Google Cloud Platform과 같은 다른 리소스 유형에는 적용되지 않습니다.

각 영역을 추가하고 제한을 적용할 때는 멤버가 자신의 클라우드 템플릿을 배포할 수 없을 정도로 너무 심하게 프로젝트 리소스를 제한하지 마십시오.

사용자가 배포 요청을 제출하면 영역을 평가하여 배포를 지원할 리소스가 있는 영역을 판단합니다. 둘 이상의 영역이 배포를 지원하는 경우에는 우선 순위가 높은 영역(최하 정수)에 워크로드가 배치됩니다.

### 5 이 프로젝트에 대해 요청된 워크로드를 배포하는 데 2시간이 넘게 걸릴 경우 **시간 초과**에 더 긴 값을 입력합니다.

기본값은 2시간입니다.

### 6 생성을 클릭합니다.

### 7 프로젝트 클라우드 영역을 사용하여 프로젝트를 테스트하려면 [프로젝트] 페이지에서 **구성 테스트**를 클릭합니다.

시뮬레이션은 프로젝트 클라우드 영역 리소스에 대해 표준화된 가상의 배포 테스트를 실행합니다. 실패하는 경우 세부 정보를 검토하고 리소스 구성을 수정할 수 있습니다.

#### 다음에 수행할 작업

클라우드 템플릿을 시작합니다. [장 6 vRealize Automation Cloud Assembly 배포 설계](#) 항목을 참조하십시오.

## vRealize Automation Cloud Assembly 프로젝트에 대해 알아보기

프로젝트는 클라우드 템플릿과 리소스 간의 컨넥터입니다. 작동 방식 및 활용하는 방법을 더 많이 이해할수록 vRealize Automation Cloud Assembly 개발 및 배포 프로세스의 효율성을 높일 수 있습니다.

## vRealize Automation Cloud Assembly 프로젝트 태그 및 사용자 지정 속성 사용

관리자는 프로젝트 요구 사항이 vRealize Automation Cloud Assembly 클라우드 템플릿과 다른 경우 프로젝트 수준 거버넌스 제약 조건 또는 사용자 지정 속성을 추가할 수 있습니다. 제약 조건 태그 외에, 프로젝트 빌딩 프로세스 중 배포된 리소스에 추가된 리소스 태그를 추가하여 리소스를 관리할 수 있습니다.

### 프로젝트 리소스 태그 소개

프로젝트 리소스 태그는 배포된 리소스를 관리하고 규정 준수를 보장하는 데 사용할 수 있는 표준화된 식별 태그로 작동합니다.

프로젝트에 정의된 리소스 태그는 해당 프로젝트의 일부로 배포된 모든 구성 요소 리소스에 추가됩니다. 이때 표준 태그 지정을 사용하면 다른 애플리케이션을 사용하여 리소스를 관리(예: CloudHealth를 사용하여 지출 비용 모니터링)하고, 무엇보다도 규정 준수를 보장할 수 있습니다.

예를 들어 클라우드 관리자는 CloudHealth와 같은 애플리케이션을 사용하여 비용을 관리하려고 합니다. 유럽연합 휴먼 리소스 도구 개발 전용 프로젝트에 `costCenter:eu-cc-1234` 태그를 추가합니다. 프로젝트 팀이 이 프로젝트에서 배포하면, 해당 태그가 배포된 리소스에 추가됩니다. 그런 다음 이 태그가 포함된 리소스를 식별하고 관리하도록 비용 산정 도구를 구성합니다. 다른 비용 센터가 있는 다른 프로젝트에는 키와 함께 사용할 대체 값이 있습니다.

### 프로젝트 제약 조건 태그 소개

프로젝트 제약 조건은 거버넌스 정의로 사용됩니다. 제약 조건은 프로젝트 클라우드 영역에서 배포 요청에 사용하거나 사용하지 않을 리소스를 정의하는 `key:value` 태그입니다.

배포 프로세스는 네트워크 및 스토리지에 대해 프로젝트 제약 조건과 일치하는 태그를 찾고, 일치하는 태그를 기반으로 배포를 수행합니다.

확장성 제약 조건은 확장성 워크플로에 사용할 vRealize Orchestrator 통합 인스턴스를 지정하는 데 사용됩니다.

프로젝트 제약 조건을 구성할 때 다음과 같은 형식을 고려하십시오.

- **key:value** 및 **key:value:hard**. 이 태그는 클라우드 템플릿을 일치하는 기능 태그를 가진 리소스에 배포해야 하는 경우에 두 가지 형식 중 하나로 사용합니다. 일치하는 태그를 찾을 수 없으면 배포 프로세스가 실패합니다. 예를 들어 프로젝트의 멤버가 배포한 클라우드 템플릿을 PCI 준수 네트워크에 프로비저닝해야 합니다. `security:pci`를 사용합니다. 프로젝트 클라우드 영역에서 네트워크를 찾을 수 없으면 안전하지 않은 배포가 없도록 하기 위해 배포가 실패합니다.
- **key:value:soft**. 이 태그는 일치하는 리소스를 원하지만 배포 프로세스가 실패 없이 진행되도록 태그가 일치하지 않는 리소스도 허용할 수 있는 경우에 사용합니다. 예를 들어 프로젝트 멤버가 클라우드 템플릿을 더 저렴한 스토리지에 배포하는 것을 선호하지만 스토리지 가용성이 배포 가능성에 방해가 되는 것을 원하지 않습니다. `tier:silver:soft`를 사용합니다. 그러면 프로젝트 클라우드 영역에 `tier:silver` 태그가 지정된 스토리지가 없는 경우에도 클라우드 템플릿이 다른 스토리지 리소스에 배포됩니다.

- **!key:value**. 일치하는 태그를 가진 리소스에 배포하지 않으려면 **hard** 또는 **soft** 옵션과 함께 이 태그를 사용합니다.

중요한 점은 프로젝트 제약 조건 태그는 클라우드 템플릿 제약 조건 태그보다 우선 순위가 높기 때문에 배포 시 클라우드 템플릿 제약 조건 태그를 재정의합니다. 이러한 재정의가 발생하면 안 되는 클라우드 템플릿이 있는 경우에는 템플릿에 `failOnConstraintMergeConflict:true`를 사용할 수 있습니다. 예를 들어 프로젝트에 네트워크 `loc:london` 제약 조건이 있지만 클라우드 템플릿의 제약 조건이 `loc:mumbai`인 경우 프로젝트 위치를 우선적으로 사용하는 대신 제약 조건 충돌 메시지와 함께 배포가 실패하도록 설정하려면 다음 샘플과 유사한 속성을 추가합니다.

```
constraints:
  - tag: 'loc:mumbai'
  failOnConstraintMergeConflict:true
```

## 사용자 지정 속성을 사용하는 방법

보고를 위한 프로젝트 사용자 지정 속성을 사용하여 확장성 작업 및 워크플로를 트리거하여 채우고 클라우드 템플릿 수준 속성을 재정의할 수 있습니다.

배포에 사용자 지정 속성을 추가하면 해당 값을 사용자 인터페이스에 사용하거나, 보고서를 생성할 수 있도록 API를 사용하여 해당 값을 검색할 수 있습니다.

확장성의 경우, 확장성 구독을 위해 사용자 지정 속성을 사용할 수도 있습니다. 확장성에 대한 자세한 내용은 [확장성을 사용하여 애플리케이션 수명 주기를 연장 및 자동화하는 방법](#) 항목을 참조하십시오.

프로젝트에 대해 변경할 특정 속성 값이 클라우드 템플릿에 포함될 수 있습니다. 다른 이름 및 값을 사용자 지정 속성으로 제공할 수 있습니다.

또한 사용자를 포함한 그 어떤 다른 사용자도 배포에 포함된 값을 볼 수 없도록 속성 값을 암호화할 수 있습니다. 예를 들어 프로젝트의 모든 사용자가 사용하지만 표시하지 않으려는 암호를 암호화할 수 있습니다. 값을 암호화하고 프로젝트를 저장한 후에는 값을 마스킹 해제하거나 바꿀 수 없습니다. **암호화된** 확인란을 지우면 값이 제거됩니다. 값을 다시 입력해야 합니다.

## vRealize Automation Cloud Assembly의 프로젝트 비용이란?

vRealize Automation Cloud Assembly 프로젝트에 사용할 수 있는 비용은 전체 프로젝트와 관련된 리소스 지출을 관리하는 데 도움이 됩니다. 프로젝트에는 개별 배포 비용도 포함됩니다.

프로젝트 및 개별 배포에 대한 비용 정보는 프로젝트와 연결된 배포가 최소 하나 프로비저닝된 후에 나타납니다. 비용은 매일 계산 및 업데이트되므로 시간 경과에 따른 배포 비용을 추적할 수 있습니다. 초기 값은 업계 벤치마크를 기반으로 합니다.

클라우드 관리자는 값을 조정하여 실제 비용을 반영할 수 있습니다.

자세한 내용은 [vRealize Automation에서 가격 책정 카드를 사용하는 방법](#) 항목을 참조하십시오.

## 배포 시간에 vRealize Automation Cloud Assembly 프로젝트가 작동하는 방식

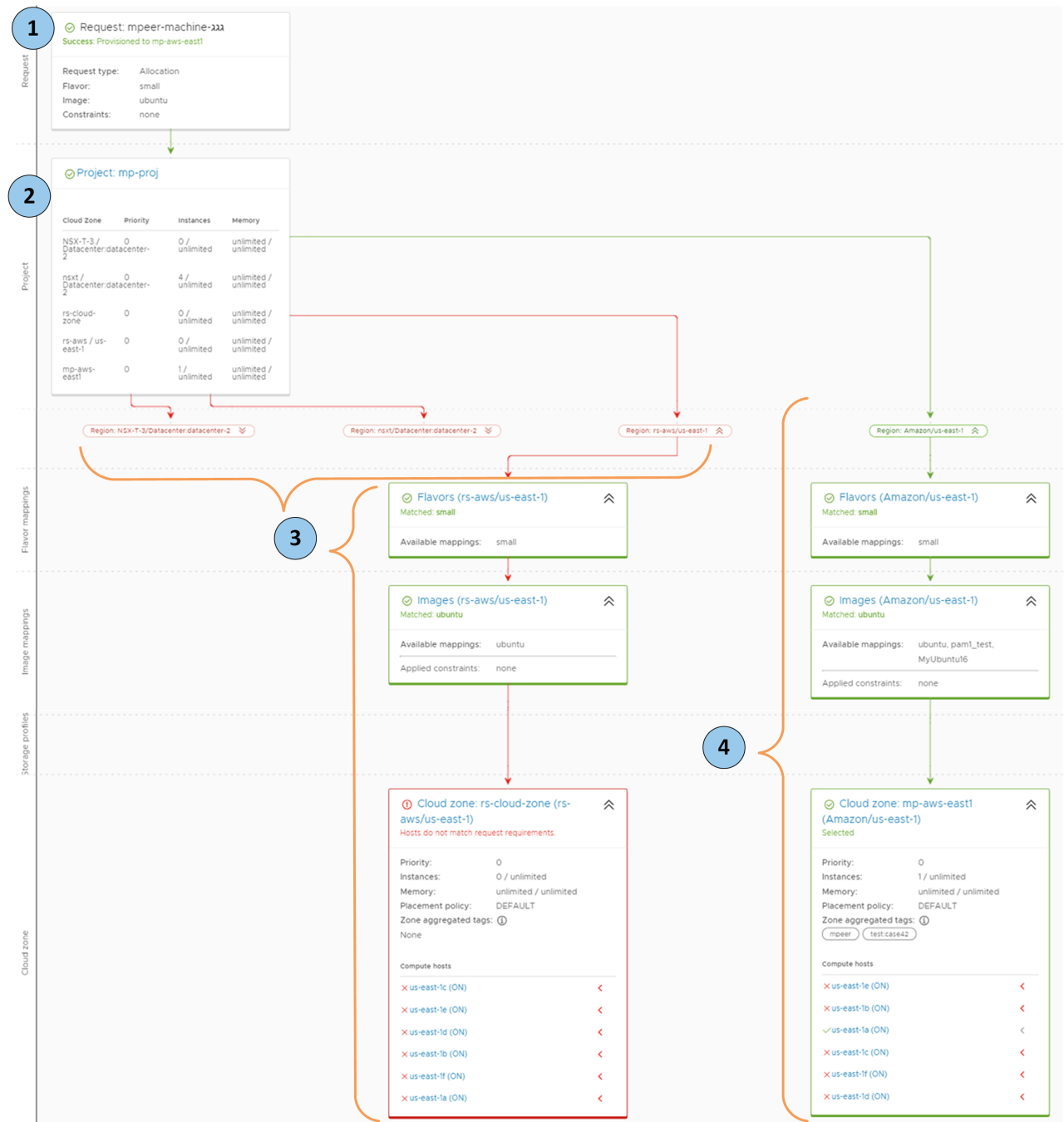
프로젝트는 프로비저닝된 리소스의 사용자 소유권 및 클라우드 영역에 대한 사용자 액세스를 제어합니다. 클라우드 관리자는 클라우드 템플릿 개발자든 배포를 관리하고 문제를 해결할 수 있으려면 배포 시간에 프로젝트가 어떻게 작동하는지 이해해야 합니다.

다양한 팀의 프로젝트를 설정하는 클라우드 관리자는 프로젝트에서 클라우드 템플릿 구성 요소가 배포되는 위치를 결정하는 방법을 이해해야 합니다. 이러한 이해는 클라우드 템플릿 개발자를 지원하는 프로젝트를 생성하고 실패한 배포 문제를 해결하는 데 도움을 줍니다.

클라우드 템플릿을 생성할 때 먼저 클라우드 템플릿을 프로젝트와 연결합니다. 배포 시, 프로젝트 클라우드 영역에 대해 클라우드 템플릿 요구 사항을 평가하여 최적의 배포 위치를 찾습니다.

다음 워크플로는 해당 프로세스를 설명합니다.

- 1 클라우드 템플릿 배포 요청을 제출합니다.
- 2 프로젝트는 템플릿 및 프로젝트 요구 사항(예: 버전, 이미지, 제약 조건 태그)을 평가합니다. 요구 사항을 프로젝트 클라우드 영역과 비교하여 요구 사항이 지원되는 영역을 찾습니다.
- 3 이러한 영역에는 요청을 지원하는 리소스가 없습니다.
- 4 이 클라우드 영역은 요청 요구 사항을 지원합니다. 템플릿이 이 클라우드 영역 계정 지역에 배포됩니다.



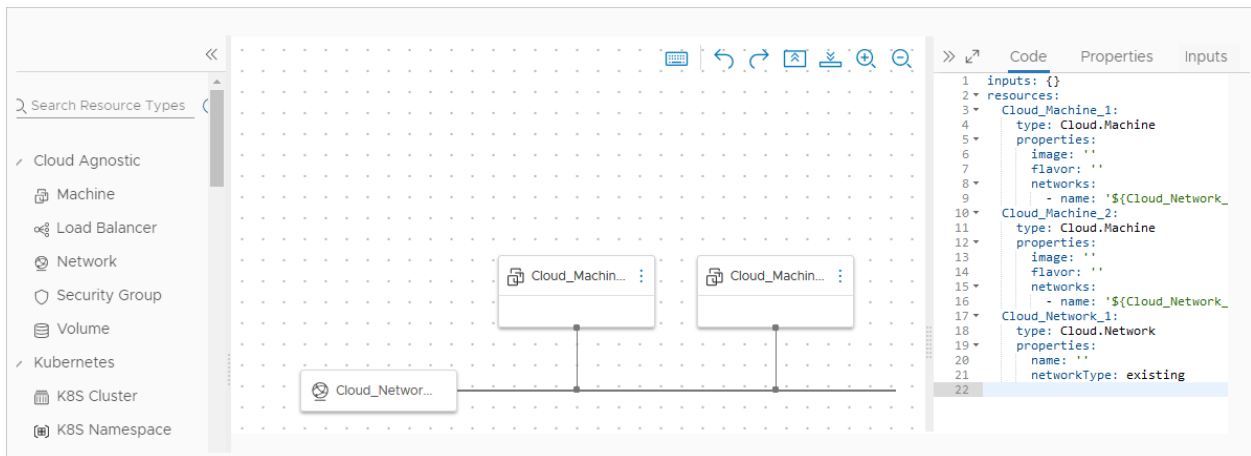
# vRealize Automation Cloud Assembly 배포 설계

# 6

배포는 vRealize Automation Cloud Assembly를 통해 클라우드 리소스에 생성하는 시스템, 애플리케이션 및 서비스를 정의하는 규격인 클라우드 템플릿(이전 명칭: Blueprint)으로 시작됩니다.

클라우드 템플릿 개발자는 특정 클라우드 벤더를 대상으로 하는 템플릿을 설계하거나, 클라우드 애그노스틱 템플릿을 설계할 수 있습니다. 프로젝트에 할당된 클라우드 영역에 따라 어느 접근 방법을 사용할지 결정됩니다. 클라우드 영역을 구성하는 리소스 종류를 이해하려면 클라우드 관리자에게 문의하십시오.

vRealize Automation Cloud Assembly 템플릿을 생성하는 것은 인프라를 코드로 구현하는 프로세스입니다. 시작하려면 설계 캔버스에서 리소스를 추가하고 연결합니다. 그런 다음 캔버스의 오른쪽에 있는 코드 편집기를 사용하여 세부 정보를 완성합니다. 코드 편집기를 사용하면 코드를 직접 입력하거나 속성 값을 양식에 입력할 수 있습니다.



## 클라우드 템플릿을 생성하기 전에

vRealize Automation Cloud Assembly 템플릿은 언제든지 생성할 수 있지만, 템플릿을 배포하려면 클라우드 리소스 인프라부터 정의해야 합니다.

### ■ 장 4 vRealize Automation Cloud Assembly 리소스 인프라 구축

또한 이러한 인프라 리소스를 클라우드 영역으로 포함하는 vRealize Automation Cloud Assembly 프로젝트도 생성해야 합니다.

### ■ 장 5 vRealize Automation Cloud Assembly 프로젝트 추가 및 관리

본 장은 다음 항목을 포함합니다.

- 클라우드 템플릿을 생성하는 방법
- 간단한 vRealize Automation Cloud Assembly 템플릿을 처음부터 생성하는 방법
- 간단한 vRealize Automation Cloud Assembly 템플릿을 향상시키는 방법
- vRealize Automation Cloud Assembly 설계에 고급 기능을 추가하는 방법
- vRealize Automation 리소스 속성 소개
- 일부 vRealize Automation Cloud Assembly 코드 예
- vRealize Automation Cloud Assembly에 Terraform 구성을 포함하는 방법
- vRealize Automation Cloud Assembly 마켓플레이스를 사용하는 방법

## 클라우드 템플릿을 생성하는 방법

vRealize Automation Cloud Assembly는 클라우드 템플릿을 코드로 생성하고 저장하기 때문에 템플릿을 설계하고 재사용하기가 쉽습니다.

빈 캔버스에서 클라우드 템플릿을 구축하거나 기존 코드를 활용할 수 있습니다.

### vRealize Automation Cloud Assembly 설계 페이지

처음부터 클라우드 템플릿을 생성하려면 **설계 > 클라우드 템플릿**으로 이동하고 **새로 만들기 > 빈 캔버스**를 클릭합니다. 리소스를 캔버스에 끌어 놓고 연결한 다음 코드 편집기에서 구성을 완료합니다.

코드 편집기를 사용하면 코드를 직접 입력하고, 잘라내고, 복사하고 붙여 넣을 수 있습니다. 코드 편집이 불편하면 설계 캔버스에서 리소스를 선택하고 코드 편집기 **속성** 탭을 클릭한 다음 그 곳에 값을 입력하면 됩니다. 입력한 속성 값이 마치 직접 입력한 것처럼 코드에 나타납니다.

```

WebTier:
  type: Cloud.Machine
  properties:
    name: wordpress
    flavor: '${input.size}'
    image: ubuntu
    count: '${input.count}'
    constraints:
      - tag: '${input.env}'
    networks:
      - network: '${resource["WP-Network-Private"].id}'
        assignPublicIpAddress: true
    storage:
      disks:
        - capacityGb: '${input.archiveDiskSize}'
          name: ArchiveDisk
    cloudConfig: |
      #cloud-config
      repo_update: true
      repo_upgrade: all

  packages:
    - apache2
    - php
    - php-mysql
    - libapache2-mod-php
  
```

The Properties panel shows the following configuration:

- Count: "\${input.count}"
- Image Type: ubuntu
- Flavor \*: \${input.size}
- Storage: (empty)
- Constraints: (empty)
- Tag: (empty)
- Maximum Capacity of the disk in GB: 1
- Size of boot disk in GB: 1
- Networks: (empty)

한 클라우드 템플릿에서 다른 클라우드 템플릿으로 코드를 복사하여 붙여 넣을 수 있습니다.

## 클라우드 템플릿 복제

템플릿을 복제하려면 **설계**로 이동하여 소스를 선택하고 **복제**를 클릭합니다. 클라우드 템플릿을 복제하여 소스의 복사본을 생성한 후 해당 클론을 새 프로젝트에 할당하거나, 새 애플리케이션의 시작 코드로 사용합니다.

## 업로드 및 다운로드

vRealize Automation Cloud Assembly 마켓플레이스에는 점프스타트를 위해 마무리된 클라우드 템플릿이 제공됩니다. [vRealize Automation Cloud Assembly 마켓플레이스를 사용하는 방법](#) 항목을 참조하십시오.

또한 사이트에 적합한 방식으로 클라우드 템플릿 YAML 코드를 업로드, 다운로드 및 공유할 수 있습니다. 외부 편집기 및 개발 환경을 사용하여 템플릿 코드를 수정할 수도 있습니다.

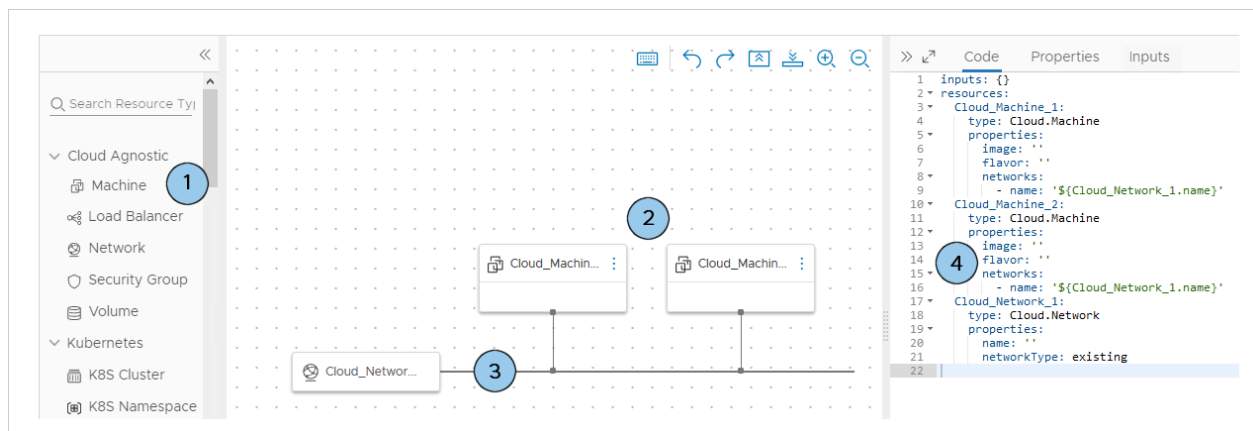
**참고** 공유 템플릿 코드를 검증하는 유용한 방법은 설계 페이지의 [vRealize Automation Cloud Assembly 코드 편집기에서 코드를 검사하는 것](#)입니다.

Cloud Templates <span>22 items</span>						
<span>NEW FROM</span> <span>SYNC REPOS</span> <span>CLONE</span> <span>DEPLOY</span> <span>DOWNLOAD</span> <span>DELETE</span> <span>Filter...</span>						
<input type="checkbox"/>	Name	Description	Source Control	Source Control - Last Sync	Project	Last Updated
<input checked="" type="checkbox"/>	ESFSE				65-Project	Aug 31, 2020, 4:41:52 PM
<input type="checkbox"/>	demo-clone		demo-01/admin-templat...	✓ New draft, version(s) ci	62-Project	Aug 31, 2020, 4:39:47 PM
<input type="checkbox"/>	aws-with-network		demo-01/admin-templat...	✓ New draft, version(s) ci	62-Project	Aug 30, 2020, 5:01:59 PM
<input type="checkbox"/>	test1		demo-01/admin-templat...	✓ New draft, version(s) ci	62-Project	Aug 28, 2020, 3:38:19 PM
<input type="checkbox"/>	test2		demo-01/admin-templat...	✓ New draft, version(s) ci	62-Project	Aug 28, 2020, 3:14:57 PM
<input type="checkbox"/>	test3		demo-01/admin-templat...	✓ New draft, version(s) ci	62-Project	Aug 28, 2020, 1:35:22 PM

## 간단한 vRealize Automation Cloud Assembly 템플릿을 처음부터 생성하는 방법

설계 페이지를 사용하면 프로비저닝할 시스템 또는 애플리케이션에 대한 vRealize Automation Cloud Assembly 템플릿 규격을 생성할 수 있습니다.

- 1 리소스를 찾습니다.
- 2 리소스를 캔버스에 끌어 옵니다.
- 3 리소스를 연결합니다.
- 4 클라우드 템플릿 코드를 편집하여 리소스를 구성합니다.



설계 페이지에서 클라우드 템플릿 이름 또는 버전을 변경하거나, 특정 버전으로 되돌리거나, 템플릿을 복제 또는 배포할 수도 있습니다.

## vRealize Automation Cloud Assembly 리소스를 선택하고 클라우드 템플릿에 추가하는 방법

vRealize Automation Cloud Assembly 리소스는 클라우드 템플릿 구축 블록입니다. 설계 페이지에서는 클라우드 애그노스틱 리소스 또는 클라우드 벤더별 리소스를 사용할 수 있습니다.

리소스는 설계 페이지의 왼쪽에 선택할 수 있게 표시됩니다.

### 클라우드 애그노스틱 리소스

클라우드 애그노스틱 리소스는 모든 클라우드 벤더에 배포할 수 있습니다. 프로비저닝 시 일치하는 클라우드별 리소스가 배포에 사용됩니다. 예를 들어 클라우드 템플릿을 AWS와 vSphere 클라우드 영역 모두에 배포하려면 클라우드 애그노스틱 리소스를 사용합니다.

### 클라우드 벤더 리소스

Amazon Web Services, Microsoft Azure, Google Cloud Platform 또는 VMware vSphere 등과 관련한 벤더 리소스는 일치하는 AWS, Azure, GCP 또는 vSphere 클라우드 영역에만 배포할 수 있습니다.

클라우드 애그노스틱 리소스를 특정 벤더를 위한 클라우드별 리소스가 포함된 클라우드 템플릿에 추가할 수 있습니다. 단, 프로젝트 클라우드 영역에서 벤더별로 지원되는 사항을 잘 알고 있어야 합니다.

### 구성 관리 리소스

구성 관리 리소스는 통합된 애플리케이션에 따라 다릅니다. 예를 들어 Puppet 리소스는 다른 리소스의 구성을 모니터링하고 적용할 수 있습니다.

## vRealize Automation Cloud Assembly에서 클라우드 템플릿 리소스를 연결하는 방법

그래픽 설계 캔버스를 사용하여 vRealize Automation Cloud Assembly 클라우드 템플릿 리소스를 연결합니다.

리소스 연결이 호환되는 경우에 리소스를 연결할 수 있습니다. 예를 들면 다음과 같습니다.

- 시스템 클러스터에 로드 밸런서 연결
- 네트워크에 시스템 연결
- 시스템에 외부 스토리지 연결

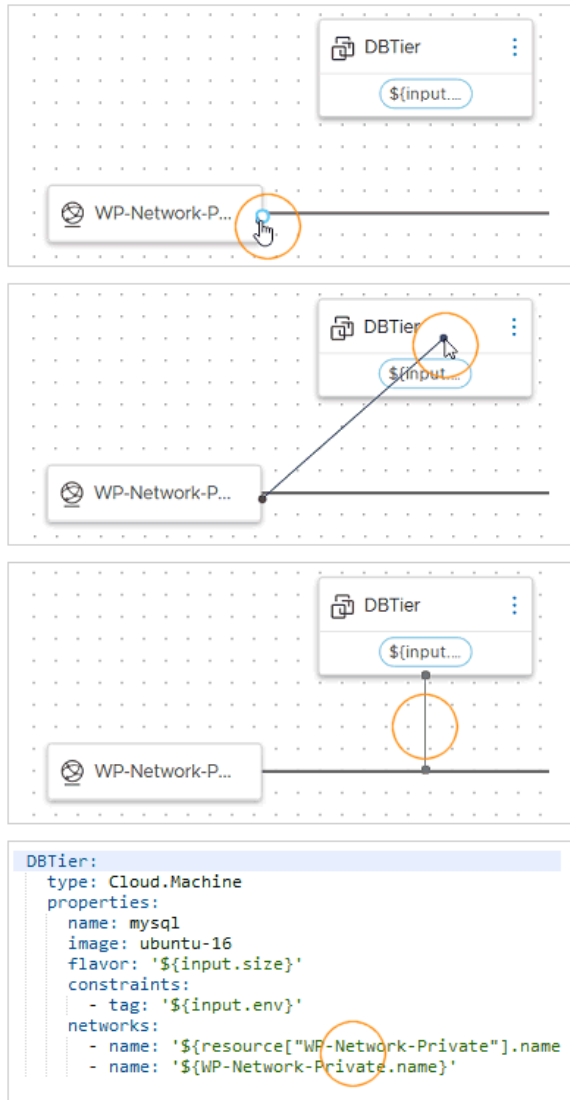
**중요** 실선 커넥터를 사용하려면 두 리소스가 동일한 클라우드 영역에 배포되어야 합니다. 충돌하는 제약 조건을 리소스에 추가하면 배포가 실패할 수 있습니다.

예를 들어 제약 조건 태그가 하나는 us-west-1의 영역에 배치하고 다른 하나는 us-east-1의 영역에 배치하도록 강제 적용하는 연결된 리소스는 배포할 수 없습니다.

실선 또는 파선 화살표는 연결이 아닌 종속성만 나타냅니다. 종속성에 대한 자세한 내용은 [vRealize Automation Cloud Assembly에서 리소스 배포 순서를 설정하는 방법](#) 항목을 참조하십시오.

리소스를 연결하려면 리소스 가장자리 위로 마우스를 이동하여 연결 버블을 표시합니다. 그런 다음 버블을 클릭한 채로 대상 리소스로 끌고 가서 마우스 버튼을 놓습니다.

코드 편집기에서 보면 소스 리소스에 대한 추가 코드가 대상 리소스 코드에 표시됩니다.



그림에서 SQL 시스템과 프라이빗 네트워크는 연결되어 있으므로 동일한 클라우드 영역에 배포해야 합니다.

## vRealize Automation Cloud Assembly에서 유효한 클라우드 템플릿 코드를 생성하는 방법

vRealize Automation Cloud Assembly 리소스를 추가하고 캔버스에서 이러한 리소스를 연결하면 시작 코드만 생성됩니다. 이러한 구성 요소를 완전하게 구성하려면 코드를 편집해야 합니다.

코드 편집기를 사용하면 코드를 직접 입력하거나 속성 값을 양식에 입력할 수 있습니다. 직접 코드 생성을 지원하기 위해 vRealize Automation Cloud Assembly 편집기에는 구문 완성 및 오류 검사 기능이 포함되어 있습니다.

### 편집기 힌트 예

사용 가능한  
값

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15
16     networks:
17       - name: small flavor
18         large flavor
19 Cloud_Network_1:
20   type: Cloud.Network
21   properties:
22     name: ''
23     networkType: existing
  
```

허용되는 속  
성

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15
16     tags: array
17
18     storage: object
19
20     remoteAccess: object
21
22     name: string
23
24     imageRef: string
25
26     count: integer
27
28     constraints: array
29
30     cloudConfig: string
  
```

하위 속성

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: ''
15     constraints:
16       -
17         net:
18           tag: string
19 Cloud_Network_1:
20   type: Cloud.Network
  
```

구문 오류

❗ Please correct errors in YAML editor before editing in canvas: row: 14, column: 17

```

10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: 'small'
15     constraints:
16       - name: '${Cloud_Network_1.name}'
17 Cloud_Network_1:
18   type: Cloud.Network
19   properties:
20     name: ''
21
22     networkType: existing
  
```

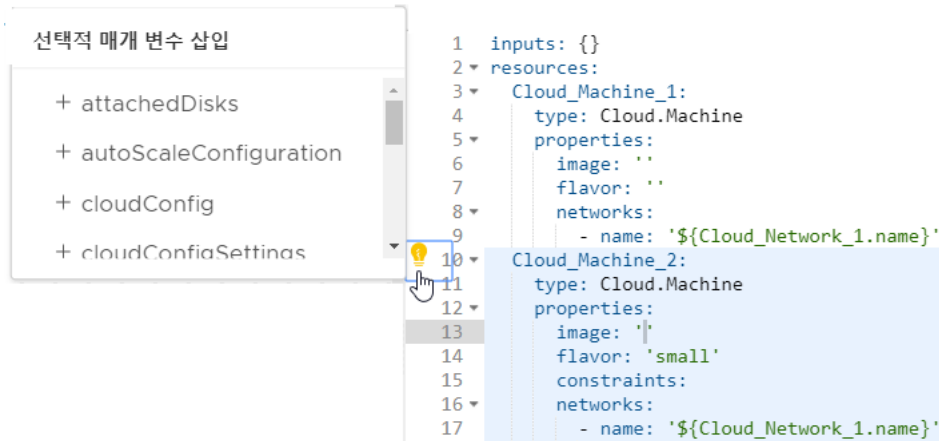
검색  
(Ctrl+F)

```

1 inputs: {}
2 resources:
3   Cloud_Machine_1:
4     type: Cloud.Machine
5     properties:
6       image: ''
7       flavor: ''
8       networks:
9         - name: '${Cloud_Network_1.name}'
10 Cloud_Machine_2:
11   type: Cloud.Machine
12   properties:
13     image: ''
14     flavor: 'small'
15     constraints:
16       - name: '${Cloud_Network_1.name}'
  
```

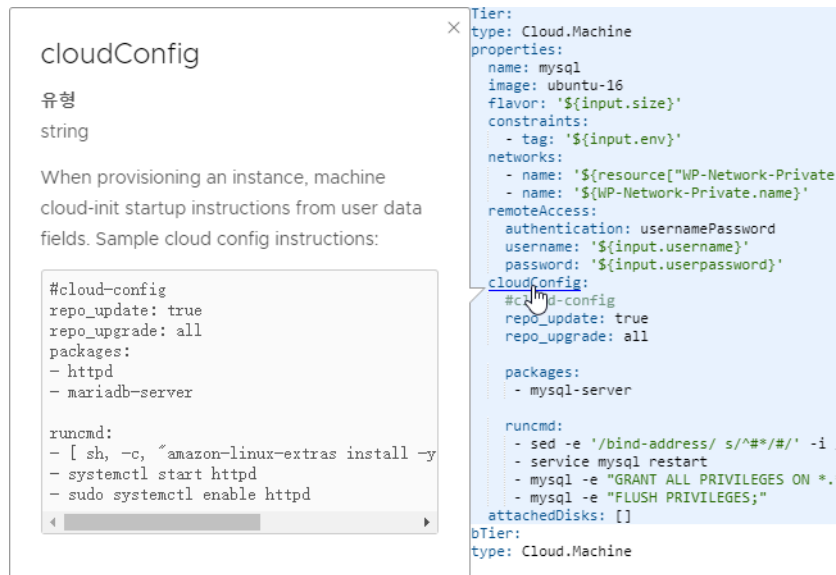
## 편집기 힌트 예

선택적 매개  
변수



스키마 도움  
말

모든 사용자 지정 속성에 대해서는 VMware (code)의 vRealize Automation 리소스 유형 스키마를 참조할 수도 있습니다.



## vRealize Automation Cloud Assembly를 사용하여 다른 버전을 저장하는 방법

클라우드 템플릿 개발자는 위험한 추가 변경을 수행하기 전에 제대로 작동하는 설계의 스냅샷을 안전하게 캡처할 수 있습니다.

배포 시, 배포할 버전을 선택할 수 있습니다.

### 클라우드 템플릿 버전 캡처

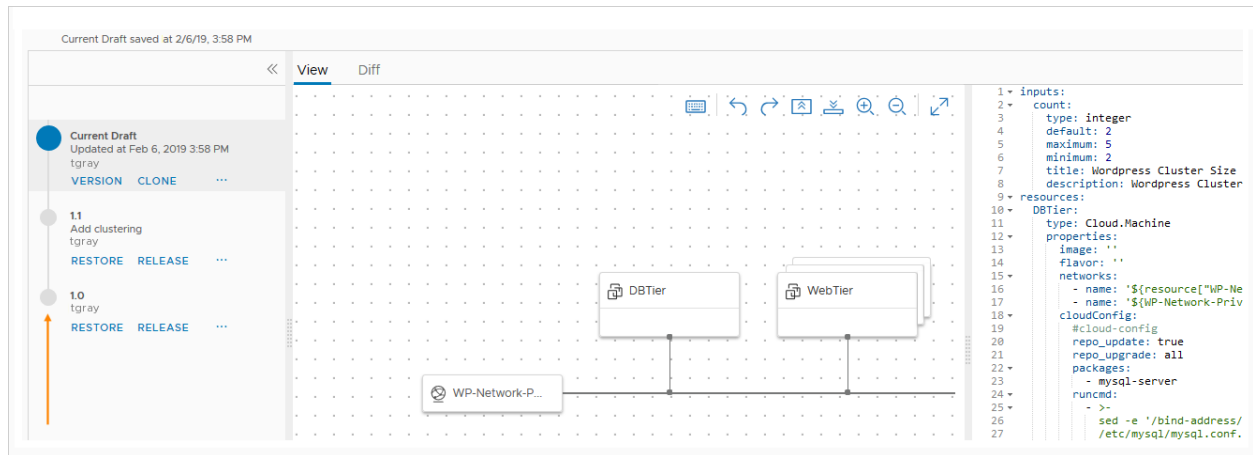
설계 페이지에서 **버전**을 클릭하고 이름을 제공합니다.

이름은 영숫자와 특수 문자(마침표, 하이픈 및 밑줄만 허용)를 사용해야 하고 공백은 사용할 수 없습니다.

## 이전 버전 복원

설계 페이지에서 **버전 기록**을 클릭합니다.

왼쪽에서 이전 버전을 선택하여 캔버스 및 코드 편집기에서 검사합니다. 원하는 버전을 찾으면 **복원**을 클릭합니다. 복원을 수행하면 명명된 버전을 제거하지 않고 현재 초안을 덮어씹습니다.



## vRealize Automation Service Broker에 버전 릴리스

설계 페이지에서 **버전 기록**을 클릭합니다.

왼쪽에서 버전을 선택하고 릴리스합니다.

현재 초안은 버전을 지정할 때까지 릴리스할 수 없습니다.

## vRealize Automation Service Broker에서 버전 다시 가져오기

카탈로그 사용자가 새 버전을 사용할 수 있도록 설정하려면 다시 가져옵니다.

vRealize Automation Service Broker에서 **컨텐츠 및 정책 > 컨텐츠 소스**로 이동합니다.

소스 목록에서 새로 릴리스된 버전의 클라우드 템플릿이 포함된 프로젝트의 소스를 클릭합니다.

**저장 및 가져오기**를 클릭합니다.

## 클라우드 템플릿 버전 비교

변경 내용과 버전이 누적되면 그 차이를 식별하는 것이 좋습니다.

vRealize Automation Cloud Assembly의 [버전 기록] 보기에서 버전을 선택하고 **차이**를 클릭합니다. 그런 다음, **비교 대상** 드롭다운에서 비교할 다른 버전을 선택합니다.

코드 차이 검토 또는 시각적 토폴로지 차이 검토 사이를 전환할 수 있습니다.

그림 6-1. 코드 차이

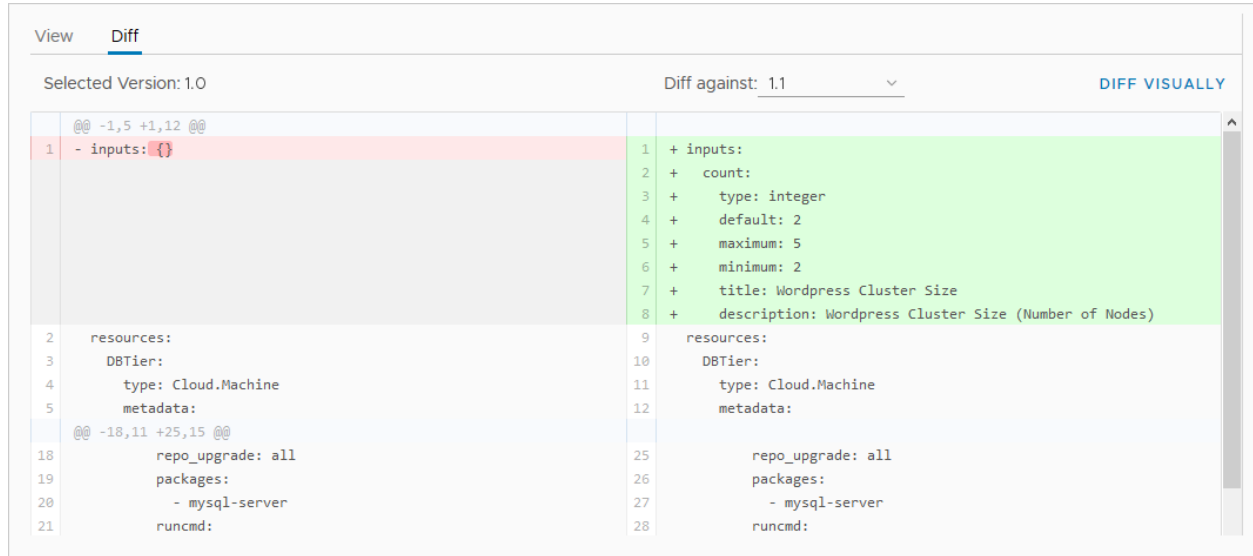
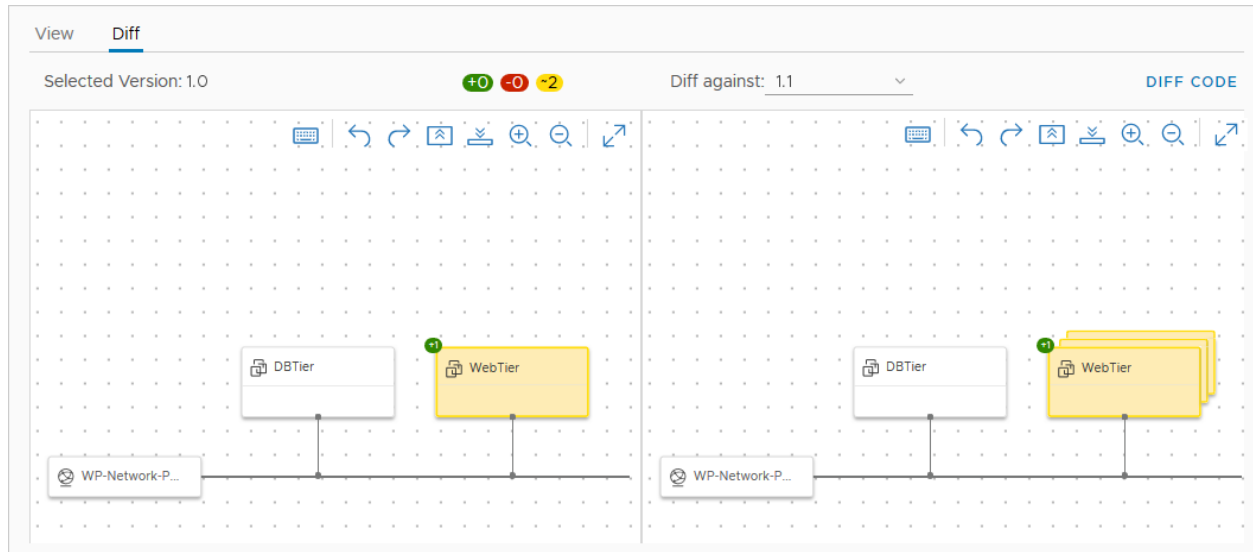


그림 6-2. 시각적 토폴로지 차이



## 클라우드 템플릿 복제

버전을 저장하는 것과 같지는 않지만 설계 페이지에서 **작업 > 복제**를 선택하면 다른 개발을 위해 현재 템플릿의 복사본이 만들어집니다.

## 간단한 vRealize Automation Cloud Assembly 템플릿을 향상시키는 방법

간단한 템플릿을 한 차원 높일 수 있는 vRealize Automation Cloud Assembly 템플릿 코드를 구현할 수 있습니다.

여기에 설명된 기술을 파악하려면 인프라 코드에 익숙해야 합니다. 다행히 vRealize Automation Cloud Assembly 코드는 사람이 읽을 수 있고 쉽게 따라 할 수 있습니다.

## vRealize Automation에서 사용자 입력이 클라우드 템플릿을 사용자 지정할 수 있는 방법

클라우드 템플릿 디자이너는 입력 매개 변수를 사용하여 사용자가 요청 시 사용자 지정 선택이 가능하도록 할 수 있습니다.

사용자가 입력을 제공할 때, 약간 다른 템플릿 복사본을 여러 개 저장할 필요가 더 이상 없습니다. 또한 입력은 2일차 작업에 대한 템플릿을 준비할 수 있습니다. [vRealize Automation 2일차 업데이트에 클라우드 템플릿 입력을 사용하는 방법](#)의 내용을 참조하십시오.

다음 입력은 MySQL 데이터베이스 서버에 대해 하나의 클라우드 템플릿을 생성하고, 이 템플릿을 사용자가 여러 클라우드 리소스 환경에 배포하여 매번 다른 용량과 자격 증명을 적용할 수 있는 방법을 보여줍니다.

The screenshot shows a 'Testing Basic' dialog box with the following fields:

- Environment:** env.dev (dropdown menu)
- Tier Machine Size:** small (dropdown menu)
- Database Username:** ouradmin (text input)
- Database Password:** masked with dots (password input)

Buttons: CANCEL, TEST

## 클라우드 템플릿 입력 매개 변수를 정의하는 방법

선택 가능한 값을 설정하는 inputs 섹션을 템플릿 코드에 추가합니다.

다음 예에서는 시스템 크기, 운영 체제 및 클러스터링된 서버의 수를 선택할 수 있습니다.

```
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)

```

코드 편집이 불편하면 코드 편집기 **입력** 탭을 클릭하고 여기에 설정을 입력할 수 있습니다. 다음 예는 앞에서 언급한 MySQL 데이터베이스에 대한 일부 입력을 보여줍니다.

The screenshot shows the 'Cloud Template Inputs' section with a table of inputs. Below the table, the 'Edit Cloud Template Input: size' dialog is open, showing fields for Name, Title, Description, Type, and Encrypted.

<input type="checkbox"/>	Name	Title	Type	Default Value
<input type="checkbox"/>	size	Tier Machine Size	string	
<input type="checkbox"/>	username	Database Username	string	
<input type="checkbox"/>	userpassword	Database Password	string	****
<input type="checkbox"/>	databaseDiskSize	MySQL Data Disk Size	number	4

**Edit Cloud Template Input: size**

Name \*

Title

Description

Type

Encrypted ☐

## 클라우드 템플릿 입력 매개 변수를 참조하는 방법

다음으로 resources 섹션에서 `${input.property-name}` 구문을 사용하여 입력 매개 변수를 참조합니다. 속성 이름에 공백이 포함되어 있는 경우 점 표기법을 사용하는 대신 대괄호와 큰따옴표로 구분합니다. `${input["속성 이름"]}`

**중요** 클라우드 템플릿 코드에서는 입력 매개 변수를 나타내는 경우가 아니면 input이라는 단어를 사용할 수 없습니다.

```

resources:
  WebTier:
    type: Cloud.Machine
    properties:

```

```
name: wordpress
flavor: '${input.wp-size}'
image: '${input.wp-image}'
count: '${input.wp-count}'
```

## 입력 속성 목록

속성	설명
const	oneOf와 함께 사용됩니다. 친숙한 제목과 연결된 실제 값입니다.
default	입력에 대해 미리 채워진 값입니다. 기본값은 올바른 유형이어야 합니다. 정수의 기본값으로 단어를 입력하지 마십시오.
description	입력에 대한 사용자 도움말 텍스트입니다.
encrypted	사용자가 입력한 내용을 암호화할지 여부를 <b>true</b> 또는 <b>false</b> 로 지정합니다. 암호는 대개 암호화됩니다. 여러 클라우드 템플릿에서 재사용 가능한 암호화된 속성을 생성할 수도 있습니다. <a href="#">비밀 vRealize Automation Cloud Assembly 속성을 생성하고 참조하는 방법</a> 의 내용을 참조하십시오.
enum	허용되는 값의 드롭다운 메뉴입니다. 다음 예시를 형식 가이드로 사용하십시오. <pre>enum:   - value 1   - value 2</pre>
format	입력에 필요한 형식을 설정합니다. 예를 들어 (25/04/19)는 날짜-시간을 지원합니다. vRealize Automation Service Broker 사용자 지정 양식에서 날짜 선택을 사용할 수 있습니다.
items	어레이 내의 항목을 선언합니다. 숫자, 정수, 문자열, 부울 또는 개체를 지원합니다.
maxItems	어레이 내에서 선택 가능한 최대 항목 수입니다.
maxLength	문자열에 허용되는 최대 문자 수입니다. 예를 들어 필드를 25자로 제한하려면 <code>maxLength: 25</code> 를 입력합니다.
maximum	숫자나 정수에 허용되는 최대값입니다.
minItems	어레이 내에서 선택 가능한 최소 항목 수입니다.
minLength	문자열에 허용되는 최소 문자 수입니다.
minimum	숫자나 정수에 허용되는 가장 작은 값입니다.

속성	설명
oneOf	사용자 입력 양식에서 친숙하지 않은 값(const)에 대한 친숙한 이름(title)을 표시할 수 있습니다. 기본값을 설정하는 경우 title 이 아니라 const를 설정합니다. 문자열, 정수 및 숫자 형식과 함께 사용할 수 있습니다.
pattern	정규식 구문에서 문자열 입력에 허용되는 문자입니다. 예: '[a-z]+' 또는 '[a-z0-9A-Z@#&]+'
속성	개체에 대한 key:value 속성 블록을 선언합니다.
readOnly	양식 레이블만 제공하는 데 사용됩니다.
title	oneOf와 함께 사용됩니다. const 값에 대한 친숙한 이름입니다. title은 배포 시 사용자 입력 양식에 표시됩니다.
type	숫자, 정수, 문자열, 부울 또는 개체의 데이터 유형입니다.
writeOnly	양식에서 별표 뒤에 키 입력을 숨깁니다. enum과 함께 사용할 수 없습니다. vRealize Automation Service Broker 사용자 지정 양식에서 암호 필드로 표시됩니다.

## 추가 예시

열거형이 포함된 문자열

```
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
```

최소 및 최대 정수

```
count:
  type: integer
  title: Machine Count
  description: The number of machines that you want to deploy.
  maximum: 5
  minimum: 1
  default: 1
```

## 개체 어레이

```
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
```

## 친숙한 이름의 문자열

```
platform:
  type: string
  oneOf:
    - title: AWS
      const: platform:aws
    - title: Azure
      const: platform:azure
    - title: vSphere
      const: platform:vsphere
  default: platform:aws
```

## 패턴 유효성 검사가 포함된 문자열

```
username:
  type: string
  title: Username
  description: The name for the user that will be created when the machine is provisioned.
  pattern: ^[a-zA-Z]+$
```

## 문자열을 암호로

```
password:
  type: string
  title: Password
  description: The initial password that will be required to logon to the machine.
  Configured to reset on first login.
  encrypted: true
  writeOnly: true
```

## 문자열을 텍스트 영역으로

```
ssh_public_key:
  type: string
  title: SSH public key
  maxLength: 256
```

부록

```
public_ip:
  type: boolean
  title: Assign public IP address
  description: Choose whether your machine should be internet facing.
  default: false
```

날짜 및 시간 일정 선택기

```
leaseDate:
  type: string
  title: Lease Date
  format: date-time
```

## vRealize Automation Cloud Assembly 리소스 플래그가 요청을 사용자 지정하는 방법

vRealize Automation Cloud Assembly에는 요청 시 리소스가 처리되는 방식을 조정하는 여러 클라우드 템플릿 설정이 포함됩니다.

리소스 플래그 설정은 리소스 개체 속성 스키마의 일부가 아닙니다. 지정된 리소스의 경우 표시된 대로 속성 섹션 외부에 플래그 설정을 추가합니다.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    preventDelete: true
    properties:
      image: coreos
      flavor: small
      attachedDisks:
        - source: '${resource.Cloud_Volume_1.id}'
  Cloud_Volume_1:
    type: Cloud.Volume
    properties:
      capacityGb: 1
```

리소스 플러그	설명
allocatePerInstance	<p><code>true</code>로 설정하면 클러스터의 각 시스템별로 리소스 할당을 사용자 지정할 수 있습니다.</p> <p>기본값은 <code>false</code>로, 이 경우 클러스터 전체에서 동등하게 리소스를 할당하여 각 시스템마다 구성이 동일합니다. 또한 개별 리소스에 대해 2일차 작업을 별도로 수행하지 못할 수 있습니다.</p> <p>인스턴스별 할당을 사용하면 <code>count.index</code>가 개별 시스템에 대해 구성을 올바르게 적용할 수 있습니다. 코드 예는 <a href="#">vRealize Automation Cloud Assembly에서 시스템 및 디스크 클러스터를 생성하는 방법</a>의 내용을 참조하십시오.</p>
createBeforeDelete	<p>일부 업데이트 작업을 수행하려면 기존 리소스를 제거하고 새 리소스를 생성해야 합니다. 기본적으로 제거가 먼저 수행되는데, 이 경우 이전 리소스는 삭제되었지만 몇 가지 이유로 새 리소스가 생성되지 않은 상태가 될 수 있습니다.</p> <p>이전 리소스를 삭제하기 전에 새 리소스가 생성되도록 하려면 이 플래그를 <code>true</code>로 설정합니다.</p>
createTimeout	<p>리소스 할당, 생성 및 계획 요청에 대한 vRealize Automation Cloud Assembly 기본 시간 초과 값은 2시간(2h)입니다. 또한 프로젝트 관리자는 프로젝트 전체에서 이러한 요청에 대해 사용자 지정 기본 시간 초과를 설정할 수 있습니다.</p> <p>이 플래그를 사용하면 모든 기본값을 재정의하고 특정 리소스 작업에 대한 개별 시간 초과를 설정할 수 있습니다.</p> <p><code>updateTimeout</code> 및 <code>deleteTimeout</code>도 참조하십시오.</p>
deleteTimeout	<p>삭제 요청에 대한 vRealize Automation Cloud Assembly 기본 시간 초과 값은 2시간(2h)입니다. 또한 프로젝트 관리자는 프로젝트 전체에서 삭제 요청에 대해 서로 다른 기본 시간 초과를 설정할 수 있습니다.</p> <p>이 플래그를 사용하면 모든 기본값을 재정의하고 특정 리소스 삭제 작업에 대한 개별 시간 초과를 설정할 수 있습니다.</p> <p><code>updateTimeout</code> 및 <code>createTimeout</code>도 참조하십시오.</p>
dependsOn	<p>이 플래그는 다음 리소스를 생성하기 전에 하나의 리소스가 반드시 존재해야 하는 리소스 간의 명시적 종속성을 식별합니다. 자세한 내용은 <a href="#">vRealize Automation Cloud Assembly에서 리소스 배포 순서를 설정하는 방법</a> 항목을 참조하십시오.</p>
dependsOnPreviousInstances	<p><code>true</code>로 설정하면 클러스터 리소스를 순차적으로 생성합니다. 기본값은 <code>false</code>로, 이 경우 클러스터에 모든 리소스를 동시에 생성합니다.</p> <p>예를 들어, 순차적 생성은 기본 노드와 보조 노드를 생성해야 하는 데이터베이스 클러스터에는 유용하지만 보조 노드 생성에는 노드를 기존 기본 노드에 연결하는 구성 설정이 필요합니다.</p>
forceRecreate	<p>모든 업데이트 작업에서 기존 리소스를 제거하고 새 리소스를 생성해야 하는 것은 아닙니다. 업데이트 기본 설정과 관계없이 업데이트를 통해 이전 리소스를 제거하고 새 리소스를 생성하려면 이 플래그를 <code>true</code>로 설정합니다.</p>

리소스 플래그	설명
ignoreChanges	리소스 사용자는 리소스를 배포된 상태에서 변경하여 재구성할 수 있습니다. 배포 업데이트를 수행하지만 변경된 리소스를 클라우드 템플릿의 구성으로 덮어쓰지 않으려는 경우 이 플래그를 <b>true</b> 로 설정합니다.
ignorePropertiesOnUpdate	리소스 사용자는 특정 속성을 사용자 지정할 수 있는데, 이러한 속성이 업데이트 작업 중에 원래 클라우드 템플릿 상태로 재설정될 수 있습니다. 업데이트 작업 중에 속성이 재설정되지 않도록 하려면 이 플래그를 <b>true</b> 로 설정합니다.
preventDelete	후속 삭제 요청에서 리소스를 보호해야 하는 경우 이 플래그를 <b>true</b> 로 설정합니다.
recreatePropertiesOnUpdate	리소스 사용자는 리소스를 배포된 상태에서 변경하여 속성을 재구성할 수 있습니다. 업데이트하는 동안 리소스가 다시 생성되거나 다시 생성되지 않을 수 있습니다. 다시 생성되지 않은 리소스는 변경된 상태의 속성과 함께 유지될 수 있습니다. 업데이트 수행 여부와 관계없이 리소스와 해당 속성이 다시 생성되도록 하려면 이 플래그를 <b>true</b> 로 설정합니다.
updateTimeout	업데이트 요청에 대한 vRealize Automation Cloud Assembly 기본 시간 초과 값은 2시간(2h)입니다. 또한 프로젝트 관리자는 프로젝트 전체에서 업데이트 요청에 대해 서로 다른 기본 시간 초과를 설정할 수 있습니다. 이 플래그를 사용하면 모든 기본값을 재정의하고 특정 리소스 업데이트 작업에 대한 개별 시간 초과를 설정할 수 있습니다. <b>deleteTimeout</b> 및 <b>createTimeout</b> 도 참조하십시오.

## vRealize Automation Cloud Assembly에서 리소스 배포 순서를 설정하는 방법

vRealize Automation Cloud Assembly 템플릿을 배포하는 경우 일부 리소스를 사용하기 위해서는 먼저 다른 리소스가 사용 가능해야 할 수 있습니다.

**중요** 실선 또는 파선 화살표는 연결이 아닌 종속성만 나타냅니다. 리소스가 서로 통신할 수 있도록 연결하려면 **vRealize Automation Cloud Assembly에서 클라우드 템플릿 리소스를 연결하는 방법** 항목을 참조하십시오.

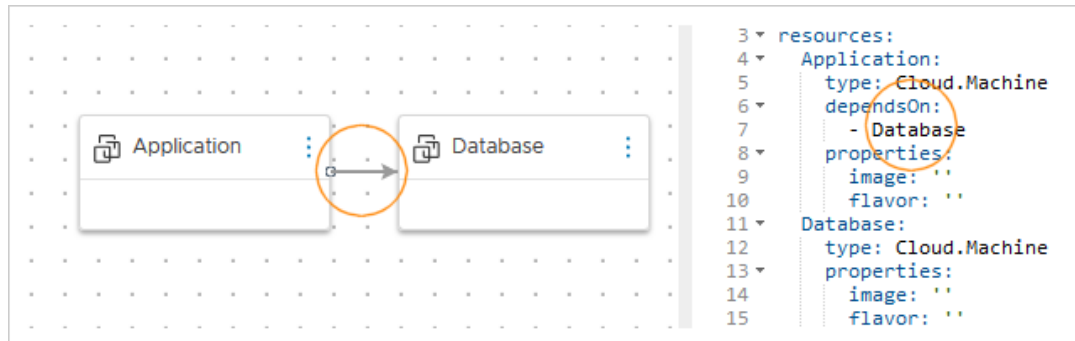
### 명시적 종속성을 생성하는 방법

때로는 다른 리소스가 먼저 배포되어야 하는 리소스가 있습니다. 예를 들어 애플리케이션 서버를 생성하여 데이터베이스에 액세스하도록 구성하려면 그 전에 먼저 데이터베이스 서버가 있어야 합니다.

명시적 종속성은 배포 시 빌드 순서를 설정하거나, 축소 또는 확장 작업에 대한 빌드 순서를 설정합니다. 그 래픽 설계 캔버스 또는 코드 편집기를 사용하여 명시적 종속성을 추가할 수 있습니다.

- 설계 캔버스 옵션 - 종속 리소스에서 시작하여 먼저 배포할 리소스에서 끝나는 연결을 그립니다.

- 코드 편집기 옵션 - 종속 리소스에 dependsOn 속성을 추가하고 먼저 배포할 리소스를 식별합니다.  
명시적 종속성은 캔버스에 실선 화살표를 생성합니다.



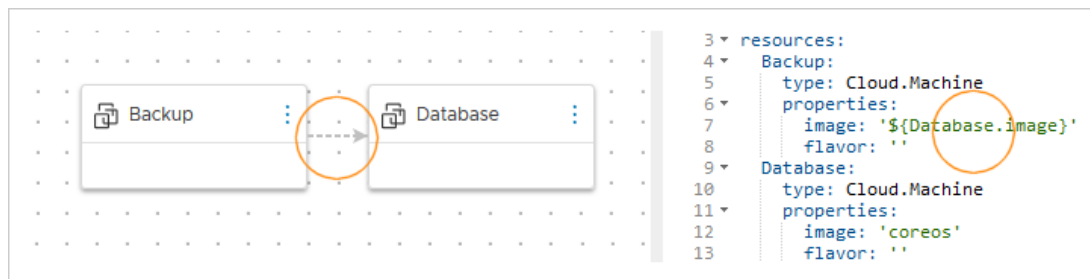
## 암시적 종속성 또는 속성 바인딩을 생성하는 방법

일부 리소스에는 다른 리소스의 속성에 있는 값이 필요한 경우가 있습니다. 예를 들어 백업 서버에는 백업 중인 데이터베이스 서버의 운영 체제 이미지가 필요할 수 있습니다. 따라서 데이터베이스 서버가 먼저 있어야 합니다.

속성 바인딩이라고도 하는 암시적 종속성은 종속 리소스를 배포하기 전에 필요한 속성을 사용할 수 있을 때까지 대기하여 빌드 순서를 제어합니다. 암시적 종속성은 코드 편집기를 사용하여 추가합니다.

- 종속 리소스를 편집하고 먼저 있어야 하는 리소스와 속성을 식별하는 속성을 추가합니다.

암시적 종속성 또는 속성 바인딩은 캔버스에 파선 화살표를 생성합니다.



## vRealize Automation Cloud Assembly에서 표현식을 사용하여 클라우드 템플릿 코드를 보다 다양하게 만드는 방법

유연성을 높이기 위해 vRealize Automation Cloud Assembly에서 클라우드 템플릿 코드에 표현식을 추가할 수 있습니다.

표현식은 다음 예시와 같이 `${expression}` 구문을 사용합니다.

예시는 중요한 라인만 표시하도록 정리되어 있습니다. 편집되지 않은 전체 클라우드 템플릿은 끝에 표시되어 있습니다.

## 예

배포 시 사용자가 원격 액세스에 필요한 암호화된 키를 붙여 넣을 수 있도록 허용:

```
inputs:
  sshKey:
    type: string
    maxLength: 500
resources:
  frontend:
    type: Cloud.Machine
    properties:
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'
```

VMware Cloud on AWS에 배포하려면 폴더 이름을 "워크로드"의 필수 이름으로 설정합니다.

```
inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
```

배포시 선택한 환경과 일치하는 "env" (모두 소문자) 태그로 시스템에 태그를 지정합니다.

```
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)}'
```

프런트 엔드 클러스터의 시스템 수를 1(소형) 또는 2(대형)로 설정합니다. 대규모 클러스터는 제거 프로세스에 의해 설정됩니다.

```
inputs:
  envsize:
    type: string
    enum:
      - Small
      - Large
resources:
  frontend:
    type: Cloud.Machine
    properties:
      count: '${input.envsize == "Small" ? 1 : 2}'
```

네트워크 리소스에 있는 속성에 바인딩하여 시스템을 동일한 "기본" 네트워크에 연결합니다.

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
      networks:
        - network: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      networks:
        - network: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing
```

API에 제출된 액세스 자격 증명 암호화:

```
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
```

API 시스템의 주소를 확인합니다.

```
resources:
  frontend:
    type: Cloud.Machine
    properties:
```

```

cloudConfig: |
  runcmd:
    - echo ${resource.apitier.networks[0].address}
apitier:
  type: Cloud.Machine
  properties:
    networks:
      - network: '${resource.Cloud_Network_1.name}'

```

## 완전한 클라우드 템플릿

```

inputs:
  environment:
    type: string
    enum:
      - AWS
      - vSphere
      - Azure
      - VMC
      - GCP
    default: vSphere
  sshKey:
    type: string
    maxLength: 500
  envsize:
    type: string
    enum:
      - Small
      - Large
resources:
  frontend:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu
      flavor: medium
      count: '${input.envsize == "Small" ? 1 : 2}'
      remoteAccess:
        authentication: publicPrivateKey
        sshKey: '${input.sshKey}'
      cloudConfig: |
        packages:
          - nginx
        runcmd:
          - echo ${resource.apitier.networks[0].address}
      constraints:
        - tag: '${"env:" + to_lower(input.environment)}'
      networks:
        - network: '${resource.Cloud_Network_1.name}'
  apitier:
    type: Cloud.Machine
    properties:
      folderName: '${input.environment == "VMC" ? "Workload" : ""}'
      image: ubuntu

```

```

    flavor: small
    cloudConfig: |
      #cloud-config
      runcmd:
        - export apikey=${base64_encode(input.username:input.password)}
        - curl -i -H 'Accept:application/json' -H 'Authorization:Basic :$apikey' http://
example.com
    remoteAccess:
      authentication: publicPrivateKey
      sshKey: '${input.sshKey}'
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'
    networks:
      - network: '${resource.Cloud_Network_1.name}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      name: Default
      networkType: existing
    constraints:
      - tag: '${"env:" + to_lower(input.environment)}'

```

## vRealize Automation Cloud Assembly의 클라우드 템플릿 표현식 구문

표현식 구문은 vRealize Automation Cloud Assembly 템플릿에서 사용할 수 있는 모든 표현식 기능을 노출합니다.

vRealize Automation Cloud Assembly에서 표현식을 사용하여 클라우드 템플릿 코드를 보다 다양하게 만드는 방법에 표시된 예시에서 구문은 일부만 표시됩니다.

### 리터럴

다음과 같은 리터럴이 지원됩니다.

- 부울(true 또는 false)
- 정수
- 부동 소수점
- String

백슬래시는 큰따옴표, 작은따옴표 및 백슬래시 자체를 이스케이프합니다.

"는 \"로 이스케이프됩니다.

'는 \'로 이스케이프됩니다.

\는 \\로 이스케이프됩니다.

다음 예제와 같이 따옴표는 동일한 유형의 따옴표로 묶인 문자열 내에서만 이스케이프해야 합니다.

```
"I am a \"double quoted\" string inside \"double quotes\"."
```

- null

## 환경 변수

환경 이름:

- orgId
- projectId
- projectName
- deploymentId
- deploymentName
- blueprintId
- blueprintVersion
- blueprintName
- requestedBy(사용자)
- requestedAt(시간)

구문:

```
env.ENV_NAME
```

예:

```
${env.blueprintId}
```

## 리소스 변수

리소스 변수를 사용하면 다른 리소스의 리소스 속성에 바인딩할 수 있습니다.

구문:

```
resource.RESOURCE_NAME.PROPERTY_NAME
```

예:

- \${resource.db.id}
- \${resource.db.networks[0].address}
- \${resource.app.id}(count가 지정되지 않은 클러스터링되지 않은 리소스에 대해 문자열을 반환합니다. 클러스터링된 리소스에 대해서는 어레이를 반환합니다.)
- \${resource.app[0].id}(클러스터링된 리소스에 대한 첫 번째 항목을 반환합니다.)

## 리소스 자체 변수

리소스 자체 변수는 할당 단계를 지원하는 리소스에만 허용됩니다. 리소스 자체 변수는 할당 단계가 완료된 후에만 사용 가능하거나 값 집합을 가집니다.

구문:

```
self.property_name
```

예:

```
${self.address}(할당 단계 중 할당된 주소를 반환합니다.)
```

resource\_x, self.property\_name 및 resource.resource\_x.property\_name이라는 이름의 리소스는 동일하며, 둘 다 자체 참조로 고려됩니다.

## 클러스터 수 인덱스

구문:

```
count.index
```

예:

```
${count.index == 0 ? "primary" : "secondary"}(클러스터링된 리소스에 대한 노드 유형을 반환합니다.)
```

제한 사항:

리소스 할당에 대한 count.index 사용은 지원되지 않습니다. 예를 들어 입력 시간에 생성된 디스크의 어레이 내에서 위치를 참조할 때 다음 용량 표현식이 실패합니다.

```
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)}'
```

## 조건

구문:

- 같음 연산자는 == 및 !=입니다.
- 관계형 연산자는 < > <= 및 >=입니다.
- 논리 연산자는 && || 및 !입니다.

## ■ 조건부 패턴 사용:

*condition-expression ? true-expression : false-expression*

예:

```
${input.count < 5 && input.size == 'small'}
```

```
${input.count < 2 ? "small" : "large"}
```

## 산술 연산자

구문:

연산자는 + - / \* 및 %입니다.

예:

```
${(input.count + 5) * 2}
```

## 문자열 연결

구문:

`${'ABC' + 'DEF'}`는 ABCDEF로 평가됩니다.

## [] 및 . 연산자

[] 및 . 연산자에 대한 처리를 통일하기 위해 표현식은 ECMAScript를 따릅니다.

따라서 `expr.identifier`는 `expr["identifier"]`에 해당합니다. 식별자는 값이 식별자인 리터럴을 구성하는 데 사용되며 [] 연산자가 해당 값과 함께 사용됩니다.

예:

```
${resource.app.networks[0].address}
```

또한 속성에 공백이 포함되어 있는 경우 점 표기법을 사용하는 대신 대괄호와 큰따옴표로 구분합니다.

틀림:

```
input.operating system
```

맞음:

```
input["operating system"]
```

## 맵 구성

구문:

```
${{'key1':'value1', 'key2':input.key2}}
```

## 어레이 구성

구문:

```
${['key1','key2']}
```

예:

```
${[1,2,3]}
```

## 함수

구문:

```
${ 함수(인수...) }
```

예:

```
${to_lower(resource.app.name)}
```

**표 6-1. 함수**

함수	설명
<code>abs(숫자)</code>	절대 숫자 값
<code>avg(어레이)</code>	숫자 어레이에서 모든 값의 평균을 반환
<code>base64_decode(문자열)</code>	디코딩된 <b>base64</b> 값을 반환
<code>base64_encode(문자열)</code>	<b>base64</b> 로 인코딩된 값을 반환
<code>ceil(숫자)</code>	인수 보다 크거나 같고 수학 정수와 동일한 최소(음의 무한대에 가장 가까움) 값을 반환
<code>contains(어레이, 값)</code>	어레이에 값이 포함되어 있는지 확인
<code>contains(문자열, 값)</code>	문자열에 값이 포함되어 있는지 확인
<code>digest(값, 유형)</code>	지원되는 유형( <b>md5</b> , <b>sha1</b> , <b>sha256</b> , <b>sha384</b> , <b>sha512</b> )을 사용하여 값의 다이제스트를 반환
<code>ends_with(주체, 접미사)</code>	주체 문자열이 접미사 문자열로 끝나는지 확인
<code>filter_by(어레이, 필터)</code>	필터 작업을 통과한 어레이 항목만 반환합니다. <pre><code>filter_by([1,2,3,4], x =&gt; x &gt;= 2 &amp;&amp; x &lt;= 3)</code></pre> <b>[2, 3]</b> 반환 <pre><code>filter_by({'key1':1, 'key2':2}, (k,v) =&gt; v != 1)</code></pre> <b>[{"key2": 2}]</b> 반환
<code>floor(숫자)</code>	인수보다 작거나 같고 수학 정수와 동일한 최대(양의 무한대에 가장 가까움) 값을 반환
<code>format(형식, 값 ...)</code>	<b>Java Class Formatter</b> 형식 및 값을 사용하여 형식이 지정된 문자열을 반환
<code>from_json(문자열)</code>	<b>json</b> 문자열을 구문 분석
<code>join(어레이, 구분 기호)</code>	구분 기호로 문자열 어레이를 연결하고 문자열을 반환
<code>json_path(값, 경로)</code>	<b>XPath for JSON</b> 을 사용하여 값에 대해 경로를 평가
<code>keys(맵)</code>	맵의 키를 반환
<code>length(어레이)</code>	어레이 길이를 반환
<code>length(문자열)</code>	문자열 길이를 반환

표 6-1. 함수 (계속)

함수	설명
map_by(어레이, 작업)	작업이 적용된 각 어레이 항목을 반환합니다. <pre>map_by([1,2], x =&gt; x * 10)</pre> <p>[10, 20] 반환</p> <pre>map_by([1,2], x =&gt; to_string(x))</pre> <p>["1", "2"] 반환</p> <pre>map_by({'key1':1, 'key2':2}, (k,v) =&gt; {k:v*10})</pre> <p>[{"key1":10}, {"key2":20}] 반환</p>
map_to_object(어레이, 키 이름)	다른 어레이의 값과 쌍으로 구성된 지정된 키 이름의 <b>key:value</b> 쌍의 어레이를 반환합니다. <pre>map_to_object(resource.Disk[*].id, "source")</pre> <p>디스크 ID 문자열과 쌍으로 구성된 <b>source</b>라는 키 필드가 있는 <b>key:value</b> 쌍의 어레이를 반환합니다.</p> <p>다음</p> <pre>map_by(resource.Disk[*].id, id =&gt; {'source':id})</pre> <p>식은 동일한 결과를 반환합니다.</p>
matches(문자열, 정규식)	문자열이 정규 표현식과 일치하는지 확인
max(어레이)	숫자 어레이에서 최대값을 반환
merge(맵, 맵)	병합된 맵을 반환
min(어레이)	숫자 어레이에서 최소값을 반환
not_null(어레이)	<b>null</b> 이 아닌 첫 번째 항목을 반환
now()	현재 시간을 <b>ISO-8601</b> 형식으로 반환
range(시작, 중지)	시작 번호로 시작하여 중지 번호 바로 앞에서 끝나는, 1씩 증가하는 일련의 숫자를 반환합니다.
replace(문자열, 대상, 교체)	대상 문자열을 포함하는 문자열을 대상 문자열로 교체
reverse(어레이)	어레이 항목의 방향을 반전
slice(어레이, 시작, 끝)	시작 인덱스에서 끝 인덱스까지 어레이 조각을 반환
split(문자열, 분 기호)	구분 기호로 문자열을 분할하고 문자열 어레이를 반환
starts_with(주체, 접두사)	주체 문자열이 접두사 문자열로 시작하는지 확인
substring(문자열, 시작, 끝)	시작 인덱스에서 끝 인덱스까지 문자열의 하위 문자열을 반환
sum(어레이)	숫자 어레이에서 모든 값의 합계를 반환
to_json(값)	값을 <b>json</b> 문자열로 직렬화
to_lower(문자열)	문자열을 소문자로 변환
to_number(문자열)	문자열을 숫자로 구문 분석
to_string(값)	값의 문자열 표현을 반환

표 6-1. 함수 (계속)

함수	설명
to_upper(문자열)	문자열을 대문자로 변환
trim(문자열)	앞뒤 공백 제거
url_encode(문자열)	URL 인코딩 규칙을 사용하여 문자열을 인코딩
uuid()	임의로 생성된 UUID를 반환
values(맵)	맵의 값을 반환

## 비밀 vRealize Automation Cloud Assembly 속성을 생성하고 참조하는 방법

비밀 vRealize Automation Cloud Assembly 속성은 프로젝트 사용자가 클라우드 템플릿 설계에 추가할 수 있는 암호화된 값이며 재사용이 가능합니다.

보안 액세스 키 및 자격 증명은 비밀 속성의 전형적인 예입니다. 생성하고 저장한 후에는 비밀 속성 값을 암호화하거나 읽을 수 없습니다.

### 비밀 속성을 생성하는 방법

- 1 프로젝트 관리자 역할 권한을 사용하여 vRealize Automation Cloud Assembly에 로그인합니다.
- 2 **인프라 > 관리 > 비밀**로 이동하여 **새 비밀**을 클릭합니다.
- 3 프로젝트를 선택합니다.
- 4 비밀의 고유한 속성 이름(공백 또는 특수 문자 제외)을 입력합니다.  
이 이름은 비밀에 표시되는 식별자입니다.
- 5 비밀 값을 입력합니다.  
입력할 때 값이 기본적으로 가려지기 때문에 화면을 공유하는 경우 값이 보호됩니다.  
필요한 경우 눈 기호를 클릭하여 값을 표시하고 확인할 수 있습니다. 하지만 저장된 후에는 비밀 값이 데이터베이스에서 암호화되고 다시 노출될 수 없습니다.
- 6 선택적으로 비밀 속성에 대한 긴 설명을 입력합니다.
- 7 **생성**을 클릭합니다.

Create Secret

Project \* admin-project ⓘ

Name \* ourPublicKey ⓘ

Value \* ..... ⓘ

Description

CANCEL CREATE

## 클라우드 템플릿에 비밀 속성을 추가하는 방법

프로젝트 사용자는 클라우드 템플릿 코드의 바인딩으로 암호 속성을 추가할 수 있습니다.

'\${secret. 문자를 입력하기 시작하면 프로젝트에 대해 생성된 비밀의 선택 목록이 표시됩니다.

```
type: Cloud.Machine
properties:
  name: ourvm
  image: mint20
  flavor: small
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: '${secret.ourPublicKey}'
    username: root
```

Terraform 구성에 비밀 속성을 추가하려면 [Terraform 구성에서 비밀 vRealize Automation Cloud Assembly 속성을 사용하는 방법](#) 항목을 참조하십시오.

## vRealize Automation Cloud Assembly 클라우드 템플릿에서 원격 액세스를 사용하도록 설정하는 방법

vRealize Automation Cloud Assembly가 배포된 시스템에 원격으로 액세스하려면 배포 전에 해당 시스템의 클라우드 템플릿에 속성을 추가합니다.

원격 액세스의 경우 다음 인증 옵션 중 하나를 구성할 수 있습니다.

**참고** 키를 복사해야 하는 경우에는 프로비저닝 시 키를 자동으로 복사하기 위해 클라우드 템플릿에 cloudConfig 섹션을 생성할 수도 있습니다. 구체적인 내용은 여기에 설명되어 있지 않지만 [vRealize Automation Cloud Assembly](#) 템플릿에서 시스템을 자동으로 초기화하는 방법에 cloudConfig에 대한 일반 정보가 제공됩니다.

## vRealize Automation Cloud Assembly 프로비저닝 시 키 쌍 생성

원격 액세스 인증을 위한 자체적인 공용-개인 키 쌍이 없는 경우 vRealize Automation Cloud Assembly에서 키 쌍을 생성할 수 있습니다.

다음 코드를 지침으로 사용합니다.

- 1 vRealize Automation Cloud Assembly에서 프로비저닝하기 전에 예시에 표시된 대로 클라우드 템플릿에 `remoteAccess` 속성을 추가합니다.

사용자 이름은 선택 사항입니다. 이 옵션을 생략하면 시스템에서 임의의 ID가 사용자 이름으로 생성됩니다.

예:

```
type: Cloud.Machine
properties:
  name: our-vm2
  image: Linux18
  flavor: small
  remoteAccess: authentication: generatedPublicPrivateKey username: testuser
```

- 2 vRealize Automation Cloud Assembly에서 시스템을 해당 클라우드 템플릿에서 프로비저닝하고 시작됨 상태로 전환합니다.

프로비저닝 프로세스를 통해 키가 생성됩니다.

- 3 **배포 > 토폴로지** 속성에서 키 이름을 찾습니다.

- 4 vSphere 클라이언트와 같은 클라우드 제공자 인터페이스를 사용하여 프로비저닝된 시스템 명령줄에 액세스합니다.

- 5 개인 키에 읽기 권한을 부여합니다.

```
chmod 600 키-이름
```

- 6 vRealize Automation Cloud Assembly 배포로 이동하여 시스템을 선택한 후 **작업 > 개인 키 얻기**를 클릭합니다.

- 7 개인 키 파일을 로컬 시스템에 복사합니다.

일반적인 로컬 파일 경로는 `/home/username/.ssh/키-이름`입니다.

- 8 원격 SSH 세션을 열고 프로비저닝된 시스템에 연결합니다.

```
ssh -i 키-이름 사용자-이름@시스템-ip
```

## vRealize Automation Cloud Assembly에 자체 공용-개인 키 쌍 제공

많은 기업이 인증을 위해 자체 공용-개인 키 쌍을 생성하고 배포합니다.

다음 코드를 지침으로 사용합니다.

- 1 로컬 환경에서 공개-개인 키 쌍을 얻거나 생성합니다.

지금은 로컬에서 키를 생성하고 저장합니다.

- 2 vRealize Automation Cloud Assembly에서 프로비저닝하기 전에 예시에 표시된 대로 클라우드 템플릿에 `remoteAccess` 속성을 추가합니다.

`sshKey`에는 공개 키 파일 `key-name.pub`에서 발견된 긴 영숫자가 포함됩니다.

사용자 이름은 선택 사항이며 로그인에 사용할 수 있도록 생성됩니다. 이 옵션을 생략하면 시스템에서 임의의 ID가 사용자 이름으로 생성됩니다.

예:

```
type: Cloud.Machine
properties:
  name: our-vm1
  image: Linux18
  flavor: small
  remoteAccess:
    authentication: publicPrivateKey
    sshKey: ssh-rsa Iq+5aQgBP3ZNT4o1baP5Ii+dstIcowRRkyobbfpA1mj9ts1f
qGxvU66PX9IeZax5hZvNWFgjw6ag+Z1zndOLhVdVoW49f274/mIRild7UUW...
    username: testuser
```

- 3 vRealize Automation Cloud Assembly에서 시스템을 해당 클라우드 템플릿에서 프로비저닝하고 시작됨 상태로 전환합니다.
- 4 클라우드 벤더 클라이언트를 사용하여 프로비저닝된 시스템에 액세스합니다.
- 5 시스템의 home 폴더에 공용 키 파일을 추가합니다. remoteAccess.sshKey에 지정한 키를 사용합니다.
- 6 개인 키 파일 해당 항목이 로컬 시스템에 있는지 확인합니다.

이 키는 일반적으로 .pub 확장명이 없는 /home/username/.ssh/key-name입니다.

- 7 원격 SSH 세션을 열고 프로비저닝된 시스템에 연결합니다.

```
ssh -i 키-이름 사용자-이름@시스템-ip
```

## vRealize Automation Cloud Assembly에 AWS 키 쌍 제공

AWS 키 쌍 이름을 클라우드 템플릿에 추가하면 vRealize Automation Cloud Assembly가 AWS에 배포하는 시스템에 원격으로 액세스할 수 있습니다.

AWS 키 쌍은 지역별로 다르다는 점에 유의하십시오. 워크로드를 us-east-1로 프로비저닝하는 경우, us-east-1에 키 쌍이 존재해야 합니다.

다음 코드를 지침으로 사용합니다. 이 옵션은 AWS 클라우드 영역에만 적용됩니다.

```
type: Cloud.Machine
properties:
  image: Ubuntu
  flavor: small
  remoteAccess: authentication: keyPairName keyPair: cas-test
constraints:
  - tag: 'cloud:aws'
```

## vRealize Automation Cloud Assembly에 사용자 이름 및 암호 제공

클라우드 템플릿에 사용자 이름과 암호를 추가하여 vRealize Automation Cloud Assembly가 배포하는 시스템에 간단하게 원격 액세스할 수 있습니다.

보안 수준은 떨어지지만 사용자 이름과 암호를 사용하여 원격으로 로그인하는 것이 상황에 따라 필요할 수 있습니다. 일부 클라우드 벤더 또는 구성에서는 이 보안 수준이 낮은 옵션을 지원하지 않을 수 있습니다.

- 1 vRealize Automation Cloud Assembly에서 프로비저닝하기 전에 예시에 표시된 대로 클라우드 템플릿에 `remoteAccess` 속성을 추가합니다.

로그인하는 데 사용할 계정으로 사용자 이름 및 암호를 설정합니다.

예:

```
type: Cloud.Machine
properties:
  name: our-vm3
  image: Linux18
  flavor: small
  remoteAccess: authentication: usernamePassword username: testuser password: admin123
```

- 2 vRealize Automation Cloud Assembly에서 시스템을 해당 클라우드 템플릿에서 프로비저닝하고 시작됨 상태로 전환합니다.
- 3 클라우드 벤더의 인터페이스로 이동하여 프로비저닝된 시스템에 액세스합니다.
- 4 프로비저닝된 시스템에서 계정을 생성하거나 사용하도록 설정합니다.
- 5 로컬 시스템에서 프로비저닝된 시스템 IP 주소 또는 FQDN에 대한 원격 세션을 열고 평소와 같이 사용자 이름과 암호를 사용하여 로그인합니다.

## 서로 다른 vRealize Automation Cloud Assembly 설계에서 동일한 속성을 재사용하는 방법

항상 함께 나타나는 vRealize Automation Cloud Assembly 속성이 있는 경우 이러한 속성을 속성 그룹으로 조합할 수 있습니다.

서로 다른 vRealize Automation Cloud Assembly 설계에 속성 그룹을 빠르게 추가하여 동일한 여러 속성을 하나씩 추가하는 시간을 절약할 수 있습니다. 또한 단일 위치에서 속성 집합을 유지 보수하거나 수정할 수 있으므로 일관성 있는 적용이 가능합니다.

vRealize Automation Cloud Assembly 관리자 역할이 있는 사용자만 속성 그룹을 생성, 업데이트 또는 삭제할 수 있습니다. 관리자는 속성 그룹을 전체 조직과 공유하거나 프로젝트 내에서만 사용하도록 제한할 수 있습니다.

**경고** 속성 그룹은 카탈로그에 이미 릴리스된 그룹을 포함하여 많은 클라우드 템플릿에 포함될 수 있습니다. 속성 그룹을 변경하면 다른 사용자에게 영향을 줄 수 있습니다.

속성 그룹에는 두 가지 유형이 있습니다.

### ■ vRealize Automation Cloud Assembly에서 입력 속성 그룹을 생성하고 사용하는 방법

입력 속성 그룹은 사용자 요청 시 연관된 속성 집합을 수집하고 적용합니다. 입력 속성 그룹에는 사용자가 추가하거나 선택할 수 있는 항목이 포함되거나, 설계에 필요한 읽기 전용 값이 포함될 수 있습니다.

사용자가 편집하거나 선택할 수 있는 속성은 읽기 또는 암호화할 수 있습니다. 읽기 전용 속성은 요청 양식에 나타나지만 편집할 수는 없습니다. 읽기 전용 값을 완전히 숨김으로 유지하려면 상수 속성 그룹을 대신 사용합니다.

#### ■ vRealize Automation Cloud Assembly에서 상수 속성 그룹을 생성하고 사용하는 방법

상수 속성 그룹은 알려진 속성을 자동으로 적용합니다. 실제로 상수 속성 그룹은 표시되지 않는 메타데이터로, 요청하는 사용자가 해당 값을 읽지 못하게 하거나 심지어 해당 값이 존재한다는 것을 알 수 없게 하는 방식으로 vRealize Automation Cloud Assembly 설계에 값을 제공합니다. 라이선스 키 또는 도메인 계정 자격 증명을 예로 들 수 있습니다.

두 속성 그룹 유형은 vRealize Automation Cloud Assembly에서 매우 다르게 처리됩니다. 속성 그룹을 생성하는 경우 먼저 입력을 생성할지 아니면 상수를 생성할지 선택해야 합니다. 혼합 속성 그룹을 생성하거나 기존 속성 집합 및 해당 속성 그룹의 유형을 다른 유형으로 변환할 수 없습니다.

### vRealize Automation Cloud Assembly에서 입력 속성 그룹을 생성하고 사용하는 방법

vRealize Automation Cloud Assembly 입력 속성 그룹은 일반적으로 사용자가 입력하거나 선택할 수 있는 관련 설정을 포함합니다. 클라우드 템플릿 설계에 필요한 읽기 전용 값도 포함될 수 있습니다.

#### 입력 속성 그룹 생성

- 1 **설계 > 속성 그룹**으로 이동하고 **새 속성 그룹**을 클릭합니다.
- 2 **입력 값**을 선택합니다.
- 3 새 속성 그룹의 이름과 설명을 입력합니다.

이름	속성 그룹 이름은 지정된 조직 내에서 고유해야 합니다. 문자, 숫자 및 밑줄만 허용됩니다.
표시 이름	요청 양식에 표시되는 전체 속성 그룹에 대한 머리글을 추가합니다.
설명	이 속성 집합의 용도를 설명합니다.
범위	관리자가 속성 그룹을 전체 조직과 공유할 수 있는지 여부를 결정합니다. 그렇지 않으면 하나의 프로젝트만 속성 그룹에 액세스할 수 있습니다.  그룹에서 항상 속성을 추가하거나 수정할 수 있지만, 범위는 영구적이며 나중에 변경할 수 없습니다.
프로젝트	범위가 프로젝트 전용이면 이 프로젝트는 속성 그룹에 액세스할 수 있습니다.

- 4 그룹에 속성을 추가하려면 **새 속성**을 클릭합니다.

새 속성을 추가하기 위한 패널은 vRealize Automation Cloud Assembly 설계 페이지 코드 편집기의 [입력] 탭과 매우 유사합니다.

이름	개별 속성의 자유 형식 이름입니다. 문자, 숫자 및 밑줄만 허용됩니다.
표시 이름	요청 양식에 표시할 개별 속성 이름을 추가합니다.

유형	문자열, 정수, 숫자, 부울(T/F), 개체 또는 어레이입니다.
기본값	요청 양식에 표시되는 미리 설정된 값 항목입니다.
암호화됨	선택하면 요청 양식에 입력할 때 그리고 나중에 배포할 때 값을 숨깁니다. 암호화된 속성은 기본값을 가질 수 없습니다.
읽기 전용	요청 양식에서 볼 수 있지만 편집할 수는 없는 값입니다. 기본값이 필요합니다.
추가 옵션	속성 유형에 따라 달라지는 옵션입니다. 드롭다운을 확장하고 추가 설정을 추가한 후 <b>생성</b> 을 클릭합니다.

다음 예에서 추가되는 속성은 운영 체제 이미지를 나타내고, 요청하는 사용자는 둘 중에서 선택할 수 있습니다.

**참고** 예제 그림에 표시된 운영 체제는 이미 구성된 vRealize Automation Cloud Assembly 인프라에 속해 있어야 합니다.

### New Property

**Name \*** image

**Display Name** Machine Image

**Description**

**Type**

STRING INTEGER NUMBER BOOLEAN OBJECT ARRAY

**Default value** coreos

**Encrypted** ☐

**Read-only** ☐ ⓘ

▼ More Options

**Format** ▼

**Minimum length**

**Maximum length**

**Pattern**

**Pairs** ☒ Values ☐ Key Values

**Enum**

Value

coreos -

ubuntu - +

- 5 그룹에 속성을 더 추가하고 완료했으면 **저장**을 클릭합니다.

### Properties 2 items

Add at least one property in order to create a property group

[+ NEW PROPERTY](#) [x DELETE](#)

<input type="checkbox"/>	Name	Display Name	Type	Default Value
<input type="checkbox"/>	image	Machine Image	string	coreos
<input type="checkbox"/>	flavor	Machine Flavor	string	small

## 클라우드 템플릿 입력에 속성 그룹 추가

속성 입력의 긴 목록에 대해서도 속성 그룹을 추가하기만 하면 요청 양식의 일부로 만들 수 있습니다.

- 1 클라우드 템플릿 설계 페이지 오른쪽의 편집 영역 위에서 **입력** 탭을 클릭합니다.
- 2 **새 클라우드 템플릿 입력**을 클릭합니다.
- 3 속성 그룹의 이름과 설명을 입력합니다.

이름	앞에서 생성한 속성 그룹 이름과 유사한 이름을 입력합니다.
표시 이름	전체 속성 그룹에 대해 이전에 생성한 것과 동일한 머리글을 입력합니다. 이것이 요청 양식에 나타납니다.
유형	<b>개체</b> 를 선택합니다.
개체 유형	<b>속성 그룹</b> 을 선택합니다.
속성 그룹 목록	원하는 속성 그룹을 선택합니다. 프로젝트에 대해 생성되고 사용 가능한 속성 그룹만 표시됩니다. 상수 속성 그룹은 표시되지 않습니다.

**New Cloud Template Input** [X]

**Name \***

**Display Name**

**Description**

**Type**

STRING INTEGER NUMBER BOOLEAN **OBJECT** ARRAY

**Select Object Type** ☐ Properties ☒ Property Groups

Select from the existing property groups

Q

Name	Description
<input checked="" type="radio"/> machine	

- 4 **생성**을 클릭합니다.

이 프로세스는 다음 예와 유사한 클라우드 템플릿 입력 코드를 생성합니다.

```
inputs:
  pgmachine:
    type: object
    title: Machine Properties
    $ref: /ref/property-groups/machine
  pgrequester:
    type: object
    title: Requester Details
    $ref: /ref/property-groups/requesterDetails
```

vRealize Automation Cloud Assembly 설계 페이지에 코드를 직접 입력하고 코드 편집기에서 \$ref: /ref/p...를 입력할 때 자동 프롬프트를 활용할 수도 있습니다.

### 속성 그룹에 클라우드 템플릿 리소스 바인딩

속성 그룹 입력 값을 사용하려면 리소스 아래에 바인딩을 추가합니다.

속성 그룹에 있는 값의 종류에 따라 이러한 값을 개별적으로 참조할 수 있습니다. 속성 그룹 이름 및 속성 이름을 기준으로 개별적으로 입력할 수 있습니다.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: '${input.pgmachine.image}'
      flavor: '${input.pgmachine.flavor}'
```

전체 속성 그룹을 참조하여 전체 값 집합을 리소스에 빠르게 추가할 수도 있습니다.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      requester: '${input.pgrequester}'
```

### 완성된 코드

입력 및 리소스를 완료하면 완성된 코드가 다음 예와 유사하게 보입니다.

```

>> Code Properties Inputs
1 formatVersion: 1
2 inputs:
3   pgmachine:
4     type: object
5     title: Machine Properties
6     $ref: /ref/property-groups/machine
7   pgrequester:
8     type: object
9     title: Requester Details
10    $ref: /ref/property-groups/requesterDetails
11  count:
12    type: integer
13    title: 'Machine Count'
14  resources:
15    Cloud_Machine_1:
16      type: Cloud.Machine
17      properties:
18        image: '${input.pgmachine.image}'
19        flavor: '${input.pgmachine.flavor}'
20        count: '${input.count}'
21        requester: '${input.pgrequester}'
22

```

배포 요청이 완료되면 요청하는 사용자가 완료하도록 속성 그룹이 나타납니다.

### Deployment Inputs

Machine Properties

Machine Image 
Machine Flavor

Requester Details

Email 
Mobile 
Internal account? ☐
PIN 
Account Type

Machine Count \*

### vRealize Automation Service Broker 사용자 지정 양식 편집기의 속성 그룹

입력 속성 그룹은 vRealize Automation Service Broker 사용자 지정 양식 인터페이스 내에 표시되며 여기에서 사용자 지정할 수 있습니다. 사용자 지정할 경우 속성 그룹 고유의 특별한 고려 사항은 없습니다. 항목의 소스가 개별적으로 생성된 속성이 아니라 속성 그룹이라는 것을 vRealize Automation Service Broker 사용자가 알아야 할 필요는 없습니다.

The screenshot shows the 'General' tab of a configuration form. It includes the following fields:

- Project**: A text input field with a dropdown arrow.
- Deployment Name**: A text input field.
- Machine Count**: A text input field.
- Machine Properties** (grouped in a dashed orange box):
  - Machine Image**: A text input field with a dropdown arrow.
  - Machine Flavor**: A text input field with a dropdown arrow.
- Requester Details** (grouped in a dashed orange box):
  - Email**: A text input field.
  - Mobile**: A text input field.
  - Internal account?**: A checkbox.
  - PIN**: A text input field.
  - Account Type**: A text input field.

자세한 내용은 [Service Broker](#) 아이콘 및 요청 양식 사용자 지정을 참조하십시오.

## vRealize Automation Cloud Assembly에서 상수 속성 그룹을 생성하고 사용하는 방법

vRealize Automation Cloud Assembly 상수를 사용하면 알려진 키-값 쌍을 설계에 자동으로 적용할 수 있습니다.

키가 클라우드 템플릿 코드에 나타나고 값이 해당 클라우드 템플릿을 기반으로 하는 배포의 일부가 됩니다. 상수에는 리소스 아래에 `propgroup` 바인딩이 필요합니다.

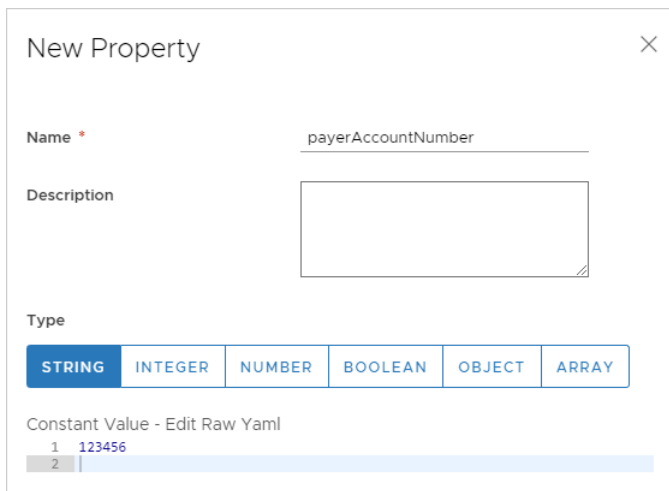
`propgroup` 바인딩은 입력 속성 그룹이 아닌 상수 속성 그룹과 함께만 사용됩니다.

### 상수 속성 그룹 생성

- 1 **설계 > 속성 그룹**으로 이동하고 **새 속성 그룹**을 클릭합니다.
- 2 **상수 값**을 선택합니다.
- 3 새 속성 그룹의 이름과 설명을 입력합니다.

이름	속성 그룹 이름은 지정된 조직 내에서 고유해야 합니다. 문자, 숫자 및 밑줄만 허용됩니다.
표시 이름	비워 둡니다. 요청 양식에 머리글이 표시되지 않습니다.
설명	이 상수 집합의 용도를 설명합니다.
범위	관리자가 속성 그룹을 전체 조직과 공유할 수 있는지 여부를 결정합니다. 그렇지 않으면 하나의 프로젝트만 속성 그룹에 액세스할 수 있습니다.  그룹에서 항상 속성을 추가하거나 수정할 수 있지만, 범위는 영구적이며 나중에 변경할 수 없습니다.
프로젝트	범위가 프로젝트 전용이면 이 프로젝트는 속성 그룹에 액세스할 수 있습니다.

- 4 그룹에 상수 속성을 추가하려면 **새 속성**을 클릭합니다.
- 5 키로 작동하는 이름과 설명을 입력합니다.
- 6 속성 유형을 선택하고 null를 원하는 값으로 바꿉니다. **생성**을 클릭합니다.

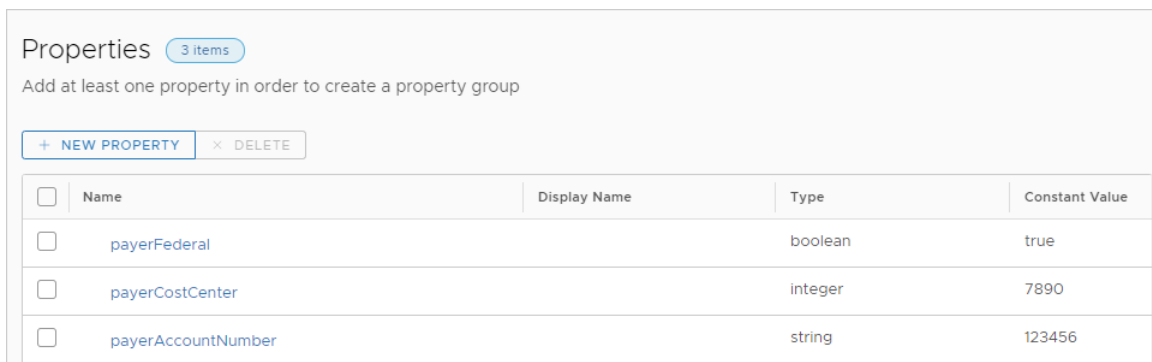


The 'New Property' dialog box contains the following fields and options:

- Name \***: A text input field containing 'payerAccountNumber'.
- Description**: A text area for additional details.
- Type**: A row of buttons for selecting the property type: STRING (selected), INTEGER, NUMBER, BOOLEAN, OBJECT, and ARRAY.
- Constant Value - Edit Raw Yaml**: A code editor showing a list of values:
 

```
1 123456
2
```

- 7 그룹에 상수를 더 추가하고 완료했으면 **저장**을 클릭합니다.



The 'Properties' panel shows a list of 3 items. Below the list are buttons for '+ NEW PROPERTY' and 'X DELETE'.

<input type="checkbox"/>	Name	Display Name	Type	Constant Value
<input type="checkbox"/>	payerFederal		boolean	true
<input type="checkbox"/>	payerCostCenter		integer	7890
<input type="checkbox"/>	payerAccountNumber		string	123456

## 속성 그룹에 클라우드 템플릿 리소스 바인딩

리소스 내에서 상수 값을 자동으로 사용하려면 리소스 아래에 `propgroup` 바인딩을 추가합니다.

속성 그룹 자체를 참조하여 전체 상수 집합을 리소스에 빠르게 추가할 수 있습니다.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      payerInfo: '${propgroup.payerDetails}'
```

또는 속성 그룹의 개별 상수를 설계의 선택한 부분에 추가할 수 있습니다.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      payerAccount: '${propgroup.payerDetails.payerAccountNumber}'
      payerCost: '${propgroup.payerDetails.payerCostCenter}'
      payerFed: '${propgroup.payerDetails.payerFederal}'
```

## vRealize Automation Cloud Assembly 템플릿에서 소켓당 코어 및 CPU 수를 정의하는 방법

vRealize Automation Cloud Assembly 템플릿 코드를 사용하면 vSphere 시스템 리소스에 대한 소켓당 코어 수를 지정할 수 있습니다.

가상 소켓당 코어 수 또는 총 소켓 수를 지정할 수 있습니다. 예를 들어 보유한 라이선싱 조건에 따라 소켓당 라이선스가 부여된 소프트웨어가 제한되거나, 사용 가능한 운영 체제에서 특정 수의 소켓만 인식하여 추가 CPU를 추가 코어에 프로비저닝해야 할 수 있습니다.

vSphere 시스템 리소스의 클라우드 템플릿에 `coreCount` 속성을 추가합니다.

`coreCount` 값은 버전 매핑 또는 클라우드 템플릿의 vSphere 시스템 리소스 코드에 지정된 CPU 수 (`cpuCount`) 값 보다 작거나 같아야 합니다. 관련 정보는 [가상 시스템의 CPU당 코어 수 설정\(1010184\)](#)을 참조하십시오.

`coreCount` 속성은 선택 사항이며 vSphere 시스템 리소스에만 사용할 수 있습니다.

예제 vSphere 시스템 리소스 코드 조각은 아래에 나와 있습니다.

```
Cloud_vSphere_Machine_1:
  type: Cloud.vSphere.Machine
  properties:
    cpuCount: 8
    coreCount: 4
```

소켓 및 소켓당 코어 수 설정에 대한 추가 정보는 블로그 문서인 [가상 시스템 vCPU 및 vNUMA 최적 크기 조정 - 지침](#)을 참조하십시오.

## vRealize Automation Cloud Assembly 설계에 고급 기능을 추가하는 방법

설계의 엔터프라이즈 준비를 향상하는 vRealize Automation Cloud Assembly 기능 및 고급 IaC(Infrastructure-as-Code) 기술이 있습니다.

여기에 설명된 일부 기능은 vRealize Automation Cloud Assembly의 설계 기능을 확장하며 다른 기능은 클라우드 템플릿 코딩 방식에 직접 적용됩니다.

### vRealize Automation Cloud Assembly를 사용하여 배포된 리소스의 이름을 사용자 지정하는 방법

클라우드 또는 프로젝트 관리자는 환경의 리소스에 대해 규정된 이름 지정 규칙을 사용하며, 배포된 리소스가 사용자 상호 작용 없이 해당 규칙을 따르도록 설정할 수 있습니다. vRealize Automation Cloud Assembly 프로젝트에서 모든 배포에 대한 이름 지정 템플릿을 생성할 수 있습니다.

예를 들어 호스트 이름 지정 규칙은 리소스에 *projectname-sitecode-costcenter-whereDeployed-identifier*를 접두사로 지정하는 것입니다. 각 프로젝트의 시스템에 대한 사용자에게 맞게 이름 지정 템플릿을 구성합니다. 템플릿 변수 중 일부는 배포될 때 시스템에서 가져오고, 다른 변수는 프로젝트 사용자 지정 속성에 기반합니다. 위 접두사에 대한 사용자 지정 명명 템플릿은 다음 예와 유사합니다.

```
${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}
```

식별자는 템플릿에 `${#####}`으로 제공되며, 6자리 식별자를 나타냅니다. 식별자는 고유성을 보장하는 카운터입니다. 카운터는 조직 전체에 적용되며 현재 프로젝트뿐만 아니라 모든 프로젝트에서 증가합니다. 여러 프로젝트가 있는 경우 현재 프로젝트의 배포에 대해 000123에서 000124로의 시퀀스를 기대하지 마십시오. 000123에서 000127로의 증가를 기대할 수 있습니다.

모든 리소스 이름은 고유해야 합니다. 증분 숫자 속성을 사용하면 고유성을 보장할 수 있습니다. vRealize Automation Cloud Assembly에서 이름이 지정된 배포를 포함하여 모든 배포에 대해 숫자가 증가합니다. 시스템이 더욱 강력해지고 사용자 지정 이름이 가상 시스템, 로드 밸런서, 보안 그룹, NAT, 게이트웨이, 리소스 그룹, 디스크를 비롯한 많은 리소스에 적용되기 때문에 번호 매기기가 무작위로 나타날 수 있지만 값은 여전히 고유성을 보장합니다. 이 숫자는 테스트 배포를 실행할 때도 증가합니다.

여기에 제공된 예제 외에도 사용자 이름, 사용되는 이미지, 기타 기본 제공 옵션 및 간단한 문자열을 추가할 수도 있습니다. 템플릿을 구축할 때 가능한 옵션에 관한 힌트가 제공됩니다.

여기에 나와 있는 일부 값은 사용 사례 예시일 뿐입니다. 사용자 환경에서 이러한 값을 그대로 사용해서는 안 됩니다. 클라우드 인프라 및 배포 관리 요구 사항에 맞게 고유한 값으로 대체하거나 예시의 값에서 추론할 수 있는 부분에 대해 생각해 보십시오.

#### 사전 요구 사항

- 프로젝트에서 배포에 사용할 이름 지정 규칙을 알고 있는지 확인합니다.
- 이 절차에서는 사용자 지정 호스트 접두사 이름 지정을 테스트하는 데 사용할 간단한 클라우드 템플릿이 있거나 생성할 수 있다고 가정합니다.

## 절차

- 1 **인프라 > 프로젝트**를 선택합니다.
- 2 기존 프로젝트를 선택하거나 새로 생성합니다.
- 3 **프로비저닝** 탭에서 [사용자 지정 속성] 섹션을 찾아서 사이트 코드 및 비용 센터 값에 대한 속성을 생성합니다.

여기에 표시된 값을 사용자 환경과 관련된 값으로 바꿀 수 있습니다.

Custom Properties

Specify the custom properties that should be added to all requests in this project. ⓘ

Define custom properties	Name	Value
	siteCode	BGL
	costCenter	IT-research

Custom Naming

Specify the naming template to be used for machines provisioned in this project.

Template: \${project.name}-\${resource.siteCode}-\${resource.co... ⓘ

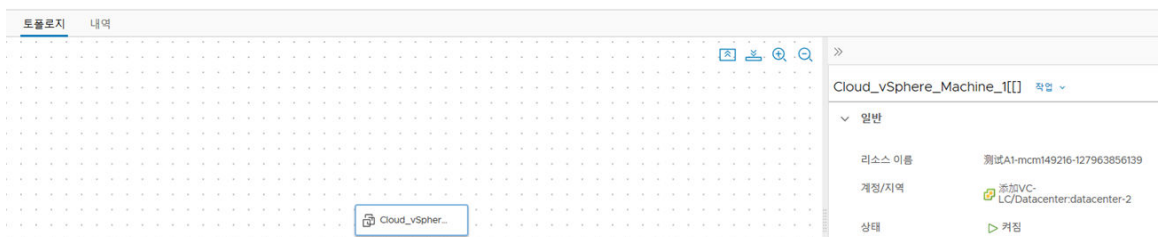
- a 이름이 **siteCode**이고 값이 **BGL**인 사용자 지정 속성을 생성합니다.
  - b 이름이 **costCenter**이고 값이 **IT-research**인 또 다른 사용자 지정 속성을 추가합니다.
- 4 [사용자에 맞게 이름 지정] 섹션을 찾아서 다음 템플릿을 추가합니다.

```
${project.name}-${resource.siteCode}-${resource.costCenter}-${endpoint.name}-${#####}
```

문자열로 복사할 수 있지만, 첫 번째 이름 지정 템플릿인 경우에는 템플릿을 빌드할 때 힌트 텍스트와 빠른 선택을 사용하는 것이 좋습니다.

- 5 프로젝트와 연결된 클라우드 템플릿을 배포하여 사용자 지정 이름이 리소스에 적용되었는지 확인합니다.
  - a **설계** 탭을 클릭한 다음 프로젝트와 연결된 클라우드 템플릿을 클릭합니다.
  - b 클라우드 템플릿을 배포합니다.
 

**배포** 탭이 열리고 진행 중인 배포가 표시됩니다.
  - c 배포가 완료되면 배포 이름을 클릭합니다.
  - d **토폴로지** 탭에서 사용자 지정 이름이 오른쪽 창에 있는 리소스 이름인지 확인합니다.



- 6 이름 지정 규칙을 확인하기 위해 테스트 클라우드 템플릿을 배포한 경우에는 배포를 삭제할 수 있습니다.

#### 다음에 수행할 작업

다른 프로젝트에 대한 사용자에게 맞게 이름 지정 템플릿을 생성합니다.

## vRealize Automation Cloud Assembly에서 시스템 및 디스크 클러스터를 생성하는 방법

vRealize Automation Cloud Assembly 템플릿 설계는 시스템 클러스터를 배포하고 디스크 클러스터를 연결할 수 있습니다.

시스템 및 디스크 클러스터를 배포하려면 클라우드 템플릿에서 `allocatePerInstance` **vRealize Automation Cloud Assembly** 리소스 플래그가 요청을 사용자 지정하는 방법과 `count.index` 및 `map_to_object` **vRealize Automation Cloud Assembly**의 클라우드 템플릿 표현식 구문을 활용하십시오.

다음 클라우드 템플릿 코드 예는 클러스터를 배포하는 설계에 대한 지침으로 사용할 수 있습니다.

### 디스크 클러스터를 공유하는 2개의 시스템:

```
resources:
  app0:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, 0,2), "source")}'
  appl:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, 2,4), "source")}'
  disk:
    type: Cloud.Volume
    allocatePerInstance: true
    properties:
      count: 4
      capacityGb: 5
```

### 각각 디스크가 하나씩 있는 가변적 시스템 수:

```
inputs:
  count:
    type: integer
    default: 2
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
```

```

    allocatePerInstance: true
  properties:
    image: ubuntu
    flavor: small
    count: '${input.count}'
    attachedDisks: '${map_to_object(slice(resource.disk[*].id, count.index, count.index +
1), "source")}'
  disk:
    type: Cloud.Volume
    allocatePerInstance: true
    properties:
      count: '${input.count}'
      capacityGb: 5

```

## 각각 디스크가 2개씩 있는 가변적 시스템 수:

```

inputs:
  count:
    type: integer
    default: 2
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:
      image: ubuntu
      flavor: small
      count: ${input.count}
      attachedDisks: '${map_to_object(slice(resource.disk[*].id, 2*count.index,
2*(count.index + 1)), "source")}'
    disk:
      type: Cloud.Volume
      allocatePerInstance: true
      properties:
        count: ${2*input.count}
        capacityGb: 5

```

## 요청 시 디스크 크기 설정:

```

inputs:
  disksize:
    type: array
    minItems: 2
    maxItems: 2
    items:
      type: object
      properties:
        size:
          type: integer
resources:
  app:
    type: Cloud.Machine
    allocatePerInstance: true
    properties:

```

```

    flavor: small
    image: ubuntu
    attachedDisks: ${map_to_object(slice(resource.disk[*].id, 0, 2), 'source')}
  disk:
    type: Cloud.Volume
    allocatePerInstance: true
    properties:
      count: 2
      capacityGb: ${input.disksize[count.index].size}

```

## vRealize Automation Cloud Assembly 템플릿에서 시스템을 자동으로 초기화하는 방법

vRealize Automation Cloud Assembly에서 명령을 직접 실행하거나 vSphere 기반 클라우드 영역에 배포하는 경우 사용자 지정 규격을 통해 시스템 초기화를 적용할 수 있습니다.

- 명령—클라우드 템플릿 코드의 `cloudConfig` 섹션에는 실행하려는 명령이 있습니다.
- 사용자 지정 규격—클라우드 템플릿 코드의 속성은 이름으로 vSphere 사용자 지정 규격을 참조합니다.

### 명령과 사용자 지정 규격이 혼합되지 않을 수 있음

vSphere에 배포하는 경우 `cloudConfig`와 사용자 지정 규격 초기화를 결합하려면 주의해서 진행해야 합니다. 공식적으로 호환되지 않으며 함께 사용하면 일관성이 없거나 원치 않는 결과가 발생할 수 있습니다.

명령과 사용자 지정 규격이 상호 작용하는 방식에 대한 예는 [vRealize Automation Cloud Assembly 클라우드 템플릿에서 vSphere 고정 IP 주소 할당](#)에서 참조하십시오.

## vRealize Automation Cloud Assembly 템플릿의 vSphere 사용자 지정 규격

vSphere 기반 클라우드 영역에 배포할 때, 사용자 지정 규격이 게스트 운영 체제 설정을 배포 시 적용할 수 있습니다.

### 사용자 지정 규격을 사용하도록 설정하는 방법

사용자 지정 규격은 배포 대상인 vSphere에 존재해야 합니다.

클라우드 템플릿 코드를 직접 편집합니다. 다음 예는 vSphere의 WordPress 호스트에 대한 `cloud-assembly-linux` 사용자 지정 규격을 가리킵니다.

```

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'
```

## 사용자 지정 규격 또는 cloudConfig 명령 사용 여부

프로비저닝 환경이 현재 vSphere에서 수행 중인 작업과 일치되도록 하려면 사용자 지정 규격을 계속 사용하는 것이 가장 좋은 방법일 수 있습니다. 단, 하이브리드 또는 다중 클라우드 프로비저닝으로 확장하기 위한 보다 중립적인 접근 방식은 cloudConfig 초기화 명령입니다.

클라우드 템플릿의 cloudConfig 섹션에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 템플릿의 구성 명령](#)의 내용을 참조하십시오.

## 명령과 사용자 지정 규격이 혼합되지 않을 수 있음

vSphere에 배포하는 경우, 내장된 cloudConfig 명령과 사용자 지정 규격 초기화를 결합하려면 주의해서 진행해야 합니다. 공식적으로 호환되지 않으며 함께 사용하면 일관성이 없거나 원치 않는 결과가 발생할 수 있습니다.

명령과 사용자 지정 규격이 상호 작용하는 방식에 대한 예는 [vRealize Automation Cloud Assembly 클라우드 템플릿에서 vSphere 고정 IP 주소 할당](#)에서 참조하십시오.

## vRealize Automation Cloud Assembly 템플릿의 구성 명령

vRealize Automation Cloud Assembly 템플릿 코드에 cloudConfig 섹션을 추가하여, 배포 시 실행되는 시스템 초기화 명령을 추가할 수 있습니다.

### cloudConfig 명령의 구성 방식

- Linux—초기화 명령은 개방형 [cloud-init](#) 표준을 따릅니다.
- Windows—초기화 명령은 [Cloudbase-init](#)를 사용합니다.

Linux [cloud-init](#) 및 Windows [Cloudbase-init](#)는 동일한 구문을 공유하지 않습니다. 한 운영 체제의 cloudConfig 섹션이 다른 운영 체제의 시스템 이미지에서 작동하지 않습니다.

### cloudConfig 명령으로 수행할 수 있는 작업

초기화 명령을 사용하여 인스턴스 생성 시 데이터 또는 설정 적용을 자동화하면 사용자, 권한, 설치 또는 기타 명령 기반 작업을 사용자 지정할 수 있습니다. 예는 다음과 같습니다.

- 호스트 이름 설정
- SSH 개인 키 생성 및 설정
- 패키지 설치

## cloudConfig 명령을 추가할 수 있는 위치

cloudConfig 섹션을 클라우드 템플릿 코드에 추가할 수 있지만 인프라를 구성할 때 시스템 이미지에 미리 추가할 수도 있습니다. 그러면 소스 이미지를 참조하는 모든 클라우드 템플릿이 동일한 초기화를 받습니다.

이미지 맵과 클라우드 템플릿 둘 다에 초기화 명령이 포함되어 있을 수 있습니다. 배포 시 명령이 병합되고, vRealize Automation Cloud Assembly가 통합된 명령을 실행합니다.

동일한 명령이 두 곳 모두에 있고 매개 변수만 다른 경우에는 이미지 맵 명령만 실행됩니다.

자세한 내용은 [vRealize Automation의 이미지 매핑에 대해 알아보기](#)의 내용을 참조하십시오.

## cloudConfig 명령 예

다음은 Linux 기반 MySQL 서버에 대한 기본 클라우드 템플릿 생성 클라우드 템플릿 코드에서 가져온 cloudConfig 섹션 예시입니다.

**참고** 명령이 제대로 해석되도록 하려면 아래와 같이 파이프 문자(cloudConfig: |)를 항상 포함해야 합니다.

```
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
```

cloud-init 스크립트가 예기치 않게 작동하는 경우 문제를 해결할 때 /var/log/cloud-init-output.log 에서 캡처된 콘솔 출력을 확인하십시오. cloud-init에 대한 자세한 내용은 [cloud-init 설명서](#)를 참조하십시오.

### 명령과 사용자 지정 규격이 혼합되지 않을 수 있음

vSphere에 배포하는 경우, 내장된 cloudConfig 명령과 사용자 지정 규격 초기화를 결합하려면 주의해서 진행해야 합니다. 공식적으로 호환되지 않으며 함께 사용하면 일관성이 없거나 원치 않는 결과가 발생할 수 있습니다.

명령과 사용자 지정 규격이 상호 작용하는 방식에 대한 예는 [vRealize Automation Cloud Assembly 클라우드 템플릿에서 vSphere 고정 IP 주소 할당](#)에서 참조하십시오.

### vRealize Automation Cloud Assembly 클라우드 템플릿에서 vSphere 고정 IP 주소 할당

vSphere에 배포하는 경우 고정 IP 주소를 할당할 수 있지만 cloudConfig 초기화 명령과 사용자 지정 규격 간에 충돌이 발생하지 않도록 주의해야 합니다.

#### 샘플 설계

다음 설계는 클라우드 템플릿 초기화 명령과 사용자 지정 규격 간의 충돌 없이 고정 IP 주소를 안전하게 적용합니다. 모두 assignment: static 네트워크 설정을 포함합니다.

## 설계

## 샘플 클라우드 템플릿 코드

cloud-init 코드가 없는  
Linux 시스템에 고정 IP  
주소 할당

```
resources:
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: public
      constraints:
        - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: linux-template
      networks:
        - name: '${wpnet.name}'
          assignment: static
          network: '${resource.wpnet.id}'
```

네트워크 할당 명령이  
포함되지 않은 cloud-  
init 코드를 사용하여  
Linux 시스템에 고정 IP  
주소를 할당합니다.

참고: vSphere 사용자  
지정 규격은

customizeGuestOs 속  
성을 true로 설정하거나  
customizeGuestOs 속  
성을 생략해도 적용됩니  
다.

## Ubuntu 샘플

```
resources:
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: public
      constraints:
        - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: ubuntu-template
      customizeGuestOs: true
      cloudConfig: |
        #cloud-config
        ssh_pwauth: yes
        chpasswd:
          list: |
            root:Pa$$w0rd
            expire: false
          write_files:
            - path: /tmpFile.txt
              content: |
                ${resource.wpnet.dns}
      runcmd:
        - hostnamectl set-hostname --pretty $
          {self.resourceName}
        - touch /etc/cloud/cloud-init.disabled
      networks:
        - name: '${wpnet.name}'
          assignment: static
          network: '${resource.wpnet.id}'
```

## CentOS 샘플

```
resources:
  wpnet:
    type: Cloud.Network
    properties:
```

## 설계

## 샘플 클라우드 템플릿 코드

```
    name: wpnet
    networkType: public
    constraints:
      - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: centos-template
      customizeGuestOs: true
      cloudConfig: |
        #cloud-config
        write_files:
          - path: /test.txt
            content: |
              deploying in power off.
              then rebooting.
    networks:
      - name: '${wpnet.name}'
        assignment: static
        network: '${resource.wpnet.id}'
```

**설계****샘플 클라우드 템플릿 코드**

네트워크 할당 명령이 포함된 cloud-init 코드를 사용하여 Linux 시스템에 고정 IP 주소를 할당합니다.

customizeGuestOs 속성은 false입니다.

**Ubuntu 샘플**

```
resources:
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: public
      constraints:
        - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: ubuntu-template
      customizeGuestOs: false
      cloudConfig: |
        #cloud-config
        write_files:
          - path: /etc/netplan/99-installer-
            config.yaml
            content: |
              network:
                version: 2
                renderer: networkd
                ethernet:
                  ens160:
                    addresses:
                      - $
                        {resource.DBTier.networks[0].address}/$
                        {resource.wpnet.prefixLength}
                    gateway4: $
                        {resource.wpnet.gateway}
                    nameservers:
                      search: $
                        {resource.wpnet.dnsSearchDomains}
                      addresses: ${resource.wpnet.dns}
        runcmd:
          - netplan apply
          - hostnamectl set-hostname --pretty $
            {self.resourceName}
          - touch /etc/cloud/cloud-init.disabled
      networks:
        - name: '${wpnet.name}'
          assignment: static
          network: '${resource.wpnet.id}'
```

**CentOS 샘플**

```
resources:
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: public
      constraints:
        - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: centos-template
```

## 설계

## 샘플 클라우드 템플릿 코드

```

    customizeGuestOs: false
    cloudConfig: |
      #cloud-config
      ssh_pwauth: yes
      chpasswd:
        list: |
          root:VMware1!
        expire: false
      runcmd:
        - nmcli con add type ethernet con-name
'custom ens192' ifname ens192 ip4 $
{self.networks[0].address}/$
{resource.wpnet.prefixLength} gw4 $
{resource.wpnet.gateway}
        - nmcli con mod 'custom ens192' ipv4.dns "$
{join(resource.wpnet.dns,' ')}"
        - nmcli con mod 'custom ens192' ipv4.dns-
search "${join(resource.wpnet.dnsSearchDomains,',')}
        - nmcli con down 'System ens192' ; nmcli
con up 'custom ens192'
        - nmcli con del 'System ens192'
        - hostnamectl set-hostname --static `dig -x
${self.networks[0].address} +short | cut -d "." -f 1`
        - hostnamectl set-hostname --pretty $
{self.resourceName}
        - touch /etc/cloud/cloud-init.disabled
    networks:
      - name: '${wpnet.name}'
        assignment: static
        network: '${resource.wpnet.id}'

```

배포가 참조된 이미지에 기반하는 경우 네트워크 할당 명령이 포함된 cloud-init 코드를 사용하여 Linux 시스템에 고정 IP 주소를 할당합니다.

customizeGuestOs 속성은 false입니다.

또한 클라우드 템플릿에 사용자 지정을 차단하는 ovfProperties 속성을 포함해서는 안 됩니다.

```

resources:
  wpnet:
    type: Cloud.Network
    properties:
      name: wpnet
      networkType: public
      constraints:
        - tag: sqa
  DBTier:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      imageRef: 'https://cloud-images.ubuntu.com/releases/focal/release/ubuntu-20.04-server-cloudimg-amd64.ova'
      customizeGuestOs: false
      cloudConfig: |
        #cloud-config
        ssh_pwauth: yes
        chpasswd:
          list: |
            root:Pa$$w0rd
            ubuntu:Pa$$w0rd
          expire: false
      write_files:
        - path: /etc/netplan/99-netcfg-vrac.yaml
          content: |
            network:
              version: 2
              renderer: networkd

```

## 설계

## 샘플 클라우드 템플릿 코드

```

ethernets:
  ens192:
    dhcp4: no
    dhcp6: no
    addresses:
      - $
    {resource.DBTier.networks[0].address}/$
    {resource.wpnet.prefixLength}
    gateway4: $
    {resource.wpnet.gateway}
    nameservers:
      search: $
    {resource.wpnet.dnsSearchDomains}
    addresses: ${resource.wpnet.dns}
  runcmd:
    - netplan apply
    - hostnamectl set-hostname --pretty $
    {self.resourceName}
    - touch /etc/cloud/cloud-init.disabled
  networks:
    - name: '${wpnet.name}'
      assignment: static
      network: '${resource.wpnet.id}'

```

## 작동하지 않거나 원치 않는 결과를 생성할 수 있는 설계

- cloud-init 코드에 네트워크 할당 명령이 포함되어 있지 않고 `customizeGuestOs` 속성은 `false`입니다.  
네트워크 설정을 구성하는 초기화 명령과 사용자 지정 규격이 둘 다 없습니다.
- cloud-init 코드에 네트워크 할당 명령이 포함되어 있지 않고 `ovfProperties` 속성이 설정되어 있습니다.  
초기화 명령이 없지만 `ovfProperties`가 사용자 지정 규격을 차단했습니다.
- cloud-init 코드에는 네트워크 할당 명령이 포함되어 있고 `customizeGuestOs` 속성이 누락되었거나 `true`로 설정되어 있습니다.  
사용자 지정 규격을 적용하면 초기화 명령과 충돌합니다.

## cloud-init 및 사용자 지정 규격에 대한 기타 해결 방법

vSphere에 배포하는 경우 cloud-init 및 사용자 지정 규격 충돌을 해결하도록 이미지를 사용자 지정할 수도 있습니다. 자세한 내용은 다음 외부 저장소를 참조하십시오.

- [vSphere 이미지 준비 스크립트](#)

## vRealize Automation Cloud Assembly 배포를 초기화 대기 상태로 만드는 방법

vRealize Automation Cloud Assembly 배포를 진행하기 전에 가상 시스템이 완전히 시작되어야 하는 경우가 있습니다.

예를 들어 아직 패키지를 설치 중인 시스템을 배포하고 웹 서버를 시작하면, 급한 사용자가 애플리케이션 사용이 가능해지기 전에 연결을 시도하는 상황이 발생할 수 있습니다.

이 기능을 사용할 때는 다음과 같은 고려 사항에 유의해야 합니다.

- 이 기능은 **cloud-init phone\_home** 모듈을 사용하며 Linux 시스템을 배포할 때 사용할 수 있습니다.
- **phone\_home**은 **Cloudbase-init** 제한 때문에 Windows에서 사용할 수 없습니다.
- **phone\_home**은 명시적 종속성과 같은 배포 순서에 영향을 줄 수 있지만 타이밍 및 처리 옵션에 대한 유연성이 더 높습니다.

vRealize Automation Cloud Assembly에서 리소스 배포 순서를 설정하는 방법 항목을 참조하십시오.

- **phone\_home**을 사용하려면 클라우드 템플릿에 **cloudConfig** 섹션이 필요합니다.
- 사용자의 창의력이 한 가지 요소입니다. 초기화 명령에는 작업 사이에 내장된 대기 시간을 포함할 수 있으며, **phone\_home**과 공동으로 사용할 수 있습니다.
- 시스템 템플릿에 **phone\_home** 모듈 설정이 이미 포함된 경우에는 클라우드 템플릿 기반 **phone\_home**이 작동하지 않습니다.
- 시스템에서 vRealize Automation Cloud Assembly로 다시 아웃바운드 통신 액세스가 가능해야 합니다.

vRealize Automation Cloud Assembly에서 **phone\_home**을 사용하여 시스템 초기화를 대기하려면 클라우드 템플릿에 **cloudConfigSettings** 섹션을 추가합니다.

```
cloudConfigSettings:
  phoneHomeShouldWait: true
  phoneHomeTimeoutSeconds: 600
  phoneHomeFailOnTimeout: true
```

속성	설명
phoneHomeShouldWait	초기화를 대기할지 여부이며, <b>true</b> 또는 <b>false</b> 입니다.
phoneHomeTimeoutSeconds	초기화가 아직 실행 중이어도 배포를 계속할지 여부를 결정하는 시기입니다. 기본값은 10분입니다.
phoneHomeFailOnTimeout	시간 초과 후 배포를 계속할지 여부이며, <b>true</b> 또는 <b>false</b> 입니다. 계속 진행하더라도 별도의 이유로 인해 배포가 계속 실패할 수 있습니다.

## Windows 게스트 사용자 지정을 수행하는 방법

배포 시 vRealize Automation Cloud Assembly에서 Windows 시스템을 자동으로 초기화하도록 하려면 Cloudbase-Init를 지원하는 이미지를 준비한 다음 적절한 명령이 포함된 클라우드 템플릿을 준비합니다.

이미지 생성 프로세스는 클라우드 벤더에 따라 다릅니다. 여기에 표시된 예제는 vSphere에 대한 예제입니다.

## vSphere에 대한 초기화 가능 Windows 이미지를 생성하는 방법

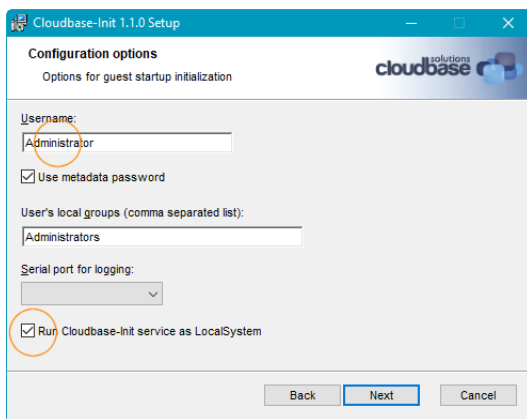
vRealize Automation Cloud Assembly에서 vSphere에 배포된 Windows 시스템을 초기화하려면 Cloudbase-Init가 설치 및 구성된 템플릿을 기반으로 해야 합니다.

- 1 vSphere를 사용하여 Windows 가상 시스템을 만들고 전원을 켭니다.
- 2 가상 시스템에서 Windows에 로그인합니다.
- 3 Cloudbase-Init를 다운로드합니다.

<https://cloudbase.it/cloudbase-init/#download>

- 4 Cloudbase-Init 설치 .msi 파일을 시작합니다.

설치 중에 **관리자**를 사용자 이름으로 입력하고 LocalSystem을 실행하기 위한 옵션을 선택합니다.



기타 설치 선택 항목은 기본값으로 유지될 수 있습니다.

- 5 설치를 실행하되 설치 마법사의 마지막 완료 페이지를 닫지 마십시오.

---

**중요** 설치 마법사의 마지막 페이지를 닫지 마십시오.

---

- 6 설치 마법사의 [완료] 페이지가 계속 열어 두고 Windows를 사용하여 Cloudbase-Init 설치 경로로 이동한 후 텍스트 편집기에서 다음 파일을 엽니다.

```
conf\cloudbase-init-unattend.conf
```

- 7 표시된 대로 metadata\_services를 OvService로 설정합니다.

```
metadata_services=cloudbaseinit.metadata.services.ovfservice.OvfService
```

- 8 cloudbase-init-unattend.conf를 저장하고 닫습니다.

- 9 동일한 폴더에서 텍스트 편집기를 사용하여 다음 파일을 엽니다.

```
conf\cloudbase-init.conf
```

- 10 표시된 대로 `first_logon_behaviour`, `metadata_services` 및 `plugins`를 설정합니다.

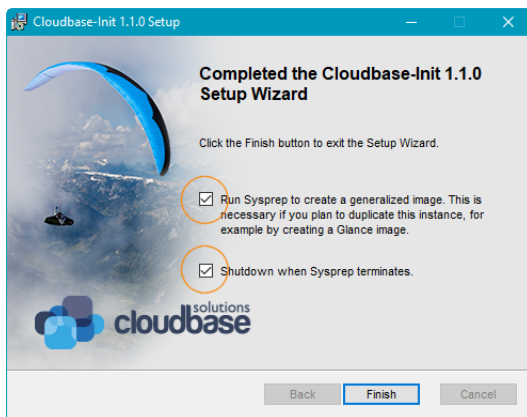
```
first_logon_behaviour=always
. . .
metadata_services=cloudbaseinit.metadata.services.ovfservice.OvfService
. . .
plugins=cloudbaseinit.plugins.windows.createuser.CreateUserPlugin,cloudbaseinit.plugins.win
dows.setuserpassword.SetUserPasswordPlugin,cloudbaseinit.plugins.common.sshpublickeys.SetUs
erSSHPublicKeysPlugin,cloudbaseinit.plugins.common.userdata.UserDataPlugin
. . .
```

- 11 `cloudbase-init.conf`를 저장하고 닫습니다.
- 12 설치 마법사의 [완료] 페이지에서 Sysprep을 실행하고 Sysprep 후 종료하는 옵션을 선택하고 **마침**을 클릭합니다.

**참고** Sysprep를 실행하면 이미지 배포가 작동하지 않는 경우가 있습니다.

배포할 때 vRealize Automation Cloud Assembly는 동적으로 생성된 사용자 지정 규격을 적용하여 네트워크 인터페이스의 연결을 끊습니다. 이미지에서 오류 중인 Sysprep 상태로 인해 사용자 지정 규격이 실패하고 배포가 연결이 끊긴 상태로 유지될 수 있습니다.

환경에서 이 문제가 발생하는 것으로 의심된다면 이미지를 생성할 때 Sysprep 옵션을 비활성화한 상태로 두십시오.



- 13 가상 시스템이 종료되면 vSphere를 사용하여 템플릿으로 되돌립니다.

#### 추가 세부 정보

다음 표에서는 설정 중에 생성된 구성 항목을 확장합니다.

구성 설정	용도
사용자 이름, CreateUserPlugin 및 SetUserPasswordPlugin	Sysprep 후에는 처음 부팅에서 CreateUserPlugin을 사용하여 빈 암호로 사용자 이름 관리자 계정을 생성합니다. SetUserPasswordPlugin을 사용하면 Cloudbase-Init를 통해 빈 암호를 클라우드 템플릿에 포함될 원격 액세스 암호로 변경할 수 있습니다.
처음 로그인 동작	이 설정은 사용자에게 처음 로그인 시 암호를 변경하라는 메시지를 표시합니다.
메타데이터 서비스	OvfService만 나열하여 Cloudbase-Init는 vCenter에서 지원되지 않는 다른 메타데이터 서비스를 찾으려고 시도하지 않습니다. 따라서 로그 파일이 더 간결해집니다. 그러지 않으면 다른 서비스를 찾을 수 없다는 항목으로 로그가 채워질 수 있기 때문입니다.
플러그인	OvfService에서 지원하는 기능이 포함된 플러그인만 나열하면 로그가 한층 더 간결해집니다. Cloudbase-Init는 지정된 순서대로 플러그인을 실행합니다.
LocalSystem으로 실행	이 설정은 전용 관리자 계정에서 실행하기 위해 Cloudbase-Init가 필요할 수 있는 고급 초기화 명령을 지원합니다.

## 클라우드 템플릿에 Cloudbase-Init 명령을 포함하는 방법

Windows 시스템을 초기화하려면 사용자가 원하는 명령을 초기화 가능 Windows 이미지가 실행하도록 vRealize Automation Cloud Assembly에서 인프라 및 클라우드 템플릿을 생성합니다.

여기에 나와 있는 예는 vSphere를 기반으로 하지만 다른 클라우드 벤더는 유사해야 합니다.

### 사전 요구 사항

- 인프라를 생성합니다. vRealize Automation Cloud Assembly에서 vSphere 클라우드 계정 및 연결된 클라우드 영역을 추가합니다.
- 버전 및 이미지 매핑을 추가하고 네트워크 및 스토리지 프로파일을 추가합니다.  
인프라에서 이미지 매핑은 Cloudbase-Init를 지원하기 위해 생성한 Windows 템플릿을 가리켜야 합니다. [vSphere에 대한 초기화 가능 Windows 이미지를 생성하는 방법](#)의 내용을 참조하십시오.  
템플릿이 나열되지 않으면 클라우드 계정으로 이동하고 이미지를 동기화합니다. 그렇지 않으면 자동 동기화는 24시간마다 실행됩니다.
- 프로젝트를 추가하고, 사용자를 추가하고, 사용자가 클라우드 영역에 프로비저닝할 수 있는지 확인합니다.

인프라 및 프로젝트 생성에 대한 자세한 내용은 [자습서: vRealize Automation Cloud Assembly에서 다중 클라우드 인프라와 배포 설정 및 테스트](#)의 예를 참조하십시오.

### 절차

- 1 vRealize Automation Cloud Assembly에서 **설계** 탭으로 이동하여 새 클라우드 템플릿을 생성합니다.
- 2 원하는 Cloudbase-init 명령을 사용하여 cloudConfig 섹션을 추가합니다.

다음 명령 예는 Windows C: 드라이브에 새 파일을 생성하고 호스트 이름을 설정합니다.

```
resources:
  Cloud_Machine_1:
    type: Cloud.Machine
    properties:
      image: cloudbase-init-win-2016
      flavor: small
      remoteAccess:
        authentication: usernamePassword
        username: Administrator
        password: Password1234@$
      cloudConfig: |
        #cloud-config
        write_files:
          content: Cloudbase-Init test
          path: C:\test.txt
        set_hostname: testname
```

자세한 내용은 [Cloudbase-init 설명서](#)를 참조하십시오.

- 3 Windows에 처음 로그인하기 위한 시스템을 구성하도록 remoteAccess 속성을 추가합니다.  
템플릿을 생성할 때 언급한 대로 메타데이터 서비스는 로그인 자격 증명을 선택하여 CreateUserPlugin 및 SetUserPasswordPlugin에 노출합니다. 암호는 Windows 암호 요구 사항을 충족해야 합니다.
- 4 vRealize Automation Cloud Assembly에서 클라우드 템플릿을 테스트하고 배포합니다.
- 5 배포 후 Windows RDP 및 템플릿의 자격 증명을 사용하여 새 Windows 시스템에 로그인하고 사용자 지정을 확인합니다.

위 예에서는 C:\test.txt 파일을 찾고 시스템 속성에서 호스트 이름을 확인합니다.

## vRealize Automation Cloud Assembly 클라우드 템플릿에서 사용할 사용자 지정 리소스 유형을 생성하는 방법

vRealize Automation Cloud Assembly에서 클라우드 템플릿을 생성할 때 리소스 유형 팔레트에는 지원되는 클라우드 계정 및 통합 끝점에 대한 리소스 유형이 포함됩니다. 리소스 유형의 확장된 목록을 기반으로 클라우드 템플릿을 생성하려는 사용 사례가 있을 수 있습니다. 사용자 지정 리소스를 생성하고, 설계 캔버스에 추가하고, 설계 및 배포 요구 사항을 지원하는 클라우드 템플릿을 생성할 수 있습니다.

### vRealize Orchestrator를 사용하여 사용자 지정 리소스 생성

각 사용자 지정 리소스는 vRealize Orchestrator SDK 인벤토리 유형을 기반으로 하며, 원하는 SDK 유형의 인스턴스인 출력이 있는 vRealize Orchestrator 워크플로에 의해 생성됩니다. Properties, Date, string 및 number 같은 기본 유형은 사용자 지정 리소스를 생성하는 데 지원되지 않습니다.

**참고** SDK 개체 유형은 플러그인 이름과 유형 이름을 구분하는 데 사용되는 콜론(":")을 사용하여 다른 속성 유형과 구분할 수 있습니다. 예를 들어 AD:UserGroup은 Active Directory 사용자 그룹을 관리하는 데 사용되는 SDK 개체 유형입니다.

vRealize Orchestrator에 있는 기본 제공 워크플로를 사용하거나 직접 생성할 수 있습니다. vRealize Orchestrator를 사용하여 모든 것을 서비스로 제공/XaaS 워크플로를 생성하는 것은 배포 시 Active Directory 사용자를 시스템에 추가하는 클라우드 템플릿을 생성하거나 사용자 지정 F5 로드 밸런서를 배포에 추가할 수 있음을 의미합니다.

## 사용자 지정 리소스 이름 및 리소스 유형

사용자 지정 리소스 이름은 클라우드 템플릿 리소스 유형 팔레트 내에서 사용자 지정 리소스를 식별합니다.

사용자 지정 리소스의 리소스 유형은 **Custom**.으로 시작해야 하며 각 리소스 유형은 고유해야 합니다. 예를 들어 Active Directory 사용자를 추가하는 사용자 지정 리소스에 대한 리소스 유형으로 Custom.ADUser를 설정할 수 있습니다. **Custom**.의 포함이 텍스트 상자에서 검증되지는 않지만 해당 문자열을 제거하면 문자열이 자동으로 추가됩니다.

## 외부 유형

외부 유형 속성은 사용자 지정 리소스의 유형을 정의합니다. vRealize Automation Cloud Assembly의 사용자 지정 리소스에서 생성 워크플로를 선택하면 외부 유형 드롭다운이 아래에 표시됩니다. 드롭다운에는 vRealize Orchestrator 워크플로의 출력 매개 변수에서 선택되는 외부 유형 속성이 포함됩니다. 드롭다운에 포함된 선택된 워크플로 출력 속성은 VC:VirtualMachine 또는 AD:UserGroup과 같은 비어레이 SDK 개체 유형이어야 합니다.

**참고** 동적 유형 플러그인을 사용하는 사용자 지정 워크플로를 생성하는 경우

DynamicTypesManager.getObject() 메서드를 사용하여 해당 변수를 생성했는지 확인합니다.

사용자 지정 리소스를 정의할 때 선택한 외부 유형의 가용성 범위도 정의합니다. 선택한 외부 유형은 다음과 같을 수 있습니다.

- 프로젝트 간에 공유됩니다.
- 선택한 프로젝트에만 사용할 수 있습니다.

정의된 범위당 하나의 외부 유형만 있을 수 있습니다. 예를 들어, 프로젝트에서 VC:VirtualMachine을 외부 유형으로 사용하는 사용자 지정 리소스를 생성하는 경우 동일한 외부 유형을 사용하는 동일한 프로젝트에 대해 다른 사용자 지정 리소스를 생성할 수 없습니다. 또한 동일한 외부 유형을 사용하는 2개의 공유 사용자 지정 리소스를 생성할 수 없습니다.

## 워크플로 입력/출력 검증

생성, 삭제 및 업데이트 워크플로를 사용자 지정 리소스에 수명 주기 작업으로 추가할 경우 vRealize Automation Cloud Assembly는 선택한 워크플로에 올바른 입력 및 출력 속성 정의가 있는지 검증합니다.

- 생성 워크플로에는 SSH:Host 또는 SQL:Database 같은 SDK 개체 유형인 출력 매개 변수가 있어야 합니다. 선택한 워크플로가 검증을 통과하지 못하는 경우에는 업데이트 또는 삭제 워크플로를 추가하거나 사용자 지정 리소스에 대한 변경 내용을 저장할 수 없습니다.
- 삭제 워크플로에는 사용자 지정 리소스의 외부 유형과 일치하는 SDK 개체 유형인 입력 매개 변수가 있어야 합니다.

- 업데이트 워크플로에는 사용자 지정 리소스의 외부 유형과 일치하는 SDK 개체 유형인 입력 및 출력 매개 변수가 모두 있어야 합니다.

## 사용자 지정 리소스 속성 스키마

사용자 지정 리소스에 vRealize Orchestrator 워크플로를 추가하면 해당 입력 및 출력 매개 변수가 속성으로 추가됩니다. **속성** 탭을 선택하여 사용자 지정 리소스 속성 스키마를 볼 수 있습니다. 스키마에는 이름, 데이터 유형, 속성 유형 및 사용 가능한 경우 지정된 속성에 대한 설명이 포함되어 있습니다. 또한 스키마는 특정 속성이 필수인지 선택 사항인지를 정의합니다.

## 2일차 작업 사용자 지정 요청 양식

다른 유형의 리소스 속성을 추가하고 수정하여 사용자 지정 리소스에 포함된 2일차 작업의 요청 양식을 간소화할 수 있습니다.

예를 들어, 요청 양식의 입력 매개 변수 값을 외부 소스에 바인딩할 수 있습니다(예: 배포 이름이나 프로젝트 이름을 검색하는 vRealize Orchestrator 작업). 또한 특정 입력 매개 변수의 값을 동일한 요청 양식에 포함된 다른 두 텍스트 상자의 계산된 값에 바인딩할 수도 있습니다.

**참고** 이 기능은 사용자 지정 리소스 및 리소스 작업 모두에서 사용할 수 있습니다. 사용자 지정 리소스 또는 리소스 작업 편집기의 **요청 매개 변수** 페이지에 있는 **값** 탭에서 요청 양식의 입력 속성 값을 사용자 지정할 수 있습니다.

## vRealize Automation Cloud Assembly에서 Active Directory에 사용자를 추가하는 클라우드 템플릿을 생성하는 방법

클라우드 템플릿을 생성할 때 사용하는 vRealize Automation Cloud Assembly 클라우드 템플릿 리소스 외에도 자체 사용자 지정 리소스를 생성할 수 있습니다.

사용자 지정 리소스는 정의된 기본 리소스 작업 워크플로를 사용하여 vRealize Automation를 통해 관리하는 vRealize Orchestrator 개체입니다. 생성 또는 삭제 작업이 트리거되면 클라우드 템플릿 서비스가 적절한 vRealize Orchestrator 워크플로를 자동으로 호출합니다. 2일차 작업으로 사용될 수 있는 vRealize Orchestrator 워크플로를 선택하여 리소스 유형의 기능을 확장할 수도 있습니다.

이 사용 사례는 vRealize Orchestrator 라이브러리에 제공된 기본 제공 워크플로를 사용합니다. 여기에는 프로세스를 수행하는 방법을 보여 주는 규범적 값 또는 문자열이 포함됩니다. 이러한 값 또는 문자열은 환경에 맞게 수정할 수 있습니다.

참조용으로 이 사용 사례는 이름이 **DevOpsTesting**인 프로젝트를 사용합니다. 이 샘플 프로젝트를 사용자 환경의 프로젝트로 교체할 수 있습니다.

### 사전 요구 사항

- vRealize Orchestrator 통합을 구성했는지 확인합니다. [Cloud Assembly에서 vRealize Orchestrator 통합 구성](#)의 내용을 참조하십시오.
- 생성, 업데이트, 삭제 및 2일차 작업에 사용 중인 워크플로가 vRealize Orchestrator에 있고 여기에서 성공적으로 실행되는지 확인합니다.

- vRealize Orchestrator에서 워크플로에 사용되는 리소스 유형을 찾습니다. 이 사용자 지정 리소스에 포함된 워크플로는 모두 동일한 리소스 유형을 사용해야 합니다. 이 사용 사례에서 리소스 유형은 AD:User입니다. 리소스 유형 검증에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 클라우드 템플릿에서 사용할 사용자 지정 리소스 유형을 생성하는 방법](#) 항목을 참조하십시오.
- vRealize Orchestrator 통합의 기본 제공 Active Directory 워크플로를 사용하여 Active Directory 서버를 구성합니다.
- 시스템 클라우드 템플릿을 구성 및 배포하는 방법을 알고 있는지 확인합니다.

## 절차

- 1 그룹에 사용자를 추가하기 위한 Active Directory 사용자 지정 리소스를 생성합니다.

이 단계에서는 사용자 지정 리소스를 클라우드 템플릿 설계 캔버스에 리소스 유형으로 추가합니다.

- a vRealize Automation Cloud Assembly에서 **설계 > 사용자 지정 리소스**를 선택하고 **새 사용자 지정 리소스**를 클릭합니다.
- b 다음 값을 제공합니다.

워크플로 이름 이외의 값은 모두 샘플 값입니다.

설정	샘플 값
이름	<b>AD user</b> 클라우드 템플릿 리소스 유형 팔레트에 표시되는 이름입니다.
리소스 유형	<b>Custom.ADUser</b> 리소스 유형은 <b>Custom.</b> 으로 시작해야 하며 각 리소스 유형은 고유해야 합니다. <b>Custom.</b> 의 포함이 텍스트 상자에서 검증되지는 않지만 해당 문자열을 제거하면 문자열이 자동으로 추가됩니다. 이 리소스 유형은 클라우드 템플릿에서 사용할 수 있도록 리소스 유형 팔레트에 추가됩니다.

- c 클라우드 템플릿 리소스 유형 목록에서 이 리소스 유형을 사용하도록 설정하려면 **활성화** 옵션이 설정되었는지 확인합니다.
- d 모든 프로젝트에서 리소스 유형을 사용할 수 있도록 하는 **범위** 설정을 선택합니다.

- e 리소스를 정의하는 워크플로와 2일차 작업을 선택합니다.

**참고** 선택한 2일차 워크플로에는 외부 유형과 동일한 유형의 입력 매개 변수가 있어야 합니다. 외부 유형 입력은 자동으로 사용자 지정 리소스에 바인딩되므로 사용자가 요청하는 2일차 사용자 지정 양식에 표시되지 않습니다.

설정	샘플 값
수명 주기 작업 - 생성	<p><b>조직 구성 단위에서 암호가 있는 사용자 생성</b> 워크플로를 선택합니다.</p> <p>여러 개의 vRealize Orchestrator 통합이 있는 경우 이러한 사용자 지정 리소스를 실행하는 데 사용하는 통합 인스턴스에서 워크플로를 선택합니다.</p> <p>워크플로를 선택하면 외부 유형 드롭다운 메뉴를 사용할 수 있게 되고 자동으로 AD:User로 설정됩니다.</p> <p><b>참고</b> 외부 소스 유형은 공유된 경우 한 번만 사용할 수 있고 프로젝트별로 한 번만 사용할 수 있습니다. 이 사용 사례에서는 모든 프로젝트에 대해 동일한 사용자 지정 리소스를 제공합니다. 이는 모든 프로젝트에서 다른 리소스 유형에 AD:User를 사용할 수 없음을 의미합니다. AD:User 유형이 필요한 다른 워크플로가 있는 경우 각 프로젝트에 대해 개별 사용자 지정 리소스를 생성해야 합니다.</p>
수명 주기 작업 - 삭제	<b>사용자 삭제</b> 워크플로를 선택합니다.
추가 작업	<p><b>사용자 암호 변경</b> 워크플로를 선택합니다.</p> <p>사용자가 작업을 요청할 때 응답하는 작업 요청 양식을 수정하려면 <b>요청 매개 변수</b> 열에서 해당 아이콘을 클릭합니다.</p> <p><b>참고</b> 추가 작업 워크플로의 경우 워크플로에 외부 유형과 동일한 유형의 입력 매개 변수가 있는지 확인합니다.</p>

이 예에는 업데이트 워크플로가 적절하게 적용된 부분이 없습니다. 프로비저닝된 사용자 지정 리소스를 변경하는 업데이트 워크플로의 일반적인 예로는 배포 축소 또는 확장을 들 수 있습니다.

- f **속성** 탭의 스키마 키 및 유형 값을 검토하여 클라우드 템플릿에서 입력을 구성할 수 있는 워크플로 입력을 이해합니다.

스키마에는 워크플로에 정의된 필수 및 선택적 입력 값이 나열됩니다. 필수 입력 값은 클라우드 템플릿 YAML에 포함되어 있습니다.

사용자 생성 워크플로에서 accountName, displayName 및 ouContainer는 필수 입력 값입니다. 다른 스키마 속성은 필수가 아닙니다. 또한 스키마를 사용하여 다른 필드 값, 워크플로 또는 작업에 대한 바인딩을 생성하려는 위치를 결정할 수 있습니다. 바인딩은 이 사용 사례에 포함되지 않습니다.

- g 사용자 지정 리소스 생성을 완료하려면 **생성**을 클릭합니다.

- 2 배포할 때 시스템에 사용자를 추가하는 클라우드 템플릿을 생성합니다.
  - a 설계 > 클라우드 템플릿을 선택하고 새로 만들기 > 빈 캔버스를 클릭합니다.
  - b 클라우드 템플릿 이름을 **Machine with an AD user**로 지정합니다.
  - c DevOpsTesting 프로젝트를 선택하고 생성을 클릭합니다.
  - d vSphere 시스템을 추가하고 구성합니다.
  - e 클라우드 템플릿 설계 페이지의 왼쪽에 있는 사용자 지정 리소스 목록에서 **AD 사용자** 리소스 유형을 캔버스로 끌어 놓습니다.

---

**참고** 아래로 스크롤하여 왼쪽 창에서 선택하거나 **리소스 유형 검색** 텍스트 상자에서 검색하여 사용자 지정 리소스를 선택할 수 있습니다. 사용자 지정 리소스가 표시되지 않으면 **리소스 유형 검색** 텍스트 상자 옆에 있는 새로 고침 버튼을 클릭합니다.

---

- f 오른쪽에서 **YAML** 코드를 편집하여 필수 입력 값과 암호를 추가합니다.

사용자가 추가하는 사용자의 이름을 제공할 수 있도록 코드에 `inputs` 섹션을 추가합니다. 다음 예에서 이러한 값 중 일부는 샘플 데이터입니다. 값이 다를 수 있습니다.

```
inputs:
  accountName:
    type: string
    title: Account name
    encrypted: true
  displayName:
    type: string
    title: Display name
  password:
    type: string
    title: Password
    encrypted: true
  confirmPassword:
    type: string
    title: Password
    encrypted: true
  ouContainer:
    type: object
    title: AD OU container
    $data: 'vro/data/inventory/AD:OrganizationalUnit'
  properties:
    id:
      type: string
    type:
      type: string
```

- g `resources` 섹션에 사용자의 선택을 요청하는 `${input.input-name}` 코드를 추가합니다.

```
resources:
  Custom_ADUser_1:
    type: Custom.ADUser
    properties:
      accountName: '${input.accountName}'
      displayName: '${input.displayName}'
      ouContainer: '${input.ouContainer}'
      password: '${input.password}'
      confirmPassword: '${input.confirmPassword}'
```

- 3 클라우드 템플릿을 배포합니다.

- a 클라우드 템플릿 디자이너 페이지에서 **배포**를 클릭합니다.
- b **배포 이름 AD User Scott**을 입력합니다.
- c **클라우드 템플릿 버전**을 선택하고 **다음**을 클릭합니다.

- d 배포 입력을 완료합니다.
  - e 배포를 클릭합니다.
- 4 프로비저닝 프로세스를 모니터링하여 사용자가 **Active Directory**에 추가되었는지 확인합니다.
- a 배포를 클릭하고 **AD User Scott** 배포를 찾습니다.
  - b 요청의 상태를 모니터링하고 배포가 성공했는지 확인합니다.
  - c 암호 변경 작업을 사용할 수 있고 작동하는지 확인합니다.

#### 다음에 수행할 작업

테스트된 클라우드 템플릿이 작동하면 다른 클라우드 템플릿과 함께 **AD user** 사용자 지정 리소스를 사용하기 시작할 수 있습니다.

### Cloud Assembly에서 SSH를 포함하는 클라우드 템플릿을 생성하는 방법

vRealize Orchestrator 워크플로를 사용하여 클라우드 템플릿을 구축하는 데 사용할 수 있는 사용자 지정 리소스를 생성할 수 있습니다. 이 사용 사례에서는 SSH 호스트를 추가하는 사용자 지정 리소스를 추가합니다. 그런 다음 리소스를 클라우드 템플릿에 포함할 수 있습니다. 또한 이 절차에서는 사용자가 개별 2일차 작업을 수행하는 대신 배포 후 SSH 구성을 변경하도록 업데이트 워크플로를 추가합니다.

사용자 지정 리소스는 정의된 기본 리소스 작업 워크플로를 사용하여 vRealize Automation를 통해 관리하는 vRealize Orchestrator 개체입니다. 생성 또는 삭제 작업이 트리거되면 클라우드 템플릿 서비스가 적절한 vRealize Orchestrator 워크플로를 자동으로 호출합니다. 2일차 작업으로 사용될 수 있는 vRealize Orchestrator 워크플로를 선택하여 리소스 유형의 기능을 확장할 수도 있습니다.

이 사용 사례는 vRealize Orchestrator 라이브러리에 제공된 기본 제공 워크플로를 사용합니다. 여기에는 프로세스를 수행하는 방법을 보여 주는 규범적 값 또는 문자열이 포함됩니다. 이러한 값 또는 문자열은 환경에 맞게 수정할 수 있습니다.

참조용으로 이 사용 사례는 이름이 **DevOpsTesting**인 프로젝트를 사용합니다. 이 프로젝트를 이미 보유한 프로젝트로 바꿀 수 있습니다.

#### 사전 요구 사항

- vRealize Orchestrator 통합을 구성했는지 확인합니다. [Cloud Assembly에서 vRealize Orchestrator 통합 구성](#)의 내용을 참조하십시오.
- 생성, 업데이트, 삭제 및 2일차 작업에 사용 중인 워크플로가 vRealize Orchestrator에 있고 여기에서 성공적으로 실행되는지 확인합니다.
- vRealize Orchestrator에서 워크플로에 사용되는 리소스 유형을 찾습니다. 이 사용자 지정 리소스에 포함된 워크플로는 모두 동일한 리소스 유형을 사용해야 합니다. 이 사용 사례에서 리소스 유형은 **SSH:Host**입니다. 리소스 유형 검증에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 클라우드 템플릿에서 사용할 사용자 지정 리소스 유형을 생성하는 방법](#) 항목을 참조하십시오.
- 시스템 클라우드 템플릿을 구성 및 배포하는 방법을 알고 있는지 확인합니다.

## 절차

- 1 SSH를 클라우드 템플릿에 추가하기 위한 SSH 호스트 사용자 지정 리소스를 생성합니다.

이 단계에서는 사용자 지정 리소스를 클라우드 템플릿 설계 캔버스에 리소스 유형으로 추가합니다.

- a vRealize Automation Cloud Assembly에서 **설계 > 사용자 지정 리소스**를 선택하고 **새 사용자 지정 리소스**를 클릭합니다.
- b 다음 값을 제공합니다.

워크플로 이름 이외의 값은 모두 샘플 값입니다.

표 6-2.

설정	샘플 값
이름	<b>SSH Host - DevOpsTesting Project</b> 클라우드 템플릿 리소스 유형 팔레트에 표시되는 이름입니다.
리소스 유형	<b>Custom.SSHHost</b> 리소스 유형은 <b>Custom.</b> 으로 시작해야 하며 각 리소스 유형은 고유해야 합니다. <b>Custom.</b> 의 포함이 텍스트 상자에서 검증되지는 않지만 해당 문자열을 제거하면 문자열이 자동으로 추가됩니다. 이 리소스 유형은 클라우드 템플릿에서 사용할 수 있도록 설계 캔버스에 추가됩니다.

- c 클라우드 템플릿 리소스 유형 목록에서 이 리소스 유형을 사용하도록 설정하려면 **활성화** 옵션이 설정되었는지 확인합니다.
- d 리소스 유형을 **DevOpsTesting** 프로젝트에서 사용할 수 있도록 하는 **범위** 설정을 선택합니다.
- e 리소스를 정의하는 워크플로를 선택합니다.

설정	설정
수명 주기 작업 - 생성	<b>SSH 호스트 추가</b> 워크플로를 선택합니다. 여러 개의 vRealize Orchestrator 통합이 있는 경우 이러한 사용자 지정 리소스를 실행하는 데 사용하는 통합 인스턴스에서 워크플로를 선택합니다. 워크플로를 선택하면 외부 유형 드롭다운 메뉴를 사용할 수 있게 되고 자동으로 SSH:Host로 설정됩니다. 외부 소스 유형은 공유된 경우 한 번만 사용할 수 있고 프로젝트별로 한 번만 사용할 수 있습니다. 이 사용 사례에서는 <b>DevOpsTesting</b> 프로젝트에만 사용자 지정 리소스를 제공합니다. SSH:Host 유형이 필요한 다른 워크플로가 있는 경우 각 프로젝트에 대해 개별 사용자 지정 리소스를 생성해야 합니다.
수명 주기 작업 - 업데이트	<b>SSH 호스트 업데이트</b> 워크플로를 선택합니다.
수명 주기 작업 - 삭제	<b>SSH 호스트 제거</b> 워크플로를 선택합니다.

- f **속성** 탭의 스키마 키 및 유형 값을 검토하여 클라우드 템플릿에서 입력을 구성할 수 있도록 워크플로 입력을 이해합니다.

스키마에는 워크플로에 정의된 필수 및 선택적 입력 값이 나열됩니다. 필수 입력 값은 클라우드 템플릿 YAML에 포함되어 있습니다.

**SSH 호스트 추가** 워크플로에서 hostname, port 및 username은 필수 입력 값입니다. 다른 스키마 속성은 필수가 아닙니다. 또한 스키마를 사용하여 다른 필드 값, 워크플로 또는 작업에 대한 바인딩을 생성하려는 위치를 결정할 수 있습니다. 바인딩은 이 사용 사례에 포함되지 않습니다.

- g 사용자 지정 리소스 생성을 완료하려면 **생성**을 클릭합니다.

## 2 배포할 때 SSH 호스트를 추가하는 클라우드 템플릿을 생성합니다.

- a **설계 > 클라우드 템플릿**을 선택하고 **새로 만들기 > 빈 캔버스**를 클릭합니다.
- b 클라우드 템플릿 이름을 **Machine with SSH Host**로 지정합니다.
- c **DevOpsTesting** 프로젝트를 선택하고 **생성**을 클릭합니다.
- d vSphere 시스템을 추가하고 구성합니다.
- e 클라우드 템플릿 [설계] 페이지의 왼쪽에 있는 사용자 지정 리소스 목록에서 **SSH Host - DevOpsTesting Project** 리소스 유형을 캔버스로 끕니다.

---

**참고** 아래로 스크롤하여 왼쪽 창에서 선택하거나 **리소스 유형 검색** 텍스트 상자에서 검색하여 사용자 지정 리소스를 선택할 수 있습니다. 사용자 지정 리소스가 표시되지 않으면 **리소스 유형 검색** 텍스트 상자 옆에 있는 새로 고침 버튼을 클릭합니다.

---

이 리소스 유형은 프로젝트에 대해 이미 구성되었기 때문에 사용할 수 있습니다. 다른 프로젝트에 대한 클라우드 템플릿을 생성하는 경우에는 이 리소스 유형을 볼 수 없습니다.

- f 오른쪽에서 **YAML** 코드를 편집하여 필수 입력 을 추가합니다.

배포 시 사용자가 사용자 이름 및 호스트 이름을 제공할 수 있도록 코드에 `inputs` 섹션을 추가합니다. 이 예에서 포트 기본값은 **22**입니다. 다음 예에서 이러한 값 중 일부는 샘플 데이터입니다. 값이 다를 수 있습니다.

```
inputs:
  hostname:
    type: string
    title: The hostname of the SSH Host
  username:
    type: string
    title: Username
```

- g `resources` 섹션에 사용자의 선택을 요청하는 `${input.input-name}` 코드를 추가합니다.

```
resources:
  Custom_SSHTest_1:
    type: Custom.SSHTest
    properties:
      port: 22
      hostname: '${input.hostname}'
      username: '${input.username}'
```

- 3 클라우드 템플릿을 배포합니다.

- a [클라우드 템플릿 디자이너] 페이지에서 **배포**를 클릭합니다.
- b **배포 이름 SSH Host Test**를 입력합니다.
- c **클라우드 템플릿 버전**을 선택하고 **다음**을 클릭합니다.
- d 배포 입력을 완료합니다.
- e **배포**를 클릭합니다.

- 4 프로비저닝 프로세스를 모니터링하여 **SSH 호스트**가 배포에 포함되어 있는지 확인합니다.

- a **배포**를 클릭하고 **SSH Host Test** 배포를 찾습니다.
- b 요청의 상태를 모니터링하고 배포가 성공했는지 확인합니다.

#### 다음에 수행할 작업

테스트된 클라우드 템플릿이 작동하면 다른 클라우드 템플릿과 함께 **SSH Host** 사용자 지정 리소스 사용을 시작할 수 있습니다.

## 2일차 변경을 준비하도록 vRealize Automation Cloud Assembly에서 설계하는 방법

vRealize Automation Cloud Assembly 리소스 유형과 이미 연결된 2일차 작업 외에도 사용자가 수행해야 하는 사용자 지정 업데이트를 미리 준비할 수 있는 설계 옵션이 있습니다.

**경고** 배포를 변경하려면 해당 클라우드 템플릿을 편집하고 다시 적용하거나 2일차 작업을 사용하면 됩니다. 하지만 대부분의 경우 두 가지 방법을 혼합하지 않도록 해야 합니다.

전원 켜기/끄기와 같은 수명 주기 2일차 변경 내용은 일반적으로 안전하지만 디스크를 추가하는 경우와 같이 다른 사항은 주의가 필요합니다.

예를 들어 2일차 작업으로 디스크를 추가한 다음 클라우드 템플릿을 다시 적용하는 혼합된 방식을 취하면 클라우드 템플릿이 2일차 변경 내용을 덮어쓸 수 있으며 이로 인해 디스크가 제거되고 데이터 손실이 발생할 수 있습니다.

2일차 준비에는 클라우드 템플릿 코드를 직접 사용하거나 vRealize Automation Cloud Assembly 설계 인터페이스를 사용할 수 있습니다.

- 클라우드 템플릿 코드의 입력을 사용하여 배포 또는 배포된 리소스를 업데이트할 때 인터페이스에서 새 값을 묻는 메시지를 표시하도록 할 수 있습니다.
- vRealize Automation Cloud Assembly를 사용하여 vRealize Orchestrator 워크플로 또는 작업을 기반으로 사용자 지정 작업을 설계할 수 있습니다. 사용자 지정 작업을 실행하면 vRealize Orchestrator가 배포 또는 배포된 리소스를 변경하게 됩니다.

### vRealize Automation 2일차 업데이트에 클라우드 템플릿 입력을 사용하는 방법

클라우드 템플릿을 설계할 때 vRealize Automation 입력 매개 변수를 사용하면 2일차 사용자가 초기 배포 요청의 선택을 다시 입력할 수 있습니다.

**경고** 일부 속성을 변경하면 리소스가 다시 생성됩니다. 예를 들어 `Cloud.Service.Azure.App.Service` 아래에서 `connection_string.name`을 변경하면 기존 리소스가 삭제되고 새 리소스가 생성됩니다.

2일차 변경을 지원하는 입력을 설계할 때 리소스를 삭제하고 다시 생성하는 입력을 허용할지 여부를 결정합니다. 리소스를 다시 생성하는 속성에 대해 알아보려면 [vRealize Automation 리소스 속성 소개](#)에서 스키마 링크를 따르십시오.

입력 생성 방법에 대한 자세한 내용은 [vRealize Automation에서 사용자 입력이 클라우드 템플릿을 사용자 지정할 수 있는 방법](#) 항목을 참조하십시오.

특정 2일차에 대한 예는 다음 섹션을 참조하십시오.

### 배포된 시스템을 다른 네트워크로 이동하는 방법

배포 및 네트워크를 유지하는 동안 vRealize Automation Cloud Assembly에 배포한 시스템을 재배포하는 기능이 필요할 수 있습니다.

예를 들어 먼저 테스트 네트워크에 배포한 다음 운영 네트워크로 이동할 수 있습니다. 여기에 설명된 기술을 사용하면 2일차 작업을 준비하기 위해 클라우드 템플릿을 미리 설계할 수 있습니다. 시스템이 이동했는지 확인합니다. 시스템은 삭제 및 다시 배포되지 않습니다.

이 절차는 **Cloud.vSphere.Machine** 리소스에만 적용됩니다. vSphere에 배포된 클라우드 애그노스틱 시스템에는 작동하지 않습니다.

### 사전 요구 사항

- vRealize Automation Cloud Assembly 네트워크 프로파일에는 시스템이 연결될 모든 서브넷이 포함되어야 합니다. vRealize Automation Cloud Assembly에서 **인프라 > 구성 > 네트워크 프로파일**로 이동하여 네트워크를 확인할 수 있습니다.

네트워크 프로파일은 사용자에게 적합한 vRealize Automation Cloud Assembly 프로젝트에 속하는 계정 및 지역에 있어야 합니다.

- 서로 다른 태그를 사용하여 두 서브넷에 태그를 지정합니다. 다음 예에서는 **test** 및 **prod**를 태그 이름으로 가정합니다.
- 배포된 시스템은 동일한 IP 할당 유형을 유지해야 합니다. 다른 네트워크로 이동할 때 정적에서 DHCP로 또는 그 반대로 변경할 수 없습니다.

### 절차

- 1 vRealize Automation Cloud Assembly에서 **설계**로 이동하고 배포에 대한 클라우드 템플릿을 생성합니다.
- 2 코드의 입력 섹션에서 사용자가 네트워크를 선택할 수 있는 항목을 추가합니다.

```
inputs:
  net-tagging:
    type: string
    enum:
      - test
      - prod
    title: Select a network
```

- 3 코드의 리소스 섹션에서 **Cloud.Network**를 추가하고 vSphere 시스템을 여기에 연결합니다.
- 4 **Cloud.Network**에서 입력의 선택을 참조하는 제약 조건을 생성합니다.

```
resources:
  ABCServer:
    type: Cloud.vSphere.Machine
    properties:
      name: abc-server
      . . .
      networks:
        - network: '${resource["ABCNet"].id}'
  ABCNet:
    type: Cloud.Network
    properties:
```

```
name: abc-network
. . .
constraints:
  - tag: '${input.net-tagging}'
```

- 5 설계를 계속 진행하고 평소와 같이 배포합니다. 배포 시 인터페이스에 **test** 또는 **prod** 네트워크를 선택하라는 메시지가 표시됩니다.
- 6 2일차 변경을 수행해야 하는 경우 **배포**로 이동하여 클라우드 템플릿에 연결된 배포를 찾습니다.
- 7 배포의 오른쪽에서 **작업 > 업데이트**를 클릭합니다.
- 8 [업데이트] 패널에서 인터페이스에 동일한 방식으로 **test** 또는 **prod** 네트워크를 선택하라는 메시지가 표시됩니다.
- 9 네트워크를 변경하려면 원하는 항목을 선택하고 **다음, 제출**을 차례로 클릭합니다.

## 가상 시스템의 vMotion 작업을 수행하기 위한 vRealize Automation Cloud Assembly 사용자 지정 작업을 생성하는 방법

클라우드 템플릿을 배포한 후 배포를 변경하는 2일차 작업을 실행할 수 있습니다. vRealize Automation Cloud Assembly에는 여러 2일차 작업이 포함되지만 다른 작업을 제공할 수도 있습니다. 사용자 지정 리소스 작업을 생성하고 사용자가 2일차 작업으로 사용할 수 있도록 할 수 있습니다.

사용자 지정 리소스 작업은 vRealize Orchestrator 워크플로를 기반으로 합니다.

사용자 지정 2일차 작업의 예는 생성 프로세스를 소개하기 위한 것입니다. 사용자 지정 작업을 효과적으로 사용하려면 필요한 작업을 실행하는 vRealize Orchestrator 워크플로와 작업을 생성할 수 있어야 합니다.

### 사전 요구 사항

- vRealize Orchestrator 통합을 구성했는지 확인합니다. [Cloud Assembly에서 vRealize Orchestrator 통합 구성](#)의 내용을 참조하십시오.
- 2일차 작업에 사용 중인 워크플로가 vRealize Orchestrator에 있고 여기에서 성공적으로 실행되는지 확인합니다.

## 절차

- 1 vMotion을 사용하여 특정 호스트에서 다른 호스트로 vSphere 가상 시스템을 이동하는 사용자 지정 리소스 작업을 생성합니다.
  - a vRealize Automation Cloud Assembly에서 **설계 > 리소스 작업**을 선택하고 **새 리소스 작업**을 클릭합니다.
  - b 다음 값을 제공합니다.  
워크플로 이름 이외의 값은 모두 샘플 값입니다.

설정	샘플 값
이름	<b>vSphere_VM_vMotion</b> 리소스 작업 목록에 표시되는 이름입니다.
표시 이름	<b>VM 이동</b> 사용자의 배포 작업 메뉴에 표시되는 이름입니다.

- c 리소스 유형과 일치하는 리소스에 대해 2일차 작업 메뉴에서 이 작업을 사용하도록 설정하려면 **활성화** 옵션을 클릭합니다.
  - d 2일차 작업을 정의하는 리소스 유형 및 워크플로를 선택합니다.

설정	샘플 값
리소스 유형	<p><b>Cloud.vSphere.Machine</b> 리소스 유형을 선택합니다.</p> <p>클라우드 템플릿 구성 요소로 배포되는 리소스 유형이며, 반드시 클라우드 템플릿에 있는 것은 아닙니다. 예를 들어, 클라우드 템플릿에 클라우드 애그노스틱 시스템이 있을 수 있지만 vCenter Server에 배포되면 시스템이 <b>Cloud.vSphere.Machine</b>입니다. 작업은 배포된 유형에 적용되므로 사용자 지정 작업을 정의할 때 클라우드 애그노스틱 유형을 사용하지 마십시오.</p> <p>이 예에서 vMotion은 vSphere 시스템에만 작동하지만 여러 리소스 유형에서 실행하려는 다른 작업이 있을 수 있습니다. 각 리소스 유형에 대한 작업을 생성해야 합니다.</p>
워크플로	<p><b>vMotion으로 가상 시스템 마이그레이션</b> 워크플로를 선택합니다.</p> <p>여러 개의 vRealize Orchestrator 통합이 있는 경우 이러한 사용자 지정 리소스 작업을 실행하는 데 사용하는 통합 인스턴스에서 워크플로를 선택합니다.</p>

- 2 vRealize Automation Cloud Assembly 스키마 속성에 대한 vRealize Orchestrator 속성의 바인딩을 생성합니다. vRealize Automation Cloud Assembly 2일차 작업은 세 가지 유형의 바인딩을 지원합니다.

바인딩 유형	설명
요청 시	기본값 바인딩 유형입니다. 선택한 경우 입력 속성이 요청 양식에 표시되고, 요청 시 사용자가 해당 값을 제공해야 합니다.
바인딩 포함 작업	이 옵션은 다음과 같은 참조 유형 입력에만 사용할 수 있습니다. <ul style="list-style-type: none"> <li>■ VC:VirtualMachine</li> <li>■ VC:Folder</li> </ul> 사용자가 바인딩을 수행하는 작업을 선택합니다. 선택한 작업은 입력 매개 변수와 동일한 유형을 반환해야 합니다. 올바른 속성 정의는 <code>\${properties.someProperty}</code> 입니다.
직접	이 옵션은 기본 데이터 유형을 사용하는 입력 속성에 사용할 수 있습니다. 선택한 경우 적절한 유형의 속성이 입력 속성의 스키마에서 직접 매핑됩니다. 사용자가 스키마 트리에서 속성을 선택합니다. 다른 유형의 속성은 사용되지 않도록 설정됩니다.

이 사용 사례에서 바인딩은 워크플로에 사용된 vRealize Orchestrator VC:VirtualMachine 입력 유형과 vRealize Automation Cloud Assembly Cloud.vSphere.Machine 리소스 유형 간에 연결하는 vRealize Orchestrator 작업입니다. 바인딩을 설정하면 vSphere VM 시스템에 대한 vMotion 작업을 요청하는 사용자에게 2일차 작업을 원활하게 수행할 수 있습니다. 시스템에서 워크플로에 이름을 지정하므로 사용자가 이름을 지정할 필요가 없습니다.

- vMotion으로 가상 시스템 마이그레이션** 워크플로를 선택한 후 **속성 바인딩** 창으로 이동합니다.
- vm 입력 속성의 바인딩을 선택합니다.
- 바인딩**에서 **바인딩 포함 작업**을 선택합니다.

**findVcVmByVcAndVmUuid** 작업이 자동으로 선택됩니다. 이 작업은 vRealize Automation Cloud Assembly의 vRealize Orchestrator 통합으로 미리 구성되어 있습니다.
- 저장**을 클릭합니다.

- 3 2일차 작업에 대한 변경 내용을 저장하려면 **생성**을 클릭합니다.

- 4 워크플로에서 다른 입력 매개 변수를 고려하기 위해 작업을 요청할 때 사용자에게 표시되는 요청 양식을 사용자 지정할 수 있습니다.

- a 리소스 작업에서 최근에 생성한 2일차 작업을 선택합니다.
- b 요청 매개 변수 편집을 클릭합니다.

요청 페이지가 사용자에게 표시되는 방식을 사용자 지정할 수 있습니다.

기본 필드 이름	화면 표시	값	제약 조건
가상 시스템의 대상 리소스 풀입니다. 기본값은 현재 리소스 풀입니다.	<ul style="list-style-type: none"> <li>레이블 = 대상 리소스 풀</li> <li>표시 유형 = 값 선택</li> </ul>		
가상 시스템을 마이그레이션할 대상 호스트	<ul style="list-style-type: none"> <li>레이블 = 대상 호스트</li> <li>표시 유형 = 값 선택</li> </ul>		필수 = 예
마이그레이션 작업의 우선 순위	레이블 = 작업의 우선 순위	값 옵션 <ul style="list-style-type: none"> <li>값 소스 = 상수</li> </ul> 텍스트 상자에 선택으로 구분된 목록을 입력합니다. <div>             lowPriority Low,defaultPriority Default,highPriority High           </div>	필수 = 예
(선택 사항) 해당 전원 크기 상태가 지정된 상태와 일치하는 경우에만 가상 시스템을 마이그레이션합니다.	이 텍스트 상자를 삭제합니다.  vMotion은 모든 전원 상태에서 시스템을 이동할 수 있습니다.		

- c 저장을 클릭합니다.

- 5 작업을 사용할 수 있는 시간을 제한하기 위해 조건을 구성할 수 있습니다.

예를 들어 시스템에 CPU가 4개 이하인 경우에만 vMotion 작업을 사용할 수 있도록 설정하려는 경우가 있을 수 있습니다.

- a 조건 필요를 설정합니다.
- b 조건을 입력합니다.

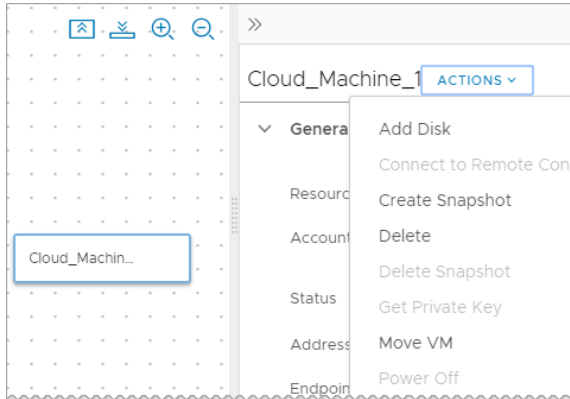
Key	연산자	값
\${properties.cpuCount}	lessThan	4

복합 조건이 필요한 경우 [vRealize Automation Cloud Assembly 사용자 지정 작업에 대한 고급 조건을 구축하는 방법](#)의 내용을 참조하십시오.

- c 업데이트를 클릭합니다.

6 조건과 일치하는 배포된 시스템에 대해 [VM 이동] 작업을 사용할 수 있는지 확인합니다.

- a 배포를 선택합니다.
- b 정의된 조건과 일치하는 배포된 시스템이 포함된 배포를 찾습니다.
- c 배포를 열고 시스템을 선택합니다.
- d 오른쪽 창에서 작업을 클릭하고 Move VM 작업이 있는지 확인합니다.

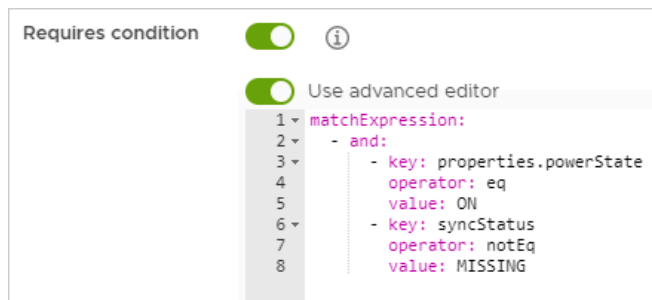


- e 작업을 실행합니다.

#### vRealize Automation Cloud Assembly 사용자 지정 작업에 대한 고급 조건을 구축하는 방법

vRealize Automation Cloud Assembly의 단순 조건 목록을 사용하는 대신 고급 편집기로 더 복잡한 조건 표현식을 조합하여 작업 사용 시기를 제어할 수 있습니다.

새 리소스 작업을 생성할 때 **조건 필요**를 선택하고 **고급 편집기 사용**을 선택합니다. 그런 다음 원하는 조건 표현식을 입력합니다.



표현식은 절 또는 절의 목록으로, 각각 키-연산자-값 형식으로 되어 있습니다. 위 그림에는 대상의 전원을 켜고 표시해야 하는 조건이 나와 있습니다.

## 절

절	설명	예
및	표현식 결과가 <b>true</b> 가 되려면 모든 하위 절이 <b>true</b> 여야 합니다.	<p>properties.powerState가 ON이고 syncStatus가 MISSING이 아닌 경우에만 <b>true</b>로 평가합니다.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: eq       value: ON     - key: syncStatus       operator: notEq       value: MISSING</pre>
또는	표현식 결과가 <b>true</b> 가 되려면 하나 이상의 하위 절이 <b>true</b> 여야 합니다.	<p>properties.powerState가 ON 또는 OFF인지 여부를 <b>true</b>로 평가합니다.</p> <pre>matchExpression:   - or:     - key: properties.powerState       operator: eq       value: ON     - key: properties.powerState       operator: eq       value: OFF</pre>

## 연산자

연산자	설명	예
eq	같음. 정확히 일치하는 항목을 찾습니다.	<p>properties.powerState가 ON일 때 <b>true</b>로 평가합니다.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: eq       value: ON</pre>
notEq	같지 않음. 정확히 일치하는 항목을 피합니다.	<p>properties.powerState가 OFF가 아닐 때 <b>true</b>로 평가합니다.</p> <pre>matchExpression:   - and:     - key: properties.powerState       operator: notEq       value: OFF</pre>

연산자	설명	예
hasAny	개체 모음에서 일치 항목을 찾습니다.	<p>storage.disks 어레이에 100 IOPS EBS 개체가 있을 때 <b>true</b>로 평가합니다.</p> <pre>matchExpression: - key: storage.disks   operator: hasAny   value:     matchExpression:       - and:           - key: iops             operator: eq             value: 100           - key: service             operator: eq             value: ebs</pre>
in	값 집합에서 일치 항목을 찾습니다.	<p>properties.powerState가 OFF 또는 SUSPEND 인 경우 <b>true</b>로 평가합니다.</p> <pre>matchExpression: - and:     - key: properties.powerState       operator: in       value: OFF, SUSPEND</pre>
notIn	일치하는 값 집합을 피합니다.	<p>properties.powerState가 OFF도 아니고 SUSPEND도 아닌 경우 <b>true</b>로 평가합니다.</p> <pre>matchExpression: - and:     - key: properties.powerState       operator: notIn       value: OFF, SUSPEND</pre>
greaterThan	지정된 임계값을 초과하는 일치 항목을 찾습니다. 숫자 값에만 적용됩니다.	<p>storage.disks 어레이의 첫 번째 개체에 50을 초과하는 IOPS가 있는 경우 <b>true</b>로 평가합니다.</p> <pre>matchExpression: - and:     - key: storage.disks[0].iops       operator: greaterThan       value: 50</pre>
lessThan	지정된 임계값 미만의 일치 항목을 찾습니다. 숫자 값에만 적용됩니다.	<p>storage.disks 어레이의 첫 번째 개체에 200 미만의 IOPS가 있는 경우 <b>true</b>로 평가합니다.</p> <pre>matchExpression: - and:     - key: storage.disks[0].iops       operator: lessThan       value: 200</pre>

연산자	설명	예
greaterThanEquals	지정된 임계값보다 크거나 같은 일치 항목을 찾습니다. 숫자 값에만 적용됩니다.	<p>storage.disks 배열의 첫 번째 개체에 100 이상의 IOPS가 있는 경우 true로 평가합니다.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator:         greaterThanEquals         value: 100</pre>
lessThanEquals	지정된 임계값보다 작거나 같은 일치 항목을 찾습니다. 숫자 값에만 적용됩니다.	<p>storage.disks 배열의 첫 번째 개체에 100 이하의 IOPS가 있는 경우 true로 평가합니다.</p> <pre>matchExpression:   - and:     - key: storage.disks[0].iops       operator: lessThanEquals         value: 100</pre>
matchesRegex	정규식을 사용하여 일치 항목을 찾습니다.	<p>properties.zone이 us-east-1a 또는 us-east-1c인 경우 true로 평가합니다.</p> <pre>matchExpression:   - and:     - key: properties.zone       operator: matchesRegex         value: (us-east-1)+(a c)     {1,2}</pre>

## 예

다음 조건 표현식은 properties.tags에 키 key1 및 값 value1의 태그가 포함되어 있을 때 true로 평가합니다.

외부 표현식은 properties.tags가 어레이이므로 hasAny를 사용하며, 어레이의 키-값 쌍에 key1=value1이 나타날 때마다 true로 평가하려고 합니다.

내부 표현식에는 두 개의 절, 즉 키 필드에 대한 절 하나와 값 필드에 대한 절 하나가 있습니다.

properties.tags 어레이는 키-값 태그 지정 쌍을 보유하며, 키 필드와 값 필드를 모두 일치시켜야 합니다.

```
matchExpression:
  - key: properties.tags
    operator: hasAny
    value:
      matchExpression:
        - and:
          - key: key
            operator: eq
            value: key1
          - key: value
            operator: eq
            value: value1
```

다음 조건 표현식은 이전 예제와 유사하지만 이제 `properties.tags`에 `key1=value1` 또는 `key2=value2`의 태그가 포함될 때마다 `true`로 평가됩니다.

```
matchExpression:
  - or:
    - key: properties.tags
      operator: hasAny
      value:
        matchExpression:
          - and:
            - key: key
              operator: eq
              value: key1
            - key: value
              operator: eq
              value: value1
    - key: properties.tags
      operator: hasAny
      value:
        matchExpression:
          - and:
            - key: key
              operator: eq
              value: key2
            - key: value
              operator: eq
              value: value2
```

## 확장성을 사용하여 애플리케이션 수명 주기를 연장 및 자동화하는 방법

확장성 작업 또는 vRealize Orchestrator 워크플로를 확장성 구독과 함께 사용하여 애플리케이션 수명 주기를 연장할 수 있습니다.

vRealize Automation Cloud Assembly 확장성을 사용하면 구독을 통해 확장성 작업 또는 vRealize Orchestrator 워크플로를 이벤트에 할당할 수 있습니다. 이벤트가 발생하면 구독은 실행할 작업 또는 워크플로를 시작하고, 모든 구독자에게 알립니다.

### 확장성 작업

확장성 작업은 작업 및 작업 수행 방법을 지정하는 데 사용되는 작은 경량 코드 스크립트입니다. 미리 정의된 vRealize Automation Cloud Assembly 작업 템플릿에서 또는 ZIP 파일에서 확장성 작업을 가져올 수 있습니다. 작업 편집기를 사용하여 확장성 작업에 대한 사용자 지정 스크립트를 생성할 수도 있습니다. 여러 작업 스크립트가 하나의 스크립트에 연결되어 있으면 작업 흐름을 생성합니다. 작업 흐름을 사용하여 순서가 지정된 작업을 생성할 수 있습니다. 작업 흐름 사용에 대한 자세한 내용은 [작업 흐름이란?](#) 항목을 참조하십시오.

### vRealize Orchestrator 워크플로

vRealize Automation Cloud Assembly를 기존 vRealize Orchestrator 환경에 통합하여 확장성 구독에서 워크플로를 사용할 수 있습니다.

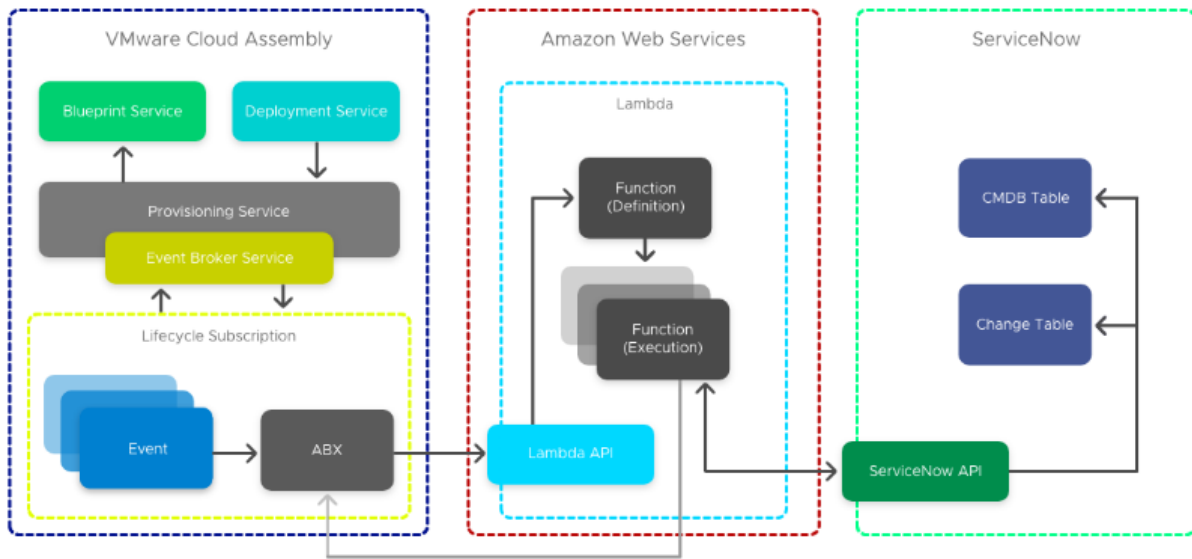
## 확장성 작업 구독

vRealize Automation Cloud Assembly 구독에 확장성 작업을 할당하여 애플리케이션 수명 주기를 연장할 수 있습니다.

**참고** 다음 구독은 사용 사례 예시이며 모든 확장성 작업 기능을 다루지는 않습니다.

### 확장성 작업을 사용하여 Cloud Assembly와 ServiceNow를 통합하는 방법

확장성 작업을 사용하면 vRealize Automation Cloud Assembly를 ServiceNow 같은 Enterprise ITSM과 통합할 수 있습니다.

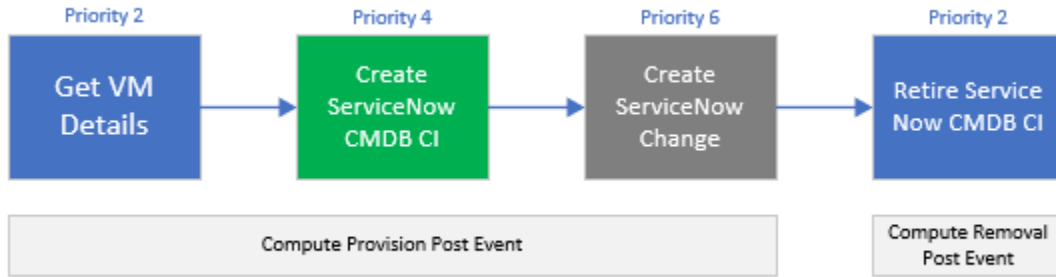


엔터프라이즈 사용자는 규정 준수를 위해 일반적으로 Cloud Management Platform을 ITSM(IT 서비스 관리) 및 CMDB(구성 관리 데이터베이스) 플랫폼과 통합합니다. 이 예시를 사용하면 확장성 작업 스크립트를 사용하여 CMDB 및 ITSM을 위해 vRealize Automation Cloud Assembly를 ServiceNow와 통합할 수 있습니다.

**참고** 또한 vRealize Orchestrator 워크플로를 사용하여 ServiceNow를 vRealize Automation Cloud Assembly와 통합할 수도 있습니다. 워크플로를 사용하여 ServiceNow를 통합하는 방법에 대한 자세한 내용은 [vRealize Orchestrator 워크플로를 사용하여 ITSM용 Cloud Assembly와 ServiceNow를 통합하는 방법](#) 항목을 참조하십시오.

이 통합을 생성하기 위해 4개의 확장성 작업 스크립트를 사용합니다. 처음 3개의 스크립트는 프로비저닝 중에 순차적으로 시작됩니다(프로비저닝 후 계산 이벤트). 4번째 스크립트는 제거 후 계산 이벤트 시점에 트리거됩니다.

이벤트 항목에 대한 자세한 내용은 [vRealize Automation Cloud Assembly](#)에서 제공되는 이벤트 항목 항목을 참조하십시오.



## VM 세부 정보 가져오기

VM 세부 정보 가져오기 스크립트는 CI 생성에 필요한 페이로드 세부 정보 및 Amazon Web Services Systems Manager(SSM) Parameter Store에 저장되어 있는 ID 토큰을 가져옵니다. 또한 이 스크립트는 나중에 사용하기 위해 추가적인 속성으로 `customProperties`를 업데이트합니다.

## ServiceNow CMDB CI 생성

ServiceNow CMDB CI 생성 스크립트는 ServiceNow 인스턴스 URL을 입력으로 전달하고, 보안 요구 사항을 충족하도록 SSM에 인스턴스를 저장합니다. 이 스크립트는 ServiceNow CMDB 고유 레코드 ID 응답(`sys_id`)도 읽습니다. 이 스크립트는 이를 출력으로 전달하고 생성 시 사용자 지정 속성 `serviceNowSysId`를 씁니다. 이 값은 인스턴스가 삭제될 때 CI를 회수된 것으로 표시하는 데 사용됩니다.

---

**참고** Lambda가 SSM Parameter Store에 액세스할 수 있도록 vRealize Automation services Amazon Web Services 역할에 추가적인 사용 권한을 할당해야 할 수도 있습니다.

---

## ServiceNow 변경 사항 생성

이 스크립트는 ServiceNow 인스턴스 URL을 입력으로 전달하고 보안 요구 사항을 충족하기 위해 ServiceNow 자격 증명을 SSM으로 저장하여 ITSM 통합을 완료합니다.

## ServiceNow 변경 사항 생성

ServiceNow CMDB CI 회수 스크립트는 ServiceNow 중지 메시지를 표시하고, 생성 스크립트에 생성된 사용자 지정 속성 `serviceNowSysId`에 기반하여 CI를 회수된 것으로 표시합니다.

## 사전 요구 사항

- 이 통합을 구성하기 전에 조건부 클라우드 템플릿 속성 `event.data["customProperties"]["enable_servicenow"] == "true"`를 사용하여 모든 이벤트 구독을 필터링합니다.

---

**참고** 이 속성은 ServiceNow 통합이 필요한 클라우드 템플릿에 있습니다.

---

- Python을 다운로드하고 설치합니다.

구독 필터링에 대한 자세한 내용은 [확장성 구독 생성](#) 항목을 참조하십시오.

## 절차

- 1 가상 시스템에서 명령줄 프롬프트를 엽니다.
- 2 VM 세부 정보 가져오기 스크립트를 실행합니다.

```
from botocore.vendored import requests
import json
import boto3
client = boto3.client('ssm','ap-southeast-2')

def handler(context, inputs):
    baseUri = inputs['url']
    casToken = client.get_parameter(Name="casToken",WithDecryption=True)

    url = baseUri + "/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 = baseUri + "/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 CMDB 구성 항목 생성 작업을 실행합니다.

```
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 생성 작업 스크립트를 실행합니다.

```

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)

```

## 결과

vRealize Automation Cloud Assembly가 ITSM ServiceNow와 통합되었습니다.

## 다음에 수행할 작업

필요한 경우, CMDB 구성 항목 회수 작업을 사용하여 CI를 회수할 수 있습니다.

```

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+"/"+{0}".format(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)

```

확장성 작업을 사용하여 vRealize Automation Cloud Assembly에 ServiceNow를 통합할 수 있는 방법에 대한 자세한 내용은 [ServiceNow](#) 통합을 위해 작업 기반 확장성을 사용하여 [Cloud Assembly](#) 확장을 참조하십시오.

## 프로비저닝하는 동안 확장성 작업을 사용하여 가상 시스템에 태그를 지정하는 방법

확장성 작업을 구독과 함께 사용하여 VM 태그 지정을 자동화하고 간소화할 수 있습니다.

클라우드 관리자는 확장성 작업 및 확장성 구독을 사용하여, 지정된 입력 및 출력으로 태그가 자동으로 지정되는 배포를 생성할 수 있습니다. VM 태그 지정 구독이 포함된 프로젝트에 대해 새 배포를 생성하면 배포 이벤트가 VM 태그 지정 스크립트의 실행을 트리거하고, 태그가 자동으로 적용됩니다. 이를 통해 시간을 절약하고 효율성을 높이는 동시에 배포를 더 쉽게 관리할 수 있습니다.

### 사전 요구 사항

- 클라우드 관리자 자격 증명에 대한 액세스.
- Lambda 함수에 대한 Amazon Web Services 역할.

### 절차

- 1 **확장성 > 라이브러리 > 작업 > 새 작업**으로 이동하고 다음 매개 변수를 사용하여 작업을 생성합니다.

매개 변수	설명
작업 이름	확장성 작업 이름이며, <b>TagVM</b> 을 접두사 또는 접미사로 사용하는 것이 좋습니다.
프로젝트	확장성 작업을 테스트할 대상 프로젝트입니다.
작업 템플릿	<b>VM 태그 지정</b>
런타임	Python
스크립트 소스	스크립트 작성

- 2 **기본 함수**으로 **Handler**를 입력합니다.
- 3 확장성 작업을 테스트하기 위한 태그 지정 입력을 추가합니다.  
예: `resourceNames = ["DB_VM"]` 및 `target = world`.
- 4 작업을 저장하려면 **저장**을 클릭합니다.
- 5 작업을 테스트하려면 **테스트**를 클릭합니다.
- 6 작업 편집기를 종료하려면 **닫기**를 클릭합니다.
- 7 **확장성 > 구독**으로 이동합니다.
- 8 **새 구독**을 클릭합니다.

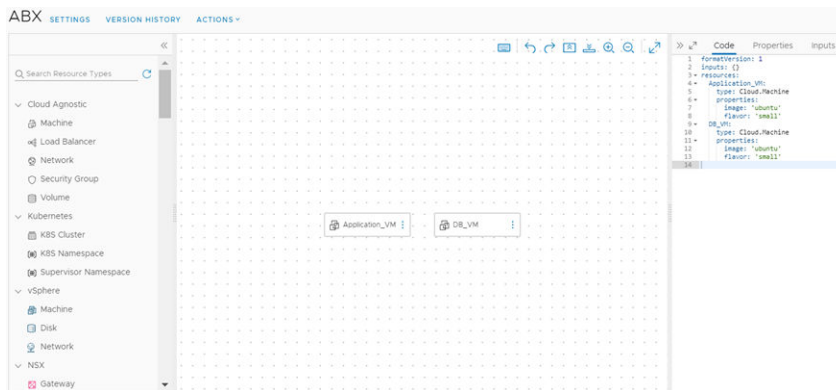
9 다음과 같은 구독 세부 정보를 입력합니다.

세부 정보	설정
이벤트 항목	VM의 태그 지정 단계와 관련된 이벤트 항목을 선택합니다. 예를 들어 계산 할당 이벤트 항목을 선택합니다.  <b>참고</b> 태그는 선택한 이벤트 항목의 이벤트 매개 변수에 포함되어야 합니다.
차단	구독에 대한 시간 초과를 1분으로 설정합니다.
작업/워크플로	확장성 작업 실행 가능 유형을 선택하고 사용자 지정 확장성 작업을 선택합니다.

10 사용자 지정 확장성 작업 구독을 저장하려면 **저장**을 클릭합니다.

11 **설계 > 클라우드 템플릿**으로 이동하여 빈 캔버스에서 클라우드 템플릿을 생성합니다.

12 클라우드 템플릿에 두 개(Application\_VM 및 DB\_VM)의 가상 시스템을 추가합니다.



13 VM을 배포하려면 **배포**를 클릭합니다.

14 배포 중에 이벤트가 시작되고 확장성 작업이 실행되는지 확인합니다.

15 태그가 올바르게 적용되었는지 확인하려면 **인프라 > 리소스 > 시스템**으로 이동합니다.

### 확장성 작업에 대해 알아보기

작업 기반 확장성은 vRealize Automation Cloud Assembly 내에서 간소화된 코드 스크립트를 사용하여 확장성 작업을 자동화합니다.

작업 기반 확장성은 스크립트 가능한 소규모 작업을 정의하고 확장성 구독에 지정된 이벤트가 발생할 때 시작하도록 구성할 수 있는 유연한 경량 런타임 엔진 인터페이스를 제공합니다.

vRealize Automation Cloud Assembly 내에서 또는 로컬 환경에 이러한 코드의 확장성 작업 스크립트를 생성하여 구독에 할당할 수 있습니다. 확장성 작업 스크립트는 작업 및 단계를 더 경량으로 간단하게 자동화하는 데 사용됩니다. vRealize Automation Cloud Assembly를 vRealize Orchestrator 서버와 통합하는 방법에 대한 자세한 내용은 [Cloud Assembly에서 vRealize Orchestrator 통합 구성](#) 항목을 참조하십시오.

작업 기반 확장성은...

- 작고 재사용할 수 있는 스크립트 가능 작업을 통해 경량 통합 및 사용자 지정을 가능하게 함으로써 vRealize Orchestrator 워크플로를 대체할 수 있는 대안을 제공합니다.
- 재사용할 수 있는 매개 변수화된 작업이 포함된 작업 템플릿을 재사용할 수 있는 방법을 제공합니다.

확장성 작업은 사용자 정의 작업 스크립트 코드를 작성하거나 미리 정의된 스크립트 코드를 .ZIP 패키지로 가져오는 방법으로 생성할 수 있습니다. 작업 기반 확장성은 Node.js, Python 및 PowerShell 런타임 환경을 지원합니다. Node.js 및 Python 런타임은 Amazon Web Services Lambda에 의존합니다. 따라서 Amazon Web Services IAM(Identity and Access Management)이 포함된 활성 구독이 있어야 하며 vRealize Automation Cloud Assembly에서 Amazon Web Services를 끝점으로 구성해야 합니다. Amazon Web Services Lambda 사용 방법에 대한 자세한 내용은 [ABX: Serverless Extensibility of Cloud Assembly Services](#)를 참조하십시오.

---

**참고** 확장성 작업은 프로젝트별로 다릅니다.

---

### 확장성 작업을 생성하는 방법

vRealize Automation Cloud Assembly를 통해 확장성 구독에 사용할 확장성 작업을 생성할 수 있습니다.

확장성 작업을 사용하면 다양하게 사용자 지정할 수 있는 경량의 유연한 방법으로 사용자 정의된 스크립트 코드와 작업 템플릿을 사용하여 애플리케이션의 수명 주기를 연장할 수 있습니다. 작업 템플릿에는 확장성 작업의 기초를 설정하기 위한 미리 정의된 매개 변수가 포함됩니다.

확장성 작업은 두 가지 방법으로 생성할 수 있습니다.

- 확장성 작업 스크립트의 사용자 정의 코드를 작성합니다.

---

**참고** 확장성 작업 편집기에서 사용자 정의 코드를 작성하려면 활성 인터넷 연결이 필요할 수 있습니다.

---

- 배포 패키지를 확장성 작업을 위한 ZIP 패키지로 가져옵니다. 확장성 작업을 위한 ZIP 패키지 생성에 대한 자세한 내용은 [Python 런타임 확장성 작업을 위한 ZIP 패키지 생성](#), [Node.js 런타임 확장성 작업을 위한 ZIP 패키지 생성](#) 또는 [PowerShell 런타임 확장성 작업을 위한 ZIP 패키지 생성](#)의 내용을 참조하십시오.

다음 단계는 Amazon Web Services를 FaaS 제공자로 사용하는 확장성 작업을 생성하는 절차를 설명합니다.

### 사전 요구 사항

- 활성 및 유효한 프로젝트의 멤버 자격이 필요합니다.
- Lambda 함수에 대해 구성된 Amazon Web Services 역할이 있어야 합니다. 예를 들어 AWSLambdaBasicExecutionRole가 표시됩니다.
- 클라우드 관리자 역할 또는 iam:PassRole 사용 권한이 설정되어 있어야 합니다.

### 절차

- 1 **확장성 > 라이브러리 > 작업**을 선택합니다.

- 2 새 작업을 클릭합니다.
- 3 작업의 이름을 입력하고 위치를 선택합니다.
- 4 (선택 사항) 작업에 대한 설명을 추가합니다.
- 5 다음을 클릭합니다.
- 6 작업 템플릿을 검색하고 선택합니다.

---

**참고** 작업 템플릿을 사용하지 않고 사용자 지정 작업을 생성하려면 **사용자 지정 스크립트**를 선택합니다.

---

구성 가능한 새 매개 변수가 나타납니다.

- 7 스크립트 작성 또는 패키지 가져오기를 선택합니다.
- 8 작업 런타임을 선택합니다.
- 9 작업 진입점에 대한 **기본 함수** 이름을 입력합니다.

---

**참고** ZIP 패키지에서 가져온 작업의 경우, 기본 함수에는 진입점이 들어 있는 스크립트 파일의 이름도 포함되어야 합니다. 예를 들어 기본 스크립트 파일의 제목이 `main.py`이고 진입점이 `handler` (`context, inputs`)인 경우 기본 함수의 이름은 *main.handler*여야 합니다.

---

- 10 작업의 입력 및 출력 매개 변수를 정의합니다.
- 11 (선택 사항) 기본 입력에 확장성 작업 비밀을 추가합니다.

---

**참고** 비밀은 `context.getSecret()` 기능을 사용하여 스크립트의 일부로 실행됩니다. 각 비밀에는 작업의 테스트 실행을 수행하고 **세부 정보 참조**를 클릭하여 찾을 수 있는 고유 식별자도 있습니다. 확장성 작업 비밀을 관리하는 방법에 대한 자세한 내용은 [확장성 작업 비밀을 생성하는 방법](#) 항목을 참조하십시오.

---

## 12 (선택 사항) 작업에 애플리케이션 종속성을 추가합니다.

**참고** PowerShell 스크립트의 경우, PowerShell 갤러리 저장소에 대해 해결되도록 애플리케이션 종속성을 정의할 수 있습니다. 애플리케이션 종속성을 정의하기 위해 공용 저장소에서 확인할 수 있는 형식은 다음과 같습니다.

```
@{
    Name = 'Version'
}

e.g.

@{
    Pester = '4.3.1'
}
```

**참고** ZIP 패키지에서 가져온 작업의 경우 애플리케이션 종속성이 자동으로 추가됩니다.

**13** 시간 초과 및 메모리 제한을 정의하려면 **사용자 지정 시간 초과 및 제한 설정** 옵션을 사용하도록 설정합니다.

**14** 작업을 테스트하려면 **저장**을 클릭한 다음 **테스트**를 클릭합니다.

### 다음에 수행할 작업

확장성 작업을 생성하고 확인한 후에는 구독에 할당할 수 있습니다.

**참고** 확장성 구독은 확장성 작업의 최신 릴리스 버전을 사용합니다. 새 버전의 작업을 생성한 후 편집기 창 오른쪽 상단에 있는 **버전**을 클릭합니다. 구독에 사용할 작업 버전을 릴리스하려면 **릴리스**를 클릭합니다.

### Python 런타임 확장성 작업을 위한 ZIP 패키지 생성

vRealize Automation Cloud Assembly 확장성 작업에서 사용되는 Python 스크립트와 종속성을 포함하는 ZIP 패키지를 생성할 수 있습니다.

확장성 작업을 위한 스크립트를 구축하는 방법에는 두 가지가 있습니다.

- vRealize Automation Cloud Assembly의 확장성 작업 편집기에서 스크립트를 직접 작성합니다.
- 로컬 환경에서 스크립트를 생성하고 이를 모든 관련 종속성과 함께 ZIP 패키지에 추가합니다.

ZIP 패키지를 사용하면 확장성 작업에 사용하기 위해 vRealize Automation Cloud Assembly로 가져올 수 있는 작업 스크립트 및 종속성의 미리 구성된 사용자 지정 템플릿을 생성할 수 있습니다.

또한 사용자 환경에서 인터넷 액세스가 부족한 경우와 같이 작업 스크립트의 종속성과 연결된 모듈을 vRealize Automation Cloud Assembly 서비스에서 확인할 수 없는 시나리오에서 ZIP 패키지를 사용할 수 있습니다.

또한 ZIP 패키지를 사용하여 여러 Python 스크립트 파일이 포함된 확장성 작업을 생성할 수 있습니다. 여러 스크립트 파일을 사용하면 확장성 작업 코드의 구조를 구성하는 데 유용할 수 있습니다.

## 사전 요구 사항

Python 3.3 이하 버전을 사용 중인 경우 PIP 패키지 설치 관리자를 다운로드하고 구성합니다. [Python 패키지 색인](#)을 참조하십시오.

## 절차

- 1 로컬 시스템에서 작업 스크립트 및 종속성에 대한 폴더를 생성합니다.

예: /home/user1/zip-action

- 2 기본 Python 작업 스크립트를 폴더에 추가합니다.

예: /home/user1/zip-action/main.py

- 3 (선택 사항) Python 스크립트에 대한 종속성을 폴더에 추가합니다.

a 종속성이 포함된 requirements.txt 파일을 생성합니다. [요구 사항 파일](#)을 참조하십시오.

b Linux 셸을 엽니다.

---

**참고** vRealize Automation Cloud Assembly의 작업 기반 확장성의 런타임은 Linux 기반입니다. 따라서 Python 종속성이 Windows 환경에서 컴파일된 경우 생성된 ZIP 패키지는 확장성 작업을 생성하는 데 사용하지 못할 수 있습니다. 따라서 Linux 셸을 사용해야 합니다.

---

- c 다음 명령을 실행하여 스크립트 폴더에 requirements.txt 파일을 설치합니다.

```
pip install -r requirements.txt --target=home/user1/zip-action
```

- 4 할당된 폴더에서 스크립트 요소와 requirements.txt 파일(해당하는 경우)을 선택하고 이를 ZIP 패키지에 압축합니다.

---

**참고** 스크립트 요소와 종속성 요소는 모두 ZIP 패키지의 루트 수준에 저장해야 합니다. Linux 환경에서 ZIP 패키지를 생성할 때 패키지 콘텐츠가 루트 수준에 저장되지 않는 문제가 발생할 수 있습니다. 이 문제가 발생하는 경우 명령줄 셸에서 `zip -r` 명령을 실행하여 패키지를 생성합니다.

---

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

## 다음에 수행할 작업

ZIP 패키지를 사용하여 확장성 작업 스크립트를 생성합니다. [확장성 작업을 생성하는 방법](#)의 내용을 참조하십시오.

Node.js 런타임 확장성 작업을 위한 ZIP 패키지 생성

vRealize Automation Cloud Assembly 확장성 작업에서 사용되는 Node.js 스크립트와 종속성을 포함하는 ZIP 패키지를 생성할 수 있습니다.

확장성 작업을 위한 스크립트를 구축하는 방법에는 두 가지가 있습니다.

- vRealize Automation Cloud Assembly의 확장성 작업 편집기에서 스크립트를 직접 작성합니다.
- 로컬 환경에서 스크립트를 생성하고 이를 모든 관련 종속성과 함께 ZIP 패키지에 추가합니다.

ZIP 패키지를 사용하면 확장성 작업에 사용하기 위해 vRealize Automation Cloud Assembly로 가져올 수 있는 작업 스크립트 및 종속성의 미리 구성된 사용자 지정 템플릿을 생성할 수 있습니다.

또한 사용자 환경에서 인터넷 액세스가 부족한 경우와 같이 작업 스크립트의 종속성과 연결된 모듈을 vRealize Automation Cloud Assembly 서비스에서 확인할 수 없는 시나리오에서 ZIP 패키지를 사용할 수 있습니다.

또한 패키지를 사용하여 여러 Node.js 스크립트 파일이 포함된 확장성 작업을 생성할 수 있습니다. 여러 스크립트 파일을 사용하면 확장성 작업 코드의 구조를 구성하는 데 유용할 수 있습니다.

## 절차

- 1 로컬 시스템에서 작업 스크립트 및 종속성에 대한 폴더를 생성합니다.

예: /home/user1/zip-action

- 2 기본 Node.js 작업 스크립트를 폴더에 추가합니다.

예: /home/user1/zip-action/main.js

- 3 (선택 사항) Node.js 스크립트에 대한 종속성을 폴더에 추가합니다.

- a 스크립트 폴더에 종속성이 있는 package.json 파일을 생성합니다. package.json 파일 생성 및 package.json 파일에서 dependencies 및 devDependencies 지정을 참조하십시오.

- b 명령줄 셸을 엽니다.

- c 작업 스크립트 및 종속성에 대해 생성한 폴더로 이동합니다.

```
cd /home/user1/zip-action
```

- d 다음 명령을 실행하여 스크립트 폴더에 package.json 파일을 설치합니다.

```
npm install --production
```

---

**참고** 이 명령은 폴더에 node\_modules 디렉토리를 생성합니다.

---

- 4 할당된 폴더에서 스크립트 요소와 node\_modules 디렉토리(해당하는 경우)를 선택하고 이를 ZIP 패키지에 압축합니다.

**참고** 스크립트 요소와 종속성 요소는 모두 ZIP 패키지의 루트 수준에 저장해야 합니다. Linux 환경에서 ZIP 패키지를 생성할 때 패키지 콘텐츠가 루트 수준에 저장되지 않는 문제가 발생할 수 있습니다. 이 문제가 발생하는 경우 명령줄 셸에서 `zip -r` 명령을 실행하여 패키지를 생성합니다.

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

### 다음에 수행할 작업

ZIP 패키지를 사용하여 확장성 작업 스크립트를 생성합니다. [확장성 작업을 생성하는 방법](#)의 내용을 참조하십시오.

PowerShell 런타임 확장성 작업을 위한 ZIP 패키지 생성

확장성 작업에 사용할 PowerShell 스크립트 및 종속성 모듈이 포함된 ZIP 패키지를 생성할 수 있습니다.

확장성 작업을 위한 스크립트를 구축하는 방법에는 두 가지가 있습니다.

- vRealize Automation Cloud Assembly의 확장성 작업 편집기에서 스크립트를 직접 작성합니다.
- 로컬 환경에서 스크립트를 생성하고 이를 모든 관련 종속성과 함께 ZIP 패키지에 추가합니다.

ZIP 패키지를 사용하면 확장성 작업에 사용하기 위해 vRealize Automation Cloud Assembly로 가져올 수 있는 작업 스크립트 및 종속성의 미리 구성된 사용자 지정 템플릿을 생성할 수 있습니다.

**참고** PowerCLI cmdlet을 종속성으로 정의하거나 ZIP 패키지로 번들로 제공할 필요가 없습니다.

PowerCLI cmdlet은 vRealize Automation Cloud Assembly 서비스의 PowerShell 런타임으로 미리 구성됩니다.

또한 사용자 환경에서 인터넷 액세스가 부족한 경우와 같이 작업 스크립트의 종속성과 연결된 모듈을 vRealize Automation Cloud Assembly 서비스에서 확인할 수 없는 시나리오에서 ZIP 패키지를 사용할 수 있습니다.

또한 ZIP 패키지를 사용하여 여러 PowerShell 스크립트 파일이 포함된 확장성 작업을 생성할 수 있습니다. 여러 스크립트 파일을 사용하면 확장성 작업 코드의 구조를 구성하는 데 유용할 수 있습니다.

### 사전 요구 사항

PowerShell 및 PowerCLI에 대해 잘 알고 있는지 확인합니다. [Docker Hub](#)에서 PowerShell Core, PowerCLI 10, PowerNSX 및 여러 커뮤니티 모듈 및 스크립트 예가 포함된 Docker 이미지를 찾을 수 있습니다.

### 절차

- 1 로컬 시스템에서 작업 스크립트 및 종속성에 대한 폴더를 생성합니다.

예: /home/user1/zip-action

- 2 .psm1 확장명을 가진 기본 PowerShell 스크립트를 폴더에 추가합니다.

다음 스크립트는 main.psm1이라는 간단한 PowerShell 함수를 보여줍니다.

```
function handler($context, $payload) {

    Write-Host "Hello " $payload.target

    return $payload
}
```

**참고** PowerShell 확장성 작업의 출력은 함수의 본문에 표시되는 마지막 변수를 기반으로 합니다. 포함된 함수에 있는 다른 모든 변수는 삭제됩니다.

- 3 (선택 사항) context 매개 변수를 사용하여 기본 PowerShell 스크립트에 프록시 구성을 추가합니다. 컨텍스트 매개 변수를 사용하여 [PowerShell 스크립트에 프록시 구성 추가](#)의 내용을 참조하십시오.
- 4 (선택 사항) PowerShell 스크립트에 대한 종속성을 추가합니다.

**참고** PowerShell 종속성 스크립트는 .psm1 확장명을 사용해야 합니다. 스크립트와 스크립트가 저장된 하위 폴더에 동일한 이름을 사용합니다.

- a Linux PowerShell 셸에 로그인합니다.

**참고** vRealize Automation Cloud Assembly의 작업 기반 확장성의 런타임은 Linux 기반입니다. Windows 환경에서 컴파일되는 모든 PowerShell 종속성은 생성된 ZIP 패키지를 사용할 수 없게 만들 수 있습니다. PowerShell 스크립트가 Photon OS에서 실행하므로 설치된 타사 종속성은 VMware Photon OS와 호환되어야 합니다.

- b /home/user1/zip-action 폴더로 이동합니다.
- c Save-Module cmdlet을 실행하여 종속성이 포함된 PowerShell 모듈을 다운로드하고 저장합니다.

```
Save-Module -Name <module name> -Path ./
```

- d 추가 종속성 모듈에 대해 이전 하위 단계를 반복합니다.

**중요** 각 종속성 모듈이 별도의 하위 폴더에 있는지 확인합니다. PowerShell 모듈 작성 및 관리에 대한 자세한 내용은 [PowerShell 스크립트 모듈을 작성하는 방법](#)을 참조하십시오.

- 5 할당된 폴더에서 스크립트 요소와 종속성 모듈 하위 폴더(해당하는 경우)를 선택하고 이를 ZIP 패키지에 압축합니다.

**참고** 스크립트 모듈 하위 폴더와 종속성 모듈 하위 폴더는 모두 ZIP 패키지의 루트 수준에 저장해야 합니다. Linux 환경에서 ZIP 패키지를 생성할 때 패키지 콘텐츠가 루트 수준에 저장되지 않는 문제가 발생할 수 있습니다. 이 문제가 발생하는 경우 명령줄 셸에서 `zip -r` 명령을 실행하여 패키지를 생성합니다.

```
cd your_script_and_dependencies_folder
zip -r ../your_action_ZIP.zip *
```

### 다음에 수행할 작업

ZIP 패키지를 사용하여 확장성 작업 스크립트를 생성합니다. [확장성 작업을 생성하는 방법](#)의 내용을 참조하십시오.

컨텍스트 매개 변수를 사용하여 PowerShell 스크립트에 프록시 구성 추가

context 매개 변수를 사용하여 PowerShell 스크립트에서 네트워크 프록시 통신을 사용하도록 설정할 수 있습니다.

특정 PowerShell cmdlet을 사용하려면 네트워크 프록시를 PowerShell 함수의 환경 변수로 설정해야 할 수 있습니다. 프록시 구성은 `$context.proxy.host` 및 `$context.proxy.port` 매개 변수를 사용하여 PowerShell 함수에 제공됩니다.

이러한 context 매개 변수를 PowerShell 스크립트의 시작 부분에 추가할 수 있습니다.

```
$proxyString = "http://" + $context.proxy.host + ":" + $context.proxy.port
$Env:HTTP_PROXY = $proxyString
$Env:HTTPS_PROXY = $proxyString
```

cmdlet이 `-Proxy` 매개 변수를 지원하는 경우 프록시 값을 특정 PowerShell cmdlet에 직접 전달할 수도 있습니다.

클라우드별 확장성 작업 구성

클라우드 계정을 사용할 수 있도록 확장성 작업을 구성할 수 있습니다.

확장성 작업을 생성할 때 작업을 구성하고 다음과 같은 다양한 클라우드 기반 계정에 연결할 수 있습니다.

- Microsoft Azure
- Amazon Web Services

### 사전 요구 사항

올바른 클라우드 계정이 필요합니다.

### 절차

- 1 **확장성 > 라이브러리 > 작업**을 선택합니다.

- 2 새 작업을 클릭합니다.
- 3 필요에 따라 작업 매개 변수를 입력합니다.
- 4 **FaaS 제공자** 드롭다운 메뉴에서 클라우드 계정 제공자를 선택하거나 **자동**을 선택합니다.

---

**참고** **자동**을 선택하는 경우 작업에서 **FaaS** 제공자를 자동으로 정의합니다.

---

- 5 **저장**을 클릭합니다.

## 결과

구성된 클라우드 계정에서 사용할 수 있도록 확장성 작업이 연결됩니다.

온-프레미스 확장성 작업 구성

Amazon Web Services나 Microsoft Azure 클라우드 계정 대신 온-프레미스 FaaS 제공자를 사용하도록 확장성 작업을 구성할 수 있습니다.

확장성 작업에 대해 온-프레미스 FaaS 제공자를 사용하면 vRealize Automation Cloud Assembly 확장성 구독에서 LDAP, CMDB 또는 vCenter 데이터 센터와 같은 온-프레미스 서비스를 사용할 수 있습니다.

## 절차

- 1 **확장성 > 라이브러리 > 작업**을 선택합니다.
- 2 새 작업을 클릭합니다.
- 3 확장성 작업의 이름과 프로젝트를 입력합니다.
- 4 (선택 사항) 확장성 작업에 대한 설명을 입력합니다.
- 5 **다음**을 클릭합니다.
- 6 확장성 작업 스크립트를 생성하거나 가져옵니다.
- 7 **FaaS 제공자** 드롭다운 메뉴를 클릭하고 **온-프레미스**를 선택합니다.
- 8 새 확장성 작업을 저장하려면 **저장**을 클릭합니다.

## 다음에 수행할 작업

생성된 확장성 작업을 vRealize Automation Cloud Assembly 확장성 구독에서 사용합니다.

## 확장성 작업 비밀을 생성하는 방법

vRealize Automation Cloud Assembly를 통해 확장성 작업에 사용할 비밀을 생성하고 저장할 수 있습니다.

확장성 작업 비밀을 사용하여 암호화된 입력 값을 확장성 작업에 추가할 수 있습니다. 암호화는 입력을 사용하여 암호 및 인증서와 같은 중요 데이터를 관리하는 사용 사례에 유용합니다. 모든 FaaS 제공자 및 런타임에 비밀을 사용할 수 있습니다.

비밀은 `context.getSecret()` 기능을 사용하여 스크립트의 일부로 실행됩니다. 이 함수는 비밀 이름을 매개 변수로 사용합니다. 예를 들어 작업에서 `abxsecret`라는 이름의 확장성 작업 비밀을 암호화된 입력 매개 변수로 사용할 수 있습니다. 작업 스크립트에 이 입력 매개 변수를 추가하려면 `context.getSecret(inputs["abxsecret"])`를 사용해야 합니다.

## 절차

- 1 **확장성 > 라이브러리 > 작업**으로 이동합니다.
- 2 **작업 비밀 관리**를 선택합니다.
- 3 암호를 생성하려면 **새 작업 비밀**을 클릭합니다.
- 4 비밀의 이름과 값을 입력하고 **저장**을 클릭합니다.

## 다음에 수행할 작업

확장성 작업에 비밀을 추가합니다. [확장성 작업을 생성하는 방법](#)의 내용을 참조하십시오.

## 공유 확장성 작업 생성

vRealize Automation Cloud Assembly 관리자는 작업을 내보내서 가져오지 않고도 프로젝트 간에 공유할 수 있는 확장성 작업을 생성할 수 있습니다.

확장성 작업 내보내기 및 가져오기에 대한 자세한 내용은 [확장성 내보내기 및 가져오기 작업](#)에서 참조하십시오.

## 사전 요구 사항

vRealize Automation Cloud Assembly 조직에서 둘 이상의 프로젝트를 생성합니다.

## 절차

- 1 **확장성 > 라이브러리 > 작업**을 선택합니다.
- 2 **새 작업**을 클릭합니다.
- 3 확장성 작업의 이름을 입력합니다.
- 4 (선택 사항) 확장성 작업에 대한 설명을 입력합니다.
- 5 확장성 작업을 생성할 프로젝트를 선택합니다.
- 6 **이 조직의 모든 프로젝트와 공유** 확인란을 선택합니다.
- 7 **다음**을 클릭합니다.
- 8 작업 스크립트를 생성하거나 가져온 후 확장성 작업을 저장합니다.

---

**참고** **설정**에서 공유를 사용하거나 사용하지 않도록 설정할 수 있습니다. 구독에서 확장성 작업이 사용되면 공유를 사용하지 않도록 설정할 수 없습니다. 공유를 사용하지 않도록 설정하려면 구독에서 확장성 작업을 제거해야 합니다.

---

- 9 확장성 구독을 생성하고, 공유 확장성 작업을 추가하고, 구독 범위를 **모든 프로젝트**로 설정합니다.

---

**참고** 확장성 구독 생성에 대한 자세한 내용은 [확장성 구독 생성](#) 항목을 참조하십시오.

---

확장성 구독은 모든 프로젝트에서 일치하는 이벤트에 의해 트리거됩니다.

## 다음에 수행할 작업

vRealize Automation Service Broker 카탈로그에서 공유 확장성 작업을 콘텐츠 소스로 가져올 수도 있습니다. 소스 프로젝트를 선택할 때 확장성 작업이 생성된 프로젝트를 입력합니다. vRealize Automation Service Broker에 확장성 작업을 추가하는 방법에 대한 자세한 내용은 [Service Broker 카탈로그에 확장성 작업 추가](#)를 참조하십시오.

## 확장성 내보내기 및 가져오기 작업

vRealize Automation Cloud Assembly를 통해 다양한 프로젝트에서 사용할 확장성 작업을 내보내고 가져올 수 있습니다.

### 사전 요구 사항

기존 확장성 작업.

### 절차

- 1 확장성 작업을 내보냅니다.
  - a **확장성 > 라이브러리 > 작업**으로 이동합니다.
  - b 확장성 작업을 선택하고 **내보내기**를 클릭합니다.  
작업 스크립트 및 해당 종속성이 로컬 환경에 ZIP 파일로 저장됩니다.
- 2 확장성 작업을 가져옵니다.
  - a **확장성 > 라이브러리 > 작업**으로 이동합니다.
  - b **가져오기**를 클릭합니다.
  - c 내보낸 확장성 작업을 선택하여 프로젝트에 할당합니다.
  - d **가져오기**를 클릭합니다.

---

**참고** 가져온 확장성 작업이 지정된 프로젝트에 이미 할당되어 있으면, 충돌 해결 정책을 선택하라는 메시지가 표시됩니다.

---



---

**대안** 작업 편집기에서 **패키지 가져오기**를 직접 선택하여 작업 스크립트를 가져올 수도 있습니다.

---

### 작업 흐름이란?

작업 흐름은 수명 주기 및 자동화를 더 확장하는 데 사용되는 확장성 작업 스크립트의 집합입니다.

모든 작업 흐름은 flow\_start로 시작하고 flow\_end로 끝납니다. 다음과 같은 작업 흐름 요소를 사용하여 여러 확장성 작업 스크립트를 함께 연결할 수 있습니다.

- **순차적 작업 흐름** - 순차적으로 실행되는 여러 확장성 작업 스크립트입니다.
- **분기 작업 흐름** - 경로를 분할하여 동일한 결과를 제공하는 여러 확장성 작업 스크립트 또는 흐름입니다.
- **참여 작업 흐름** - 서로 조인되어 동일한 결과를 제공하는 여러 확장성 작업 스크립트 또는 흐름입니다.
- **조건부 작업 흐름** - 조건이 충족되었을 때 실행되는 여러 확장성 작업 스크립트 또는 흐름입니다.

## 순차적 작업 흐름

순차적으로 실행되는 여러 확장성 작업 스크립트입니다.

```
version: "1"
flow:
  flow_start:
    next: action1
  action1:
    action: <action_name>
    next: action2
  action2:
    action: <action_name>
    next: flow_end
```

**참고** 이전 작업을 `next:` 작업으로 할당하여 이전 작업으로 루프백할 수 있습니다. 예를 들어 이 예에서는 `next: flow_end` 대신 `next: action1`을 입력하여 **action1**을 다시 실행하고 작업 시퀀스를 다시 시작할 수 있습니다.



## 분기 작업 흐름

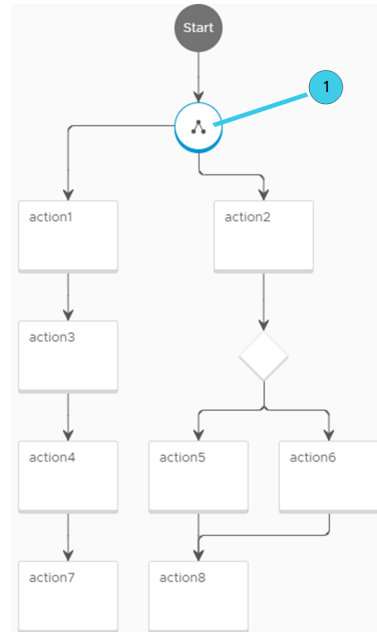
경로를 분할하여 동일한 출력을 제공하는 여러 확장성 작업 스크립트 또는 흐름입니다.

```

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>

```

**참고** 이전 작업을 `next: 작업으로 할당하여 이전 작업으로 루프백`할 수 있습니다. 예를 들어 `next: flow_end`를 사용하여 작업 흐름을 종료하는 대신 `next: action1`을 입력하면 `action1`을 다시 실행하여 작업 시퀀스를 다시 시작할 수 있습니다.



① 분기 요소

### 참여 작업 흐름

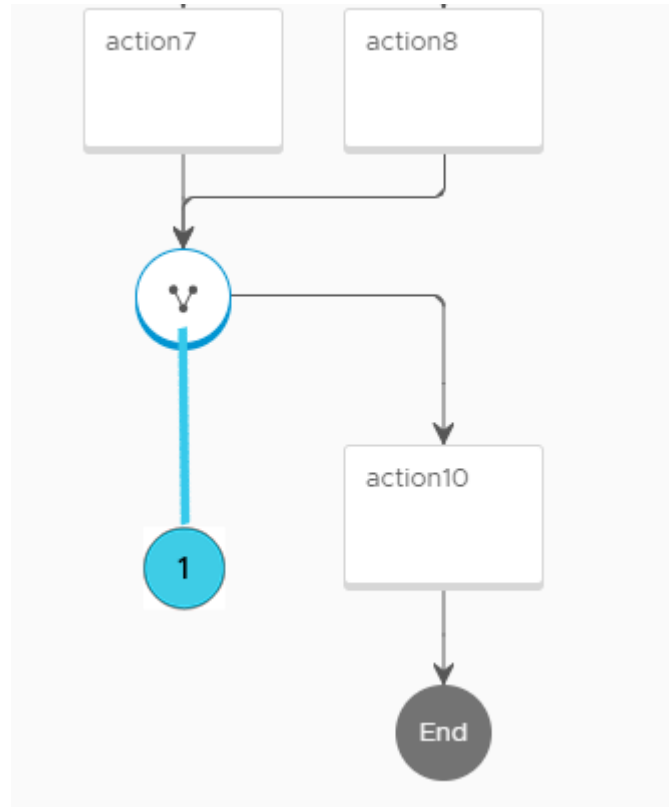
경로를 함께 조인하여 동일한 출력을 제공하는 여러 확장성 작업 스크립트 또는 흐름입니다.

```

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

```

**참고** 이전 작업을 `next: 작업으로 할당하여` 이전 작업으로 루프백할 수 있습니다. 예를 들어 이 예에서는 `next: flow_end` 대신 `next: action1`을 입력하여 **action1**을 다시 실행하고 작업 시퀀스를 다시 시작할 수 있습니다.



① 조인 요소

### 조건부 작업 흐름

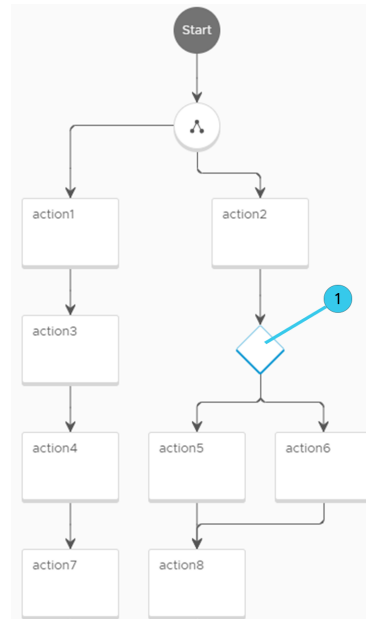
스위치 요소를 사용하여 조건이 충족되면 실행되는 여러 확장성 작업 스크립트 또는 흐름입니다.

경우에 따라서는 조건이 `true`와 같아야 작업이 실행됩니다. 그 외의 경우에는 이 예시에 나와 있듯이 매개 변수 값이 충족되어야 작업이 실행될 수 있습니다. 조건이 충족되지 않으면 작업 흐름이 실패합니다.

```

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 스위치 요소

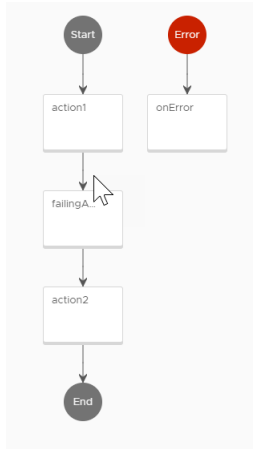
**참고** 이전 작업을 `next`: 작업으로 할당하여 이전 작업으로 루프백할 수 있습니다. 예를 들어 `next`: `flow_end`를 사용하여 작업 흐름을 종료하는 대신 `next`: `action1`을 입력하면 `action1`을 다시 실행하여 작업 시퀀스를 다시 시작할 수 있습니다.

작업 흐름에서 오류 처리기를 사용하는 방법

오류 처리기 요소를 사용하여 지정된 흐름 단계에서 오류를 생성하도록 작업 흐름을 구성할 수 있습니다.

오류 처리기 요소에는 두 가지 입력이 필요합니다.

- 실패한 작업의 지정된 오류 메시지.
- 작업 흐름 입력.



흐름의 작업이 실패하고 작업 흐름에 오류 처리기 요소가 포함되어 있는 경우 오류 메시지가 생성되면서 작업 실패를 경고합니다. 오류 처리기는 자체 작업입니다. 다음 스크립트는 작업 흐름에서 사용될 수 있는 오류 처리기의 예입니다.

```
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
```

[작업 실행] 창에서 성공한 실행과 실패한 실행을 볼 수 있습니다.

상태	작업	작업 ID
완료됨	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
실패	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
완료됨	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6
완료됨	AWS-ABX	8a769ecc6df809c7016e01a83fe204e6

이 예에서 오류 처리기 요소가 들어 있는 **flow-with-handler** 작업 흐름이 성공적으로 실행되었습니다. 하지만 흐름의 작업 중 하나가 실패하여 오류 처리기가 시작되었고 오류가 생성되었습니다.

## 작업 실행을 추적하는 방법

작업 실행 탭에는 구독에서 트리거된 확장성 작업 및 해당 상태에 대한 로그가 표시됩니다.

**확장성 > 작업 > 작업 실행**을 사용하여 작업 실행 로그를 볼 수 있습니다. 또한 하나 이상의 속성을 한 번에 사용하여 작업 실행 목록을 필터링할 수도 있습니다. 실행 ID를 클릭하면 개별 작업 실행에 대한 추가적인 세부 정보를 볼 수 있습니다.

## 실패한 확장성 작업 실행 문제 해결

확장성 작업 실행이 실패하면 문제 해결 단계를 수행하여 문제를 해결할 수 있습니다.

작업 실행이 실패하면 오류 메시지, 실패 상태 및 실패 로그가 수신될 수 있습니다. 작업 실행이 실패했다면 그 원인은 배포 실패이거나 코드 실패입니다.

문제	해결 방법
배포 실패	이러한 실패는 클라우드 계정 구성, 작업 배포 또는 작업의 배포를 방해할 수 있는 기타 종속성과 관련된 문제의 결과입니다. 사용한 프로젝트가 구성된 클라우드 계정 내에 정의되어 있는지 그리고 기능을 실행할 수 있는 사용 권한이 있는지 확인합니다. 작업을 다시 시작하기 전에 작업의 세부 정보 페이지 내에서 특정 프로젝트를 대상으로 작업을 테스트할 수 있습니다.
코드 실패	이러한 실패는 잘못된 스크립트 또는 코드의 결과입니다. [작업 실행] 로그를 사용하여 문제를 해결하고 잘못된 스크립트를 수정합니다.

## 확장성 워크플로 구독

vRealize Orchestrator에 호스팅된 워크플로를 vRealize Automation Cloud Assembly와 함께 사용하여 애플리케이션 수명 주기를 연장할 수 있습니다.

## vRealize Orchestrator 워크플로 구독을 사용하여 가상 시스템 속성을 수정하는 방법

기존 vRealize Orchestrator 워크플로를 사용하여 가상 시스템 속성을 수정하고 가상 시스템을 Active Directory에 추가할 수 있습니다.

이벤트 항목 매개 변수는 EBS(Event Broker Service) 메시지에 대한 페이로드의 형식을 정의합니다. 워크플로 내에서 EBS 메시지 페이로드를 받아서 사용하려면 inputProperties 워크플로 입력 매개 변수를 정의해야 합니다.

## 사전 요구 사항

- 클라우드 관리자 사용자 역할이 있어야 합니다.
- 기존 vRealize Orchestrator 온-프레미스 워크플로.
- 성공적인 통합 및 vRealize Orchestrator 클라이언트 서버에 대한 연결이 있어야 합니다.

## 절차

- 1 **확장성 > 구독**을 선택합니다.
- 2 **새 구독**을 클릭합니다.

3 다음 매개 변수를 사용하여 구독을 생성합니다.

매개 변수	값
이름	RenameVM
이벤트 항목	원하는 vRealize Orchestrator 통합에 적합한 이벤트 항목을 선택합니다. 예를 들어 계산 할당 이벤트 항목을 선택합니다.
차단/비차단	비차단
작업/워크플로	vRealize Orchestrator Runnable 유형을 선택합니다. 원하는 워크플로를 선택합니다. 예: Set VM name

4 구독을 저장하려면 **저장**을 클릭합니다.

5 클라우드 템플릿을 생성하거나 기존 클라우드 템플릿을 배포하여 구독을 할당하고 활성화합니다.

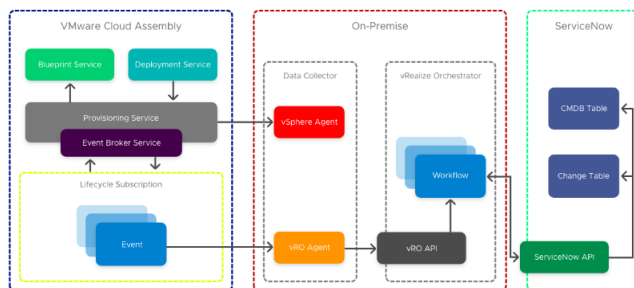
#### 다음에 수행할 작업

다음 방법 중 하나를 사용하여 워크플로가 성공적으로 시작되었는지 확인합니다.

- 워크플로 실행이 기록되는지 확인합니다(**확장성 > 작업 > 워크플로 실행**).
- vRealize Orchestrator 클라이언트를 열고 워크플로로 이동한 후 상태를 확인하거나, 해당하는 로그 탭을 열어서 워크플로 상태를 확인합니다.

#### vRealize Orchestrator 워크플로를 사용하여 ITSM용 Cloud Assembly와 ServiceNow를 통합하는 방법

vRealize Orchestrator 호스팅된 워크플로를 사용하여 ITSM 규정 준수를 위해 vRealize Automation Cloud Assembly와 ServiceNow를 통합할 수 있습니다.



엔터프라이즈 사용자는 규정 준수를 위해 일반적으로 Cloud Management Platform을 ITSM(IT 서비스 관리) 및 CMDB(구성 관리 데이터베이스) 플랫폼과 통합합니다. 이 예시를 사용하면 vRealize Orchestrator 호스팅된 워크플로를 사용하여 CMDB 및 ITSM을 위해 vRealize Automation Cloud Assembly를 ServiceNow와 통합할 수 있습니다. vRealize Orchestrator 통합 및 워크플로를 사용할 때 기능 태그는 서로 다른 환경에 대해 인스턴스가 여러 개 있는 경우에 특히 유용합니다. 기능 태그에 대한 자세한 내용은 [vRealize Automation Cloud Assembly에서 기능 태그 사용 항목](#)을 참조하십시오.

**참고** 확장성 작업 스크립트를 사용하여 ServiceNow를 vRealize Automation Cloud Assembly와 통합할 수도 있습니다. 확장성 작업 스크립트를 사용한 ServiceNow 통합에 대한 자세한 내용은 [확장성 작업을 사용하여 Cloud Assembly와 ServiceNow를 통합하는 방법](#) 항목을 참조하십시오.

이 예시에서 ServiceNow 통합은 3개의 최상위 수준 워크플로로 구성되며, 각 워크플로에는 자체 구독이 포함되어 있어 각 구성 요소를 개별적으로 업데이트하고 반복할 수 있습니다.

- 이벤트 구독 진입점 - 기본 로깅을 사용하며, 요청하는 사용자 및 vCenter VM(해당하는 경우)을 식별합니다.
- 통합 워크플로 - 개체 및 피드 입력을 기술 워크플로로 분리하고 로깅, 속성 및 출력 업데이트를 처리합니다.
- 기술 워크플로 - 페이로드 이외의 추가적인 가상 시스템 속성을 사용하여 CMDB CI, CR 및 vRealize Automation Cloud Assembly IaaS API를 생성하기 위한 ServiceNow API용 다운스트림 시스템 통합입니다.

### 사전 요구 사항

- 독립형 또는 클러스터링된 vRealize Orchestrator 환경.
- vRealize Automation Cloud Assembly의 vRealize Orchestrator 통합. vRealize Automation Cloud Assembly와 독립형 vRealize Orchestrator 통합에 대한 자세한 내용은 [Cloud Assembly에서 vRealize Orchestrator 통합 구성](#) 항목을 참조하십시오.

### 절차

- 1 여러 워크플로에 사용되는 공통 구성이 포함된 구성 파일을 vRealize Orchestrator에서 생성하고 저장합니다.
- 2 vRealize Automation Cloud Assembly API 토큰을 1단계의 구성 파일과 동일한 위치에 저장합니다.

---

**참고** vRealize Automation Cloud Assembly API 토큰에는 만료 기간이 있습니다.

---

- 3 제공된 스크립트 요소를 사용하여 vRealize Orchestrator에서 워크플로를 생성합니다. 이 스크립트는 REST 호스트를 참조하고 해당 위치를 찾습니다. 또한 추가적인 인증 헤더로 추가되는 토큰의 선택적 매개 변수를 사용하는 REST 작업을 표준화합니다.

```
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"];

```

이 스크립트는 cpuCount 및 memoryMB 출력을 상위 워크플로에 보내고 기존 customProperties 속성을 업데이트합니다. 이러한 값은 CMDB를 생성할 때 이후 워크플로에서 사용할 수 있습니다.

- 4 ServiceNow CMDB CI 생성 스크립트 요소를 워크플로에 추가합니다. 이 요소는 구성 항목을 사용하여 ServiceNow REST 호스트의 위치를 찾고, cmdb\_ci\_vmware\_instance 표의 REST 작업을 생성하며, 사후 데이터를 위해 워크플로 입력에 기반한 콘텐츠 개체 문자열을 생성한 후, 반환되는 sys\_id를 출력합니다.

```
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'];
```

- 5 하위 워크플로의 출력을 사용하여 기존 `customProperties`를 사용하는 속성 개체를 생성하고 ServiceNow의 값으로 `serviceNowSysId` 속성을 덮어씁니다. 이 고유 ID는 삭제 시 CMDB에서 인스턴스를 회수된 것으로 표시하기 위해 사용됩니다.

## 결과

vRealize Automation Cloud Assembly가 ITSM ServiceNow와 통합되었습니다. 워크플로를 사용하여 vRealize Automation Cloud Assembly에 ServiceNow를 통합할 수 있는 방법에 대한 자세한 내용은 [ServiceNow 통합을 위해 vRealize Orchestrator를 사용하여 Cloud Assembly 확장을 참조하십시오](#).

## 워크플로 구독에 대해 알아보기

vRealize Orchestrator와 vRealize Automation Cloud Assembly를 통합을 사용하면 워크플로를 사용하여 애플리케이션의 수명 주기를 확장할 수 있습니다.

vRealize Automation에는 내장형 vRealize Orchestrator 배포가 포함되어 있습니다. 구독에서 내장형 vRealize Orchestrator 배포의 워크플로 라이브러리를 사용할 수 있습니다. vRealize Orchestrator 클라이언트를 사용하여 워크플로를 생성, 수정 및 삭제할 수 있습니다.

vRealize Automation Cloud Assembly에서 외부 vRealize Orchestrator 배포를 통합할 수도 있습니다. "내장된 vRealize Orchestrator 클라이언트 사용" 에서 "외부 vRealize Orchestrator 클라이언트를 통합하는 방법" 을 참조하십시오.

### vRealize Orchestrator 워크플로 생성 모범 사례

워크플로 구독은 특정 이벤트 항목 및 해당 항목의 이벤트 매개 변수를 기반으로 합니다. 구독에서 vRealize Orchestrator 워크플로를 시작할 수 있으려면 이벤트 데이터를 사용하도록 올바른 입력 매개 변수로 워크플로를 구성해야 합니다.

#### 워크플로 입력 매개 변수

사용자 지정 워크플로에는 모든 매개 변수 또는 페이로드의 모든 데이터를 사용하는 단일 매개 변수가 포함될 수 있습니다.

단일 매개 변수를 사용하려면 유형이 `Properties`이고 이름이 `inputProperties`인 하나의 매개 변수를 구성합니다.

#### 워크플로 출력 매개 변수

사용자 지정 워크플로에는 회신 이벤트 항목 유형에 필요한 후속 이벤트와 관련된 출력 매개 변수가 포함될 수 있습니다.

이벤트 항목에 회신이 필요한 경우 워크플로 출력 매개 변수가 회신 스키마의 매개 변수와 일치해야 합니다.

### 워크플로 실행을 추적하는 방법

**워크플로 실행** 창에는 구독에서 트리거한 워크플로와 해당 상태에 대한 로그가 표시됩니다.

**확장성 > 작업 > 워크플로 실행**으로 이동하여 워크플로 실행에 대한 로그를 볼 수 있습니다.

### 실패한 워크플로 구독 문제 해결

워크플로 구독이 실패한 경우 문제 해결 단계를 수행하여 문제를 해결할 수 있습니다.

워크플로 실행이 실패하면 워크플로 구독이 시작되지 않거나 완료되지 않을 수 있습니다. 워크플로 실행 실패는 몇 가지 일반적인 문제로 인해 발생할 수 있습니다.

문제	원인	해결 방법
vRealize Orchestrator 워크플로 구독이 시작되지 않거나 완료되지 않습니다.	이벤트 메시지가 수신되면 사용자 지정 워크플로를 실행하도록 워크플로 구독을 구성했지만 워크플로가 실행되지 않거나 완료되지 않습니다.	<ol style="list-style-type: none"> <li>1 워크플로 구독이 올바르게 저장되었는지 확인합니다.</li> <li>2 워크플로 구독 조건이 올바르게 구성되었는지 확인합니다.</li> <li>3 vRealize Orchestrator에 지정된 워크플로가 포함되어 있는지 확인합니다.</li> <li>4 vRealize Orchestrator에 워크플로가 올바르게 구성되었는지 확인합니다.</li> </ol>
승인 요청 vRealize Orchestrator 워크플로 구독이 실행되지 않습니다.	vRealize Orchestrator 워크플로를 실행하도록 사전 승인 또는 사후 승인 워크플로 구독을 구성했습니다. 서비스 카탈로그에서 정의된 조건과 일치하는 시스템이 요청될 때 워크플로가 실행되지 않습니다.	<p>승인 워크플로 구독을 성공적으로 실행하려면 모든 구성 요소가 올바르게 구성되어 있는지 확인해야 합니다.</p> <ol style="list-style-type: none"> <li>1 승인 정책이 활성화되고 올바르게 적용되었는지 확인합니다.</li> <li>2 워크플로 구독이 올바르게 구성 및 저장되었는지 확인합니다.</li> <li>3 이벤트 로그에서 승인 관련 메시지를 검토합니다.</li> </ol>
승인 요청 vRealize Orchestrator 워크플로 구독이 거부됩니다.	<p>지정된 vRealize Orchestrator 워크플로를 실행하는 사전 승인 또는 사후 승인 워크플로 구독을 구성했지만 요청이 외부 승인 수준에서 거부됩니다.</p> <p>한 가지 가능한 원인은 vRealize Orchestrator의 내부 워크플로 실행 오류입니다. 예를 들어 워크플로가 누락되었거나 vRealize Orchestrator 서버가 실행 중이지 않을 수 있습니다.</p>	<ol style="list-style-type: none"> <li>1 로그에서 승인 관련 메시지를 검토합니다.</li> <li>2 vRealize Orchestrator 서버가 실행 중인지 확인합니다.</li> <li>3 vRealize Orchestrator에 지정된 워크플로가 포함되어 있는지 확인합니다.</li> </ol>

## 확장성 구독에 대해 알아보기

확장성 작업 또는 vRealize Orchestrator에 호스팅된 워크플로를 확장성 구독과 함께 사용하여 애플리케이션 수명 주기를 연장할 수 있습니다.

환경에서 트리거 이벤트가 발생하면 구독이 시작되고 지정된 워크플로 또는 확장성 작업이 실행됩니다. 시스템 이벤트는 이벤트 로그에서 보고, 워크플로 실행은 워크플로 실행 창에서 보고 작업 실행은 작업 실행 창에서 볼 수 있습니다. 구독은 프로젝트별로 다릅니다. 즉, 지정된 프로젝트를 통해 클라우드 템플릿 및 배포에 연결됩니다.

## 확장성 용어

vRealize Automation Cloud Assembly에서 확장성 및 구독을 사용할 때 구독과 이벤트 브로커 서비스에만 해당하는 용어가 나올 수 있습니다.

표 6-3. 확장성 용어

용어	설명
이벤트 항목	논리적 의도와 구조가 동일한 일련의 이벤트를 설명합니다. 모든 이벤트는 이벤트 항목의 인스턴스입니다. 특정 이벤트 항목에 차단 매개 변수를 할당할 수 있습니다. 자세한 내용은 <a href="#">차단 이벤트 항목</a> 항목을 참조하십시오.
이벤트	생산자 또는 생산자가 관리하는 엔티티의 상태 변경을 나타냅니다. 이벤트는 이벤트 발생에 대한 정보를 기록하는 엔티티입니다.
이벤트 브로커 서비스	생산자가 구독 소비자에게 게시하는 메시지를 디스패치하는 서비스입니다.
페이로드	해당 이벤트 항목에 관련된 모든 관련 속성이 포함된 이벤트 데이터입니다.
구독	이벤트 항목을 구독하고 알림을 트리거하는 조건을 정의하여 구독자가 이벤트 관련 알림을 받는 것에 관심이 있음을 나타냅니다. 구독은 애플리케이션 수명 주기의 일부를 자동화하는 데 사용되는 트리거 이벤트에 확장성 작업 또는 워크플로를 연결합니다.
구독자	구독 정의를 기반으로 이벤트 브로커 서비스에 게시된 이벤트를 통해 알림을 받는 사용자입니다. 구독자를 소비자라고 할 수도 있습니다.
시스템 관리자	vRealize Automation Cloud Assembly를 사용하여 테넌트 워크플로 구독 및 시스템 워크플로 구독을 생성하고, 읽고, 업데이트하고, 삭제할 수 있는 권한을 가진 사용자입니다.
워크플로 구독	vRealize Orchestrator 워크플로를 트리거하는 이벤트 항목 및 조건을 지정합니다.
작업 구독	확장성 작업 실행을 트리거하는 이벤트 항목 및 조건을 지정합니다.
워크플로	vRealize Automation Cloud Assembly 내에 통합된 vRealize Orchestrator 워크플로입니다. 이러한 워크플로를 구독 내의 이벤트에 연결할 수 있습니다.
확장성 작업	구독에서 이벤트가 트리거된 후 실행할 수 있는 간소화된 코드 스크립트입니다. 확장성 작업은 워크플로와 비슷하지만 더 간단합니다. 확장성 작업은 vRealize Automation Cloud Assembly 내에서 사용자 지정할 수 있습니다.
작업 실행	<b>작업 실행</b> 탭을 통해 액세스할 수 있습니다. 작업 실행은 트리거 이벤트에 대한 응답으로 실행된 확장성 작업의 세부 로그입니다.

### 차단 이벤트 항목

일부 이벤트 항목은 차단 이벤트를 지원합니다. 확장성 구독의 동작은 항목이 이러한 이벤트 유형을 지원하는지 여부와 구독을 구성하는 방법에 따라 다릅니다.

vRealize Automation Cloud Assembly 확장성 구독은 비차단 및 차단 이벤트 항목이라는 두 가지의 광범위한 이벤트 항목 유형을 사용할 수 있습니다. 이벤트 항목 유형은 확장성 구독의 동작을 정의합니다.

## 비차단 이벤트 항목

비차단 이벤트 항목은 비차단 구독만 생성할 수 있도록 허용합니다. 비차단 구독은 비동기식으로 트리거되고 구독이 트리거되는 순서에 의존할 수 없습니다.

## 차단 이벤트 항목

일부 이벤트 항목은 차단을 지원합니다. 구독이 차단으로 표시되는 경우 설정된 조건을 충족하는 모든 메시지는 차단 구독의 **Runnable** 항목이 실행될 때까지 일치하는 조건이 있는 다른 구독에서 수신되지 않습니다.

차단 구독은 우선 순위에 따라 실행됩니다. 가장 높은 우선 순위 값은 0입니다. 동일한 이벤트 항목에 대해 우선 순위 수준이 같은 둘 이상의 차단 구독이 있는 경우 구독은 구독 이름을 기반으로 알파벳 역순으로 실행됩니다. 모든 차단 구독이 처리되고 나면 메시지가 모든 비차단 구독에 동시 전송됩니다. 차단 구독은 동기식으로 실행되기 때문에 후속 구독에 알림이 표시될 때 변경된 이벤트 페이로드에 업데이트된 이벤트가 포함됩니다.

차단 이벤트 항목을 사용하여 서로 종속 관계에 있는 여러 구독을 관리할 수 있습니다.

예를 들어 두 번째 구독이 첫 번째 구독의 결과에 따라 달라지는 두 개의 프로비저닝 워크플로 구독을 가질 수 있다고 가정하면 첫 번째 구독은 프로비저닝 중에 속성을 변경하고 두 번째 구독은 시스템 이름과 같은 새 속성을 파일 시스템에 기록합니다. **ChangeProperty** 구독은 0으로 우선 순위가 지정되고

**RecordProperty**는 두 번째 구독이 첫 번째 구독의 결과를 사용하므로 1로 우선 순위가 지정됩니다. 시스템이 프로비저닝되면 **ChangeProperty** 구독이 실행을 시작합니다. **RecordProperty** 구독 조건은 사후 프로비저닝 조건을 기반으로 하기 때문에 이벤트가 **RecordProperty** 구독을 트리거합니다. 하지만

**ChangeProperty** 워크플로는 차단 워크플로이므로 완료되기 전까지 이벤트가 수신되지 않습니다. 시스템 이름이 변경되고 첫 번째 워크플로 구독이 완료되면 두 번째 워크플로 구독이 실행되면서 파일 시스템에 이름을 기록합니다.

## 복구 Runnable 항목

차단 이벤트 항목의 경우 복구 **Runnable** 항목을 구독에 추가할 수 있습니다. 구독의 복구 **Runnable** 항목은 기본 **Runnable** 항목이 실패하는 경우 실행됩니다. 예를 들어 기본 **Runnable** 항목이 **ServiceNow**와 같은 **CMDB** 시스템에서 기록을 생성하는 워크플로인 워크플로 구독을 생성할 수 있습니다. 워크플로 구독이 실패하더라도 일부 기록이 **CMDB** 시스템에서 생성될 수 있습니다. 이 시나리오에서는 복구 **Runnable** 항목을 사용하여 실패한 **Runnable** 항목으로 인해 **CMDB** 시스템에 남아 있는 기록을 정리할 수 있습니다.

서로 종속 관계에 있는 여러 개의 구독이 포함된 사용 사례의 경우 `ebs.recover.continuation` 속성을 복구 **Runnable** 항목에 추가할 수 있습니다. 이 속성을 사용하면 현재 구독이 실패하는 경우 체인의 다음 구독으로 확장성 서비스를 계속해야 할지 지시할 수 있습니다.

## vRealize Automation Cloud Assembly에서 제공되는 이벤트 항목

vRealize Automation Cloud Assembly에는 미리 정의된 이벤트 항목이 포함되어 있습니다.

## 이벤트 항목

이벤트 항목은 비슷한 이벤트를 그룹화하는 범주입니다. 구독에 할당된 이벤트 항목은 구독을 트리거할 이벤트를 정의합니다. 다음의 이벤트 항목은 vRealize Automation Cloud Assembly에 기본적으로 제공됩니다. 모든 항목은 리소스의 사용자 지정 속성 또는 태그를 추가하거나 업데이트하는 데 사용될 수 있습니다. vRealize Orchestrator 워크플로 또는 확장성 작업이 실패하면 해당하는 작업도 실패합니다.

**표 6-4. Cloud Assembly 이벤트 항목**

이벤트 항목	차단 가능	설명
Cloud template configuration	아니요	클라우드 템플릿 생성 또는 삭제와 같은 클라우드 템플릿 구성 이벤트가 발생한 경우에 생성됩니다. 이 이벤트 항목은 외부 시스템에 이러한 이벤트를 알리는 데 유용할 수 있습니다.
Cloud template version configuration	아니요	버전 생성, 릴리스, 릴리스 취소 또는 복원과 같은 새로운 클라우드 템플릿 버전 관리 이벤트가 발생할 때 생성됩니다. 이 이벤트 항목은 타사 버전 제어 시스템 통합에 유용할 수 있습니다.
Compute allocation	예	resourcenames 및 hostselections가 할당되기 전에 생성됩니다. 이러한 속성 둘 모두 이 단계에서 수정할 수 있습니다.
Compute post provision	예	리소스 프로비저닝이 완료된 이후 생성됩니다.
Compute post removal	예	계산 리소스가 제거된 이후 생성됩니다.
Compute provision	예	하이퍼바이저 계층에서 리소스가 프로비저닝되기 전에 생성됩니다. <b>참고</b> 할당된 IP 주소를 변경할 수 있습니다.
Compute removal	예	리소스가 제거되기 전에 생성됩니다.
Compute reservation	예	예약 시 생성됩니다. <b>참고</b> 배치 순서를 변경할 수 있습니다.
Deployment action completed	예	배포 작업의 완료 후에 생성됩니다.
Deployment action requested	예	배포 작업의 완료 전에 생성됩니다.
Deployment completed	예	클라우드 템플릿 또는 카탈로그 요청 배포 후에 생성됩니다.
Deployment onboarded	아니요	새 배포가 온보딩될 때 생성됩니다.
Deployment requested	예	클라우드 템플릿 또는 카탈로그 요청 배포 전에 생성됩니다.

표 6-4. Cloud Assembly 이벤트 항목 (계속)

이벤트 항목	차단 가능	설명
Deployment resource action completed	예	리소스 작업의 배포 후에 생성됩니다.
Deployment resource action requested	예	리소스 작업의 배포 전에 생성됩니다.
Deployment resource completed	예	배포 리소스의 프로비저닝 후에 생성됩니다.
Deployment resource requested	예	배포 리소스의 프로비저닝 전에 생성됩니다.
Disk allocation	예	디스크 리소스의 사전 할당을 위해 생성됩니다.
Disk attach	예	<p>디스크가 시스템에 연결되기 전에 생성됩니다. Disk attach는 읽기 및 쓰기 이벤트입니다. 다시 쓰기를 위해 지원되는 디스크 속성은 다음과 같습니다.</p> <ul style="list-style-type: none"> <li>■ diskFullPaths</li> <li>■ diskDatastoreNames</li> <li>■ diskParentDirs</li> </ul> <p>업데이트에는 세 가지 vSphere 특정 디스크 속성이 모두 필요합니다. 다른 모든 속성은 읽기 전용입니다.</p> <p><b>참고</b> 다시 쓰기는 vSphere First Class Disk에 대해 선택 사항입니다.</p>
Disk detach	예	디스크가 시스템에서 분리된 후에 생성됩니다. Disk detach는 읽기 전용 이벤트입니다.
Disk post removal	예	디스크 리소스의 삭제 후에 생성됩니다.
Disk post resize	예	디스크 리소스의 크기 조정 후에 생성됩니다.
EventLog	예	로깅 관련 이벤트에 대해 생성됩니다.
Kubernetes cluster allocation	예	Kubernetes 클러스터에 대한 리소스 사전 할당에 대해 생성됩니다.
Kubernetes cluster post provision	예	Kubernetes 클러스터의 프로비저닝 후에 생성됩니다.
Kubernetes cluster post removal	예	Kubernetes 클러스터의 삭제 후에 생성됩니다.
Kubernetes cluster provision	예	Kubernetes 클러스터의 프로비저닝 전에 생성됩니다.
Kubernetes cluster removal	예	Kubernetes 클러스터 삭제 프로세스의 시작 전에 생성됩니다.

표 6-4. Cloud Assembly 이벤트 항목 (계속)

이벤트 항목	차단 가능	설명
Kubernetes namespace allocation	예	Kubernetes 네임스페이스 리소스에 대한 사전 할당 중에 생성됩니다.
Kubernetes namespace post provision	예	Kubernetes 네임스페이스 리소스의 프로비저닝 후에 생성됩니다.
Kubernetes namespace post removal	예	Kubernetes 네임스페이스 리소스의 제거 후에 생성됩니다.
Kubernetes namespace provision	예	Kubernetes 네임스페이스의 프로비저닝 전에 생성됩니다.
Kubernetes namespace removal	예	네임스페이스 클러스터 리소스의 제거 전에 생성됩니다.
Kubernetes supervisor namespace allocation	예	Kubernetes 감독자 네임스페이스 리소스에 대한 사전 할당 중에 생성됩니다.
Kubernetes supervisor namespace post provision	예	감독자 네임스페이스의 프로비저닝 후에 생성됩니다.
Kubernetes supervisor namespace post removal	예	감독자 네임스페이스 리소스의 제거 후에 생성됩니다.
Kubernetes supervisor namespace provision	예	감독자 네임스페이스의 프로비저닝 전에 생성됩니다.
Kubernetes supervisor namespace removal	예	감독자 네임스페이스 리소스의 제거 전에 생성됩니다.
Load balancer post provision	예	로드 밸런서의 프로비저닝 후에 생성됩니다.
Load balancer post removal	예	로드 밸런서의 제거 후에 생성됩니다.
Load balancer provision	예	로드 밸런서의 프로비저닝 전에 생성됩니다.
Load balancer removal	예	로드 밸런서를 제거하기 전에 생성됩니다.
Network Configure	예	계산을 할당하는 동안 네트워크가 구성되면 생성됩니다.  <b>참고</b> 네트워크 구성 항목은 여러 IP 주소/NIC를 지원합니다.
Network post provisioning	예	네트워크 리소스의 프로비저닝 후에 생성됩니다.
Network post removal	예	네트워크 리소스가 제거된 후에 생성됩니다.
Network provisioning	예	네트워크 리소스의 프로비저닝 전에 생성됩니다.
Network removal	예	네트워크 리소스가 제거되기 전에 생성됩니다.

표 6-4. Cloud Assembly 이벤트 항목 (계속)

이벤트 항목	차단 가능	설명
Security group post provisioning	예	보안 그룹의 프로비저닝 후에 생성됩니다.
Security group post removal	예	보안 그룹이 제거된 후에 생성됩니다.
Security group provisioning	예	보안 그룹의 프로비저닝 전에 생성됩니다.
Security group removal	예	보안 그룹이 제거되기 전에 생성됩니다.
Project Lifecycle	아니오	프로젝트가 생성, 업데이트 또는 삭제되면 생성되는 이벤트입니다.
Provisioning request	예	프로비저닝 서비스가 2일차 작업과 리소스 할당과 같은 요청을 수신할 때 실행됩니다.

### 이벤트 매개 변수

이벤트 항목을 추가한 후 해당 이벤트 항목의 매개 변수를 볼 수 있습니다. 이러한 이벤트 매개 변수는 이벤트의 페이로드(inputProperties)의 구조를 정의합니다. 특정 이벤트 매개 변수는 수정할 수 없으며 읽기 전용으로 표시됩니다. 매개 변수의 오른쪽에 있는 정보 아이콘을 클릭하여 이러한 읽기 전용 매개 변수를 식별할 수 있습니다.

### 확장성 이벤트 로그

확장성 이벤트 페이지에는 환경 내에서 발생한 모든 이벤트의 목록이 표시됩니다.

**확장성 > 이벤트**로 이동하여 확장성 이벤트 로그를 볼 수 있습니다. 하나 이상의 속성을 사용하여 이벤트 목록을 필터링할 수도 있습니다. 개별 이벤트에 대한 추가적인 세부 정보를 보려면 이벤트 ID를 선택합니다.

ID	Timestamp	Event Topic	User Name	Target ID	Description
cb3156ce-a324-f5ae-5dd1-66d1e59f11a6	04/28/20, 1:10 PM	N/A	N/A	endpoints	CREATE
efe21151-2906-dce2-14ab-68c17132d756	03/25/20, 4:22 PM	N/A	N/A	endpoints	CREATE
468e8b55-cf27-e77e-0179-1b5b736717b3	03/25/20, 10:12 AM	N/A	N/A	endpoints	CREATE
d9482883-d1ae-5899-fb06-852c202cc178	03/20/20, 2:41 PM	N/A	N/A	endpoints	CREATE
38584d40-p663-6311-7098-3747aa528d12	01/30/20, 5:35 PM	N/A	N/A	endpoints	CREATE

### 확장성 구독 생성

vRealize Automation Cloud Assembly와 함께 확장성 작업을 사용하거나 vRealize Orchestrator 통합을 사용하여 애플리케이션을 확장하는 구독을 생성할 수 있습니다.

확장성 구독을 사용하면 특정 수명주기 이벤트에서 워크플로 또는 작업을 트리거하여 애플리케이션을 확장할 수 있습니다. 또한 구독에 필터를 적용하여 지정된 이벤트에 대해 부울 조건을 설정할 수도 있습니다. 예를 들어 부울 식이 'true'인 경우에만 이벤트 및 워크플로 또는 작업이 트리거됩니다. 따라서, 이벤트, 작업 또는 워크플로가 트리거되는 시기를 제어하려는 시나리오에 유용합니다.

### 사전 요구 사항

- 클라우드 관리자 사용자 역할이 있어야 합니다.
- vRealize Orchestrator 워크플로를 사용하는 경우:
  - 내장된 vRealize Orchestrator 클라이언트의 라이브러리 또는 통합된 외부 vRealize Orchestrator 인스턴스의 라이브러리.
- 확장성 작업을 사용하는 경우:
  - 기존 확장성 작업 스크립트가 있어야 합니다. 자세한 내용은 [확장성 작업을 생성하는 방법](#) 항목을 참조하십시오.

### 절차

- 1 **확장성 > 구독**을 선택합니다.
- 2 **새 구독**을 클릭합니다.
- 3 구독의 세부 정보를 입력합니다.
- 4 **이벤트 항목**을 선택합니다.
- 5 (선택 사항) 이벤트 항목에 대한 조건을 설정합니다.

---

**참고** 조건은 JavaScript 구문 표현식을 사용하여 생성할 수 있습니다. 이 표현식에는 "&&" (AND), "||" (OR), "^" (XOR) 및 "!" (NOT) 같은 부울 연산자가 포함될 수 있습니다. 또한 "==" (equal to), "!=" (not equal to), ">=" (greater than or equal), "<=" (less than or equal), ">" (greater than) 및 "<" (lesser than) 같은 산술 연산자를 사용할 수 있습니다. 보다 복잡한 부울 표현식은 더 간단한 표현식으로 작성할 수 있습니다. 지정된 항목 매개 변수에 따라 이벤트의 페이로드 (데이터)에 액세스하려면 'event.data' 또는 이벤트의 헤더 속성( sourceType, sourceIdentity, timeStamp, eventType, eventTopicId, correlationType, correlationId, description, targetType, targetId, userName, orgId)을 사용합니다.

---

- 6 **작업/워크플로**에서 확장성 구독에 대한 실행 가능 항목을 선택합니다.
- 7 (선택 사항) 해당하는 경우 이벤트 항목에 대한 차단 동작을 구성합니다.
- 8 (선택 사항) 확장성 구독에 대한 프로젝트 범위를 정의하려면 **모든 프로젝트**를 사용하지 않도록 설정하고 **프로젝트 추가**를 클릭합니다.
- 9 구독을 저장하려면 **저장**을 클릭합니다.

## 결과

구독이 생성되었습니다. 선택된 이벤트 항목별로 분류된 이벤트가 발생하면 연결된 vRealize Orchestrator 워크플로 또는 확장성 작업이 시작되고 모든 구독자에게 알림이 제공됩니다.

## 다음에 수행할 작업

구독을 생성한 후에는 클라우드 템플릿을 생성 또는 배포하여 구독을 연결하고 사용할 수 있습니다.

vRealize Automation Cloud Assembly 내의 **확장성** 탭에서 워크플로 실행 상태를 확인할 수도 있습니다.

vRealize Orchestrator 워크플로를 포함하는 구독의 경우 vRealize Orchestrator 클라이언트에서 실행 및 워크플로 상태를 모니터링할 수도 있습니다.

## 확장성 구독 문제 해결

확장성 구독 실패 문제를 해결합니다.

구독이 실패하는 경우에는 일반적으로 워크플로 또는 확장성 작업 스크립트의 오류가 원인일 수 있습니다.

### 항목 매개 변수 및 페이로드 보기

덤프 구독 항목 매개 변수 스크립트를 사용하여 지정된 이벤트 단계에서 가상 시스템의 특정 매개 변수 및 페이로드를 볼 수 있습니다.

기본적으로 이 스크립트는 vRealize Orchestrator 워크플로에 사용 가능한 입력을 디버깅하고 확인하는데 유용합니다. 가상 시스템의 모든 매개 변수를 보려면 워크플로에 다음 스크립트를 사용합니다.

```
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)

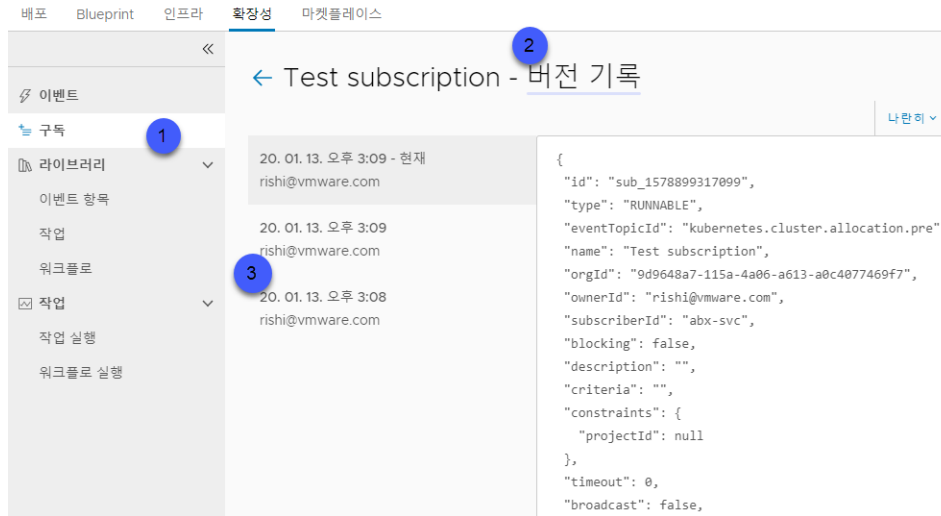
customProps = inputProperties.get("customProperties")
```

## 구독 버전 기록

구독이 실패하는 경우 버전 기록을 볼 수 있습니다.

### 구독 버전 기록 보기

[버전 기록] 탭은 사용자 및 변경 날짜와 함께 구독의 변경 기록을 보여줄 수 있습니다. 구독이 실패하거나 잘못 실행 중인 경우 버전 기록을 확인하면 원인을 식별하는 데 도움이 될 수 있습니다.



1

구독 탭에서 구독을 엽니다.

2

버전 기록을 보려면 **버전 기록**을 클릭합니다.

3

각 변경 항목을 클릭하면 변경과 연결된 해당 구독 코드를 볼 수 있습니다.

## vRealize Automation 리소스 속성 소개

vRealize Automation IaC(Infrastructure-as-Code) 편집기에서 마우스로 클릭하거나 가리켜서 구문 및 코드 완성에 필요한 도움을 받을 수 있습니다. 종종 사용자 지정 속성이라고도 하는 클라우드 템플릿 리소스 속성 집합을 보려면 통합된 리소스 스키마를 참조하십시오.

이 스키마는 VMware {code} 사이트에서 사용할 수 있습니다. 링크를 따르고 **모델**을 클릭하여 클라우드 템플릿(이전 이름: Blueprint)에 사용할 수 있는 리소스 개체를 나열합니다.

- [VMware {code}에 대한 vRealize Automation 리소스 유형 스키마](#)

## 일부 vRealize Automation Cloud Assembly 코드 예

vRealize Automation Cloud Assembly의 클라우드 템플릿 코드는 조합 및 응용 가능성에 제한이 거의 없습니다.

종종 성공적인 코드 예시는 더 많은 개발을 위한 최고의 출발점이 됩니다. 예시를 따를 때에는 리소스 이름, 값 등의 측면에서 사용자 사이트 설정을 적용하기 위해 대체가 필요합니다.

## vRealize Automation Cloud Assembly 클라우드 템플릿의 vSphere 리소스 예시

이러한 코드 예시는 vRealize Automation Cloud Assembly 클라우드 템플릿 내에서 vSphere 시스템 리소스를 보여줍니다.

리소스	클라우드 템플릿 예시
CPU, 메모리 및 운영 체제가 포함된 vSphere 가상 시스템	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 1       totalMemoryMB: 1024       image: ubuntu</pre>
데이터스토어 리소스가 포함된 vSphere 시스템	<pre>resources:   demo-vsphere-disk-001:     type: Cloud.vSphere.Disk     properties:       name: DISK_001       type: 'HDD'       capacityGb: 10       dataStore: 'datastore-01'       provisioningType: thick</pre>
연결된 디스크가 포함된 vSphere 시스템	<pre>resources:   demo-vsphere-disk-001:     type: Cloud.vSphere.Disk     properties:       name: DISK_001       type: HDD       capacityGb: 10       dataStore: 'datastore-01'       provisioningType: thin   demo-machine:     type: Cloud.vSphere.Machine     properties:       name: demo-machine       cpuCount: 2       totalMemoryMB: 2048       imageRef: &gt;-         https://bintray.com/vmware/photon/         download_file?file_path=2.0%2FRC%2Fova%2Fphoton-         custom-hw11-2.0-31bb961.ova       attachedDisks:         - source: '\${demo-vsphere-disk-001.id}'</pre>

**리소스****클라우드 템플릿 예시**

동적 디스크 수가 있는  
vSphere 시스템

```
inputs:
  disks:
    type: array
    title: disks
    items:
      title: disk
      type: object
      properties:
        size:
          type: integer
          title: size
    maxItems: 15
resources:
  Cloud_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: centos7
      flavor: small
      attachedDisks: '$
{map_to_object(resource.Cloud_Volume_1[*].id,
"source")}'
  Cloud_Volume_1:
    type: Cloud.Volume
    allocatePerInstance: true
    properties:
      capacityGb: '${input.disks[count.index].size}'
      count: '${length(input.disks)}'
```

스냅샷 이미지에 있는  
vSphere 시스템입니다.  
슬래시와 스냅샷 이름을  
추가합니다. 스냅샷 이  
미지는 연결된 클론일  
수 있습니다.

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      imageRef: 'demo-machine/snapshot-01'
      cpuCount: 1
      totalMemoryMB: 1024
```

vCenter의 특정 폴더에  
있는 vSphere 시스템

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      name: demo-machine
      cpuCount: 2
      totalMemoryMB: 1024
      imageRef: ubuntu
      resourceGroupName: 'myFolder'
```

**리소스****클라우드 템플릿 예시**

여러 NIC가 포함된  
vSphere 시스템

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      image: ubuntu
      flavor: small
      networks:
        - network: '${network-01.name}'
          deviceIndex: 0
        - network: '${network-02.name}'
          deviceIndex: 1
    network-01:
      type: Cloud.vSphere.Network
      properties:
        name: network-01
    network-02:
      type: Cloud.vSphere.Network
      properties:
        name: network-02
```

vCenter의 연결된 태그  
가 포함된 vSphere 시  
스템

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      flavor: small
      image: ubuntu
      tags:
        - key: env
          value: demo
```

**리소스****클라우드 템플릿 예시**

사용자 지정 규격이 포함된 vSphere 시스템

```
resources:
  demo-machine:
    type: Cloud.vSphere.Machine
    properties:
      name: demo-machine
      image: ubuntu
      flavor: small
      customizationSpec: Linux
```

원격 액세스가 포함된 vSphere 시스템

```
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}
```

## 문서화된 vRealize Automation Cloud Assembly 클라우드 템플릿 예

이 예에서는 상세한 주석 집합을 포함하여 vRealize Automation Cloud Assembly 템플릿(이전 명칭: Blueprint)에 있는 섹션의 구조 및 용도를 검토할 수 있습니다.

```
# *****
#
# This WordPress cloud template is enhanced with comments to explain its
# parameters.
```

```

#
# Try cloning it and experimenting with its YAML code. If you're new to
# YAML, visit yaml.org for general information.
#
# The cloud template deploys a minimum of 3 virtual machines and runs scripts
# to install packages.
#
# *****
#
# -----
# Templates need a descriptive name and version if
# source controlled in git.
# -----
name: WordPress Template with Comments
formatVersion: 1
version: 1
#
# -----
# Inputs create user selections that appear at deployment time. Inputs
# can set placement decisions and configurations, and are referenced
# later, by the resources section.
# -----
inputs:
#
# -----
# Choose a cloud endpoint. 'Title' is the visible
# option text (oneOf allows for the friendly title). 'Const' is the
# tag that identifies the endpoint, which was set up earlier, under the
# Cloud Assembly Infrastructure tab.
# -----
platform:
  type: string
  title: Deploy to
  oneOf:
    - title: AWS
      const: aws
    - title: Azure
      const: azure
    - title: vSphere
      const: vsphere
  default: vsphere
#
# -----
# Choose the operating system. Note that the Cloud Assembly
# Infrastructure must also have an AWS, Azure, and vSphere Ubuntu image
# mapped. In this case, enum sets the option that you see, meaning there's
# no friendly title feature this time. Also, only Ubuntu is available
# here, but having this input stubbed in lets you add more operating
# systems later.
# -----
osimage:
  type: string
  title: Operating System
  description: Which OS to use
  enum:

```

```

- Ubuntu

#
# -----
# Set the number of machines in the database cluster. Small and large
# correspond to 1 or 2 machines, respectively, which you see later,
# down in the resources section.
# -----
dbenvsize:
  type: string
  title: Database cluster size
  enum:
    - Small
    - Large

#
# -----
# Dynamically tag the machines that will be created. The
# 'array' of objects means you can create as many key-value pairs as
# needed. To see how array input looks when it's collected,
# open the cloud template and click TEST.
# -----
Mtags:
  type: array
  title: Tags
  description: Tags to apply to machines
  items:
    type: object
    properties:
      key:
        type: string
        title: Key
      value:
        type: string
        title: Value

#
# -----
# Create machine credentials. These credentials are needed in
# remote access configuration later, in the resources section.
# -----
username:
  type: string
  minLength: 4
  maxLength: 20
  pattern: '[a-z]+'
  title: Database Username
  description: Database Username
userpassword:
  type: string
  pattern: '[a-z0-9A-Z@#&$]+'
  encrypted: true
  title: Database Password
  description: Database Password

#
# -----
# Set the database storage disk size.
# -----

```

```

databaseDiskSize:
  type: number
  default: 4
  maximum: 10
  title: MySQL Data Disk Size
  description: Size of database disk
#
# -----
# Set the number of machines in the web cluster. Small, medium, and large
# correspond to 2, 3, and 4 machines, respectively, which you see later,
# in the WebTier part of the resources section.
# -----
clusterSize:
  type: string
  enum:
    - small
    - medium
    - large
  title: Wordpress Cluster Size
  description: Wordpress Cluster Size
#
# -----
# Set the archive storage disk size.
# -----
archiveDiskSize:
  type: number
  default: 4
  maximum: 10
  title: Wordpress Archive Disk Size
  description: Size of Wordpress archive disk
#
# -----
# The resources section configures the deployment of machines, disks,
# networks, and other objects. In several places, the code pulls from
# the preceding interactive user inputs.
# -----
resources:
#
# -----
# Create the database server. Choose a cloud agnostic machine 'type' so
# that it can deploy to AWS, Azure, or vSphere. Then enter its property
# settings.
# -----
  DBTier:
    type: Cloud.Machine
    properties:
#
# -----
# Descriptive name for the virtual machine. Does not become the hostname
# upon deployment.
# -----
    name: mysql
#
# -----
# Hard-coded operating system image to use. To pull from user input above,

```

```

# enter the following instead.
# image: '${input.osimage}'
# -----
#     image: Ubuntu
#
# -----
# Hard-coded capacity to use. Note that the Cloud Assembly
# Infrastructure must also have AWS, Azure, and vSphere flavors
# such as small, medium, and large mapped.
# -----
#     flavor: small
#
# -----
# Tag the database machine to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with a site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
#     constraints:
#       - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Also tag the database machine with any free-form tags that were created
# during user input.
# -----
#     tags: '${input.Mtags}'
#
# -----
# Set the database cluster size by referencing the dbenvsize user
# input. Small is one machine, and large defaults to two.
# -----
#     count: '${input.dbenvsize == "Small" ? 1 : 2}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
#     networks:
#       - network: '${resource.WP_Network.id}'
#
# -----
# Enable remote access to the database server. Reference the credentials
# from the user input.
# -----
#     remoteAccess:
#       authentication: usernamePassword
#       username: '${input.username}'
#       password: '${input.userpassword}'
#
# -----
# You are free to add custom properties, which might be used to initiate
# an 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: '${env.requestedBy}'
#
# -----
# You are free to add custom properties, which might be used to initiate
# an extensibility subscription, for example.
# -----
#     ABC-Company-ID: 9393
#
# -----
# Tag the web server to deploy to the cloud vendor chosen from the
# user input. Tags are case-sensitive, so 'to_lower' forces the tag to
# lowercase to ensure a match with your site's tagging convention. It's
# important if platform input were to contain any upper case characters.
# -----
#     constraints:
#       - tag: '${"env:" + to_lower(input.platform)}'
#
# -----
# Add a variable to connect the machine to a network resource based on
# a property binding to another resource. In this case, it's the
# 'WP_Network' network that gets defined further below.
# -----
#     networks:
#       - network: '${resource.WP_Network.id}'
#
# -----
# Run OS commands or scripts to further configure the web server,
# with operations such as setting a hostname, generating SSH private keys,
# or installing packages.
# -----
#     cloudConfig: |
#       #cloud-config
#       repo_update: true
#       repo_upgrade: all
#       packages:
#         - apache2
#         - php
#         - php-mysql
#         - libapache2-mod-php
#         - php-mcrypt
#         - mysql-client
#       runcmd:
#         - mkdir -p /var/www/html/mywordpresssite && cd /var/www/html && wget https://
wordpress.org/latest.tar.gz && tar -xzf /var/www/html/latest.tar.gz -C /var/www/html/
mywordpresssite --strip-components 1
#         - i=0; while [ $i -le 5 ]; do mysql --connect-timeout=3 -h $
{DBTier.networks[0].address} -u root -pmysqlpassword -e "SHOW STATUS;" && break || sleep 15;
i=$((i+1)); done
#         - mysql -u root -pmysqlpassword -h ${resource.DBTier.networks[0].address} -e
"create database wordpress_blog;"

```

```

- mv /var/www/html/mywordpresssite/wp-config-sample.php /var/www/html/
mywordpresssite/wp-config.php
- sed -i -e s/"define('DB_NAME', 'database_name_here');"/"define('DB_NAME',
'wordpress_blog');"/ /var/www/html/mywordpresssite/wp-config.php && sed -i -e
s/"define('DB_USER', 'username_here');"/"define('DB_USER', 'root');"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define('DB_PASSWORD',
'password_here');"/"define('DB_PASSWORD', 'mysqlpassword');"/ /var/www/html/
mywordpresssite/wp-config.php && sed -i -e s/"define('DB_HOST',
'localhost');"/"define('DB_HOST', '${resource.DBTier.networks[0].address}');"/ /var/www/html/
mywordpresssite/wp-config.php
- service apache2 reload

#
# -----
# Create the network that the database and web servers connect to.
# Choose a cloud agnostic network 'type' so that it can deploy to AWS,
# Azure, or vSphere. Then enter its property settings.
# -----
WP_Network:
  type: Cloud.Network
  properties:
#
# -----
# Descriptive name for the network. Does not become the network name
# upon deployment.
# -----
    name: WP_Network
#
# -----
# Set the networkType to an existing network. You could also use a
# constraint tag to target a specific, like-tagged network.
# The other network types are private or public.
# -----
    networkType: existing
#
# *****
#
# VMware hopes that you found this commented template useful. Note that
# you can also access an API to create templates, or query for input
# schema that you intend to request. See the following Swagger
# documentation.
#
# www.mgmt.cloud.vmware.com/blueprint/api/swagger/swagger-ui.html
#
# *****

```

## vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예

클라우드 템플릿 설계 및 배포에서 네트워킹, 보안 및 로드 밸런서 리소스와 설정을 사용할 수 있습니다.

클라우드 템플릿 설계 코드 옵션의 요약은 [vRealize Automation 리소스 유형 스키마](#)를 참조하십시오.

관련 정보는 다음을 참조하십시오.

- [vRealize Automation 클라우드 템플릿에서 네트워크 리소스 사용](#)
- [vRealize Automation 클라우드 템플릿에서 보안 그룹 리소스 사용](#)
- [vRealize Automation 클라우드 템플릿에서 로드 밸런서 리소스 사용](#)

이러한 예에서는 기본 클라우드 템플릿 설계 내의 샘플 네트워크, 보안 그룹 및 로드 밸런서 리소스를 보여 줍니다.

리소스 시나리오	예제 클라우드 템플릿 설계 코드
NSX 네트워크 리소스에 연결된 여러 NIC가 있는 vSphere 시스템입니다.	<pre>resources:   demo-machine:     type: Cloud.vSphere.Machine     properties:       image: ubuntu       flavor: small       networks:         - network: '\$ {resource.Cloud_vSphere_Network_1.id}'   Cloud_vSphere_Network_1:     type: Cloud.vSphere.Network     properties:       networkType: existing   Cloud_vSphere_Network_2:     type: Cloud.NSX.Network     properties:       networkType: existing</pre>
Azure VM 배포를 위한 고정 IP 주소가 있는 전용 네트워크 추가	<pre>formatVersion: 1 inputs: {} resources:   Cloud_Azure_Machine_1:     type: Cloud.Azure.Machine     properties:       image: photon       flavor: Standard_B1ls       networks:         - network: '\$ {resource.Cloud_Network_1.id}'       assignment: static       address: 10.0.0.45       assignPublicIpAddress: false   Cloud_Network_1:     type: Cloud.Network     properties:       networkType: existing</pre>

## 리소스 시나리오

기존 배포를 위한  
Cloud.NSX.NAT 리소스에서  
NAT 및 DNAT 포트 포워딩 규칙  
을 추가하거나 편집합니다.

## 예제 클라우드 템플릿 설계 코드

```
resources:
  gw:
    type: Cloud.NSX.NAT
    properties:
      networks:
        - '${resource.akout.id}'
      natRules:
        - translatedInstance: '${
{resource.centos.networks[0].id}'
          index: 0
          protocol: TCP
          kind: NAT44
          type: DNAT
          sourceIPs: any
          sourcePorts: 80
          translatedPorts: 8080
          destinationPorts: 8080
          description: edit
        - translatedInstance: '${
{resource.centos.networks[0].id}'
          index: 1
          protocol: TCP
          kind: NAT44
          type: DNAT
          sourceIPs: any
          sourcePorts: 90
          translatedPorts: 9090
          destinationPorts: 9090
          description: add
      centos:
        type: Cloud.vSphere.Machine
        properties:
          image: WebTinyCentOS65x86
          flavor: small
          customizationSpec: Linux
          networks:
            - network: '${resource.akout.id}'
              assignment: static
      akout:
        type: Cloud.NSX.Network
        properties:
          networkType: outbound
          constraints:
            - tag: nsxt-nat-1-M2
```

로드 밸런싱 로깅 수준, 알고리즘  
및 크기를 지정합니다.

로깅 수준, 알고리즘 및 크기의 사용을 보여 주는 샘플 NSX 로드 밸  
런서:

```
resources:
  Cloud_LoadBalancer_1:
    type: Cloud.NSX.LoadBalancer
    properties:
      name: myapp-lb
      network: '${appnet-public.name}'
      instances: '${wordpress.id}'
      routes:
        - protocol: HTTP port: '80'
          loggingLevel: CRITICAL
          algorithm: LEAST_CONNECTION
          type: MEDIUM
```

## 리소스 시나리오

로드 밸런서를 명명된 시스템 또는 명명된 시스템 NIC와 연결합니다. machine ID 또는 machine network ID를 지정하여 시스템을 로드 밸런서 풀에 추가할 수 있습니다. 인스턴스 속성은 시스템(machine by ID) 및 NIC(machine by network ID)를 모두 지원합니다.

첫 번째 예에서 배포는 machine by ID 설정을 사용하여 시스템이 네트워크에 배포된 경우 시스템을 로드 밸런싱합니다.

두 번째 예에서 배포는 machine by network ID 설정을 사용하여 시스템이 명명된 시스템 NIC에 배포된 경우에만 시스템을 로드 밸런싱합니다.

세 번째 예에서는 동일한 instances 옵션에 사용되는 두 설정을 보여 줍니다.

NSX 로드 밸런서에 상태 점검 설정을 추가합니다. 추가 옵션에는 httpMethod, requestBody 및 responseBody가 포함됩니다.

## 예제 클라우드 템플릿 설계 코드

instances 속성을 사용하여 시스템 ID 또는 시스템 네트워크 ID를 정의할 수 있습니다.

### ■ 시스템 ID

```
Cloud_LoadBalancer_1:
  type: Cloud.LoadBalancer
  properties:
    network: '${resource.Cloud_Network_1.id}'
    instances: '$
{resource.Cloud_Machine_1.id}'
```

### ■ 시스템 네트워크 ID

```
Cloud_LoadBalancer_1:
  type: Cloud.LoadBalancer
  properties:
    network: '${resource.Cloud_Network_1.id}'
    instances: '$
{resource.Cloud_Machine_1.networks[0].id}'
```

### ■ 로드 밸런서 포함을 위해 지정된 시스템 하나와 로드 밸런서 포함을 위해 지정된 다른 시스템 NIC:

```
instances:
  - resource.Cloud_Machine_1.id
  - resource.Cloud_Machine_2.networks[2].id
```

```
myapp-lb:
  type: Cloud.NSX.LoadBalancer
  properties:
    name: myapp-lb
    network: '${appnet-public.name}'
    instances: '${wordpress.id}'
    routes:
      - protocol: HTTP
        port: '80'
        algorithm: ROUND_ROBIN
        instanceProtocol: HTTP
        instancePort: '80'
    healthCheckConfiguration:
      protocol: HTTP
      port: '80'
      urlPath: /mywordpresssite/wp-admin/
      install.php
      intervalSeconds: 60
      timeoutSeconds: 10
      unhealthyThreshold: 10
      healthyThreshold: 2
      connectionLimit: '50'
      connectionRateLimit: '50'
      maxConnections: '500'
      minConnections: ''
      internetFacing: true{code}
```

리소스 시나리오	예제 클라우드 템플릿 설계 코드
<p>공용 IP 대신 내부 IP를 사용하는 공용 클라우드 시스템. 이 예에서는 특정 네트워크 ID를 사용합니다.</p> <p>참고: <code>networks</code>: 설정에서 <code>network</code>: 옵션을 사용하여 대상 네트워크 ID를 지정합니다.</p> <p><code>networks</code>: 설정의 <code>name</code>: 옵션은 폐기되었으며 사용하면 안 됩니다.</p>	<pre>resources:   wf_proxy:     type: Cloud.Machine     properties:       image: ubuntu 16.04       flavor: small       constraints:         - tag: 'platform:vsphere'     networks:       - network: '\${resource.wf_net.id}'         assignPublicIpAddress: false</pre>
<p>NSX 네트워크 리소스 유형을 사용하여 NSX-V 또는 NSX-T에 대해 라우팅된 네트워크입니다.</p>	<pre>Cloud_NSX_Network_1:   type: Cloud.NSX.Network   properties:     networkType: routed</pre>
<p>클라우드 템플릿의 시스템 NIC 리소스에 태그를 추가합니다.</p>	<pre>formatVersion: 1 inputs: {} resources:   Cloud_Machine_1:     type: Cloud.vSphere.Machine     properties:       flavor: small       image: ubuntu     networks:       - name: '\${resource.Cloud_Network_1.name}'         deviceIndex: 0         tags:           - key: 'nic0'             value: null           - key: internal             value: true       - name: '\${resource.Cloud_Network_2.name}'         deviceIndex: 1         tags:           - key: 'nic1'             value: null           - key: internal             value: false</pre>
<p>아웃바운드 네트워크에 대한 NSX-T 논리적 스위치에 태그를 지정합니다.</p> <p>태그 지정은 NSX-T 및 VMware Cloud on AWS에 대해 지원됩니다.</p> <p>이 시나리오에 대한 자세한 내용은 커뮤니티 블로그 게시물 <a href="#">Creating Tags in NSX with Cloud Assembly</a>를 참조하십시오.</p>	<pre>Cloud_NSX_Network_1:   type: Cloud.NSX.Network   properties:     networkType: outbound     tags:       - key: app         value: opencart</pre>

## 리소스 시나리오

## 예제 클라우드 템플릿 설계 코드

시스템 NIC에 제약 조건 태그가 적용된 기존 보안 그룹입니다.

기존 보안 그룹을 사용하려면 `securityGroupType` 속성에 *existing*을 입력합니다.

태그 제약 조건을 사용하여 기존 보안 그룹을 할당하기 위해

`Cloud.SecurityGroup` 리소스에 태그를 할당할 수 있습니다. 태그가 포함되지 않은 보안 그룹은 클라우드 템플릿 설계에서 사용할 수 없습니다.

`securityGroupType:`  
`existing` 보안 그룹 리소스에 대해 제약 조건 태그를 설정해야 합니다. 이러한 제약 조건은 기존 보안 그룹에 설정된 태그와 일치해야 합니다.

`securityGroupType: new` 보안 그룹 리소스에 대해 제약 조건 태그를 설정할 수 없습니다.

```
formatVersion: 1
inputs: {}
resources:
  allowSsh_sg:
    type: Cloud.SecurityGroup
    properties:
      securityGroupType: existing
      constraints:
        - tag: allowSsh
  compute:
    type: Cloud.Machine
    properties:
      image: centos
      flavor: small
      networks:
        - network: '${resource.prod-net.id}'
          securityGroups:
            - '${resource.allowSsh_sg.id}'
  prod-net:
    type: Cloud.Network
    properties:
      networkType: existing
```

## 리소스 시나리오

Allow 및 Deny 액세스 옵션을 설명하는 2개의 방화벽 규칙이 포함된 주문형 보안 그룹입니다.

## 예제 클라우드 템플릿 설계 코드

```
resources:
  Cloud_SecurityGroup_1:
    type: Cloud.SecurityGroup
    properties:
      securityGroupType: new
      rules:
        - ports: 5000
          source:
            'fc00:10:000:000:000:56ff:fe89:48b4'
            access: Allow
            direction: inbound
            name: allow_5000
            protocol: TCP
        - ports: 7000
          source:
            'fc00:10:000:000:000:56ff:fe89:48b4'
            access: Deny
            direction: inbound
            name: deny_7000
            protocol: TCP
  Cloud_vSphere_Machine_1:
    type: Cloud.vSphere.Machine
    properties:
      image: photon
      cpuCount: 1
      totalMemoryMB: 256
      networks:
        - network: '$
{resource.Cloud_Network_1.id}'
          assignIPv6Address: true
          assignment: static
          securityGroups:
            - '$
{resource.Cloud_SecurityGroup_1.id}'
  Cloud_Network_1:
    type: Cloud.Network
    properties:
      networkType: existing
```

## 리소스 시나리오

다음에 포함하여 2개의 보안 그룹이 있는 복잡한 클라우드 템플릿:

- 기존 보안 그룹 1개
- 여러 방화벽 규칙 예가 있는 주문형 보안 그룹 1개
- vSphere 시스템 1개
- 기존 네트워크 1개

이 샘플은 다양한 조합의 프로토콜 및 포트, 서비스, IP CIDR(소스 및 대상으로), IP 범위(소스 또는 대상으로)를 비롯해 임의, IPv6 및 (::/0)에 대한 옵션을 설명합니다.

시스템 NIC의 경우 연결된 네트워크 및 보안 그룹을 지정할 수 있습니다. NIC 인덱스 또는 IP 주소를 지정할 수도 있습니다.

## 예제 클라우드 템플릿 설계 코드

```
formatVersion: 1
inputs: {}
resources:
  DEMO_ESG : "existing security group -
security group 1)"
    type: Cloud.SecurityGroup
    properties:
      constraints:
        - tag: BlockAll
      securityGroupType: existing
    ( "designation of existing for security group
1" )
  DEMO_ODSG: ( "on-demand security group -
security group 2" )
    type: Cloud.SecurityGroup
    properties:
      rules: ( "multiple firewall rules in this
section" )
        - name: IN-ANY ( "rule 1" )
          source: any
          service: any
          direction: inbound
          access: Deny
        - name: IN-SSH ( "rule 2" )
          source: any
          service: SSH
          direction: inbound
          access: Allow
        - name: IN-SSH-IP ( "rule 3" )
          source: 33.33.33.1-33.33.33.250
          protocol: TCP
          ports: 223
          direction: inbound
          access: Allow
        - name: IPv-6-ANY-SOURCE ( "rule 4" )
          source: ':::/0'
          protocol: TCP
          ports: 223
          direction: inbound
          access: Allow
        - name: IN-SSH-IP ( "rule 5" )
          source: 44.44.44.1/24
          protocol: UDP
          ports: 22-25
          direction: inbound
          access: Allow
        - name: IN-EXISTING-SG ( "rule 6" )
          source: '${resource["DEMO_ESG"].id}'
          protocol: ICMPv6
          direction: inbound
          access: Allow
        - name: OUT-ANY ( "rule 7" )
          destination: any
          service: any
          direction: outbound
          access: Deny
        - name: OUT-TCP-IPv6 ( "rule 8" )
          destination:
'2001:0db8:85a3::8a2e:0370:7334/64'
          protocol: TCP
          ports: 22
          direction: outbound
```

## 리소스 시나리오

## 예제 클라우드 템플릿 설계 코드

```

        access: Allow
        - name: IPv6-ANY-DESTINATION ( "rule
9" )
        destination: '::/0'
        protocol: UDP
        ports: 23
        direction: outbound
        access: Allow
        - name: OUT-UDP-SERVICE ( "rule 10" )
        destination: any
        service: NTP
        direction: outbound
        access: Allow
        securityGroupType: new ( "designation of
on-demand for security group 2" )
        DEMO_VC_MACHINE: ( "machine resource" )
        type: Cloud.vSphere.Machine
        properties:
            image: PHOTON
            cpuCount: 1
            totalMemoryMB: 1024
        networks: ( "Machine network NICs" )
        - network: '${resource.DEMO_NW.id}'
          securityGroups:
            - '${resource.DEMO_ODSG.id}'
            - '${resource.DEMO_ESG.id}'
        DEMO_NETWORK: ( "network resource" )
        type: Cloud.vSphere.Network
        properties:
            networkType: existing
            constraints:
                - tag: nsx62

```

**리소스 시나리오****예제 클라우드 템플릿 설계 코드**

원 암(one arm) 로드 밸런서가 있는 주문형 네트워크입니다.

```
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
```

로드 밸런서가 있는 기존 네트워크입니다.

```
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:
```

## 리소스 시나리오

## 예제 클라우드 템플릿 설계 코드

```

- 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

```

## 자세히 알아보기

네트워크 및 보안 구현 시나리오의 예는 다음과 같은 VMware 블로그를 참조하십시오.

- [NSX-T를 사용하는 vRealize Automation Cloud Assembly 로드 밸런서 심화 연구](#)
- [Cloud Assembly 및 NSX를 통한 네트워크 자동화 - 1부\(NSX-T, vCenter 클라우드 계정 및 네트워크 CIDR 포함\)](#)
- [Cloud Assembly 및 NSX를 통한 네트워크 자동화 - 2부\(기본 및 아웃바운드 네트워크 유형 사용 포함\)](#)
- [Cloud Assembly 및 NSX를 통한 네트워크 자동화 - 3부\(기본 및 주문형 보안 그룹 사용 포함\)](#)
- [Cloud Assembly 및 NSX를 통한 네트워크 자동화 - 4부\(기본 및 주문형 로드 밸런서 사용 포함\)](#)

## vRealize Automation 클라우드 템플릿에서 네트워크 리소스 사용

vRealize Automation 클라우드 템플릿 설계를 생성하거나 편집할 때 목적에 가장 적합한 네트워크 리소스를 사용합니다. 클라우드 템플릿에서 사용할 수 있는 NSX 및 클라우드 애그노스틱 네트워크 옵션에 대해 알아봅니다.

vRealize Automation 클라우드 템플릿 설계에서 시스템 및 관련 조건을 기반으로 사용 가능한 네트워크 리소스 유형 중 하나를 선택합니다.

## 클라우드 애그노스틱 네트워크 리소스

클라우드 템플릿 **설계** 페이지의 **클라우드 애그노스틱 > 네트워크** 리소스를 사용하여 클라우드 애그노스틱 네트워크를 추가합니다. 리소스가 클라우드 템플릿 코드에 `Cloud.Network` 리소스 유형으로 표시됩니다. 기본 리소스는 다음과 같이 표시됩니다.

```
Cloud_Network_1:
  type: Cloud.Network
  properties:
    networkType: existing
```

NSX 네트워크에 연결되지 않았거나 연결되지 않았을 수 있는 대상 시스템 유형에 대한 네트워킹 특성을 지정하려는 경우 클라우드 애그노스틱 네트워크를 사용합니다.

다음 리소스 유형에 대해 클라우드 애그노스틱 네트워크 리소스를 사용할 수 있습니다.

- 클라우드 애그노스틱 시스템
- vSphere
- GCP(Google Cloud Platform)
- AWS(Amazon Web Services)
- Microsoft Azure
- VMC(VMware Cloud on AWS)

다음 네트워크 유형(networkType) 설정에 대해 클라우드 애그노스틱 네트워크 리소스를 사용할 수 있습니다.

- 공용
- 전용
- 아웃바운드
- 기존

## vSphere 네트워크 리소스

클라우드 템플릿 **설계** 페이지의 **vSphere > 네트워크** 리소스를 사용하여 vSphere 네트워크를 추가합니다. 리소스가 클라우드 템플릿 코드에 `Cloud.vSphere.Network` 리소스 유형으로 표시됩니다. 기본 리소스는 다음과 같이 표시됩니다.

```
Cloud_vSphere_Network_1:
  type: Cloud.vSphere.Network
  properties:
    networkType: existing
```

vSphere 시스템 유형(`Cloud.vSphere.Machine`)에 대한 네트워킹 특성을 지정하려는 경우 vSphere 네트워크를 사용합니다.

vSphere 네트워크 리소스는 `Cloud.vSphere.Machine` 시스템 유형에 대해서만 사용할 수 있습니다.

다음 네트워크 유형(networkType) 설정에 대해 vSphere 리소스를 사용할 수 있습니다.

- 공용
- 전용
- 기존

예제에 대해서는 vRealize Automation에서 클라우드 템플릿 및 네트워크 프로파일의 네트워크 설정 사용 항목을 참조하십시오.

## NSX 네트워크 리소스

클라우드 템플릿 **설계** 페이지의 **NSX > 네트워크** 리소스를 사용하여 NSX 네트워크를 추가합니다. 리소스가 클라우드 템플릿 코드에 Cloud.NSX.Network 리소스 유형으로 표시됩니다. 기본 리소스는 다음과 같이 표시됩니다.

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

NSX-V 또는 NSX-T 클라우드 계정에 연결된 하나 이상의 시스템에 네트워크 리소스를 연결하려는 경우 NSX 네트워크를 사용합니다. NSX 네트워크 리소스를 사용하면 NSX-V 또는 NSX-T 클라우드 계정에 연결된 vSphere 시스템 리소스에 대한 NSX 네트워킹 특성을 지정할 수 있습니다.

다음 네트워크 유형(networkType) 설정에 대해 Cloud.NSX.Network 리소스를 사용할 수 있습니다.

- 공용
- 전용
- 아웃바운드
- 기존
- 라우팅됨 - 라우팅된 네트워크는 NSX-V 및 NSX-T에만 사용할 수 있습니다.

여러 개의 아웃바운드 또는 라우팅된 네트워크에서 동일한 NSX-T Tier-1 라우터 또는 NSX-V ESG(Edge 서비스 게이트웨이)를 공유하도록 하려면 초기 배포 전에 템플릿의 연결된 네트워크에 단일 NSX 게이트웨이 리소스(Cloud.NSX.Gateway)를 연결합니다. 배포 후에 게이트웨이를 2일차 또는 반복적인 개발 작업으로 추가하면 각 네트워크가 자체 라우터를 생성합니다.

템플릿에서 NSX NAT 리소스를 사용하여 NAT 및 DNAT 포트 포워딩 규칙을 지원할 수 있습니다.

## Azure, AWS 또는 GCP 배포 의도가 있는 클라우드 애그노스틱 네트워크 리소스

공용 클라우드 제공자 VM에는 NSX 또는 vSphere 기반 시스템 배포에 반드시 필요한 것은 아닌 특정 클라우드 템플릿 속성 조합이 필요할 수 있습니다. 이러한 시나리오 중 일부를 지원하는 클라우드 템플릿 코드의 예는 vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예에서 참조하십시오.

## NSX 게이트웨이 리소스

클라우드 템플릿에서 게이트웨이 리소스(Cloud.NSX.Gateway)를 사용하여 단일 배포 내에서 단일 NSX-T Tier-1 라우터 또는 NSX-V ESG(Edge 서비스 게이트웨이)를 재사용하거나 공유할 수 있습니다. 게이트웨이 리소스는 Tier-1 또는 ESG를 나타내며 배포의 여러 네트워크에 연결할 수 있습니다. 게이트웨이 리소스는 아웃바운드 또는 라우팅된 네트워크에서만 사용할 수 있습니다.

Cloud.NSX.Gateway 리소스를 사용하면 배포에서 연결된 아웃바운드 네트워크 또는 라우팅된 네트워크 간에 NSX-T Tier-1 라우터 또는 NSX-V ESG(Edge 서비스 게이트웨이)를 공유할 수 있습니다.

게이트웨이는 주로 단일 아웃바운드 또는 라우팅된 네트워크에 연결됩니다. 그러나 게이트웨이가 여러 네트워크에 연결된 경우 이러한 네트워크는 동일한 유형이어야 합니다(예를 들어 모두 아웃바운드 네트워크이거나 모두 라우팅된 네트워크여야 함). 게이트웨이는 동일한 아웃바운드 또는 라우팅된 네트워크에 연결된 여러 시스템 또는 로드 밸런서에 연결될 수 있습니다. 게이트웨이에서 생성된 NSX-T Tier-1 라우터 또는 NSX-V ESG(Edge 서비스 게이트웨이)를 재사용할 수 있도록 게이트웨이를 공유된 주문형 네트워크의 로드 밸런서에 연결해야 합니다.

여러 개의 아웃바운드 또는 라우팅된 네트워크가 동일한 T1 라우터 또는 Edge를 공유하도록 허용하려면 먼저 모든 네트워크에 단일 Cloud.NSX.Gateway 게이트웨이 리소스를 연결합니다. 클라우드 템플릿을 배포하기 전에 모든 대상 네트워크 및 단일 게이트웨이를 함께 연결해야 합니다. 그렇지 않으면 각 네트워크가 자체 라우터를 생성합니다.

연결된 계산 게이트웨이 리소스가 포함된 NSX 네트워크의 경우, 게이트웨이 설정이 배포의 모든 연결된 네트워크에 적용됩니다. 단일 NSX-T Tier-1 논리적 라우터가 각 배포에 대해 생성되고 배포의 모든 주문형 네트워크 및 로드 밸런서에서 공유됩니다. 각 배포에 대해 단일 NSX-V Edge가 생성되고 배포의 모든 주문형 네트워크 및 로드 밸런서에서 공유됩니다.

게이트웨이 리소스를 네트워크에 반복적인 배포 업데이트로 연결할 수 있습니다. 그러나 그렇게 하면 T1 또는 Edge 라우터가 생성되지 않고 초기 네트워크 배포에서 라우터가 생성됩니다.

연결된 게이트웨이 리소스를 사용하지 않는 NSX-T 네트워크의 경우, 클라우드 템플릿의 여러 주문형 네트워크가 배포에 여러 Tier-1 논리적 라우터를 계속해서 생성합니다.

게이트웨이에 NAT 규칙이 포함된 경우 Tier-1 라우터 또는 Edge 라우터에 대한 NAT 또는 DNAT 규칙을 재구성하거나 삭제할 수 있습니다. 게이트웨이가 NAT 규칙 없이 처음 배포되는 경우 2일차 작업을 사용할 수 없습니다. 를 사용합니다.

## NSX NAT 리소스

Cloud.NSX.NAT 리소스를 사용하면 게이트웨이 리소스를 통해 연결된 모든 아웃바운드 네트워크에 DNAT 규칙 및 포트 포워딩을 연결할 수 있습니다. DNAT 규칙을 구성해야 하는 게이트웨이 리소스에 NAT 리소스를 연결할 수 있습니다.

**참고** Cloud.NSX.Gateway 리소스는 원래 DNAT 규칙에 사용할 수 있었습니다. 그러나 DNAT 규칙 및 포트 포워딩을 정의하기 위한 수단으로서의 Cloud.NSX.Gateway 사용은 폐기되었습니다. 이전 버전과의 호환성을 위해 가용 상태는 유지됩니다. DNAT 규칙 및 포트 포워딩에는 Cloud.NSX.NAT 클라우드 템플릿 리소스를 사용합니다. NAT 규칙 규칙에 Cloud.NSX.Gateway 리소스 유형을 사용하려고 하면 클라우드 템플릿에 주의가 표시됩니다.

Cloud.NSX.NAT 리소스는 아웃바운드 NSX-V 또는 NSX-T 네트워크에 연결된 경우 DNAT 규칙 및 포트 포워딩을 지원합니다.

리소스의 NAT 규칙 설정은 natRules:입니다. 게이트웨이 리소스에 NAT 리소스를 연결하여 게이트웨이에서 natRules: 항목을 구성할 수 있습니다. 리소스에 지정된 DNAT 규칙은 연결된 시스템 또는 로드 밸런서를 해당 대상으로 사용합니다.

DNAT 포트 포워딩 규칙을 추가, 재정렬, 편집 또는 삭제함으로써 기존 배포에서 시스템 NIC 또는 계산 게이트웨이를 재구성하여 해당 natRules: 설정을 수정할 수 있습니다. DNAT 규칙은 클러스터링된 시스템과 함께 사용할 수 없습니다. 2일차 작업의 일부로 클러스터 내의 개별 시스템에 대해 DNAT 규칙을 지정할 수 있습니다.

## 외부 IPAM 통합 옵션

클라우드 템플릿 설계 및 배포에서 Infloblox IPAM 통합에 사용할 수 있는 속성에 대한 정보는 [vRealize Automation 클라우드 템플릿에서 IPAM 통합을 위해 Infloblox 특정 속성 및 확장 가능한 특성 사용의 내용](#)을 참조하십시오.

## 사용 가능한 2일차 작업

클라우드 템플릿 및 배포 리소스에 사용할 수 있는 일반 2일차 작업의 목록은 [vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업의 내용](#)을 참조하십시오.

특정 네트워크에서 다른 네트워크로 이동하는 방법에 대한 예는 [배포된 시스템을 다른 네트워크로 이동하는 방법의 내용](#)을 참조하십시오.

## 자세히 알아보기

관련 정보 및 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#) 항목을 참조하십시오.

네트워크 리소스 정의에 대한 자세한 내용은 [vRealize Automation의 네트워크 리소스의 내용](#)을 참조하십시오.

네트워크 프로파일 정의에 대한 자세한 내용은 [vRealize Automation의 네트워크 프로파일에 대해 알아보기의 내용](#)을 참조하십시오.

샘플 네트워크 리소스 및 설정을 보여 주는 클라우드 템플릿 설계에 대한 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)의 내용을 참조하십시오.

## vRealize Automation 클라우드 템플릿에서 보안 그룹 리소스 사용

vRealize Automation 클라우드 템플릿을 생성하거나 편집할 때 목적에 가장 적합한 보안 그룹 리소스를 사용합니다. 클라우드 템플릿에서 사용할 수 있는 보안 그룹 옵션에 대해 알아봅니다.

## 클라우드 애그노스틱 보안 그룹 리소스

현재 한 가지 유형의 보안 그룹 리소스만 있습니다. 클라우드 템플릿 [설계] 페이지의 **클라우드 애그노스틱 > 보안 그룹** 리소스를 사용하여 보안 그룹 리소스를 추가합니다. 리소스가 클라우드 템플릿 코드에 Cloud.SecurityGroup 리소스 유형으로 표시됩니다. 기본 리소스는 다음과 같이 표시됩니다.

```
Cloud_SecurityGroup_1:
  type: Cloud.SecurityGroup
  properties:
    constraints: []
    securityGroupType: existing
```

클라우드 템플릿 설계에서 보안 그룹 리소스를 기존(securityGroupType: existing) 또는 주문형(securityGroupType: new)으로 지정합니다.

기존 보안 그룹을 클라우드 템플릿 설계에 직접 추가하거나 네트워크 프로파일에 추가된 기존 보안 그룹을 사용할 수 있습니다. 기존 보안 그룹은 다양한 클라우드 계정 유형에 대해 지원됩니다.

NSX-V 및 NSX-T의 경우, 클라우드 템플릿을 설계하거나 수정할 때 기존 보안 그룹을 추가하거나 새 보안 그룹을 정의할 수 있습니다. 주문형 보안 그룹은 NSX-T 및 NSX-V에 대해서만 지원됩니다.

Microsoft Azure를 제외한 모든 클라우드 계정 유형의 경우 하나 이상의 보안 그룹을 시스템 NIC에 연결할 수 있습니다. Microsoft Azure 가상 시스템 NIC("machineName")는 하나의 보안 그룹에만 연결할 수 있습니다.

기본적으로 보안 그룹 속성 securityGroupType은 existing으로 설정되어 있습니다. 주문형 보안 그룹을 생성하려면 securityGroupType 속성에 대해 new를 입력합니다. 주문형 보안 그룹에 대한 방화벽 규칙을 지정하려면 보안 그룹 리소스의 Cloud.SecurityGroup 섹션에서 rules 속성을 사용합니다.

### 기존 보안 그룹

기존 보안 그룹은 NSX-T 또는 Amazon Web Services 같은 소스 클라우드 계정 리소스에서 생성됩니다. 이는 소스에서 vRealize Automation에 의해 수집된 데이터입니다. vRealize Automation 네트워크 프로파일의 일부로 사용 가능한 리소스 목록에서 기존 보안 그룹을 선택할 수 있습니다. 클라우드 템플릿 설계에서 보안 그룹 리소스의 securityGroupType: existing 설정을 사용하여 이름별로 구체적으로 또는 지정된 네트워크 프로파일에서 멤버 자격별로 근본적으로 기존 보안 그룹을 지정할 수 있습니다. 네트워크 프로파일에 보안 그룹을 추가하는 경우 네트워크 프로파일에 기능 태그를 하나 이상 추가합니다. 클라우드 템플릿 설계에서 주문형 보안 그룹 리소스를 사용하려면 제약 조건 태그가 필요합니다.

클라우드 템플릿 설계의 보안 그룹 리소스를 하나 이상의 시스템 리소스에 연결할 수 있습니다.

---

**참고** 클라우드 템플릿 설계의 시스템 리소스를 사용하여 Microsoft Azure 가상 시스템 NIC("machineName")에 프로비저닝하려는 경우 시스템 리소스를 단일 보안 그룹에만 연결해야 합니다.

---

### 주문형 NSX-V 및 NSX-T 보안 그룹

보안 그룹 리소스 코드의 securityGroupType: new 설정을 사용하여 클라우드 템플릿 설계를 정의하거나 수정할 때 주문형 보안 그룹을 정의할 수 있습니다.

주문형 NSX-V 또는 NSX-T 보안 그룹을 사용하여 특정 방화벽 규칙 집합을 네트워크 시스템 리소스나 그룹화된 리소스 집합에 적용할 수 있습니다. 각 보안 그룹에는 명명된 방화벽 규칙을 여러 개 포함할 수 있습니다. 주문형 보안 그룹을 사용하여 서비스 또는 프로토콜 및 포트를 지정할 수 있습니다. 서비스나 프로토콜 중 하나만 지정할 수 있습니다. 프로토콜에 더해 포트를 지정할 수 있습니다. 서비스를 지정하는 경우에는 포트를 지정할 수 없습니다. 규칙에 서비스 또는 프로토콜이 포함되어 있지 않은 경우 기본 서비스 값은 [임의]입니다.

방화벽 규칙에서 IP 주소 및 IP 범위를 지정할 수도 있습니다. 일부 방화벽 규칙 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)에 나와 있습니다.

NSX-V 또는 NSX-T 주문형 보안 그룹에서 방화벽 규칙을 생성할 때 기본값은 지정된 네트워크 트래픽을 허용하고 다른 네트워크 트래픽도 허용하는 것입니다. 네트워크 트래픽을 제어하려면 각 규칙에 대한 액세스 유형을 지정해야 합니다. 규칙 액세스 유형은 다음과 같습니다.

- 허용(기본값) - 이 방화벽 규칙에 지정된 네트워크 트래픽을 허용합니다.
- 거부 - 이 방화벽 규칙에 지정된 네트워크 트래픽을 차단합니다. 연결이 거부되었음을 클라이언트에게 능동적으로 알립니다.
- 삭제 - 이 방화벽 규칙에 지정된 네트워크 트래픽을 거부합니다. 수신기가 온라인 상태가 아닌 경우에도 자동으로 패킷을 삭제합니다.

access: Allow 및 access: Deny 방화벽 규칙을 사용하는 설계 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)의 내용을 참조하십시오.

---

**참고** 클라우드 관리자는 NSX 주문형 보안 그룹만 포함한 클라우드 템플릿 설계를 생성하고 해당 조직의 멤버가 네트워크 프로파일 및 클라우드 템플릿 설계에 기존 보안 그룹으로 추가할 수 있는 재사용 가능한 기존 보안 그룹 리소스를 생성하도록 해당 설계를 배포할 수 있습니다.

---

방화벽 규칙은 소스 및 대상 IP 주소에 대한 IPv4 또는 IPv6 형식 CIDR 값을 지원합니다. 방화벽 규칙에서 IPv6 CIDR 값을 사용하는 설계 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)의 내용을 참조하십시오.

### 주문형 보안 그룹 방화벽 규칙에서 App 분리 정책 사용

App 분리 정책을 사용하여 클라우드 템플릿에서 프로비저닝된 리소스 간의 내부 트래픽만 허용할 수 있습니다. App 분리 정책을 사용하면 클라우드 템플릿에서 프로비저닝된 시스템이 다른 시스템과 통신할 수 있지만 방화벽 외부에 연결할 수는 없습니다. 네트워크 프로파일에서 App 분리 정책을 생성할 수 있습니다. 또한 방화벽 거부 규칙이 있는 주문형 보안 그룹을 사용하거나 전용 또는 아웃바운드 네트워크를 사용하여 클라우드 템플릿 설계에 App 분리를 지정할 수도 있습니다.

App 분리 정책은 낮은 우선 순위로 생성됩니다. 여러 정책을 적용하는 경우 가중치가 더 높은 정책이 우선 적용됩니다.

App 분리 정책을 생성하면 정책에 자동 생성된 정책 이름이 할당됩니다. 또한 이 정책은 연결된 리소스 끝점 및 프로젝트와 관련된 다른 클라우드 템플릿 설계 및 설계 반복에서 재사용할 수 있습니다. App 분리 정책 이름은 클라우드 템플릿 설계 코드에 표시되지 않지만, 클라우드 템플릿 설계를 배포한 후에 [프로젝트] 페이지([인프라 > 관리 > 프로젝트](#))에서 사용자 지정 속성으로 표시됩니다.

프로젝트에서 동일한 연결된 끝점의 경우, App 분리를 위해 주문형 보안 그룹이 필요한 모든 배포는 동일한 App 분리 정책을 사용할 수 있습니다. 정책이 생성되면 삭제되지 않습니다. App 분리 정책을 지정하면 vRealize Automation가 프로젝트 내에서 그리고 연결된 끝점과 관련하여 정책을 검색합니다. 정책을 찾으면 재사용하고, 정책을 찾지 못하면 생성합니다. App 분리 정책 이름은 프로젝트의 사용자 지정 속성 목록에서 초기 배포 후에만 표시됩니다.

## 반복적인 클라우드 템플릿 개발에서 보안 그룹 사용

반복적인 개발 중에 보안 그룹 제약 조건을 변경하면 보안 그룹이 클라우드 템플릿의 시스템에 연결되어 있지 않은 경우 보안 그룹이 지정된 대로 반복에서 업데이트됩니다. 그러나 보안 그룹이 이미 시스템에 연결되어 있으면 다시 배포에 실패합니다. 반복적인 클라우드 템플릿 개발 중에는 연결된 시스템에서 기존의 보안 그룹 및/또는 securityGroupType 리소스 속성을 분리하고 각 다시 배포 간에 다시 연결해야 합니다. 처음에 클라우드 템플릿을 배포했다고 가정할 때 필요한 워크플로는 다음과 같습니다.

- 1 Cloud Assembly 템플릿 디자이너에서, 보안 그룹을 클라우드 템플릿의 연결된 모든 시스템에서 분리합니다.
- 2 **기존 배포 업데이트**를 클릭하여 템플릿을 다시 배포합니다.
- 3 템플릿에서 기존의 보안 그룹 제약 조건 태그 및/또는 securityGroupType 속성을 제거합니다.
- 4 템플릿에 새 보안 그룹 제약 조건 태그 및/또는 securityGroupType 속성을 추가합니다.
- 5 새 보안 그룹 제약 조건 태그 및/또는 securityGroupType 속성 인스턴스를 템플릿의 시스템에 연결합니다.
- 6 **기존 배포 업데이트**를 클릭하여 템플릿을 다시 배포합니다.

## 사용 가능한 2일차 작업

클라우드 템플릿 및 배포 리소스에 사용할 수 있는 일반 2일차 작업의 목록은 [vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업](#)의 내용을 참조하십시오.

## 자세히 알아보기

네트워크 격리를 위한 보안 그룹 사용에 대한 관련 정보는 [vRealize Automation의 보안 리소스](#)의 내용을 참조하십시오.

네트워크 프로파일에서 보안 그룹 설정 사용에 대한 자세한 내용은 [vRealize Automation의 네트워크 프로파일](#)에 대해 [알아보기](#) 및 [vRealize Automation Cloud Assembly에서 네트워크 프로파일 및 클라우드 템플릿 설계의 보안 그룹 설정 사용](#)의 내용을 참조하십시오.

샘플 보안 리소스 및 설정을 보여 주는 클라우드 템플릿 설계에 대한 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)의 내용을 참조하십시오.

## vRealize Automation 클라우드 템플릿에서 로드 밸런서 리소스 사용

vRealize Automation 클라우드 템플릿을 생성하거나 편집할 때 목적에 가장 적합한 로드 밸런서 리소스를 사용합니다.

클라우드 템플릿에서 **NSX** 및 클라우드 애그노스틱 로드 밸런서 리소스를 사용하여 배포의 로드 밸런싱을 제어할 수 있습니다.

클라우드 애그노스틱 로드 밸런서를 여러 클라우드에 배포할 수 있습니다. 클라우드별 로드 밸런서는 특정 클라우드/토폴로지에만 사용할 수 있는 고급 설정 및 기능을 지정할 수 있습니다. 클라우드별 속성은 **NSX** 로드 밸런서(**Cloud.NSX.LoadBalancer**) 리소스 유형에서 사용할 수 있습니다. 이러한 속성을 클라우드 애그노스틱 로드 밸런서(**Cloud.LoadBalancer**)에 추가하는 경우, 이러한 속성은 예를 들어 **Amazon Web Services** 또는 **Microsoft Azure** 로드 밸런서가 프로비저닝되는 경우에는 무시되지만 **NSX-V** 또는 **NSX-T** 로드 밸런서가 프로비저닝되는 경우에는 유지됩니다. **vRealize Automation** 클라우드 템플릿의 조건을 기반으로 사용 가능한 로드 밸런서 리소스 유형 중 하나를 선택합니다.

로드 밸런서 리소스는 설계 캔버스의 보안 그룹 리소스에 직접 연결할 수 없습니다.

### 클라우드 애그노스틱 로드 밸런서 리소스

모든 유형의 대상 시스템에 대한 네트워킹 특성을 지정하려는 경우 클라우드 애그노스틱 로드 밸런서를 사용합니다.

[클라우드 템플릿 설계] 페이지의 **클라우드 애그노스틱 > 로드 밸런서** 리소스를 사용하여 클라우드 애그노스틱 로드 밸런서를 추가합니다. 리소스가 클라우드 템플릿 코드에 **Cloud.LoadBalancer** 리소스 유형으로 표시됩니다. 기본 리소스는 다음과 같이 표시됩니다.

```
Cloud_LoadBalancer_1:
  type: Cloud.LoadBalancer
  properties:
    routes: []
    network: ''
    instances: []
    internetFacing: false
```

### NSX 로드 밸런서 리소스

클라우드 템플릿에 **NSX-V** 또는 **NSX-T**(정책 API 또는 Manager API 메서드)와 관련된 특성이 포함된 경우 **NSX** 로드 밸런서를 사용합니다. 하나 이상의 로드 밸런서를 **NSX-V** 또는 **NSX-T** 네트워크에 연결하거나 **NSX-V** 또는 **NSX-T** 네트워크에 연결된 시스템에 연결할 수 있습니다.

**NSX** 로드 밸런서는 **NSX > 로드 밸런서** 리소스를 사용하여 추가합니다. 리소스가 클라우드 템플릿 코드에 **Cloud.NSX.LoadBalancer** 리소스 유형으로 표시됩니다. 기본 리소스는 다음과 같이 표시됩니다.

```
Cloud_NSX_LoadBalancer_1:
  type: Cloud.NSX.LoadBalancer
  properties:
    routes: []
    network: ''
    instances: []
```

## 클라우드 템플릿 코드의 로드 밸런서 옵션

클라우드 템플릿에 하나 이상의 로드 밸런서 리소스를 추가하면 다음과 같은 설정을 지정할 수 있습니다. 몇 가지 예는 **vRealize Automation** 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예에 있습니다.

HTTP 프로토콜은 모든 주문형 로드 밸런서에 대해 지원됩니다.

HTTPS 프로토콜은 NSX 모드가 **정책**으로 설정된 NSX-T 클라우드 계정에 연결된 주문형 로드 밸런서에 대해서만 지원됩니다. NSX 모드가 **관리자**로 설정된 NSX-T 클라우드 계정은 HTTPS 프로토콜을 사용할 수 없습니다.

### ■ 시스템 규격

로드 밸런싱 풀에 참여할 명명된 시스템 리소스를 지정할 수 있습니다. 또는 특정 시스템 NIC가 로드 밸런서 풀에 참여하도록 지정할 수도 있습니다.

이 옵션은 **NSX** 로드 밸런서 리소스(Cloud.NSX.LoadBalancer)에만 사용할 수 있습니다.

이 옵션은 existing 및 public 네트워크 유형에 사용할 수 있습니다. 주문형 private, routed 및 outbound 네트워크 유형도 지원됩니다.

#### ■ resource.Cloud\_Machine\_1.id

클라우드 템플릿 코드에서 *Cloud\_Machine\_1*로 식별된 시스템이 로드 밸런서에 포함되도록 지정합니다.

#### ■ resource.Cloud\_Machine\_2.networks[2].id

로드 밸런서가 시스템 NIC *Cloud\_Machine\_2.networks[2]*에 배포될 때 클라우드 템플릿 코드에서 *Cloud\_Machine\_2*로 식별된 시스템만 로드 밸런서에 포함되도록 지정합니다.

### ■ 로깅 수준

로깅 수준 값은 오류 로그의 심각도 수준을 지정합니다. 옵션에는 없음, 긴급, 경고, 위험, 오류, 주의, 정보, 디버그 및 알림이 있습니다. 로깅 수준 값은 클라우드 템플릿의 모든 로드 밸런서에 적용됩니다. 이 옵션은 NSX에만 해당됩니다. 상위 값이 있는 로드 밸런서의 경우 상위 로깅 수준 설정이 해당 하위 항목의 모든 로깅 수준 설정을 재정의합니다.

관련 정보는 NSX 제품 설명서에서 **로드 밸런서 추가** 등의 항목을 참조하십시오.

### ■ 유형

로드 밸런서 유형을 사용하여 규모의 크기를 지정합니다. 기본값은 소형입니다. 이 옵션은 NSX에만 해당됩니다. 상위 값이 있는 로드 밸런서의 경우 상위 유형 설정이 해당 하위 항목의 모든 유형 설정을 재정의합니다.

#### ■ 소형

NSX-V의 컴팩트 및 NSX-T의 소형과 상관 관계가 있습니다.

#### ■ 중형

NSX-V의 대형 및 NSX-T의 중형과 상관 관계가 있습니다.

#### ■ 대형

NSX-V의 4배 대형 및 NSX-T의 대형과 상관 관계가 있습니다.

- 초대형

NSX-V의 2배 대형 및 NSX-T의 대형과 상관 관계가 있습니다.

관련 정보는 NSX 제품 설명서에서 **로드 밸런서 리소스 크기 조정** 등의 항목을 참조하십시오.

이 옵션은 **NSX** 로드 밸런서 리소스(Cloud.NSX.LoadBalancer)에만 사용할 수 있습니다.

- 알고리즘(서버 풀)

알고리즘 밸런싱 방법을 사용하여 수신 연결이 서버 풀 멤버 간에 배포되는 방식을 제어합니다. 이 알고리즘은 서버 풀이나 서버에서 직접 사용할 수 있습니다. 모든 로드 밸런싱 알고리즘은 다음 조건 중 하나라도 충족하는 서버를 건너뛸입니다.

- 관리 상태가 [사용 안 함]으로 설정되어 있습니다.
- 관리 상태가 [정상적으로 사용 안 함]으로 설정되어 있고 일치하는 지속성 항목이 없습니다.
- 액티브 또는 패시브 상태 점검 상태가 [다운됨]입니다.
- 서버 풀 최대 동시 연결에 대한 연결 제한에 도달했습니다.

이 옵션은 NSX에만 해당됩니다.

- IP\_HASH

소스 IP 주소의 해시와 실행 중인 모든 서버의 총 가중치에 따라 서버를 선택합니다.

NSX-V 및 NSX-T의 IP-HASH와 상관 관계가 있습니다.

- LEAST\_CONNECTION

서버에 이미 있는 연결 수를 기반으로 클라이언트 요청을 여러 서버로 분산합니다. 새 연결은 연결 수가 가장 적은 서버로 전송됩니다. 서버 풀 멤버 가중치가 구성되어 있어도 무시합니다.

NSX-V의 LEASTCONN 및 NSX-T의 LEAST\_CONNECTION과 상관 관계가 있습니다.

- ROUND\_ROBIN

수신 클라이언트 요청은 요청을 처리할 수 있는 사용 가능한 서버 목록을 통해 순환됩니다. 서버 풀 멤버 가중치가 구성되어 있어도 무시합니다. 기본값입니다.

NSX-V 및 NSX-T의 ROUND\_ROBIN과 상관 관계가 있습니다.

- WEIGHTED\_LEAST\_CONNECTION

각 서버에 풀의 다른 서버에 비해 해당 서버가 어떻게 수행하는지를 나타내는 가중치 값이 할당됩니다. 이 값은 풀에 있는 다른 서버와 비교하여 서버로 전송되는 클라이언트 요청 수를 결정합니다. 이 로드 밸런싱 알고리즘은 가중치 값을 사용하여 사용 가능한 서버 리소스 간에 로드를 적절히 분산하는 데 중점을 둡니다. 기본적으로 가중치 값이 구성되어 있지 않고 느린 시작이 사용되도록 설정되어 있는 경우 가중치는 1입니다.

NSX-T의 WEIGHTED\_LEAST\_CONNECTION과 상관 관계가 있습니다. NSX-V에 상관 관계가 없습니다.

## ■ WEIGHTED\_ROUND\_ROBIN

각 서버에 풀의 다른 서버에 비해 해당 서버가 어떻게 수행하는지를 나타내는 가중치 값이 할당됩니다. 이 값은 풀에 있는 다른 서버와 비교하여 서버로 전송되는 클라이언트 요청 수를 결정합니다. 이 로드 밸런싱 알고리즘은 사용 가능한 서버 리소스 간에 로드를 균등하게 분산하는 데 중점을 둡니다.

NSX-T의 WEIGHTED\_ROUND\_ROBIN과 상관 관계가 있습니다. NSX-V에 상관 관계가 없습니다.

## ■ URI

URI의 왼쪽 부분이 해시되고 실행 중인 서버의 총 가중치로 나누어집니다. 결과에 따라 요청을 받을 서버가 지정됩니다. 이렇게 하면 켜지거나 꺼지는 서버가 없는 경우 URI는 항상 동일한 서버로 연결됩니다. URI 알고리즘 매개 변수에는 두 가지 옵션(uriLength=<len> 및 uriDepth=<dep>)이 있습니다. length 매개 변수 범위는  $1 \leq \text{len} < 256$  이어야 합니다. depth 매개 변수 범위는  $1 \leq \text{dep} < 10$  이어야 합니다. length 및 depth 매개 변수 다음에는 양의 정수가 옵니다. 이 옵션은 URI 시작 부분만을 기반으로 서버 밸런스를 유지합니다. length 매개 변수는 알고리즘에서 해시 계산을 위해 URI 시작 부분에 정의된 문자만 고려해야 함을 나타냅니다. depth 매개 변수는 해시 계산에 사용될 최대 디렉토리 깊이를 나타냅니다. 요청의 각 슬래시는 1개의 수준으로 계산됩니다. 두 매개 변수를 모두 지정하면 둘 중 하나에 도달할 때 평가가 중지됩니다.

NSX-V의 URI와 상관 관계가 있습니다. NSX-T에 상관 관계가 없습니다.

## ■ HTTPHEADER

HTTP 헤더 이름은 각 HTTP 요청에서 조회됩니다. 괄호 안의 헤더 이름은 대/소문자를 구분하지 않습니다. 헤더가 없거나 값을 포함하지 않으면 라운드 로빈 알고리즘이 적용됩니다.

HTTPHEADER 알고리즘 매개 변수에는 하나의 옵션 headerName=<name>이 있습니다.

NSX-V의 HTTPHEADER와 상관 관계가 있습니다. NSX-T에 상관 관계가 없습니다.

## ■ URL

인수에 지정된 URL 매개 변수는 각 HTTP GET 요청의 쿼리 문자열에서 조회됩니다. 매개 변수 다음에 등호(=)와 값이 나오면 해당 값은 해시되고 실행 중인 서버의 총 가중치로 나누어집니다. 결과에 따라 요청을 받을 서버가 지정됩니다. 이 프로세스는 요청의 사용자 식별자를 추적하는 데 사용되고, 켜지거나 꺼지는 서버가 없는 한 동일한 사용자 ID가 항상 동일한 서버로 전송되도록 합니다. 값이나 매개 변수가 없으면 라운드 로빈 알고리즘이 적용됩니다. URL 알고리즘 매개 변수에는 하나의 옵션 urlParam=<url>이 있습니다.

NSX-V의 URL과 상관 관계가 있습니다. NSX-T에 상관 관계가 없습니다.

관련 정보는 NSX 제품 설명서에서 [로드 밸런싱을 위한 서버 풀 추가](#) 등의 항목을 참조하십시오.

## ■ 상태 모니터

상태 모니터 옵션을 사용하여 서버가 사용 가능한지 여부를 테스트합니다. HTTP, ICMP, TCP 및 UDP 프로토콜에 대한 액티브 상태 모니터링이 지원됩니다. 패시브 상태 모니터링은 NSX-T에서만 사용할 수 있습니다.

이 옵션은 NSX에만 해당됩니다.

#### ■ httpMethod

상태 점검 요청에 대한 서버 상태를 감지하는 데 사용할 HTTP 메서드입니다. 메서드는 GET, HOST, OPTIONS, HEAD 또는 PUT입니다.

#### ■ requestBody

상태 점검 요청 본문 콘텐츠입니다. HTTP, TCP 및 UDP 프로토콜에서 사용되며 이러한 프로토콜에 필요합니다.

#### ■ responseBody

상태 점검에 필요한 응답 본문 콘텐츠입니다. 수신된 문자열이 이 응답 본문과 일치하면 서버가 정상 상태로 간주됩니다. HTTP, TCP 및 UDP 프로토콜에서 사용되며 이러한 프로토콜에 필요합니다.

---

**참고** UDP 모니터 프로토콜을 사용하는 경우 UDP Data Sent 및 UDP Data Expected 매개 변수가 필요합니다. requestBody 및 responseBody 속성은 이러한 매개 변수에 매핑됩니다.

---

이 옵션은 NSX 로드 밸런서 리소스(Cloud.NSX.LoadBalancer)에만 사용할 수 있습니다.

관련 정보는 NSX 제품 설명서에서 [액티브 상태 모니터 구성](#) 등의 항목을 참조하십시오.

#### ■ 상태 점검

상태 점검 옵션을 사용하여 로드 밸런서가 상태 점검을 수행하는 방법을 지정합니다.

이 옵션은 NSX 로드 밸런서 리소스(Cloud.NSX.LoadBalancer)에만 사용할 수 있습니다.

[vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)에서 사용 가능한 상태 점검 설정 샘플을 참조하십시오.

### NSX-V 및 NSX-T 네트워크 유형과 로드 밸런서 옵션

로드 밸런서 옵션은 클라우드 템플릿에서 로드 밸런서 리소스가 연결되어 있는 네트워크에 따라 다릅니다. 네트워크 유형 및 네트워크 조건에 상대적으로 로드 밸런서를 구성할 수 있습니다.

#### ■ 주문형 아웃바운드 네트워크

로드 밸런서 계산이 주문형 outbound 네트워크에 연결되어 있는 경우 주문형 네트워크의 Tier-1 라우터에 대해 로드 밸런서가 생성됩니다.

#### ■ 주문형 전용 네트워크

로드 밸런서 계산이 주문형 private 네트워크에 연결되어 있는 경우 새로운 Tier-1 라우터가 생성되고 네트워크 프로파일에 지정된 Tier-0 라우터에 연결됩니다. 이후 로드 밸런서는 Tier-1 라우터에 연결됩니다. VIP가 existing 네트워크에 있는 경우 Tier-1 라우터 VIP 알림이 사용되도록 설정됩니다.

DHCP에 대해 private 네트워크가 구성된 경우 전용 네트워크 및 로드 밸런서는 Tier-1 라우터를 공유합니다.

#### ■ 기존 네트워크

로드 밸런서가 existing 네트워크에 연결되어 있는 경우 로드 밸런서는 기존 네트워크의 Tier-1 라우터와 함께 생성됩니다. Tier-1 라우터에 연결된 로드 밸런서가 없는 경우 새로운 로드 밸런서 서비스가 생성됩니다. 로드 밸런서가 이미 있다면 새로운 가상 서버가 여기에 연결됩니다. existing 네트워크가 Tier-1 라우터에 연결되어 있지 않은 경우 새로운 Tier-1 라우터가 생성되어 네트워크 프로파일에 정의된 Tier-0 라우터에 연결되고 Tier-1 라우터 VIP 알람은 사용되도록 설정되지 않습니다.

#### ■ 네트워크 프로파일에 정의된 네트워크 격리

outbound 또는 private 네트워크 유형의 경우 네트워크 프로파일에서 네트워크 격리 설정을 지정하여 새 보안 그룹을 예물레이션할 수 있습니다. 시스템이 기존 네트워크에 연결되고 프로파일에서 격리 설정이 정의되므로 이 옵션은 기존 네트워크에 생성된 로드 밸런서와 유사합니다. 차이점은 데이터 경로를 사용하도록 설정하기 위해 Tier-1 업링크 포트 IP가 격리 보안 그룹에 추가된다는 점입니다.

클라우드 템플릿 설계에서 NSX 로드 밸런서 리소스를 사용하여 NSX에 연결된 네트워크에 대한 로드 밸런서 설정을 지정할 수 있습니다.

자세한 내용은 VMware 블로그 게시물 [NSX-T를 포함하는 vRA Cloud Assembly 로드 밸런서 심화 연구](#)를 참조하십시오.

#### 여러 로드 밸런서가 NSX-T Tier 1 또는 NSX-V Edge를 공유할 때 로깅 수준 또는 유형 설정 재구성

NSX-T 끝점에서 Tier-1 라우터를 공유하거나 NSX-V 끝점에서 Edge 라우터를 공유하는 여러 로드 밸런서가 포함된 클라우드 템플릿을 사용하는 경우, 로드 밸런서 리소스 중 하나에서 로깅 수준 또는 유형 설정을 재구성해도 다른 로드 밸런서에 대한 설정이 업데이트되지 않습니다. 설정이 일치하지 않으면 NSX에서 불일치가 발생합니다. 이러한 로깅 수준 및/또는 유형 설정을 재구성할 때 불일치를 방지하려면 연결된 해당 NSX 끝점에서 Tier 1 또는 Edge를 공유하는 클라우드 템플릿의 모든 로드 밸런서 리소스에 대해 동일한 재구성 값을 사용합니다.

#### 사용 가능한 2일차 작업

로드 밸런서가 포함된 배포를 축소하거나 확장하는 경우 로드 밸런서는 새로 추가된 시스템을 포함하거나 분해를 목표로 하는 로드 밸런싱 시스템을 중지하도록 구성됩니다.

클라우드 템플릿 및 배포에 사용할 수 있는 일반 2일차 작업의 목록은 [vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업](#)의 내용을 참조하십시오.

#### 자세히 알아보기

네트워크 프로파일의 로드 밸런서 설정 정의에 대한 자세한 내용은 [vRealize Automation의 네트워크 프로파일에 대해 알아보기](#)의 내용을 참조하십시오.

로드 밸런서를 포함하는 클라우드 템플릿 설계에 대한 예는 [vRealize Automation 클라우드 템플릿의 네트워크, 보안 및 로드 밸런서 예](#)의 내용을 참조하십시오.

#### 사용자 이름 및 암호 액세스를 사용하는 Puppet 지원 클라우드 템플릿

이 예에서는 사용자 이름 및 암호 액세스를 사용하여 Puppet 구성 관리를 vCenter 계산 리소스에 배포된 클라우드 템플릿에 추가합니다.

이 절차는 사용자 이름 및 암호 인증이 필요한 배포 가능한 Puppet 지원 리소스를 생성할 수 있는 방법에 대한 하나의 예를 보여 줍니다. 사용자 이름 및 암호 액세스란 사용자가 계산 리소스에서 Puppet 기본 시스템으로 수동으로 로그인해야 Puppet 구성 관리를 호출할 수 있음을 의미합니다.

선택적으로, 계산 리소스가 Puppet 기본 시스템을 사용하여 인증을 처리하도록 클라우드 템플릿에서 구성 관리를 설정하는 원격 액세스 인증을 구성할 수 있습니다. 원격 액세스를 사용하도록 설정하면 계산 리소스가 암호 인증을 만족하는 키를 자동으로 생성합니다. 이 경우에도 올바른 사용자 이름은 필요합니다.

vRealize Automation Cloud Assembly Blueprint에서 서로 다른 Puppet 시나리오를 구성할 수 있는 방법에 대한 더 많은 예를 보려면 [AWS Puppet 구성 관리 클라우드 템플릿 예제](#) 및 [vCenter Puppet 구성 클라우드 템플릿 예](#) 항목을 참조하십시오.

### 사전 요구 사항

- 올바른 네트워크에 Puppet Enterprise 인스턴스를 설정합니다.
- 통합 기능을 사용하여 Puppet Enterprise 인스턴스를 vRealize Automation Cloud Assembly에 추가합니다. [vRealize Automation Cloud Assembly에서 Puppet Enterprise 통합 구성](#) 항목을 참조하십시오.
- vSphere 계정 및 vCenter 계산 리소스를 설정합니다.

## 절차

- 1 Puppet 구성 관리 구성 요소를 원하는 클라우드 템플릿 캔버스의 vSphere 계산 리소스에 추가합니다.
  - a 인프라 > 관리 > 통합을 선택합니다.
  - b 통합 추가를 클릭하고 Puppet을 선택합니다.
  - c [Puppet 구성] 페이지에 적절한 정보를 입력합니다.

구성	설명	예시 값
호스트 이름	Puppet 기본 시스템의 호스트 이름 또는 IP 주소	Puppet-Ubuntu
SSH 포트	vRealize Automation Cloud Assembly와 Puppet 기본 시스템 간 통신을 위한 SSH 포트입니다. (선택 사항)	해당 없음
Autosign 암호	Autosign 인증서 요청 지원을 위해 노드에서 제공해야 하는 Puppet 기본 시스템에서 구성된 공유 암호입니다.	사용자 특정
위치	Puppet 기본 시스템이 전용 클라우드에 있는지 아니면 공용 클라우드에 있는지 나타냅니다.  <b>참고</b> 교차 클라우드 배포는 배포 계산 리소스와 Puppet 기본 시스템 간에 연결이 있는 경우에만 지원됩니다.	
Cloud proxy	Microsoft Azure 또는 Amazon Web Services와 같은 공용 클라우드 계정에는 필요하지 않습니다. vCenter 기반 클라우드 계정을 사용 중이라면 계정에 적절한 cloud proxy를 선택합니다.	해당 없음
사용자 이름	Puppet 기본 시스템에 대한 SSH 및 RBAC 사용자 이름입니다.	사용자 특정. YAML 값은 '\$ {input.username}'입니다.
암호	Puppet 기본 시스템에 대한 SSH 및 RBAC 암호입니다.	User 특정. YAML 값은 '\$ {input.password}'입니다.
이 사용자에게 대해 sudo 명령 사용	procidd에 대해 sudo 명령을 사용하려면 선택합니다.	true
이름	Puppet 기본 시스템 이름입니다.	PEMasterOnPrem
설명		

- 2 다음 예에 나와 있는 것과 같이 Puppet YAML에 사용자 이름 및 암호를 추가합니다.
- 3 아래 예에 나와 있는 것과 같이 Puppet 클라우드 템플릿 YAML에 대한 remoteAccess 속성 값이 authentication: username and password로 설정되어 있는지 확인합니다.

## 예제: vCenter 사용자 이름 및 암호 YAML 코드

다음 예는 vCenter 계산 리소스에서 사용자 이름 및 암호 인증을 추가하기 위한 대표 YAML 코드를 보여줍니다.

```
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}'
```

## AWS Puppet 구성 관리 클라우드 템플릿 예제

AWS 계산 리소스에서 Puppet 기반 구성 관리를 지원하는 클라우드 템플릿을 구성하기 위한 몇 가지 옵션이 있습니다.

## 사용자 이름 및 암호를 사용하여 AWS에서 Puppet 관리

예시...	샘플 Blueprint YAML
지원되는 모든 Amazon 시스템 이미지의 클라우드 구성 인증	<pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 resources:   Webserver:     type: Cloud.AWS.EC2.Instance     properties:       flavor: small       image: centos       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa                 AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6/+vGbmKXoRpX                 dmettem@dmettem-m01.vmware.com             runcmd:               - echo "Defaults:\${input.username} !requiretty"               &gt;&gt; /etc/sudoers.d/\${input.username}   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEOAWS       environment: production       role: 'role::linux_webserver'       host: '\${Webserver.*}'       osType: linux       username: '\${input.username}'       password: '\${input.password}'       useSudo: true </pre>
기존 사용자를 사용하여 사용자 지정 Amazon 시스템 이미지의 클라우드 구성 인증	<pre> inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 </pre>

예시...	샘플 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>

## 생성된 PublicPrivateKey를 사용하여 AWS에서 Puppet 관리

예제...	샘플 Blueprint YAML
<p>generatedPublicPrivateKey 액세 스를 사용하여 AWS에서 remoteAccess.authentication 인증</p>	<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 Puppet 구성 클라우드 템플릿 예

vCenter 계산 리소스에서 Puppet 기반 구성 관리를 지원하는 클라우드 템플릿을 구성하기 위한 몇 가지 옵션이 있습니다.

## 사용자 이름 및 암호 인증을 사용하는 vSphere의 Puppet

다음 예제는 사용자 이름 및 암호 인증을 사용하는 vSphere OVA의 Puppet에 대한 YAML 코드 예를 보여줍니다.

표 6-5.

예제...	샘플 Blueprint YAML
<p>사용자 이름 및 암호 인증을 사용하는 vSphere OVA의 Puppet에 대한 YAML 코드</p>	<pre>inputs:   username:     type: string     title: Username     default: puppet   password:     type: string     title: Password     encrypted: true     default: VMware@123 resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'   Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024       imageRef: &gt;- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX dmettem@dmettem-m01.vmware.com       runcmd:         - echo "Defaults:\${input.username}"</pre>
<p>계산 리소스에서 사용자 이름 및 암호 인증을 사용하는 vSphere OVA의 Puppet에 대한 YAML 코드</p>	<pre>inputs:   username:     type: string     title: Username     default: puppet</pre>

표 6-5. (계속)

예제...	샘플 Blueprint YAML
	<pre> password:   type: string   title: Password   encrypted: true   default: VMware@123 resources:   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEonAWS       environment: dev       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       useSudo: true       host: '\${Webserver.*}'       osType: linux       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'   Webserver:     type: Cloud.vSphere.Machine     properties:       cpuCount: 1       totalMemoryMB: 1024       imageRef: &gt;= https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       cloudConfig:           #cloud-config         ssh_pwauth: yes         chpasswd:           list:               \${input.username}:\${input.password}           expire: false         users:           - default           - name: \${input.username}             lock_passwd: false             sudo: ['ALL=(ALL) NOPASSWD:ALL']             groups: [wheel, sudo, admin]             shell: '/bin/bash'             ssh-authorized-keys:               - ssh-rsa                 AAAAB3NzaC1yc2EAAAADAQABAAQDytVL+Q6+vGbmKXoRpX                 dmettem@dmettem-m01.vmware.com             runcmd:               - echo "Defaults:\${input.username} </pre>
<p>제산 리소스에서 원격 액세스 지원 암호 인증을 사용하는 vCenter의 Puppet에 대한 YAML 코드</p>	<pre> inputs:   username:     type: string     title: Username     description: Username to use to install Puppet agent     default: puppet   password:     type: string     title: Password     default: VMware@123     encrypted: true </pre>

표 6-5. (계속)

예제...	샘플 Blueprint YAML
	<pre> description: Password for the given username to install Puppet agent resources:   Puppet-Ubuntu:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;-         https://cloud-images.ubuntu.com/releases/16.04/         release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       remoteAccess:         authentication: usernamePassword         username: '\${input.username}'         password: '\${input.password}'   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: PEMasterOnPrem       environment: production       role: 'role::linux_webserver'       username: '\${input.username}'       password: '\${input.password}'       host: '\${Puppet-Ubuntu.*}'       useSudo: true       agentConfiguration:         certName: '\${Puppet-Ubuntu.address}' </pre>

## 생성된 PublicPrivateKey 인증을 사용하는 vSphere의 Puppet

표 6-6.

예제...	샘플 Blueprint YAML
계산 리소스에서 생성된 PublicPrivateKey 인증을 사용하는 vSphere OVA의 Puppet에 대한 YAML 코드.	<pre> inputs: {} resources:   Machine:     type: Cloud.vSphere.Machine     properties:       flavor: small       imageRef: &gt;- https://cloud-images.ubuntu.com/releases/16.04/ release-20170307/ubuntu-16.04-server-cloudimg-amd64.ova       remoteAccess:         authentication: generatedPublicPrivateKey   Puppet_Agent:     type: Cloud.Puppet     properties:       provider: puppet-BlueprintProvisioningITSuite       environment: production       role: 'role::linux_webserver'       host: '\${Machine.*}'       osType: linux       username: ubuntu       useSudo: true       agentConfiguration:         runInterval: 15m         certName: '\${Machine.address}'         - echo "Defaults:\${input.username} </pre>

## vRealize Automation Cloud Assembly에 Terraform 구성을 포함하는 방법

vRealize Automation Cloud Assembly의 클라우드 템플릿에 Terraform 구성을 리소스로 포함할 수 있습니다.

### vRealize Automation Cloud Assembly Terraform 런타임 환경 준비

Terraform 구성이 포함된 설계에는 vRealize Automation Cloud Assembly 온-프레미스 제품과 통합하는 Terraform 런타임 환경에 대한 액세스 권한이 필요합니다.

### Terraform 런타임을 추가하는 방법

런타임 환경은 Terraform CLI 명령을 실행하여 요청된 작업을 수행하는 Kubernetes 클러스터로 구성됩니다. 또한 런타임은 로그를 수집하고 Terraform CLI 명령의 결과를 반환합니다.

vRealize Automation 온-프레미스 제품을 사용하려면 사용자가 자체 Terraform 런타임 Kubernetes 클러스터를 구성해야 합니다. 조직당 하나의 Terraform 런타임만 지원됩니다. 해당 조직에 대한 모든 Terraform 배포는 동일한 런타임을 사용합니다.

1 Terraform CLI를 실행할 Kubernetes 클러스터가 있는지 확인합니다.

- 모든 사용자는 kubeconfig 파일을 제공하여 관리되지 않는 Kubernetes 클러스터에서 Terraform CLI를 실행할 수 있습니다.
- 엔터프라이즈 라이선스 사용자는 vRealize Automation에서 관리되는 Kubernetes 클러스터에서 Terraform CLI를 실행할 수 있습니다.

vRealize Automation Cloud Assembly에서 **인프라 > 리소스 > Kubernetes** 로 이동하여 Kubernetes 클러스터가 있는지 확인합니다. 추가해야 하는 경우에는 [vRealize Automation Cloud Assembly에서 Kubernetes를 사용하는 방법](#) 항목을 참조하십시오.

2 Kubernetes 클러스터가 새로 추가되거나 수정된 경우 데이터 수집이 완료될 때까지 기다립니다.

데이터 수집은 네임스페이스 및 기타 정보 목록을 검색하며, 제공자에 따라 최대 5분이 소요될 수 있습니다.

3 데이터 수집이 완료되면 **인프라 > 연결 > 통합 > 통합 추가**로 이동하고 **Terraform 런타임** 카드를 선택합니다.

4 설정을 입력합니다.

그림 6-3. Terraform 런타임 통합 예제

## New Integration

**Name \*** OurOrg TF Runtime

**Description**

### Terraform Runtime Integration

**Runtime type \***

☒ Managed kubernetes cluster  
☐ External kubeconfig

**Kubernetes cluster \*** OurK8Cluster ⓘ

**Kubernetes namespace \*** OurK8Namespace ⓘ

### Runtime Container Settings

**Image** projects.registry.vmware.com/vra/terraform:latest ⓘ

**CPU request (Millicores)** 250

**CPU limit (Millicores)** 250

**Memory request (MB)** 512

**Memory limit (MB)** 512

VALIDATE

설정	설명
이름	런타임 통합에 고유한 이름을 부여합니다.
설명	통합의 목적을 설명합니다.
Terraform 런타임 통합:	
런타임 유형(엔터프라이즈 전용)	엔터프라이즈 라이선스 사용자는 vRealize Automation에서 관리되는 Kubernetes 클러스터에서 Terraform CLI를 실행할지 또는 관리되지 않는 클러스터에서 실행할지 선택할 수 있습니다.
Kubernetes kubeconfig(모든 사용자)	<p>관리되지 않는 Kubernetes 클러스터의 경우 외부 클러스터에 대한 kubeconfig 파일의 전체 콘텐츠를 붙여넣습니다.</p> <p>프록시 서버와 함께 외부 Kubernetes 런타임을 사용하려면 <a href="#">프록시 지원을 추가하는 방법을 참조하십시오</a>.</p> <p>이 옵션은 모든 사용자가 사용할 수 있습니다.</p>

설정	설명
Kubernetes 클러스터(엔터프라이즈 전용)	vRealize Automation에서 관리되는 Kubernetes의 경우 Terraform CLI를 실행할 클러스터를 선택합니다. 클러스터 및 해당 kubeconfig 파일에 연결할 수 있어야 합니다. /cmx/api/resources/k8s/clusters/{clusterId}/kube-config에서 GET을 사용하여 kubeconfig에 대한 액세스를 검증할 수 있습니다. 이 옵션은 엔터프라이즈 라이선스에서만 사용할 수 있습니다.
Kubernetes 네임스페이스	Terraform CLI를 실행하는 포드를 생성하기 위해 클러스터 내에서 사용할 네임스페이스를 선택합니다.
런타임 컨테이너 설정	
이미지	실행하려는 Terraform 버전의 컨테이너 이미지에 대한 경로를 입력합니다.  <b>참고</b> 검증 버튼은 컨테이너 이미지를 확인하지 않습니다.
CPU 요청	실행 중인 컨테이너의 CPU 양을 입력합니다. 기본값은 250밀리코어입니다.
CPU 제한	실행 중인 컨테이너에 대해 허용되는 최대 CPU를 입력합니다. 기본값은 250밀리코어입니다.
메모리 요청	실행 중인 컨테이너의 메모리 양을 입력합니다. 기본값은 512MB입니다.
메모리 제한	실행 중인 컨테이너에 대해 허용되는 최대 메모리를 입력합니다. 기본값은 512MB입니다.

5 **검증**을 클릭하고 필요에 따라 설정을 조정합니다.

6 **추가**를 클릭합니다.

설정이 캐시됩니다. 통합을 추가한 후에는 클러스터 또는 네임스페이스와 같은 설정을 수정할 수 있지만 변경 내용이 감지되고 Terraform CLI가 새 설정에서 실행되는 데 최대 5분이 걸릴 수 있습니다.

## Terraform 런타임 문제 해결

일부 Terraform 구성 배포 문제가 런타임 통합과 관련이 있을 수 있습니다.

문제	원인	해결 방법
네임스페이스가 잘못되었음을 나타내는 오류와 함께 검증이 실패합니다.	UI에서 클러스터를 수정했지만 이전 네임스페이스를 종료했습니다.	클러스터 선택을 수정한 후 항상 네임스페이스를 다시 선택합니다.
네임스페이스 드롭다운이 비어 있거나 새로 추가된 네임스페이스를 나열하지 않습니다.	클러스터에 대한 데이터 수집이 완료되지 않았습니다. 데이터 수집은 클러스터를 입력하거나 수정한 후 최대 5분이 소요되고 네임스페이스를 입력하거나 수정할 때 최대 10분이 소요됩니다.	<p>기존 네임스페이스가 포함된 새 클러스터의 경우 데이터 수집이 완료될 때까지 5분까지 기다립니다.</p> <p>기존 클러스터의 새 네임스페이스의 경우 데이터 수집이 완료될 때까지 10분까지 기다립니다.</p> <p>문제가 계속되면 클러스터를 제거하고 <b>인프라 &gt; 리소스 &gt; Kubernetes</b>에 다시 추가합니다.</p>
통합 계정이 업데이트된 후에도 Terraform CLI 컨테이너가 이전 클러스터, 이전 네임스페이스 또는 이전 런타임 설정으로 생성됩니다.	vRealize Automation에서 사용하는 Kubernetes API 클라이언트는 5분 동안 캐시됩니다.	변경 내용을 적용하려면 5분까지 소요될 수 있습니다.
검증 또는 Terraform 배포 작업이 kubeconfig를 사용할 수 없다는 오류와 함께 실패합니다.	<p>vRealize Automation에서 클러스터에 연결할 수 없기 때문에 이러한 오류가 발생하는 경우가 있습니다.</p> <p>다른 경우에는 사용자 자격 증명, 토큰 또는 인증서가 잘못되었습니다.</p>	kubeconfig 오류는 여러 가지 이유로 인해 발생할 수 있으며 문제 해결을 위해 기술 지원과 관련한 참여가 필요할 수 있습니다.

## 프록시 지원을 추가하는 방법

외부 Kubernetes 런타임 클러스터가 프록시 서버를 통해 연결되도록 하려면 다음 단계를 수행합니다.

- 1 외부 Kubernetes 클러스터 서버에 로그인합니다.
- 2 빈 폴더를 생성합니다.
- 3 새 폴더에서 Dockerfile이라는 새 파일에 다음 줄을 추가합니다.

```
FROM projects.registry.vmware.com/vra/terraform:latest as final
ENV https_proxy=protocol://username:password@proxy_host:proxy_port
ENV http_proxy=protocol://username:password@proxy_host:proxy_port
ENV no_proxy=.local,.localdomain,localhost
```

- 4 인터넷에 액세스하는 데 사용하는 프록시 서버 설정이 https\_proxy 및 http\_proxy 환경 변수에 포함되도록 자리 표시자 값을 수정합니다.

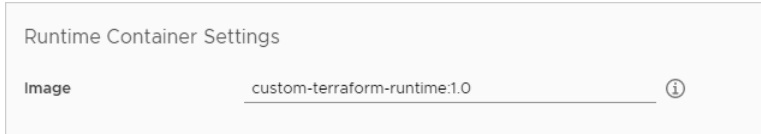
*protocol*은 프록시 서버가 무엇을 사용하느냐에 따라 http 또는 https가 되며 https\_proxy 또는 http\_proxy의 환경 변수 이름과 일치하지 않을 수 있습니다.

- 5 Dockerfile을 저장하고 닫습니다.
- 6 빈 폴더에서 다음 명령을 실행합니다. 계정 권한에 따라 sudo 모드에서 명령을 실행해야 할 수도 있습니다.

```
docker build --file Dockerfile --tag custom-terraform-runtime:1.0 .
```

이 명령은 로컬 custom-terraform-runtime: 1.0 Docker 이미지를 생성합니다.

- 7 vRealize Automation Cloud Assembly의 **인프라 > 연결 > 통합**에서 Terraform 런타임 통합으로 이 동합니다.
- 8 custom-terraform-runtime:1.0 이미지를 사용하도록 런타임 컨테이너 설정을 생성하거나 편집합니다.



## vRealize Automation Cloud Assembly 인터넷에 액세스할 수 없는 Terraform 런타임

인터넷 연결이 끊어진 상태에서 Terraform 통합을 설계하고 실행해야 하는 vRealize Automation Cloud Assembly 사용자는 이 예제에 따라 런타임 환경을 설정할 수 있습니다.

**참고** 설정하는 동안 일시적으로 인터넷에 연결해야 합니다.

이 프로세스에서는 자체 Docker 레지스트리가 있고 인터넷에 연결하지 않고도 해당 저장소에 액세스할 수 있다고 가정합니다.

### 사용자 지정 컨테이너 이미지 생성

- 1 Terraform 제공자 플러그인 이진 파일이 포함된 사용자 지정 컨테이너 이미지를 구축합니다.

다음 Dockerfile은 Terraform GCP 제공자를 사용하여 사용자 지정 이미지를 생성하는 예시를 보여줍니다.

Dockerfile에서 다운로드한 기본 이미지 `projects.registry.vmware.com/vra/terraform:latest`는 VMware Harbor 레지스트리(`projects.registry.vmware.com`)에 대한 임시 인터넷 액세스가 필요합니다.

방화벽 설정 또는 프록시 설정으로 인해 이미지 빌드가 실패할 수 있습니다. 또한 Terraform 제공자 플러그인 이진 파일을 다운로드하려면 `releases.hashicorp.com`에 임시 액세스가 가능하도록 설정해야 할 수도 있습니다. 하지만 개인 레지스트리를 사용하여 플러그인 이진을 옵션으로 제공할 수 있습니다.

```
FROM projects.registry.vmware.com/vra/terraform:latest as final

# Create provider plug-in directory
ARG plugins=/tmp/terraform.d/plugin-cache/linux_amd64
RUN mkdir -m 777 -p $plugins

# Download and unzip all required provider plug-ins from hashicorp to provider directory
RUN cd $plugins \
  && wget -q https://releases.hashicorp.com/terraform-provider-google/3.58.0/terraform-provider-google_3.58.0_linux_amd64.zip \
  && unzip *.zip \
  && rm *.zip
```

```
# For "terraform init" configure terraform CLI to use provider plug-in directory and not
download from internet
ENV TF_CLI_ARGS_init="-plugin-dir=$plugins -get-plugins=false"
```

- 2 사용자 지정 컨테이너 이미지를 구축하고 태그를 지정하고 자체 Docker 저장소로 푸시합니다.
- 3 vRealize Automation Cloud Assembly의 **인프라 > 연결 > 통합**에서 Terraform 런타임 통합으로 이 통합합니다.
- 4 런타임 컨테이너 설정을 생성 또는 편집하여 사용자 지정 컨테이너 이미지에 대한 저장소를 추가합니다. 예제의 구축된 사용자 지정 컨테이너 이미지 이름은 `registry.ourcompany.com/project1/image1:latest`입니다.

Runtime Container Settings	
Image	<input type="text" value="registry.ourcompany.com/project1/image1:latest"/> ⓘ

### Terraform CLI를 로컬에서 호스팅

- 1 Terraform CLI 이진 파일을 다운로드합니다.
- 2 Terraform CLI 이진 파일을 로컬 웹 서버에 업로드합니다.
- 3 vRealize Automation Cloud Assembly에서 **인프라 > 구성 > Terraform 버전**으로 이동합니다.
- 4 로컬 웹 서버에서 호스팅되는 Terraform CLI 이진 파일에 대한 URL을 포함하도록 Terraform 버전을 생성하거나 편집합니다.

0.12.29 DELETE	
Version *	<input type="text" value="0.12.29"/> ⓘ
Description	<input type="text"/>
Enabled	<input checked="" type="checkbox"/> ⓘ
URL *	<input type="text" value="http://host1.ourcompany.com:8080/tf/0.12.29/terraform_0.12.29_linux_amd64.zip"/> ⓘ
SHA256 Checksum *	<input type="text" value="872245d9c6302b24dc0d98a1e010aef1e4ef60865a2d1f60102c8ad03e9d5a1d"/> ⓘ

### Terraform 구성 설계 및 배포

런타임이 준비되면 Terraform 구성 파일을 git에 추가하고 클라우드 템플릿을 설계하고 배포할 수 있습니다.

시작하려면 vRealize Automation Cloud Assembly에서 **Terraform 구성 준비** 항목을 참조하세요.

## 문제 해결

배포할 때 vRealize Automation Cloud Assembly에서 배포를 엽니다. [기록] 탭에서 Terraform 이벤트를 찾아서 오른쪽의 **로그 표시**를 클릭합니다. 로컬 Terraform 제공자가 작동 중이면 로그에 다음 메시지가 나타납니다.

```
Initializing provider plugins
```

```
Terraform has been successfully initialized
```

보다 탄탄한 로그를 위해 클라우드 템플릿 코드를 수동으로 편집하여 다음 예와 같이 **TF\_LOG: DEBUG**를 추가할 수 있습니다.

```
resources:
  terraform:
    type: Cloud.Terraform.Configuration
    properties:
      providers:
        - name: google
          # List of available cloud zones: gcp/us-west1
          cloudZone: gcp/us-west1
      environment:
        # Configure terraform CLI debug log settings
        TF_LOG: DEBUG
    terraformVersion: 0.12.29
    configurationSource:
      repositoryId: fc569ef7-f013-4489-9673-6909a2791071
      commitId: 3e00279a843a6711f7857929144164ef399c7421
      sourceDirectory: gcp-simple
```

## vRealize Automation Cloud Assembly에서 Terraform 구성 준비

vRealize Automation Cloud Assembly 템플릿에 Terraform 구성을 추가하기 전에 버전 제어 저장소를 설정하고 통합합니다.

- 1 사전 요구 사항
- 2 버전 제어 저장소에 Terraform 구성 파일 저장
- 3 클라우드 영역 매핑 사용
- 4 저장소를 vRealize Automation Cloud Assembly와 통합

### 사전 요구 사항

vRealize Automation 온-프레미스 제품에서 Terraform 작업을 실행하려면 Terraform 런타임 통합이 필요합니다. [vRealize Automation Cloud Assembly Terraform 런타임 환경 준비](#)의 내용을 참조하십시오.

### 버전 제어 저장소에 Terraform 구성 파일 저장

vRealize Automation Cloud Assembly는 Terraform 구성에 대해 다음과 같은 버전 제어 저장소를 지원합니다.

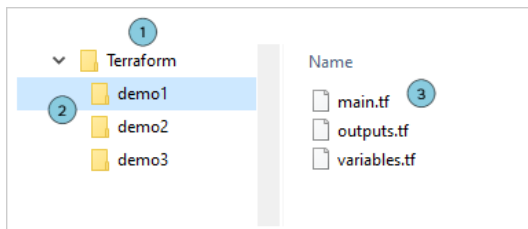
- GitHub 클라우드, GitHub Enterprise 온-프레미스

- GitLab 클라우드, GitLab Enterprise 온-프레미스
- Bitbucket 온-프레미스

버전 제어 저장소에서 한 계층의 하위 디렉토리가 있고 각 하위 디렉토리에 **Terraform** 구성 파일을 포함하는 기본 디렉토리를 생성합니다. Terraform 구성당 하나의 하위 디렉토리를 생성합니다.

- 1 기본 디렉토리
- 2 단일 하위 디렉토리 계층
- 3 배포 준비된 Terraform 구성 파일

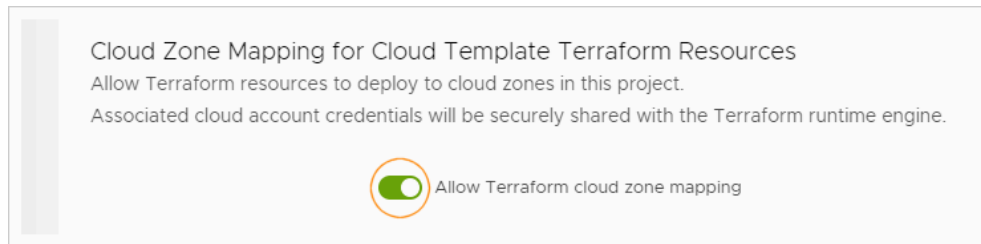
구성 파일이 포함된 Terraform 상태 파일을 포함하지 마십시오. terraform.tfstate가 있으면 배포 중에 오류가 발생합니다.



## 클라우드 영역 매핑 사용

클라우드 계정에 배포하려는 경우, Terraform 런타임 엔진에 이러한 클라우드 영역 자격 증명이 필요합니다.

프로젝트 **프로비저닝** 탭에서 **Terraform 클라우드 영역 매핑 허용**을 사용하도록 설정합니다.



자격 증명이 안전하게 전송되더라도 추가적인 보안을 위해 프로젝트 사용자가 클라우드 계정에 배포할 필요가 없으면 이 옵션을 비활성화된 상태로 두어야 합니다.

## 저장소를 vRealize Automation Cloud Assembly와 통합

vRealize Automation Cloud Assembly에서 **인프라 > 연결 > 통합**으로 이동합니다.

Terraform 구성인 GitHub, GitLab 또는 Bitbucket을 저장한 저장소 오퍼링 유형에 통합을 추가합니다.

통합에 프로젝트를 추가하는 경우 **Terraform 구성** 유형을 선택하고 저장소 및 분기를 식별합니다.

**폴더**는 이전 구조의 기본 디렉토리입니다.

Add Repository: testProject

Configure a repository to be used for this project.

Type \*

Terraform Configurations

▼ ⓘ

Repository \*

parnassusdemo/repository1

ⓘ

Branch \*

master

Folder

/Terraform

## vRealize Automation Cloud Assembly에서 Terraform 구성에 대한 설계

저장소 및 Terraform 구성 파일을 보유한 경우 이에 대한 vRealize Automation Cloud Assembly 템플릿을 설계할 수 있습니다.

- 1 사전 요구 사항
- 2 Terraform 런타임 버전 사용
- 3 설계에 Terraform 리소스 추가
- 4 클라우드 템플릿 배포

### 사전 요구 사항

버전 제어 저장소를 설정하고 통합합니다. vRealize Automation Cloud Assembly에서 Terraform 구성 준비의 내용을 참조하십시오.

### Terraform 런타임 버전 사용

Terraform 구성을 배포할 때 사용자가 사용할 수 있는 Terraform 런타임 버전을 정의할 수 있습니다. Terraform 구성에는 내부적으로 코딩된 버전 제약 조건도 포함될 수 있습니다.

허용되는 버전 목록을 생성하려면 **인프라 > 구성 > Terraform 버전**으로 이동합니다. 버전 0.12.x만 지원됩니다.

### 설계에 Terraform 리소스 추가

Terraform 구성이 포함된 클라우드 템플릿을 생성합니다.

- 1 vRealize Automation Cloud Assembly에서 **설계 > 클라우드 템플릿**으로 이동한 후 **새로 만들기 > Terraform**을 클릭합니다.

Terraform 구성 마법사가 나타납니다.

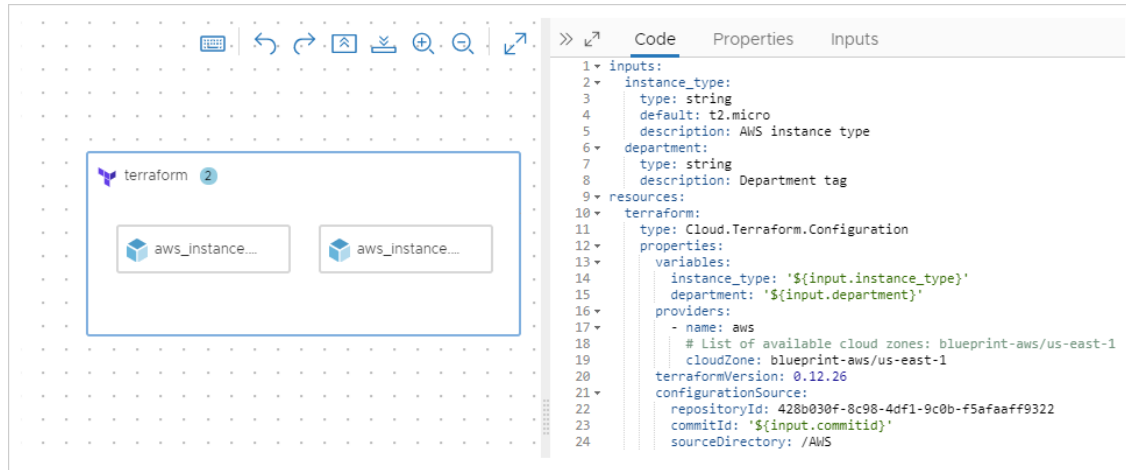
- 2 표시되는 메시지를 따릅니다.

마법사 페이지	설정	값
새 클라우드 템플릿	이름	설계에 식별 이름을 지정합니다.
	설명	설계의 목적을 설명합니다.

마법사 페이지	설정	값
구성 소스	프로젝트	Terraform 구성이 저장된 저장소 통합이 포함된 프로젝트를 선택합니다.
	저장소	Terraform 구성을 저장한 통합된 저장소를 선택합니다.
	커밋	저장소 커밋을 선택하거나, 항목을 비워 두어 저장소 헤드의 Terraform 구성을 사용합니다. Bitbucket 제한 사항 - Bitbucket 저장소 서버 구성으로 인해 선택 가능한 커밋 수가 잘릴 수 있습니다.
	소스 디렉토리	생성한 저장소 구조에서 하위 디렉토리를 선택합니다. 이전 설정에 표시된 예시 하위 디렉토리는 demo1, demo2 및 demo3입니다.
구성 완료	저장소	올바른 저장소 선택을 확인합니다.
	소스 디렉토리	올바른 디렉토리 선택을 확인합니다.
	Terraform 버전	Terraform 구성을 배포할 때 실행할 Terraform 런타임 버전을 선택합니다.
	제공자	Terraform 구성에 제공자 블록이 포함된 경우 이 클라우드 템플릿이 배포될 제공자 및 클라우드 영역을 확인합니다.  제공자가 없는 경우에는 문제가 발생하지 않습니다. 마법사를 완료한 후에는 템플릿 속성의 제공자 및 클라우드 영역을 편집하여 배포 대상을 추가하거나 변경하면 됩니다.
	변수	암호화할 중요 값(예: 암호)을 선택합니다.
	출력	설계 코드가 추가로 참조할 수 있는 표현식으로 변환되는 Terraform 구성에서 출력을 확인합니다.

### 3 생성을 클릭합니다.

Terraform 리소스는 배포할 Terraform 구성을 반영하는 vRealize Automation Cloud Assembly 코드와 함께 클라우드 템플릿 캔버스에 표시됩니다.



원하는 경우 다른 vRealize Automation Cloud Assembly 리소스를 클라우드 템플릿에 추가하여 Terraform 코드와 Terraform이 아닌 코드를 하이브리드 설계에 결합할 수 있습니다.

**참고** 저장소에서 Terraform 구성을 업데이트해도 변경 내용이 클라우드 템플릿으로 동기화되지 않습니다. 자동 동기화는 새로 추가된 중요 변수와 같은 보안 위험을 야기할 수 있습니다.

Terraform 구성 변경 내용을 캡처하려면 마법사를 다시 실행하고 새 커밋을 선택한 후 새 중요 변수를 식별합니다.

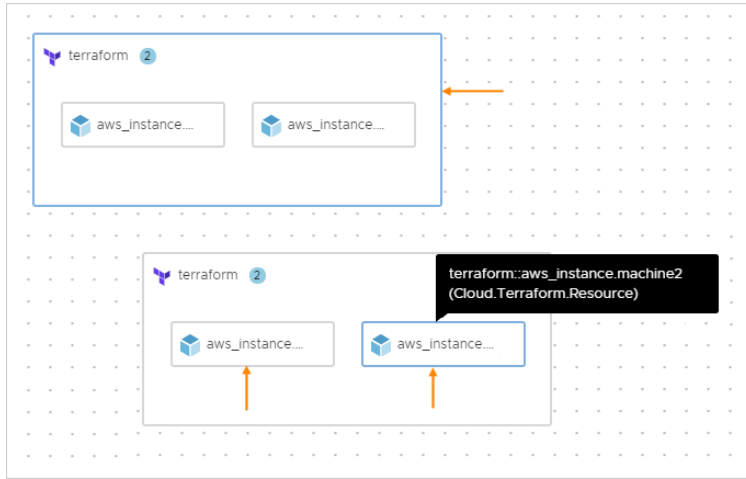
## 클라우드 템플릿 배포

클라우드 템플릿을 배포할 때 배포 **내역** 탭에서 할당 또는 생성 단계와 같은 이벤트를 확장하여 Terraform CLI의 메시지 로그를 검사할 수 있습니다.

승인 - 계획, 할당 또는 생성과 같은 필요한 Terraform 단계 외에도 승인 단계를 통해 vRealize Automation Cloud Assembly가 거버넌스를 도입합니다. 요청 승인에 대한 자세한 내용은 [Service Broker 승인 정책을 구성하는 방법](#)을 참조하십시오.

Timestamp	Status	Resource type	Resource name	Details
Aug 3, 202...	PLAN_FINISHED	Cloud.Terraform.Configurati...	terraform	Creating 2 Terraform resources, updating 0 Terraform resources, deleting 0 Terraform resources
Aug 3, 202...	PLAN_IN_PROGRESS	Cloud.Terraform.Configurati...	terraform	<a href="#">Hide Logs</a> <div> 2:24:23 PM * provider.random: version = "~&gt; 2.3"  2:24:23 PM Terraform has been successfully initialized!  2:24:28 PM Refreshing Terraform state in-memory prior to plan...  2:24:28 PM The refreshed state will be used to calculate this plan, but will not be persisted to local or remote state storage.  <a href="#">View as plain text</a> </div>
Aug 3, 202...	INITIALIZATION_FINISH...			
Aug 3, 202...	INITIALIZATION_IN_PRO...			

배포한 후에는 전체 Terraform 구성 요소를 나타내는 바깥쪽 리소스와 그 안에 Terraform에서 생성한 개별 구성 요소를 나타내는 하위 리소스를 볼 수 있습니다. 상위 Terraform 리소스는 하위 리소스의 수명 주기를 제어합니다.



## Terraform 구성에서 비밀 vRealize Automation Cloud Assembly 속성을 사용하는 방법

vRealize Automation Cloud Assembly 클라우드 템플릿 설계에 추가하는 Terraform 구성에 비밀, 암호화된 값을 적용할 수 있습니다.

- 1 Git 저장소에서 비밀 속성을 변수로 참조하는 Terraform 구성 소스 파일을 추가합니다.

아래 Terraform 구성 소스 예제에서는 API 및 애플리케이션 키가 비밀 변수입니다.

```
variable "datadog_api_key" {
  description = "Datadog API Key"
}

variable "datadog_app_key" {
  description = "Datadog App Key"
}

provider "datadog" {
  api_key = "${var.datadog_api_key}"
  app_key = "${var.datadog_app_key}"
}

# Create a new monitor
resource "datadog_monitor" "default" {
  # ...
}

# Create a new timeboard
resource "datadog_timeboard" "default" {
  # ...
}
```

- 2 vRealize Automation Cloud Assembly에서 **인프라 > 관리 > 비밀**로 이동하고 비밀 속성 값을 입력합니다.

비밀 이름과 해당 값을 추가합니다. 이름의 경우에는 Terraform 소스의 변수 이름과 동일한 이름을 입력하는 것이 가장 간편합니다.

필요한 경우 비밀 vRealize Automation Cloud Assembly 속성을 생성하고 참조하는 방법에서 자세한 내용을 참조하십시오.

Name	Project	Value
datadog_api_key	Terraform	*****
datadog_app_key	Terraform	*****

- 3 vRealize Automation Cloud Assembly에서 클라우드 템플릿에 사용할 Terraform 구성을 가져옵니다.

**설계 > 클라우드 템플릿**으로 이동하여 **새로 만들기 > Terraform**을 클릭합니다.

**참고** 마법사의 마지막 페이지에서 선택할 수 있도록 변수가 표시되더라도 비밀 변수를 민감한 것으로 설정할 필요는 없습니다. 비밀 vRealize Automation Cloud Assembly 변수는 이미 암호화되어 있으므로 마법사가 적용하는 암호화가 필요하지 않습니다.

필요한 경우 vRealize Automation Cloud Assembly에서 Terraform 구성에 대한 설계에서 자세한 내용을 참조하십시오.

예제 클라우드 템플릿은 다음 코드와 유사합니다.

```
inputs:
  datadog_api_key:
    type: string
    description: Datadog API Key
  datadog_app_key:
    type: string
    description: Datadog App Key
resources:
  terraform:
    type: Cloud.Terraform.Configuration
    properties:
      variables:
        datadog_api_key: '${input.datadog_api_key}'
        datadog_app_key: '${input.datadog_app_key}'
      providers: []
      terraformVersion: 0.12.29
      configurationSource:
        repositoryId: 0fbf8f5e-54e1-4da3-9508-2b701gf25f51
        commitId: ed12424b249aa50439kr1c268942a4616bd751b6
        sourceDirectory: datadog
```

4 코드 편집기에서 비밀 값에 대해 아래와 같이 input을 secret으로 수동으로 변경합니다.

```
terraform:
  type: Cloud.Terraform.Configuration
  properties:
    variables:
      datadog_api_key: '${secret.datadog_api_key}'
      datadog_app_key: '${secret.datadog_app_key}'
```

5 코드의 inputs: 섹션에서 secret 속성에 대한 바인딩으로 대체된 input 항목을 제거합니다.

## vRealize Automation의 Terraform 구성에 대해 알아보기

Terraform 구성을 vRealize Automation에 리소스로 포함하는 경우 특정 제한 사항 및 문제 해결에 대해 숙지하십시오.

### Terraform 구성에 대한 제한 사항

- Terraform 구성을 사용하여 설계를 검증하는 경우 테스트 버튼은 vRealize Automation Cloud Assembly 구문을 확인하지만 기본 Terraform 코드 구문은 확인하지 않습니다.  
또한 테스트 버튼은 Terraform 구성과 연결된 커밋 ID를 검증하지 않습니다.
- Terraform 구성이 포함된 클라우드 템플릿의 경우 템플릿을 다른 프로젝트에 복제하려면 다음 해결 방법이 필요합니다.
  - a 새 프로젝트의 **통합** 탭에서 통합에 대한 repositoryId를 복사합니다.
  - b 클론 템플릿을 엽니다. 코드 편집기에서 repositoryId를 복사한 항목으로 바꿉니다.
- 버전 제어 저장소에 구성 파일이 포함된 Terraform 상태 파일을 포함하지 않습니다.  
terraform.tfstate가 있으면 배포 중에 오류가 발생합니다.

### 상위 Terraform 리소스에 대해 지원되는 2일차 작업

상위 Terraform 리소스의 경우 Terraform 상태 파일을 보거나 새로 고칠 수 있습니다. 상태 파일 작업에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업](#)에서 전체 작업 목록을 참조하십시오.

### 하위 리소스에 대해 지원되는 2일차 작업

Terraform 구성을 배포한 후 하위 리소스에서 2일차 작업을 사용할 수 있게 되기까지 최대 20분이 소요될 수 있습니다.

Terraform 구성의 하위 리소스의 경우 다음과 같은 일부 2일차 작업만 지원됩니다. 작업에 대한 자세한 내용은 [vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업](#)의 전체 작업 목록에서 확인하십시오.

제공자	Terraform 리소스 유형	지원되는 2일차 작업
AWS	aws_instance	전원 켜기
		전원 끄기
		재부팅
		재설정
Azure	azurerm_virtual_machine	전원 켜기
		전원 끄기
		다시 시작
		일시 중단
vSphere	vsphere_virtual_machine	전원 켜기
		전원 끄기
		재부팅
		재설정
		종료
		일시 중단
		스냅샷 생성
		스냅샷 삭제
GCP	google_compute_instance	스냅샷 되돌리기
		전원 켜기
		전원 끄기
		스냅샷 생성
		스냅샷 삭제

## 2일차 작업 가용성 문제 해결

누락되었거나 비활성화된 OOTB(기본 제공) 2일차 작업은 문제 해결이 필요할 수 있습니다.

문제	원인	해결 방법
Terraform 리소스의 [작업] 메뉴에 필요한 OOTB 2일차 작업이 없습니다.	이전 목록에 언급된 것처럼 제공자 및 리소스 유형에 대해 작업이 지원되지 않을 수 있습니다. 또는 리소스 검색 및 리소스 캐싱의 타이밍으로 인해 작업이 표시되는 데 최대 20분이 소요될 수 있습니다.	설계에서 제공자 및 리소스 유형을 확인합니다. 데이터 수집이 완료될 때까지 20분 정도 기다립니다.
데이터 수집을 수행하는 데 20분이 소요될 경우에도 Terraform 리소스에는 필요한 2일차 작업이 없습니다.	리소스 검색 문제로 인해 작업이 표시되지 않습니다. 이 문제는 리소스를 프로젝트 외부의 클라우드 영역에서 실수로 생성한 경우에 발생합니다. 예를 들어 프로젝트에 클라우드 계정 및 지역 us-east-1 클라우드 영역만 포함되어 있지만 Terraform 구성에는 us-west-1의 제공자 블록이 포함되어 있으며 설계 시 이를 변경하지 않았습다. 또 다른 가능성은 데이터 수집이 작동하지 않기 때문입니다.	설계에서 클라우드 영역을 기준으로 프로젝트 클라우드 영역을 확인합니다. <b>인프라 &gt; 연결 &gt; 클라우드 계정</b> 으로 이동하고 클라우드 계정에 대한 데이터 수집 상태 및 마지막 성공 수집 시간을 확인합니다.
리소스 상태와 데이터 수집에 명백한 문제가 없더라도 2일차 작업이 비활성화됩니다(회색).	가끔씩 간헐적으로 타이밍 문제 및 데이터 수집 실패가 발생하는 것으로 알려져 있습니다.	이 문제는 20분 내에 해결되어야 합니다.
리소스 상태에 따라 활성이어야 하는, 잘못된 2일차 작업이 비활성화되었습니다. 예를 들어, 제공자 인터페이스를 사용하여 리소스 전원이 꺼진 경우에도 전원 켜기가 사용되도록 설정되고 전원 켜기가 비활성화되었습니다.	데이터 수집 타이밍으로 인해 일시적인 불일치가 발생할 수 있습니다. 외부 vRealize Automation에서 전원 상태를 변경하는 경우 변경 내용을 올바르게 반영하는 데 시간이 소요됩니다.	20분까지 기다립니다.

## vRealize Automation에서 사용자 지정 Terraform 제공자 사용

사용자 지정 Terraform 제공자를 생성한 경우 이를 사용하려면 다음 단계를 수행합니다.

- 1 Git 버전 제어 저장소의 기본 Terraform 디렉토리 아래에 다음 하위 디렉토리 구조를 추가합니다.

```
terraform.d/plugins/linux_amd64
```

- 2 사용자 지정 Terraform 제공자 Go 바이너리를 linux\_amd64 디렉토리에 추가합니다.

기본적으로 terraform init는 해당 디렉토리에서 사용자 지정 제공자 플러그인을 검색합니다.

**참고** VMware는 사용자 지정 Terraform 제공자가 실행되지 못하고 no such file or directory 메시지를 게시하는 경우를 확인합니다.

이런 경우 CGO를 비활성화(0으로 설정)한 상태에서 사용자 지정 제공자 Go 바이너리를 다시 컴파일해보십시오. CGO는 C 코드를 호출하는 Go 패키지용입니다.

## vRealize Automation Cloud Assembly 마켓플레이스를 사용하는 방법

리소스 라이브러리를 점프스타트하려면 vRealize Automation Cloud Assembly 마켓플레이스에서 파일을 다운로드합니다. 마켓플레이스에는 완성된 클라우드 템플릿과 개방형 가상화 이미지가 제공됩니다.

## 마켓플레이스에 액세스하는 방법

vRealize Automation Cloud Assembly에서 **인프라 > 연결 > 통합**을 선택합니다. **통합 추가**를 클릭하고 **My VMware**를 클릭한 후 My VMware 계정 자격 증명을 제공합니다.

## 마켓플레이스 클라우드 템플릿 파일을 다운로드하고 사용하는 방법

**마켓플레이스** 탭에서 **가져오기**를 클릭하고 클라우드 템플릿 EULA에 동의합니다. 그런 다음, 템플릿을 vRealize Automation Cloud Assembly 프로젝트에 추가하거나 간단히 다운로드할 수 있습니다. **설계** 탭에서 클라우드 템플릿을 업로드할 수 있습니다.

프로젝트 기반 예제에서는 빅데이터 작업의 프로젝트 관리자라고 가정합니다. 팀을 지원하기 위해, 팀 프로젝트에 추가할 마켓플레이스 Hadoop 템플릿을 찾습니다. 그리고 리소스 환경에 맞게 클라우드 템플릿을 사용자 지정하고 릴리스합니다. 그런 다음, 팀에서 배포할 수 있도록 템플릿을 vRealize Automation Service Broker 카탈로그로 가져옵니다.

## 마켓플레이스 이미지 파일을 다운로드하고 사용하는 방법

**마켓플레이스** 탭에서 **가져오기**를 클릭하고 OVF 또는 OVA 이미지 EULA에 동의합니다. 나중에, OVF 또는 OVA 이미지를 다운로드하여 클라우드 템플릿 코드에서 참조할 수 있습니다.

이전 예시를 계속 진행하다 보면, 팀에 Hadoop 자체 버전에 대한 액세스 권한이 필요할 수 있습니다. Hadoop OVF를 다운로드하여 vCenter Server 콘텐츠 라이브러리와 같은 클라우드 계정 리소스에 추가합니다. 그런 다음, OVF 이미지를 가리켜야 하는 템플릿 코드를 업데이트합니다.

# vRealize Automation Cloud Assembly 배포 관리

# 7

vRealize Automation Cloud Assembly 클라우드 템플릿 개발자는 [배포] 탭을 사용하여 배포를 관리합니다. 실패한 프로비저닝 프로세스의 문제를 해결하고, 내용을 변경하고, 사용되지 않는 배포를 제거할 수 있습니다.

배포는 클라우드 템플릿의 프로비저닝된 인스턴스입니다. [배포] 탭에는 성공한 배포와 실패한 배포가 나열됩니다. 이 페이지를 사용하여 성공한 배포를 관리하거나 실패한 요청에 대한 문제 해결을 시작할 수 있습니다.

## 배포 카드 사용

카드 목록을 사용하여 배포를 찾고 관리할 수 있습니다. 특정 배포를 필터링하거나 검색한 후 해당 배포에 대해 작업을 실행할 수 있습니다.

### 1 특성을 기반으로 요청을 필터링합니다.

예를 들어 소유자, 프로젝트, 리스 만료 날짜 또는 기타 필터링 옵션을 기반으로 필터링할 수 있습니다. 또는 특정 태그가 있는 두 프로젝트에 대한 모든 배포를 찾을 수도 있습니다. 프로젝트 및 태그 예제에 대한 필터를 구성하는 경우 결과는 다음 조건을 준수합니다. (Project1 OR Project2) AND Tag1.

필터 창에 표시되는 값은 현재 배포 목록에 따라 달라집니다.

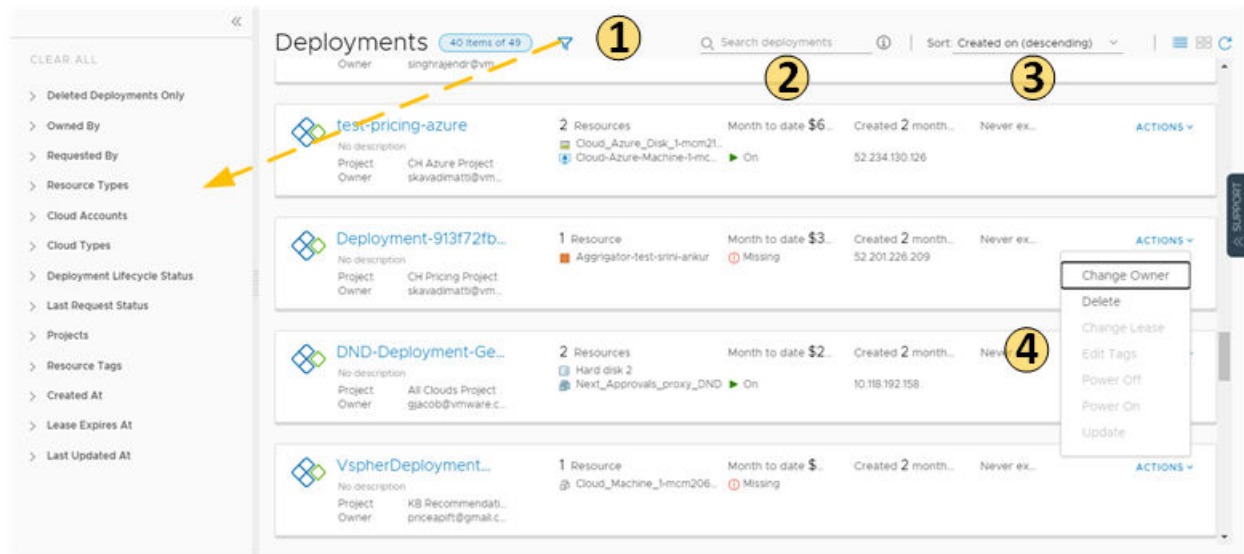
대부분의 필터와 필터를 사용하는 방식은 비교적 명확합니다. 하지만 배포 수명 주기 상태 및 마지막 요청 상태 필터를 효과적으로 사용하려면 이러한 필터의 작동 방식을 이해해야 합니다. 이러한 필터에 대한 추가 정보가 아래에 나와 있습니다.

### 2 키워드 또는 요청자를 기반으로 배포를 검색합니다.

### 3 시간 또는 이름별로 목록을 정렬합니다.

### 4 사용하지 않는 배포를 삭제하여 리소스를 회수하는 작업을 포함하여, 배포에 대해 배포 수준의 작업을 실행합니다.

배포 비용, 만료 날짜 및 상태도 볼 수 있습니다.



## [배포 수명 주기 상태] 및 [마지막 요청 상태] 필터 사용

[배포 수명 주기 상태] 및 [마지막 요청 상태] 필터는 개별적으로 또는 조합하여 사용할 수 있습니다(특히 많은 수의 배포를 관리하는 경우).

- [배포 수명 주기 상태]는 관리 작업을 기반으로 배포의 현재 상태를 필터링합니다.

이 필터는 삭제된 배포에 사용할 수 없습니다.

필터 창에 표시되는 값은 나열된 배포의 현재 상태에 따라 달라집니다. 가능한 모든 값이 표시되지 않을 수 있습니다. 다음 목록은 가능한 모든 값입니다. 업데이트에는 2일차 작업이 포함됩니다.

- 생성 - 성공
  - 생성 - 진행 중
  - 생성 - 실패
  - 업데이트 - 성공
  - 업데이트 - 진행 중
  - 업데이트 - 실패
  - 삭제 - 진행 중
  - 삭제 - 실패
  - [마지막 요청 상태]는 배포에서 실행된 마지막 작업을 필터링합니다.
- 이 필터는 삭제된 배포에 사용할 수 없습니다.

필터 창에 표시되는 값은 나열된 배포에서 실행된 마지막 작업에 따라 달라집니다. 가능한 모든 값이 표시되지 않을 수 있습니다. 다음 목록은 가능한 모든 값입니다.

- 보류 중. 요청의 첫 번째 단계로, 작업이 제출되었지만 배포 프로세스가 아직 시작되지 않았습니다.
- 실패. 배포 프로세스의 특정 단계에서 요청이 실패했습니다.
- 취소됨. 배포 프로세스 처리 중에(아직 완료 전) 사용자가 요청을 취소했습니다.
- 성공. 요청이 배포를 생성, 업데이트 또는 삭제했습니다.
- 진행 중. 배포 프로세스가 현재 실행되는 중입니다. 추가 배포 상태(예: [배포 기록] 탭에 표시되는 초기화 및 완료)는 필터로 제공되지 않지만 [진행 중] 필터를 사용하여 해당 상태의 배포를 찾을 수 있습니다.
- 승인 보류 중. 요청이 하나 이상의 승인 정책을 트리거했습니다. 프로세스가 승인 요청에 대한 응답을 기다리는 중입니다.
- 승인 거부됨. 트리거된 승인 정책에서 승인자가 요청을 거부했습니다. 요청이 계속되지 않습니다.

다음 예는 필터를 개별적으로 또는 함께 사용하는 방법을 설명합니다.

- 실패한 모든 삭제 요청을 찾으려면 [배포 수명 주기 상태] 필터에서 **삭제 - 실패**를 선택합니다.
- 승인 대기 중인 모든 요청을 찾으려면 [마지막 요청 상태] 필터에서 **승인 보류 중**을 선택합니다.
- 승인 요청이 아직 보류 중인 삭제 요청을 찾으려면 [배포 수명 주기 상태] 필터에서 **삭제 - 진행 중**을 선택하고 [마지막 요청 상태] 필터에서 **승인 보류 중**을 선택합니다.

본 장은 다음 항목을 포함합니다.

- vRealize Automation Cloud Assembly에서 배포를 모니터링하는 방법
- vRealize Automation Cloud Assembly 배포 실패 시 수행할 수 있는 작업
- 완료된 vRealize Automation Cloud Assembly 배포의 수명주기를 관리하는 방법
- vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업

## vRealize Automation Cloud Assembly에서 배포를 모니터링하는 방법

vRealize Automation Cloud Assembly 클라우드 템플릿을 배포한 후, 요청을 모니터링하여 리소스가 프로비저닝되고 실행 중인지 확인할 수 있습니다. 배포 카드부터 리소스 프로비저닝을 확인할 수 있습니다. 다음으로, 배포 세부 정보를 검토할 수 있습니다. 마지막으로, 삭제된 배포는 삭제 후 최대 90일 동안 볼 수 있고 필터링할 수 있습니다.

### 절차

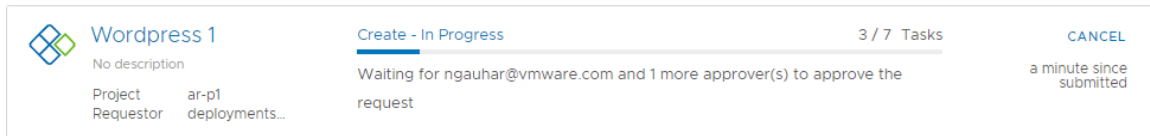
- 1 **배포**를 클릭하고 필요한 경우 필터 및 검색을 사용하여 처리 중인 배포 카드를 찾습니다.

## 2 카드 상태를 검토합니다.

배포가 진행 중인 경우 프로세스 표시줄에 남은 작업 수가 표시됩니다. 배포가 성공적으로 완료되면 카드에 배포에 대한 기본 세부 정보가 표시됩니다.

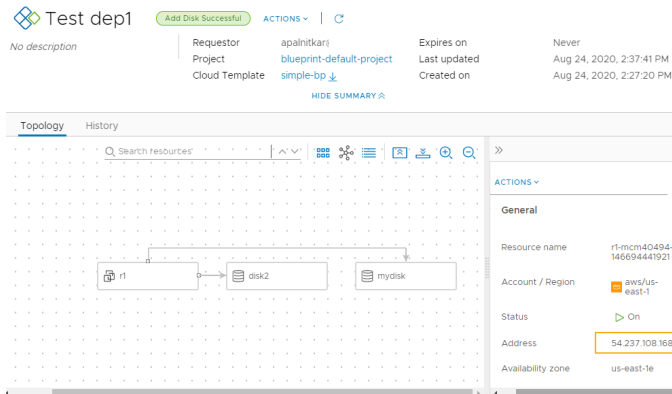


요청에 대한 승인 정책이 트리거되면 하나 이상의 승인자 이름과 함께 요청이 진행 중 상태로 표시됩니다. 승인 정책은 관리자가 vRealize Automation Service Broker에서 정의합니다. 승인자는 정책에 정의되어 있습니다. 승인자는 vRealize Automation Service Broker에서 요청을 승인합니다. 2일차 작업에 대한 승인이 발생할 수도 있습니다.



## 3 리소스가 배포된 위치를 확인하려면 배포 이름을 클릭하고 [토폴로지] 페이지에서 세부 정보를 검토합니다.

기본 구성 요소의 IP 주소가 필요할 가능성이 높습니다. 각 구성 요소를 클릭하면 해당 구성 요소와 관련된 정보가 제공됩니다. 이 예에서는 IP 주소가 강조 표시됩니다.



외부 링크를 사용할 수 있는지 여부는 클라우드 제공자에 따라 다릅니다. 외부 링크를 사용할 수 있는 경우에는 해당 제공자의 자격 증명이 있어야 구성 요소에 액세스할 수 있습니다.

### 다음에 수행할 작업

- 배포를 변경할 수 있습니다. 완료된 vRealize Automation Cloud Assembly 배포의 수명주기를 관리하는 방법의 내용을 참조하십시오.
- 배포에 실패하는 경우 vRealize Automation Cloud Assembly 배포 실패 시 수행할 수 있는 작업 항목을 참조하십시오.

## vRealize Automation Cloud Assembly 배포 실패 시 수행할 수 있는 작업

배포 요청은 여러 가지 이유로 실패할 수 있습니다. 네트워크 트래픽 문제, 대상 클라우드 제공자의 리소스 부족 또는 배포 규격 결함 때문일 수 있습니다. 또는 배포가 성공했지만 작동하지 않는 것처럼 보일 수도 있습니다. vRealize Automation Cloud Assembly를 사용하면 배포를 검사하고, 오류 메시지를 검토하여 문제가 환경 때문인지, 요청된 워크로드 규격 때문인지 아니면 다른 이유 때문인지 파악할 수 있습니다.

워크플로를 사용하여 조사를 시작합니다. 진행 결과 일시적인 환경 문제가 실패의 원인일 수 있습니다. 이러한 유형의 문제는 상태가 개선된 것을 확인한 이후에 요청을 다시 배포하여 해결할 수 있습니다. 그 이외의 경우에는 조사할 때 다른 부분을 더 세부적으로 검토해야 할 수 있습니다.

프로젝트 멤버는 vRealize Automation Cloud Assembly에서 요청 세부 정보를 검토할 수 있습니다.

### 절차

- 요청이 실패했는지 확인하려면 **배포** 탭을 클릭하고 배포 카드를 찾습니다.

The screenshot shows a card for a deployment named 'WP-ROR1.0'. On the left, it lists '설명 없음' (No description), '프로젝트 test' (Project test), and '요청자 smasaru...' (Requester smasaru...). The main area has a red header 'Create Failed' and a progress indicator '2 / 8 작업' (2 / 8 tasks). Below this, a red error message is displayed: 'java.lang.IllegalArgumentException: com.vmware.xenon.common.LocalizableValidationException: Cannot find a profile for compute network [Cloud\_Network\_2-mcm112748-99510009947] of type [PRIVATE] that would match endpoint...'. A '표시' (Show) link with a chevron icon is at the bottom left. On the right, there is a '작업' (Task) dropdown menu and a note '5일 제한한 후' (After 5-day limit).

실패한 배포는 카드에 표시됩니다.

- 오류 메시지를 검토합니다.
- 자세한 내용을 보려면 배포 이름을 클릭하여 배포 세부 정보를 표시합니다.

## 2 배포 세부 정보 페이지에서 기록 탭을 클릭합니다.

WP - ROR2 Create Failed ACTIONS ⌵ ⌵

No description

Requestor: fritz  
Project: Tiger Team  
Cloud Template: WordPress Template [⌵](#)

Expires on: Never  
Last updated: Sep 9, 2020, 12:06:42 PM  
Created on: Sep 9, 2020, 12:06:38 PM

[HIDE SUMMARY](#)

Topology History

Sep 9, 2020, 12:06:42 PM ! CREATE fritz 2.a

Create Failed Requested by: fritz [Provisioning diagram](#) 2.c

Events Request details

Timestamp	Status	Resource type	Resource name	Details <span>2.b</span>
Sep 9, 2020, ...	REQUEST_FAILED			Could not find any profile to match network 'WP-Network-Private' of type 'EXISTING' with constraints '[type:isolated-net, env:dev]'.
Sep 9, 2020, ...	COMPLETION_FINISHED			
Sep 9, 2020, ...	COMPLETION_IN_PROGRE...			
Sep 9, 2020, ...	ALLOCATE_FAILED	Cloud.Network	WP-Network-Private	Could not find any profile to match...

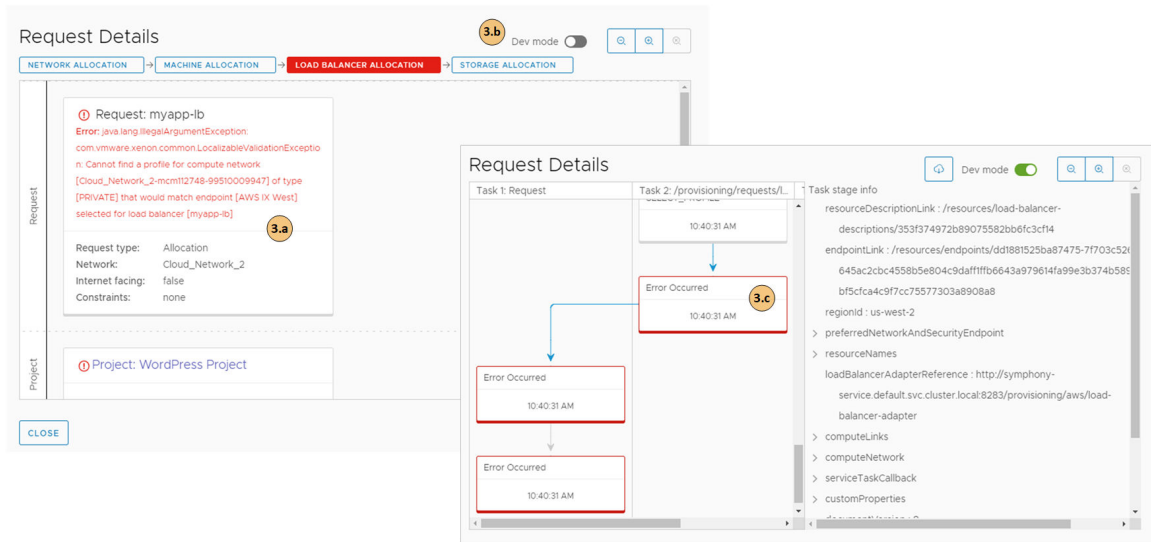
8 Events

- 이벤트 트리를 검토하여 프로비저닝 프로세스가 실패한 위치를 확인합니다. 이 트리는 배포를 수정했지만 변경이 실패한 경우에 유용합니다.  
  
이 트리는 배포 작업을 실행할 때도 표시됩니다. 트리를 사용하여 실패한 변경 사항 문제를 해결할 수 있습니다.
- 세부 정보에는 오류 메시지의 자세한 버전이 제공됩니다.
- 요청된 항목이 vRealize Automation Cloud Assembly 클라우드 템플릿인 경우 메시지 오른쪽의 링크를 클릭하면 **요청 세부 정보**를 볼 수 있는 vRealize Automation Cloud Assembly가 열립니다.

## 3 요청 세부 정보에는 문제를 조사할 수 있도록 실패한 구성 요소에 대한 프로비저닝 워크플로가 제공됩니다.

요청 기록은 48시간 동안 보존됩니다.

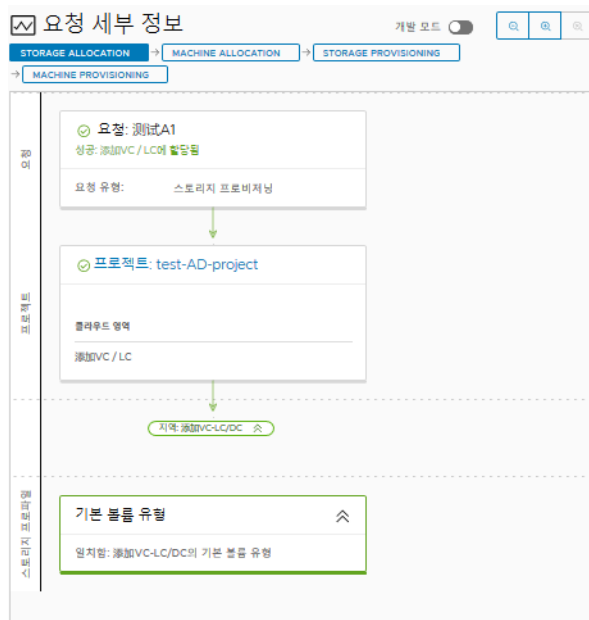
삭제 후 최대 90일 동안 삭제된 배포 기록 보기 및 필터링



- a 오류 메시지를 검토합니다.
  - b 개발 모드를 켜면 간단한 프로비저닝 워크플로와 더 자세한 순서도 간에 전환할 수 있습니다.
  - c 카드를 클릭하여 배포 스크립트를 검토합니다.
- 4 오류를 해결하고 클라우드 템플릿을 다시 배포합니다.
- 템플릿 구조에 오류가 있거나 인프라 구성 방식과 관련이 있을 수 있습니다.

#### 다음에 수행할 작업

오류가 해결되고 클라우드 템플릿이 배포되면 [요청 세부 정보]에서 다음 예와 유사한 정보를 볼 수 있습니다. 요청 세부 정보를 보려면 **인프라 > 활동 > 요청**을 선택하십시오.



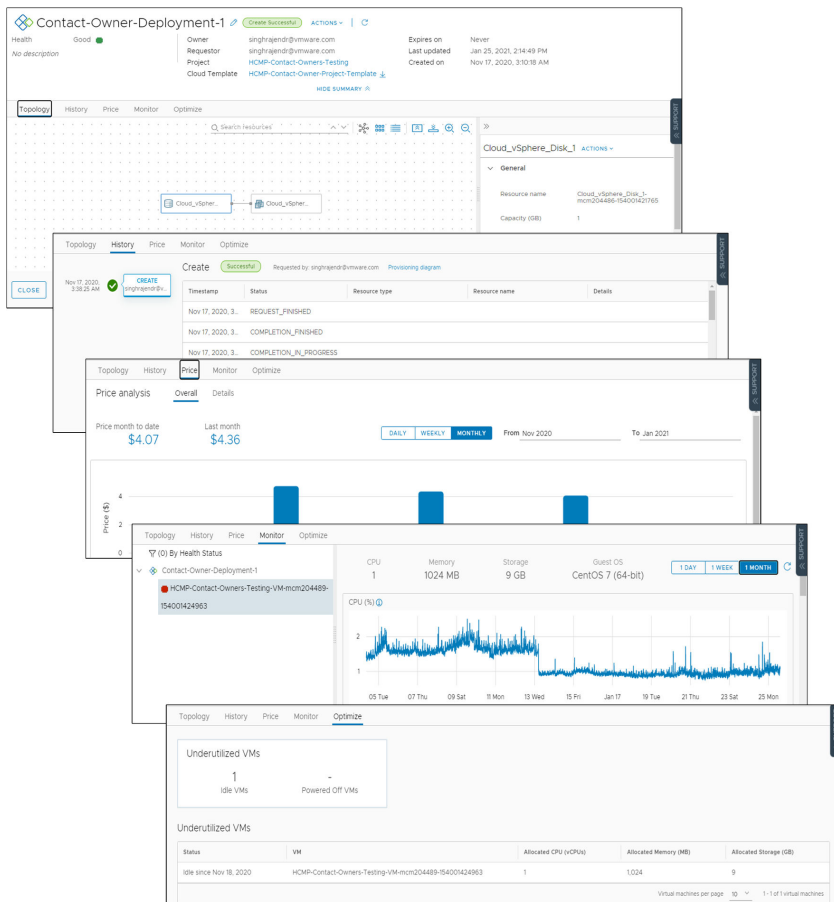
# 완료된 vRealize Automation Cloud Assembly 배포의 수명주기를 관리하는 방법

배포를 프로비저닝하고 실행한 후 배포를 관리하기 위해 실행할 수 있는 몇 가지 작업이 있습니다. 수명주기 관리에는 전원 켜기 또는 끄기, 배포 크기 조정 및 삭제 등이 포함될 수 있습니다. 개별 구성 요소에 대해 다양한 작업을 실행하여 관리할 수도 있습니다.

## 절차

- 1 배포를 클릭하고 배포를 찾습니다.
- 2 배포 세부 정보에 액세스하려면 배포 이름을 클릭합니다.

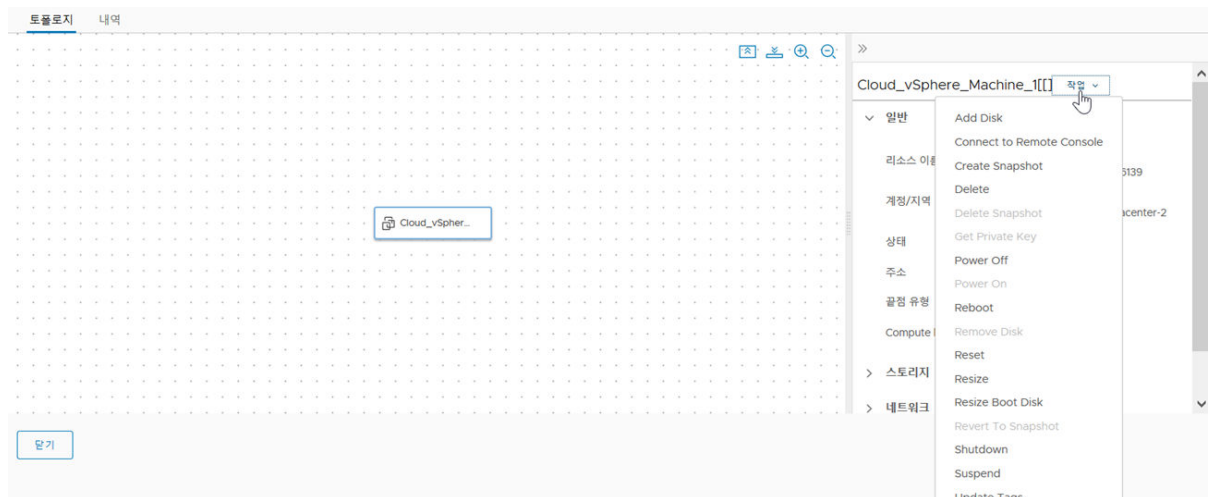
배포 세부 정보를 사용하여 리소스가 배포되는 방식과 변경된 사항을 파악합니다. 가격 책정 정보, 배포의 현재 상태 및 수정해야 하는 리소스가 있는지도 볼 수 있습니다.



- **토폴로지** 탭. [토폴로지] 탭을 사용하면 배포 구조와 리소스를 이해할 수 있습니다.
- **기록** 탭. [기록] 탭에는 모든 프로비저닝 이벤트 및 요청된 항목이 배포된 이후에 실행하는 작업과 관련된 모든 이벤트가 포함됩니다. 프로비저닝 프로세스에 문제가 있는 경우, [기록] 탭에 있는 이벤트가 장애를 해결하는 데 도움이 됩니다.

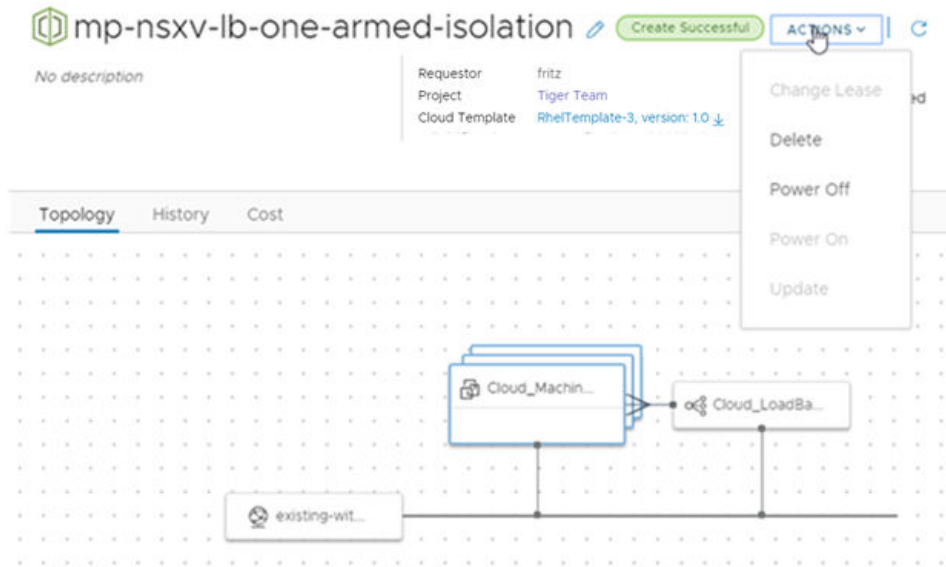
- **가격 책정** 탭. 가격 책정 카드를 사용하여 배포로 인해 조직에 비용이 얼마나 드는지 이해할 수 있습니다. 가격 책정 정보는 vRealize Operations Manager 또는 CloudHealth 통합을 기반으로 합니다.
  - **모니터링** 탭. [모니터링] 탭 데이터는 vRealize Operations Manager의 데이터를 기반으로 배포 상태에 대한 정보를 제공합니다.
  - **최적화** 탭. [최적화] 탭은 배포에 대한 활용률 정보를 제공하며 리소스 사용을 최적화하기 위해 리소스를 회수하거나 달리 수정하기 위한 제안을 제공합니다. 최적화 정보는 vRealize Operations Manager의 데이터를 기반으로 합니다.
- 3 현재 구성에서 배포 비용이 너무 많이 들어서 구성 요소의 크기를 조정하려는 경우, [토폴로지] 페이지에서 구성 요소를 선택한 다음 구성 요소 페이지에서 **작업 > 크기 조정**을 선택합니다.

사용 가능한 작업은 구성 요소, 클라우드 계정 및 사용 권한에 따라 다릅니다.



- 4 개발 수명주기의 일부로 배포 중 하나가 더 이상 필요하지 않습니다. 배포를 제거하고 리소스를 회수하려면 **작업 > 삭제**를 선택합니다.

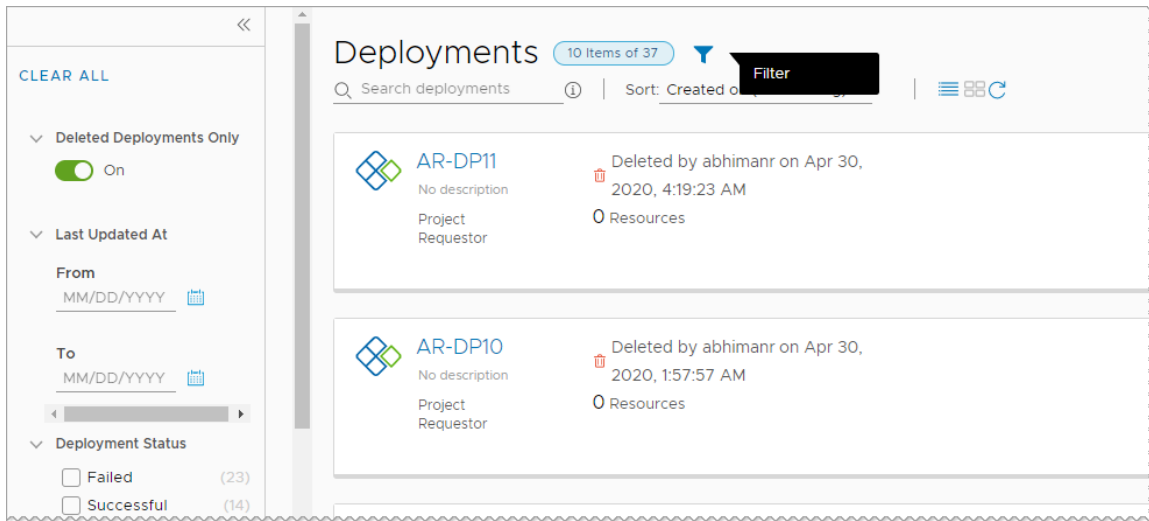
사용 가능한 작업은 배포 상태에 따라 다릅니다.



- 5 삭제된 배포를 보려면 **배포** 탭에서 필터를 클릭한 다음, **삭제된 배포만** 토글을 켭니다.

이제 배포 목록이 삭제된 것으로 제한됩니다. 특정 배포의 기록을 검토할 수 있습니다. 예를 들어 삭제된 시스템의 이름을 검색할 수 있습니다.

삭제된 배포는 90일 동안 나열됩니다.



#### 다음에 수행할 작업

가능한 작업에 대한 자세한 내용은 vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업 항목을 참조하십시오.

## vRealize Automation Cloud Assembly 배포에서 실행할 수 있는 작업

클라우드 템플릿을 배포한 후 vRealize Automation Cloud Assembly에서 작업을 실행하여 리소스를 관리할 수 있습니다. 사용 가능한 작업은 리소스 유형 그리고 해당 작업이 특정 클라우드 계정 또는 통합 플랫폼에서 지원되는지 여부에 따라 달라집니다.

가능한 작업도 관리자가 권한을 부여하는 대상에 따라 달라집니다.

관리자나 프로젝트 관리자는 vRealize Automation Service Broker에서 2일차 작업 정책을 설정할 수 있습니다. [소비자에게 Service Broker 2일차 작업 정책에 대한 권한을 부여하는 방법](#)을 참조하십시오.

목록에 포함되지 않은 작업이 표시될 수도 있습니다. 관리자가 추가한 사용자 지정 작업일 수 있습니다.

예: 가상 시스템의 vMotion 작업을 수행하기 위한 vRealize Automation Cloud Assembly 사용자 지정 작업을 생성하는 방법.

표 7-1. 가능한 작업 목록

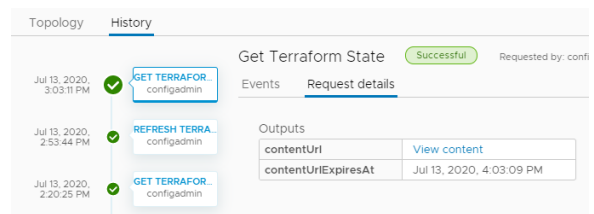
작업	적용 대상 리소스 유형	대상 클라우드 계정 또는 통합	설명
디스크 추가	시스템	<ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>기존 가상 시스템에 디스크를 추가합니다.</p> <p>Azure 시스템에 디스크를 추가하면 디스크가 해당 시스템이 포함되어 있는 리소스 그룹에 배포됩니다. 영구 디스크인 경우 새 리소스 그룹이 생성됩니다.</p> <p>VMware SDRS(Storage DRS)를 사용하고 스토리지 프로파일에 데이터스토어 클러스터가 구성된 경우, SDRS의 디스크를 vSphere 시스템에 추가할 수 있습니다.</p>
취소	<ul style="list-style-type: none"> <li>배포</li> <li>배포의 다양한 리소스 유형</li> </ul>	<ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>요청이 처리되는 동안 배포 또는 리소스에 대한 2일차 작업 또는 배포를 취소합니다.</p> <p>배포 카드나 배포 세부 정보에서 요청을 취소할 수 있습니다. 요청을 취소하는 경우 해당 요청은 <b>배포</b> 탭에 실패한 요청으로 표시됩니다. <b>삭제</b> 작업을 사용하여 배포된 리소스를 해제하고 배포 목록을 정리합니다.</p> <p>너무 오래 실행되고 있다고 생각되는 요청을 취소하는 것도 배포 시간을 관리하는 한 가지 방법입니다. 그러나 프로젝트에서 <b>요청 시간 초과</b>를 설정하는 것이 더 효율적입니다. 기본 시간 초과 값은 2시간입니다. 프로젝트에 대한 워크로드 배포에 더 많은 시간이 필요한 경우 더 긴 시간을 설정할 수 있습니다.</p>
리스 변경	배포	<ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>리스 만료 날짜 및 시간을 변경합니다.</p> <p>리스가 만료되면 배포가 삭제되고 리소스가 회수됩니다.</p> <p>리스 정책은 vRealize Automation Service Broker에 설정됩니다.</p>

표 7-1. 가능한 작업 목록 (계속)

작업	적용 대상 리소스 유형	대상 클라우드 계정 또는 통합	설명
소유자 변경	배포	<ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>배포 소유자를 선택한 사용자로 변경합니다. 선택한 사용자는 요청을 배포한 동일한 프로젝트의 멤버여야 합니다. 서비스 관리자 또는 프로젝트 관리자를 소유자로 할당하려면 이들을 프로젝트 멤버로 추가해야 합니다.</p> <p>클라우드 템플릿 디자이너가 템플릿을 배포하는 경우, 디자이너는 요청자이자 소유자입니다. 하지만 요청자는 다른 프로젝트 멤버를 소유자로 만들 수 있습니다.</p> <p>정책을 사용하여 소유자가 배포를 통해 수행할 수 있는 작업을 제어하고 더 제한적이거나 덜 제한적인 사용 권한을 부여할 수 있습니다.</p>
보안 그룹 변경	시스템	<ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>	<p>배포의 시스템 네트워크에 보안 그룹을 연결하고 분리할 수 있습니다. 변경 작업은 <b>NSX-V</b> 및 <b>NSX-T</b>에 대한 기존 및 주문형 보안 그룹에 적용됩니다. 이 작업은 시스템 클러스터가 아닌 단일 시스템에만 사용할 수 있습니다.</p> <p>보안 그룹을 시스템 네트워크에 연결하려면 배포에 보안 그룹이 있어야 합니다.</p> <p>배포에 있는 모든 시스템의 모든 네트워크에서 보안 그룹을 분리하는 경우 보안 그룹이 배포에서 제거되지 않습니다.</p> <p>이러한 변경 내용은 네트워크 프로파일의 일부로 적용된 보안 그룹에는 영향을 주지 않습니다.</p> <p>이 작업은 시스템을 다시 생성하지 않고 시스템의 보안 그룹 구성을 변경합니다. 이는 비파괴적인 변경입니다.</p> <ul style="list-style-type: none"> <li>시스템의 보안 그룹 구성을 변경하려면 토폴로지 창에서 시스템을 선택하고 오른쪽 창에서 <b>작업</b> 메뉴를 클릭한 다음 <b>보안 그룹 변경</b>을 선택합니다. 이제 시스템 네트워크를 사용하여 보안 그룹에서 연결을 추가하거나 제거할 수 있습니다.</li> </ul>
원격 콘솔에 연결	시스템	<ul style="list-style-type: none"> <li>VMware vSphere</li> </ul>	<p>선택한 시스템에서 원격 세션을 엽니다.</p> <p>성공적인 연결을 위해 다음 요구 사항을 검토합니다.</p> <ul style="list-style-type: none"> <li>배포 소비자는, 프로비저닝된 시스템의 전원이 켜져 있는지 확인합니다.</li> </ul>
스냅샷 생성	시스템	<ul style="list-style-type: none"> <li>Google Cloud Platform</li> <li>VMware vSphere</li> </ul>	<p>가상 시스템의 스냅샷을 생성합니다.</p> <p>vSphere에서 2개의 스냅샷만 허용되고 이미 스냅샷이 2개 있는 경우, 스냅샷을 삭제할 때까지 이 명령을 사용할 수 없습니다.</p>
삭제	배포	<ul style="list-style-type: none"> <li>Amazon Web Service</li> <li>Google Cloud Platform</li> <li>Microsoft Azure</li> <li>VMware vSphere</li> </ul>	<p>배포를 제거합니다.</p> <p>모든 리소스가 삭제되고 회수됩니다.</p> <p>삭제가 실패하면 배포에 대한 삭제 작업을 두 번째로 실행할 수 있습니다. 두 번째 시도 중에 <b>삭제 실패 무시</b>를 선택할 수 있습니다. 이 옵션을 선택하면 배포가 삭제되지만 리소스가 회수되지 않을 수 있습니다. 모든 리소스가 제거되었는지 확인하려면 배포가 프로비저닝된 시스템을 확인해야 합니다. 그렇지 않은 경우 해당 시스템의 나머지 리소스를 수동으로 삭제해야 합니다.</p>
	NSX 게이트웨이	<ul style="list-style-type: none"> <li>NSX</li> </ul>	<p>NSX-T 또는 NSX-V 게이트웨이에서 NAT 포트 포워딩 규칙을 삭제합니다.</p>

표 7-1. 가능한 작업 목록 (계속)

작업	적용 대상 리소스 유형	대상 클라우드 계정 또는 통합	설명
	시스템 및 로드 밸런서	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> <li>■ VMware NSX</li> </ul>	배포에서 시스템 또는 로드 밸런서를 삭제합니다. 이 작업을 수행하면 배포가 불가능해질 수 있습니다.
	보안 그룹	<ul style="list-style-type: none"> <li>■ NSX-T</li> <li>■ NSX-V</li> </ul>	<p>보안 그룹이 배포의 어떤 시스템과도 연결되어 있지 않으면 프로세스는 배포에서 보안 그룹을 제거합니다.</p> <ul style="list-style-type: none"> <li>■ 보안 그룹이 주문형인 경우 끝점에서 삭제됩니다.</li> <li>■ 보안 그룹이 공유되는 경우 작업이 실패합니다.</li> </ul>
스냅샷 삭제	시스템	<ul style="list-style-type: none"> <li>■ VMware vSphere</li> <li>■ Google Cloud Platform</li> </ul>	가상 시스템의 스냅샷을 삭제합니다.
태그 편집	배포	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	개별 배포 리소스에 적용되는 리소스 태그를 추가하거나 수정합니다.
Terraform 상태 가져오기	Terraform 구성	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	<p>Terraform 상태 파일을 표시합니다.</p> <p>배포된 클라우드 플랫폼에서 Terraform 시스템에 대해 변경한 내용을 확인하고 배포를 업데이트하려면, 먼저 [Terraform 상태 새로 고침] 작업을 실행한 다음 [Terraform 상태 가져오기] 작업을 실행합니다.</p> <p>대화 상자에 파일이 표시되는 경우, 새로 고침 작업을 새로 실행하기 전에 약 1시간 동안 파일을 사용할 수 있습니다. 나중을 위해 필요한 경우 복사할 수 있습니다.</p> <p>배포 기록 탭에서 파일을 볼 수도 있습니다. [이벤트] 탭에서 [Terraform 상태 가져오기] 이벤트를 선택하고 <b>요청 세부 정보</b>를 클릭합니다. 파일이 만료되지 않았으면 <b>컨텐츠 보기</b>를 클릭합니다. 파일이 만료된 경우 [새로 고침] 및 [가져오기] 작업을 다시 실행합니다.</p>
전원 끄기	배포	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	게스트 운영 체제를 종료하지 않고 배포 전원을 끕니다.



구성에 내장된 Terraform 리소스에 대해 다른 2일차 작업을 실행할 수 있습니다. 가능한 작업은 리소스 유형, 배포된 클라우드 플랫폼, 2일차 정책에 따라 작업을 실행할 권한이 있는지 여부에 따라 다릅니다.

표 7-1. 가능한 작업 목록 (계속)

작업	적용 대상 리소스 유형	대상 클라우드 계정 또는 통합	설명
	시스템	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	게스트 운영 체제를 종료하지 않고 시스템 전원을 끕니다.
전원 켜기	배포	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	배포 전원을 켭니다. 리소스가 일시 중단된 경우, 리소스가 일시 중단된 지점에서 정상 작업이 재개됩니다.
	시스템	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	시스템 전원을 켭니다. 시스템이 일시 중단된 경우, 시스템이 일시 중단된 지점에서 정상 작업이 재개됩니다.
재부팅	시스템	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ VMware vSphere</li> </ul>	가상 시스템에서 게스트 운영 체제를 재부팅합니다. vSphere 시스템의 경우, 이 작업을 사용하려면 해당 시스템에 VMware Tools가 설치되어 있어야 합니다.
재구성	로드 밸런서	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware NSX</li> </ul>	로드 밸런서 크기 및 로깅 수준을 변경합니다. 또한 경로를 추가 또는 제거하고 프로토콜, 포트, 상태 구성 및 멤버 풀 설정을 변경할 수 있습니다. NSX 로드 밸런서의 경우 상태 점검을 사용하거나 사용하지 않도록 설정하고 상태 옵션을 수정할 수 있습니다. NSX-T의 경우 점검을 액티브 또는 패시브로 설정할 수 있습니다. NSX-V에서는 패시브 상태 점검을 지원하지 않습니다.
	NSX 게이트웨이 포트 포워딩	<ul style="list-style-type: none"> <li>■ NSX-T</li> <li>■ NSX-V</li> </ul>	NSX-T 또는 NSX-V 게이트웨이에서 NAT 포트 포워딩 규칙을 추가, 편집 또는 삭제합니다.
	보안 그룹	<ul style="list-style-type: none"> <li>■ NSX-T</li> </ul>	NSX-T 주문형 보안 그룹에 대한 방화벽 규칙을 추가, 편집 또는 제거합니다. NSX-V 또는 기존 보안 그룹에서 작업을 실행할 수 없습니다. <ul style="list-style-type: none"> <li>■ 규칙을 추가하거나 제거하려면 토폴로지 창에서 보안 그룹을 선택하고 오른쪽 창에서 <b>작업</b> 메뉴를 클릭한 다음, <b>재구성</b>을 선택합니다. 이제 규칙을 추가, 편집 또는 제거할 수 있습니다.</li> </ul>
Terraform 상태 새로 고침	Terraform 구성	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	Terraform 상태 파일의 최신 반복을 검색합니다. 배포된 클라우드 플랫폼에서 Terraform 시스템에 대한 변경 사항을 검색하고 배포를 업데이트하려면 먼저 [Terraform 상태 새로 고침] 작업을 실행합니다. 파일을 보려면 구성에서 <b>Terraform 상태 가져오기</b> 작업을 실행합니다. 배포 기록 탭을 사용하여 새로 고침 프로세스를 모니터링합니다.

표 7-1. 가능한 작업 목록 (계속)

작업	적용 대상 리소스 유형	대상 클라우드 계정 또는 통합	설명
디스크 제거	시스템	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	기존 가상 시스템에서 디스크를 제거합니다.
재설정	시스템	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	게스트 운영 체제를 종료하지 않고 가상 시스템을 강제로 다시 시작합니다.
크기 조정	시스템	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	가상 시스템의 CPU 및 메모리를 늘리거나 줄입니다.
부팅 디스크 크기 조정	시스템	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	부팅 디스크 매체의 크기를 늘리거나 줄입니다.
디스크 크기 조정	스토리지 디스크	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Google Cloud Platform</li> </ul>	스토리지 디스크의 용량을 늘립니다.
다시 시작	시스템	<ul style="list-style-type: none"> <li>■ Microsoft Azure</li> </ul>	실행 중인 시스템을 종료하고 다시 시작합니다.
스냅샷으로 되돌리기	시스템	<ul style="list-style-type: none"> <li>■ Google Cloud Platform</li> <li>■ VMware vSphere</li> </ul>	시스템의 이전 스냅샷으로 되돌립니다. 이 작업을 사용하려면 기존 스냅샷이 있어야 합니다.
Puppet 작업 실행	관리되는 리소스	<ul style="list-style-type: none"> <li>■ Puppet Enterprise</li> </ul>	배포 내의 시스템에서 선택한 작업을 실행합니다. 작업은 Puppet 인스턴스에 정의됩니다. 작업을 식별하고 입력 매개 변수를 제공할 수 있어야 합니다.
종료	시스템	<ul style="list-style-type: none"> <li>■ VMware vSphere</li> </ul>	게스트 운영 체제를 종료한 후 시스템 전원을 끕니다. 이 작업을 사용하려면 시스템에 VMware Tools가 설치되어 있어야 합니다.
일시 중단	시스템	<ul style="list-style-type: none"> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	시스템이 사용될 수 없고 현재 사용 중인 스토리지 이외의 시스템 리소스를 사용하지 않도록 시스템을 일시 중지합니다.
업데이트	배포	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	입력 매개 변수에 따라 배포를 변경합니다. 예를 보려면 <a href="#">배포된 시스템을 다른 네트워킹으로 이동하는 방법</a> 항목을 참조하십시오.
태그 업데이트	시스템 및 디스크	<ul style="list-style-type: none"> <li>■ Amazon Web Service</li> <li>■ Microsoft Azure</li> <li>■ VMware vSphere</li> </ul>	개별 리소스에 적용된 태그를 추가, 수정 또는 삭제합니다.