

# vRealize Automation 7.3용 vRealize Orchestrator 플러그인 사 용

vRealize Orchestrator 7.3  
vRealize Automation 7.3  
vRealize Automation 7.3

VMware 웹 사이트 (<https://docs.vmware.com/kr/>) 에서 최신 기술 문서를 확인할 수 있습니다.  
또한 VMware 웹 사이트에서 최신 제품 업데이트를 제공합니다.  
이 문서에 대한 의견이 있으면 [docfeedback@vmware.com](mailto:docfeedback@vmware.com)으로 사용자 의견을 보내주십시오.

Copyright © 2012–2017 VMware, Inc. 판권 소유. [저작권 및 상표 정보](#).

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

# 목차

	vRealize Automation 용 vRealize Orchestrator 플러그인 사용	5
<b>1</b>	vRealize Automation 용 VMware vRealize Orchestrator 플러그인 소개	7
	vRealize Automation 플러그인을 사용하는 vRealize Orchestrator 의 역할	7
<b>2</b>	vRealize Automation 플러그인 구성	9
	구성 워크플로	9
	vRealize Automation 호스트 추가	10
	IaaS 호스트 추가	11
<b>3</b>	vRealize Automation 플러그인 워크플로 사용	13
	제거 작업 제한 사항	13
	제한된 작업	14
	vRealize Automation 플러그인 인벤토리 사용	15
	vRealize Automation 플러그인 관리 워크플로 사용	16
	vRealize Automation 플러그인 인프라 관리 워크플로 사용	21
	vRealize Automation IaaS 모델 엔티티 생성	23
	vRealize Automation IaaS 모델 엔티티 읽기	24
	vRealize Automation 플러그인 요청 워크플로 사용	24
	vRealize Automation 플러그인 샘플 워크플로 사용	25
	vRealize Automation 플러그인 API 액세스	25
<b>4</b>	예제 vRealize Automation 플러그인 스크립트	27
	CRUD 인프라 관리 작업 예제 스크립트	27
	vRealize Automation 엔티티 찾기 예제 스크립트	31
	vRealize Automation 예제 스크립트를 통해 프로비저닝된 리소스 가져오기	32
	일반 작업 예제 스크립트	33
	색인	37



# vRealize Automation 용 vRealize Orchestrator 플러그인 사용

---

vRealize Automation용 vRealize Orchestrator 플러그인 사용에서는 VMware vRealize Automation용 VMware® vRealize Orchestrator 플러그인을 구성하고 사용하기 위한 정보와 지침을 제공합니다.

## 대상 사용자

vRealize Automation용 vRealize Orchestrator 플러그인 사용에 나와 있는 정보는 가상 시스템 기술, Orchestrator 워크플로 개발 및 VMware vRealize Automation에 익숙한 숙련된 사용자를 위해 작성되었습니다.

Orchestrator에 대한 자세한 내용은

[http://www.vmware.com/support/pubs/orchestrator\\_pubs.html](http://www.vmware.com/support/pubs/orchestrator_pubs.html)을 참조하십시오.

vRealize Automation에 대한 자세한 내용은

<http://www.vmware.com/support/pubs/vrealize-automation.html> 항목을 참조하십시오.

## VMware 기술 자료 용어집

VMware 기술 자료 사이트에서는 새로운 용어를 정리한 용어집을 제공하고 있습니다. VMware 기술 설명서에 사용된 용어에 대한 정의를 보려면 <http://www.vmware.com/support/pubs>를 참조하십시오.



# vRealize Automation 용 VMware vRealize Orchestrator 플러그인 소개

# 1

vRealize Automation용 VMware vRealize Orchestrator 플러그인은 vRealize Orchestrator와 vRealize Automation 간의 상호 작용을 지원합니다.

vRealize Automation 플러그인을 사용하여 다음 vRealize Automation 기능에 대한 워크플로를 생성하고 실행할 수 있습니다.

- XaaS 사용자 지정 리소스 및 Blueprint 관리
- 카탈로그 항목 및 리소스 관리와 요청
- 사용 권한 구성
- 승인 정책 구성
- 작업 항목 상호 작용
- vSphere 및 vCloud Director 가상 시스템 프로비저닝 작업과 사후 프로비저닝 작업
- vRealize Automation IaaS 모델에 대한 CRUD(생성, 읽기, 업데이트 및 삭제) 작업

## vRealize Automation 플러그인을 사용하는 vRealize Orchestrator의 역할

Orchestrator 클라이언트를 사용하여 워크플로를 실행 및 생성하고 플러그인 API에 액세스할 수 있습니다. vRealize Automation 설치에 포함된 vRealize Orchestrator 인스턴스 또는 외부 vRealize Orchestrator 서버를 사용할 수 있습니다.

vRealize Orchestrator는 vRealize Automation 플러그인을 지원합니다. vRealize Orchestrator는 VMware 클라우드 스택 및 타사 기술을 관리할 수 있는 확장 가능한 워크플로 라이브러리를 제공하는 개발 및 프로세스 자동화 플랫폼입니다.

vRealize Orchestrator는 개방형 플러그인 아키텍처를 통해 관리 및 운영 솔루션과의 통합을 지원합니다.





# vRealize Automation 플러그인 구성

# 2

vRealize Automation 호스트와 IaaS 호스트를 추가하여 플러그인을 구성할 수 있습니다.

## 구성 워크플로

구성 워크플로 범주의 워크플로를 사용하여 vRealize Automation 호스트를 관리할 수 있습니다.

### vRealize Automation 호스트

이러한 워크플로는 Orchestrator 클라이언트의 **워크플로** 보기에서 플러그인 라이브러리의 **구성** 하위 디렉토리를 통해 액세스할 수 있습니다.

워크플로 이름	설명
vRA 호스트 추가	플러그인 인벤토리에 vRealize Automation 호스트를 추가합니다. 테넌트 관리 및 운영 작업의 경우 <b>인벤토리</b> 보기를 사용하여 각 테넌트에서 워크플로를 실행할 수 있습니다. 테넌트에 대해 플러그인의 전체 기능을 사용하려면 각 테넌트에 대해 전용 vRealize Automation 호스트를 생성합니다.
구성 요소 레지스트리를 사용하여 vRA 호스트 추가	사용자별 세션 연결을 사용하여 플러그인 인벤토리에 vRealize Automation 호스트를 추가합니다. vRealize Automation 시스템 관리자의 자격 증명을 사용하여 Orchestrator 클라이언트에 로그인해야 합니다. 외부 vRealize Orchestrator 서버에서 이 기능을 사용하려면 vRealize Automation 구성 요소 레지스트리에 Orchestrator 서버를 등록해야 합니다.
vRA 호스트의 IaaS 호스트 추가	플러그인 인벤토리에 선택한 vRealize Automation 호스트의 IaaS 호스트를 추가합니다.
vRA 호스트 제거	플러그인 인벤토리에서 vRealize Automation 호스트를 제거합니다.
vRA 호스트 업데이트	플러그인 인벤토리에서 vRealize Automation 호스트를 업데이트합니다.
vRA 호스트 유효성 검사	vRealize Automation 호스트와 해당 연결의 유효성을 검사합니다.

**참고** vRealize Orchestrator 서버가 vRealize Automation 구성 요소 레지스트리에 등록된 경우 이름이 Default인 vRealize Automation 호스트가 자동으로 추가됩니다. Default 호스트는 기본 테넌트에 대한 사용자별 세션 연결을 사용합니다. vRealize Automation 설치에 포함된 Orchestrator 서버는 기본적으로 vRealize Automation 구성 요소 레지스트리에 등록됩니다.

### vRealize Automation IaaS 호스트

이러한 워크플로는 Orchestrator 클라이언트의 **워크플로** 보기에서 플러그인 라이브러리의 **인프라 관리 > 구성** 하위 디렉토리를 통해 액세스할 수 있습니다.

vRealize Automation 설치에 포함된 vRealize Orchestrator 서버는 기본적으로 vRealize Automation 구성 요소 레지스트리에 등록됩니다.

워크플로 이름	설명
IaaS 호스트 추가	플러그인 인벤토리에 vRealize Automation IaaS 호스트를 추가합니다. 이 워크플로는 vRA 호스트의 IaaS 호스트 추가와 기능적으로 동일하지만 vRealize Automation 호스트가 필요 없습니다.
IaaS 호스트 제거	플러그인 인벤토리에서 vRealize Automation IaaS 호스트를 제거합니다.
IaaS 호스트 업데이트	플러그인 인벤토리에서 vRealize Automation IaaS 호스트를 업데이트합니다.
IaaS 호스트 유효성 검사	vRealize Automation IaaS 호스트와 해당 연결의 유효성을 검사합니다.

## vRealize Automation 호스트 추가

워크플로를 실행하여 vRealize Automation 호스트를 추가하고 호스트 연결 매개 변수를 구성할 수 있습니다.

### 프로시저

- 1 Orchestrator 클라이언트의 드롭다운 목록에서 **실행** 또는 **설계**를 선택합니다.
- 2 **워크플로** 보기를 클릭합니다.
- 3 **라이브러리 > vRealize Automation > 구성**을 확장합니다.
- 4 **vRA 호스트 추가** 워크플로를 마우스 오른쪽 버튼으로 클릭하고 **워크플로 시작**을 선택합니다.
- 5 **호스트 이름** 텍스트 상자에 호스트의 고유한 이름을 입력합니다.
- 6 **호스트 URL** 텍스트 상자에 호스트의 URL 주소를 입력합니다.  
예: https://hostname
- 7 **테넌트** 텍스트 상자에 테넌트의 이름을 입력합니다.  
테넌트에 대해 플러그인의 전체 기능을 사용하려면 각 테넌트에 대해 전용 vRealize Automation 호스트를 생성합니다.
- 8 사용자의 확인 없이 SSL 인증서를 자동으로 설치할지 여부를 선택합니다.
- 9 (선택 사항) vRealize Orchestrator가 vRealize Automation로부터 응답 또는 연결을 기다리는 시간을 구성하려면 **연결 시간 제한(초)** 및 **작업 시간 제한(초)** 텍스트 상자에 시간 초과 간격을 입력합니다.
- 10 **세션 모드** 드롭다운 메뉴에서 호스트에 대한 연결 유형을 선택합니다.

옵션	작업
공유 세션	<b>인증 사용자 이름</b> 및 <b>인증 암호</b> 텍스트 상자에 vRealize Automation 사용자에게 대한 자격 증명을 입력합니다.
사용자별 세션	현재 로그인되어 있는 사용자의 자격 증명을 사용하여 연결합니다. vRealize Automation 시스템 관리자의 자격 증명을 사용하여 Orchestrator 클라이언트에 로그인해야 합니다. 외부 vRealize Orchestrator 서버에서 이 옵션을 사용하려면 vRealize Automation 구성 요소 레지스트리에 Orchestrator 서버를 등록해야 합니다.

- 11 **제출**을 클릭합니다.

### 후속 작업

vRealize Automation 인프라 관리 호스트를 추가합니다.

## IaaS 호스트 추가

워크플로를 실행하여 vRealize Automation 호스트의 IaaS 호스트를 추가하고 연결 매개 변수를 구성할 수 있습니다.

### 프로시저

- 1 Orchestrator 클라이언트의 드롭다운 목록에서 **실행** 또는 **설계**를 선택합니다.
- 2 **워크플로** 보기를 클릭합니다.
- 3 **라이브러리 > vRealize Automation > 인프라 관리 > 구성**을 확장합니다.
- 4 **IaaS 호스트 추가**를 마우스 오른쪽 버튼으로 클릭하고 **워크플로 시작**을 선택합니다.
- 5 **vCAC 호스트** 드롭다운 메뉴에서 IaaS 호스트를 구성하려는 vRealize Automation 호스트를 선택합니다.
- 6 **호스트 이름** 텍스트 상자에 호스트의 고유한 이름을 입력합니다.
- 7 Model Manager가 설치되어 있는 시스템의 URL을 입력합니다.  
예를 들어 `https://model_manager_machine.com`을 입력합니다.
- 8 SSL 인증서를 설치하려면 **예**를 선택합니다.
- 9 프록시를 사용하여 Model Manager 시스템에 액세스하려면 **예**를 선택합니다.  
이 옵션을 선택하는 경우 다음 페이지에서 프록시 호스트와 프록시 포트를 제공해야 합니다.
- 10 **다음**을 클릭합니다.
- 11 명시적 프록시를 구성 중인 경우 프록시 호스트 URL과 포트를 제공합니다.
- 12 **다음**을 클릭합니다.
- 13 고유한 시간 제한 값을 구성하려면 **아니요**를 클릭합니다.
- 14 (선택 사항) vRealize Orchestrator가 vRealize Automation로부터 응답 또는 연결을 기다리는 시간을 구성하려면 **연결 시간 제한(초)** 및 **작업 시간 제한(초)** 텍스트 상자에 시간 초과 간격을 입력합니다.
- 15 **다음**을 클릭합니다.
- 16 호스트의 인증 유형을 선택합니다.

옵션	설명
<b>SSO</b>	vCenter Single Sign-On을 사용하려면 이 옵션을 선택합니다.
<b>NTLM</b>	사용 중인 Active Directory 인프라에서 NTLM 인증을 사용할 경우에만 이 옵션을 선택하여 NTLM(NT LAN Manager) 프로토콜 기반 인증을 사용하도록 설정합니다. 이 옵션을 선택할 경우 추가 NTLM 자격 증명 및 인증 옵션을 사용해야 합니다.

- 17 NTLM을 선택했으면 **다음**을 클릭하고 Workstation 시스템 이름과 NetBIOS 도메인 이름을 입력합니다.
- 18 **제출**을 클릭합니다.



# vRealize Automation 플러그인 워크플로 사용

# 3

vRealize Automation 플러그인 워크플로 라이브러리에는 카탈로그와의 상호 작용, 인프라 관리, 테넌트 및 서비스 생성 등의 일반 작업에 사용할 수 있는 워크플로가 포함됩니다.

vRealize Automation 관련 헤더 작업 및 ID와 같은 사용자 지정 HTTP 헤더를 사용하고, CRUD, 프로비저닝 및 사후 프로비저닝 워크플로에서 적용할 수 있습니다.

이 장에서는 다음 주제에 대해 설명합니다.

- “제거 작업 제한 사항,” (13 페이지)
- “vRealize Automation 플러그인 인벤토리 사용,” (15 페이지)
- “vRealize Automation 플러그인 관리 워크플로 사용,” (16 페이지)
- “vRealize Automation 플러그인 인프라 관리 워크플로 사용,” (21 페이지)
- “vRealize Automation 플러그인 요청 워크플로 사용,” (24 페이지)
- “vRealize Automation 플러그인 샘플 워크플로 사용,” (25 페이지)
- “vRealize Automation 플러그인 API 액세스,” (25 페이지)

## 제거 작업 제한 사항

버전 7.0부터 일부 생성, 읽기, 업데이트 및 삭제 작업이 제한됩니다. 이전 버전에서 사용한 워크플로의 작업은 7.0 이상에서 작동하지 않습니다. 워크플로를 지원되는 작업으로 업데이트하거나 필요한 작업을 다시 사용하도록 설정할 수 있습니다.

작업을 다시 사용하도록 설정하려면 `operations.properties` 파일에서 해당 작업을 제거해야 합니다. 파일의 작업 목록은 “제한된 작업,” (14 페이지) 항목을 참조하십시오.

### 프로시저

- 1 vRealize Orchestrator의 드롭다운 메뉴에서 **디자인**을 선택합니다.
- 2 **리소스** 보기를 클릭합니다.
- 3 리소스 계층에서 **라이브러리 > VCAC > 유틸리티**를 확장합니다.
- 4 백업을 만들고 `operations.properties` 파일을 수정합니다.
  - a `operations.properties`를 마우스 오른쪽 버튼으로 클릭하고 **파일에 저장**을 선택합니다.
  - b 사본을 백업으로 저장합니다.
  - c 새 사본을 만들고 다시 사용하도록 설정할 작업을 삭제합니다.
  - d 새 파일을 저장합니다.

- 5 vRealize Orchestrator에서 기존 파일을 대체합니다.
  - a vRealize Orchestrator에서 **유틸리티** 폴더를 마우스 오른쪽 버튼으로 클릭하고 **리소스 가져오기**를 클릭합니다.
  - b operations.properties 파일의 새 버전을 찾아서 **열기**를 클릭합니다.
  - c **한 번 바꾸기**를 클릭하여 수정된 버전을 저장합니다.
- 6 vRealize Orchestrator 서버를 다시 시작합니다.
- 7 operations.properties 파일을 선택하고 **뷰어** 탭을 클릭합니다.
- 8 사용하도록 설정 중인 작업이 더 이상 파일에 없는지 확인합니다.

파일에서 제거한 작업이 이제 이전 워크플로에서 작동합니다.

### 후속 작업

새 워크플로를 만들 때 제한된 작업을 사용하지 않도록 합니다.

## 제한된 작업

operations.properties 파일의 콘텐츠에는 제한된 작업이 포함되어 있습니다. 작업을 다시 사용하도록 설정하려면 파일에서 해당 작업을 제거해야 합니다.

다음 텍스트는 operations.properties 파일의 기본 버전입니다. 작업을 다시 사용하도록 설정하려면 “[제거 작업 제한 사항](#),” (13 페이지) 항목을 참조하십시오.

```
#Blueprints
operation.create=ManagementModelEntities.svc@VirtualMachineTemplates
operation.update=ManagementModelEntities.svc@VirtualMachineTemplates
operation.delete=ManagementModelEntities.svc@VirtualMachineTemplates
#Blueprint properties
operation.create=ManagementModelEntities.svc@VirtualMachineProperties
operation.read=ManagementModelEntities.svc@VirtualMachineProperties
operation.update=ManagementModelEntities.svc@VirtualMachineProperties
operation.delete=ManagementModelEntities.svc@VirtualMachineProperties
#Global profiles
operation.create=ManagementModelEntities.svc@GlobalProfiles
operation.read=ManagementModelEntities.svc@GlobalProfiles
operation.update=ManagementModelEntities.svc@GlobalProfiles
operation.delete=ManagementModelEntities.svc@GlobalProfiles
#Global profile properties
operation.create=ManagementModelEntities.svc@GlobalProfileProperties
operation.read=ManagementModelEntities.svc@GlobalProfileProperties
operation.update=ManagementModelEntities.svc@GlobalProfileProperties
operation.delete=ManagementModelEntities.svc@GlobalProfileProperties
#PropertySetXml
operation.create=ManagementModelEntities.svc@PropertySetXml
operation.read=ManagementModelEntities.svc@PropertySetXml
operation.update=ManagementModelEntities.svc@PropertySetXml
operation.delete=ManagementModelEntities.svc@PropertySetXml
#Property definitions
operation.create=ManagementModelEntities.svc@PropertyDefinitions
operation.read=ManagementModelEntities.svc@PropertyDefinitions
operation.update=ManagementModelEntities.svc@PropertyDefinitions
operation.delete=ManagementModelEntities.svc@PropertyDefinitions
#Property attributes
operation.create=ManagementModelEntities.svc@PropertyAttributes
```

```

operation.read=ManagementModelEntities.svc@PropertyAttributes
operation.update=ManagementModelEntities.svc@PropertyAttributes
operation.delete=ManagementModelEntities.svc@PropertyAttributes
#Property Attribute Types
operation.create=ManagementModelEntities.svc@PropertyAttributeTypes
operation.read=ManagementModelEntities.svc@PropertyAttributeTypes
operation.update=ManagementModelEntities.svc@PropertyAttributeTypes
operation.delete=ManagementModelEntities.svc@PropertyAttributeTypes
#Control layouts
operation.create=ManagementModelEntities.svc@ControlLayouts
operation.read=ManagementModelEntities.svc@ControlLayouts
operation.update=ManagementModelEntities.svc@ControlLayouts
operation.delete=ManagementModelEntities.svc@ControlLayouts
#Amazon Virtual Machine Templates
operation.create=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.read=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.update=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
operation.delete=AmazonWSModelEntities.svc@AmazonVirtualMachineTemplates
#Openstack Virtual Machine Templates
operation.create=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.read=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.update=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
operation.delete=OpenStackModelEntities.svc@OpenstackVirtualMachineTemplates
#Endpoint credentials
operation.create=ManagementModelEntities.svc@ConnectionCredentials
operation.update=ManagementModelEntities.svc@ConnectionCredentials
operation.delete=ManagementModelEntities.svc@ConnectionCredentials
#Management endpoints
operation.create=ManagementModelEntities.svc@ManagementEndpoints
operation.update=ManagementModelEntities.svc@ManagementEndpoints
operation.delete=ManagementModelEntities.svc@ManagementEndpoints
#Management endpoint properties
operation.create=ManagementModelEntities.svc@ManagementEndpointProperties
operation.read=ManagementModelEntities.svc@ManagementEndpointProperties
operation.update=ManagementModelEntities.svc@ManagementEndpointProperties
operation.delete=ManagementModelEntities.svc@ManagementEndpointProperties

```

## vRealize Automation 플러그인 인벤토리 사용

인벤토리 보기를 사용하여 vRealize Automation 개체에서 워크플로를 실행할 수 있습니다.

인벤토리 개체에 사용할 수 있는 워크플로를 표시하려면 **도구 > 사용자 기본 설정 > 인벤토리**로 이동한 후 **인벤토리에서 컨텍스트 메뉴 사용** 확인란을 선택합니다. 이 옵션을 사용하도록 설정하면 Orchestrator 인벤토리에서 개체를 마우스 오른쪽 버튼으로 클릭했을 때 해당 개체에 사용할 수 있는 모든 워크플로가 표시됩니다.

## vRealize Automation 플러그인 관리 워크플로 사용

관리 워크플로를 사용하여 vRealize Automation 서비스, 테넌트, 승인 정책, 사용 권한, 비즈니스 그룹, 카탈로그 항목 및 고급 서비스 구성 요소를 관리할 수 있습니다.

일부 워크플로에는 vRealize Automation 호스트인 vCACCAFE:VCACHost에 대한 입력 매개 변수가 포함되어 있습니다. vRealize Automation 호스트 연결을 구성한 방법에 따라 사용자가 워크플로를 실행할 때 역할이 적용되는 방식이 결정됩니다.

- 연결을 공유 세션으로 구성한 경우 공유 세션의 사용자 계정에 워크플로를 실행하는 데 필요한 역할이 있어야 합니다.
- 연결을 사용자별 세션으로 구성한 경우 vRealize Automation 사용자 인터페이스에서와 마찬가지로 워크플로를 실행하는 각 사용자에게 필요한 역할이 있어야 합니다.

이러한 워크플로는 vRealize Orchestrator 클라이언트의 **워크플로** 보기에서 **라이브러리 > vRealize Automation > 관리** 하위 디렉토리를 통해 확인할 수 있습니다.

**승인 정책** 하위 디렉토리의 워크플로를 사용하여 승인 정책을 생성하고 관리할 수 있습니다.

**표 3-1.** 승인 정책

워크플로	설명
승인 정책 활성화	승인 정책을 활성화합니다. 승인 정책을 활성화하면 읽기 전용 상태가 됩니다.
승인 수준 추가	항상 필요한 승인 수준을 승인에 추가합니다. 승인자에 대한 특정 사용자 및 그룹을 선택해야 합니다.
승인 정책 복사	승인 정책을 복사합니다.
승인 정책 생성	수준이 없거나 승인자가 없는 초안 승인 정책을 생성합니다. 승인 수준을 생성하고 정책의 승인자를 지정하려면 승인 수준 추가 워크플로를 실행합니다.
승인 정책 비활성화	승인 정책을 비활성화합니다. 또한 승인 정책과 연결된 모든 기존 사용 권한을 삭제할 수 있습니다.
승인 정책 삭제	초안 상태의 승인 정책을 삭제합니다. 활성 승인 정책은 읽기 전용입니다.

**비즈니스 그룹** 하위 디렉토리의 워크플로를 사용하여 비즈니스 그룹 및 비즈니스 그룹 사용자 지정 속성을 생성하고 관리할 수 있습니다.

**표 3-2.** 비즈니스 그룹

워크플로	설명
사용자 지정 속성 추가	비즈니스 그룹에 사용자 지정 속성을 추가합니다.
비즈니스 그룹 생성	비즈니스 그룹을 생성합니다.
비즈니스 그룹 삭제	비즈니스 그룹을 삭제합니다.
사용자 지정 속성 삭제	비즈니스 그룹에서 사용자 지정 속성을 제거합니다.
비즈니스 그룹 업데이트	기본 시스템 접두사, 활성 디렉토리 컨테이너, 사용자 역할 등 비즈니스 그룹에 대한 세부 정보를 업데이트합니다.
사용자 지정 속성 업데이트	비즈니스 그룹에 대한 사용자 지정 속성을 업데이트합니다.

관리 하위 디렉토리에는 vRealize Automation 7.0 이전 버전에서 작동하는 **비즈니스 그룹(더 이상 사용되지 않음)** 하위 디렉토리가 있습니다. 기본 폴더에 있는 동일한 이름의 워크플로를 사용하십시오.

**카탈로그 항목** 하위 디렉토리의 워크플로를 사용하여 카탈로그 항목을 관리할 수 있습니다.



**표 3-3.** 카탈로그 항목

워크플로	설명
카탈로그 항목 활성화	카탈로그 항목을 활성화합니다. 카탈로그 항목을 활성화하고 서비스에 할당해야 사용자가 해당 항목을 요청할 수 있습니다.
서비스에 카탈로그 항목 추가	서비스에 카탈로그 항목을 추가합니다. 카탈로그 항목을 활성화하고 서비스에 할당해야 사용자가 해당 항목을 요청할 수 있습니다.
카탈로그 항목 비활성화	사용자가 요청할 수 없도록 카탈로그 항목을 비활성화하고 서비스 카탈로그에서 제거합니다.

**복합 Blueprint** 하위 디렉토리의 워크플로를 사용하여 디자인 캔버스에서 생성한 복합 Blueprint를 관리할 수 있습니다.

**표 3-4.** 복합 Blueprint

워크플로	설명
복합 Blueprint 삭제	디자인 Blueprint 목록에서 게시되지 않은 Blueprint를 삭제합니다.
복합 Blueprint 가져오기	YAML 파일에서 복합 Blueprint를 가져옵니다.
복합 Blueprint 게시	초안 상태의 복합 Blueprint를 게시합니다.
복합 Blueprint 게시 취소	게시된 복합 Blueprint의 게시를 취소합니다.

**컨텐츠** 하위 디렉토리 워크플로는 더 이상 사용되지 않습니다. Cloud Client를 사용하여 가져오기 및 내보내기 작업을 수행할 수 있습니다. Cloud Client 다운로드 및 설명서는 <https://developercenter.vmware.com/tool/cloudclient>에서 제공됩니다.

**표 3-5.** 컨텐츠

워크플로	설명
컨텐츠 내보내기(더 이상 사용되지 않음)	Cloud Client를 사용하여 가져오기 및 내보내기 작업을 수행할 수 있습니다. Cloud Client 다운로드 및 설명서는 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 에서 제공됩니다.
컨텐츠 가져오기(더 이상 사용되지 않음)	Cloud Client를 사용하여 가져오기 및 내보내기 작업을 수행할 수 있습니다. Cloud Client 다운로드 및 설명서는 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 에서 제공됩니다.
컨텐츠 전송(더 이상 사용되지 않음)	Cloud Client를 사용하여 가져오기 및 내보내기 작업을 수행할 수 있습니다. Cloud Client 다운로드 및 설명서는 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 에서 제공됩니다.
컨텐츠 유효성 검사(더 이상 사용되지 않음)	Cloud Client를 사용하여 가져오기 및 내보내기 작업을 수행할 수 있습니다. Cloud Client 다운로드 및 설명서는 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 에서 제공됩니다.

**사용 권한** 하위 디렉토리의 워크플로를 사용하여 사용 권한을 생성하고 관리할 수 있습니다.

**표 3-6.** 사용 권한

워크플로	설명
사용 권한 활성화	사용 권한을 활성화합니다.
사용 권한에 카탈로그 항목 할당	하나 이상의 카탈로그 항목을 사용 권한에 할당합니다. 이 워크플로를 사용하여 승인 정책을 할당할 수도 있습니다.
사용 권한에 즉시 실행 작업 할당	하나 이상의 즉시 실행 작업을 사용 권한에 할당합니다. 즉시 실행 작업은 요청을 생성하지 않습니다.
사용 권한에 리소스 작업 할당	하나 이상의 리소스 작업을 사용 권한에 할당합니다. 이 워크플로를 사용하여 승인 정책을 할당할 수도 있습니다.
사용 권한에 서비스 할당	하나 이상의 서비스를 사용 권한에 할당합니다. 이 워크플로를 사용하여 승인 정책을 할당할 수도 있습니다.
사용 권한에 사용자 및 그룹 할당	하나 이상의 사용자 또는 그룹을 사용 권한에 할당합니다.
사용 권한 생성(더 이상 사용되지 않음)	사용 권한을 생성합니다. 하위 테넌트에 대한 사용 권한 생성을 사용하지 않습니다.
하위 테넌트에 대한 사용 권한 생성	사용 권한을 생성합니다.
사용 권한 비활성화	사용 권한을 비활성화합니다.
사용 권한에서 사용자 및 그룹의 할당 취소	사용 권한에 대한 사용자 목록에서 사용자 및 그룹을 제거합니다.

**속성** 하위 디렉토리의 워크플로를 사용하여 속성 정의 및 속성 그룹을 관리할 수 있습니다. vRealize Automation 속성과의 충돌을 피하려면 모든 사용자 지정 속성 이름에 회사 또는 기능 이름과 같은 접두사 뒤에 점을 추가하여 사용하십시오.

**표 3-7.** 속성 정의

워크플로	설명
속성 정의 생성	사용자 지정 속성을 생성합니다.
속성 정의 삭제	사용자 지정 속성을 삭제합니다.

속성 그룹은 속성 정의의 컬렉션입니다.

**표 3-8.** 속성 그룹

워크플로	설명
그룹에 속성 추가	정의된 사용자 지정 속성을 그룹에 추가합니다.
속성 그룹 생성	정의된 사용자 지정 속성을 추가할 수 있는 속성 그룹을 생성합니다.
속성 그룹 삭제	속성 그룹을 삭제합니다.
그룹에서 속성 제거	속성 그룹에서 정의된 사용자 지정 속성을 제거합니다.
속성 그룹 업데이트	속성 그룹의 이름 또는 설명을 수정합니다.
그룹에서 속성 업데이트	속성 그룹의 속성에 대한 이름, 값 및 동작을 수정합니다.

**서비스** 하위 디렉토리의 워크플로를 사용하여 서비스를 관리할 수 있습니다.

**표 3-9.** 서비스

워크플로	설명
서비스 활성화	서비스를 활성화합니다.
서비스에 카탈로그 항목 할당	하나 이상의 카탈로그 항목을 서비스에 할당합니다.
서비스 복사	서비스를 복사합니다.

**표 3-9.** 서비스 (계속)

워크플로	설명
서비스 생성	서비스를 생성합니다.
서비스 비활성화	서비스를 비활성화합니다.
서비스 삭제	서비스를 삭제합니다.

**테넌트** 하위 디렉토리의 워크플로를 사용하여 테넌트를 생성하고 관리할 수 있습니다.

ID 저장소 워크플로는 더 이상 사용되지 않습니다. Directories Management API용으로 vRealize Automation에서 변경된 대체 워크플로가 사용됩니다.

**표 3-10.** 테넌트

워크플로	설명
관리자 추가	하나 이상의 테넌트 관리자 및 인프라 관리자를 테넌트에 추가합니다.
테넌트에 ID 저장소 추가	vRealize Automation 호스트의 테넌트에 ID 저장소를 추가합니다. 이 워크플로는 테넌트를 구성하는 시스템 관리자만 실행할 수 있습니다.
테넌트에 ID 저장소 추가(더 이상 사용되지 않음)	테넌트에 ID 저장소 추가 워크플로를 사용하십시오.
vCAC 호스트에 ID 저장소 추가	vRealize Automation 호스트로 구성된 테넌트에 ID 저장소를 추가합니다. 이 워크플로는 테넌트에 대한 ID 저장소를 구성하는 테넌트 관리자만 실행할 수 있습니다.
vCAC 호스트에 ID 저장소 추가(더 이상 사용되지 않음)	vCAC 호스트에 ID 저장소 추가 워크플로를 사용하십시오.
테넌트 생성	테넌트를 생성합니다. 시스템 관리자 자격 증명으로 추가된 vRealize Automation 호스트를 선택해야 합니다.
테넌트에서 ID 저장소 삭제	vRealize Automation 호스트의 테넌트에서 ID 저장소를 삭제합니다. 이 워크플로는 테넌트를 구성하는 시스템 관리자만 실행할 수 있습니다.
vCAC 호스트에서 ID 저장소 삭제	vRealize Automation 호스트로 구성된 테넌트에서 ID 저장소를 삭제합니다. 이 워크플로는 테넌트에 대한 ID 저장소를 구성하는 테넌트 관리자만 실행할 수 있습니다.
테넌트 삭제	테넌트를 삭제합니다.
관리자 제거	하나 이상의 테넌트 관리자 및 인프라 관리자를 테넌트에서 제거합니다.
테넌트에 대한 ID 저장소 업데이트	vRealize Automation 호스트의 테넌트에 대한 기존 ID 저장소를 업데이트합니다. 이 워크플로는 테넌트를 구성하는 시스템 관리자만 실행할 수 있습니다.
테넌트에 대한 ID 저장소 업데이트(더 이상 사용되지 않음)	테넌트에 대한 ID 저장소 업데이트 워크플로를 사용하십시오.
vCAC 호스트에 대한 ID 저장소 업데이트	vRealize Automation 호스트로 구성된 테넌트에 대한 ID 저장소를 업데이트합니다. 이 워크플로는 테넌트에 대한 ID 저장소를 구성하는 테넌트 관리자만 실행할 수 있습니다.
vCAC 호스트에 대한 ID 저장소 업데이트(더 이상 사용되지 않음)	vCAC 호스트에 대한 ID 저장소 업데이트 워크플로를 사용하십시오.
테넌트 업데이트	기존 테넌트의 이름, 설명 및 연락처 이메일 주소를 업데이트합니다.

**워크플로 구독** 하위 디렉토리의 워크플로를 사용하여 이벤트 워크플로 구독을 관리할 수 있습니다.

**표 3-11.** 워크플로 구독

워크플로	설명
워크플로 구독 삭제	게시되지 않은 워크플로 구독을 삭제합니다. 이 워크플로는 시스템 및 테넌트 워크플로 구독에 적용됩니다.
시스템 워크플로 구독 내보내기	시스템 워크플로 구독을 내보내고 JSON 형식의 vRealize Orchestrator 리소스 요소로 저장합니다. 시스템 워크플로 구독은 시스템 이벤트 및 모든 테넌트의 이벤트에 반응하는 특수한 워크플로 구독입니다.
테넌트 워크플로 구독 내보내기	테넌트 워크플로 구독을 내보내고 JSON 형식의 리소스 요소로 저장합니다. 테넌트 관련 워크플로를 실행하는 특수한 워크플로 구독입니다.
시스템 워크플로 구독 가져오기	JSON 파일에서 시스템 워크플로 구독을 가져옵니다. 시스템 워크플로 구독은 시스템 이벤트에 대해 트리거되며 여러 테넌트에 걸쳐 적용될 수 있습니다.
테넌트 워크플로 구독 가져오기	JSON 파일에서 내보낸 워크플로 구독을 가져옵니다. 이러한 워크플로 구독은 테넌트와 관련이 있습니다.
워크플로 구독 게시	초안 또는 게시되지 않은 상태의 워크플로 구독을 게시합니다. 이 워크플로는 시스템 및 테넌트 워크플로 구독에 적용됩니다.
시스템 워크플로 구독 등록	시간 초과 및 우선 순위 값을 포함하여 시스템 워크플로 구독을 생성합니다.
테넌트 워크플로 구독 등록	시간 초과 및 우선 순위 값을 포함하여 테넌트 관련 워크플로 구독을 생성합니다.
워크플로 구독 게시 취소	게시된 워크플로 구독의 게시를 취소합니다. 이 워크플로는 시스템 및 테넌트 워크플로 구독에 적용됩니다.
워크플로 구독 업데이트	이름, 설명, vRealize Orchestrator 워크플로, 구독 조건, 시간 초과 값, 상태 값 및 우선 순위 값을 변경합니다. 이벤트 항목 또는 차단 상태는 업데이트할 수 없습니다.

**XaaS 사용자 지정 리소스** 하위 디렉토리의 워크플로를 사용하여 XaaS 사용자 지정 리소스를 생성하고 삭제할 수 있습니다.

**표 3-12.** XaaS 사용자 지정 리소스

워크플로	설명
사용자 지정 리소스 생성	사용자 지정 리소스를 생성합니다.
사용자 지정 리소스 삭제	사용자 지정 리소스를 제거합니다.

**XaaS 리소스 작업** 하위 디렉토리의 워크플로를 사용하여 XaaS 리소스 작업을 생성하고 관리할 수 있습니다.

**표 3-13.** XaaS 리소스 작업

워크플로	설명
리소스 작업 복제	기존 리소스 작업의 사본을 생성합니다.
리소스 작업 생성	리소스 작업을 생성합니다.
리소스 작업 삭제	리소스 작업을 삭제합니다.
리소스 작업 게시	리소스 작업을 게시합니다.
리소스 작업 게시 취소	리소스 작업의 게시를 취소합니다.

**XaaS 리소스 매핑** 하위 디렉토리의 워크플로를 사용하여 비 XaaS 리소스에 대한 XaaS 매핑을 생성하고 관리할 수 있습니다.

**표 3-14.** XaaS 리소스 매핑

워크플로	설명
리소스 매핑 생성	카탈로그 리소스 유형을 vRealize Orchestrator 유형에 매핑합니다.
리소스 매핑 삭제	리소스 매핑을 삭제합니다.
대상 기준 설정	리소스 매핑의 가용성을 결정하는 조건을 지정합니다.

**XaaS 서버 구성** 하위 디렉토리의 워크플로를 사용하여 대상 Orchestrator 인스턴스를 관리할 수 있습니다.

**표 3-15.** XaaS 서버 구성

워크플로	설명
Orchestrator 서버 구성 업데이트	포트, 호스트, 사용자 이름 및 암호 등 서버 설정을 수정합니다.
Orchestrator 서버 구성 유효성 검사	vRealize Orchestrator 설정이 유효한지 확인합니다. 구성이 유효하면 TRUE 값이 반환되고, 구성이 유효하지 않으면 FALSE 값이 반환됩니다.

**XaaS 서비스 Blueprint** 하위 디렉토리의 워크플로를 사용하여 XaaS Blueprint를 생성하고 관리할 수 있습니다.

**표 3-16.** XaaS Blueprint

워크플로	설명
서비스 Blueprint 복제	서비스 Blueprint의 사본을 생성합니다.
서비스 Blueprint 생성	서비스 Blueprint를 생성합니다.
서비스 Blueprint 삭제	서비스 Blueprint를 삭제합니다.
서비스 Blueprint 게시	서비스 Blueprint를 게시합니다.
서비스 Blueprint 게시 취소	서비스 Blueprint의 게시를 취소합니다.

## vRealize Automation 플러그인 인프라 관리 워크플로 사용

인프라 관리 워크플로를 사용하여 기본 작업을 실행할 수 있습니다. 확장성 패키지를 사용하여 vRealize Orchestrator 워크플로를 프로비저닝 프로세스의 일부로 호출하거나 사용자 지정 작업을 통해 호출하는 기능으로 vRealize Automation을 사용자 지정할 수 있습니다.

인프라 관리 워크플로는 Orchestrator 클라이언트의 **워크플로** 보기에서 플러그인 라이브러리의 **인프라 관리** 하위 디렉토리를 통해 확인할 수 있습니다.

인프라 관리 워크플로를 사용하여 가상 시스템을 프로비저닝하고 기본 생성, 읽기, 업데이트 또는 삭제 작업을 실행할 수 있습니다.

표 3-17. 인프라 관리

워크플로 이름	설명
가상 시스템 상태 변경 대기	가상 시스템 집합의 상태 변경을 대기합니다. 모든 가상 시스템이 성공 상태에 있는 경우 트리거가 호출되고 워크플로가 성공적으로 종료됩니다. 지정된 가상 시스템 중 하나 이상이 실패 상태이거나 존재하지 않는 경우에는 워크플로가 실패합니다. 다음 옵션에서 선택하여 성공 및 실패 상태를 입력해야 합니다. <ul style="list-style-type: none"> <li>■ Requested</li> <li>■ AwaitingApproval</li> <li>■ RegisterMachine</li> <li>■ BuildingMachine</li> <li>■ AddingDisks</li> <li>■ MachineProvisioned</li> <li>■ MachineActivated</li> <li>■ InstallTools(VMware에만 해당)</li> <li>■ On</li> <li>■ Off</li> <li>■ TurningOn</li> <li>■ TurningOff</li> <li>■ ShuttingDown</li> <li>■ Suspending</li> <li>■ Resetting</li> <li>■ Rebooting</li> <li>■ Expired</li> <li>■ DeactivateMachine</li> <li>■ UnprovisionMachine</li> <li>■ Disposing</li> <li>■ Finalized</li> </ul>
IaaS 모델 엔티티 생성	지정된 vRealize Automation 모델의 엔티티를 생성하고 유지합니다.
IaaS 모델 엔티티 삭제	지정된 vRealize Automation 모델 엔티티를 삭제합니다.
프로비저닝 이후 작업 호출(더 이상 사용되지 않음)	리소스 작업 요청 워크플로를 사용하십시오.
Blueprint에서 가상 시스템 프로비저닝 (vRealize Automation 7.0에서 제거됨)	카탈로그 항목 요청 또는 프로비저닝 요청과 함께 카탈로그 항목 요청으로 대체되었습니다.
사용자 필터를 사용하여 IaaS 엔티티 읽기	사용자 필터를 사용하여 vRealize Automation 엔티티 목록을 읽습니다. 필터를 지정하지 않은 경우 모든 엔티티가 결과로 반환됩니다.
시스템 쿼리를 사용하여 IaaS 엔티티 읽기	OData 시스템 필터를 사용하여 vRealize Automation 엔티티 목록을 읽습니다. 시스템 필터는 OData URI 규칙에 적용됩니다.
IaaS 모델 엔티티 읽기	해당 ID를 사용하여 vRealize Automation 모델 엔티티를 읽습니다.
IaaS 모델 엔티티 업데이트	해당 ID를 사용하여 vRealize Automation 모델 엔티티를 업데이트합니다.

**확장성** 하위 디렉토리의 워크플로를 사용하여 vRealize Orchestrator 워크플로를 프로비저닝 프로세스의 일부로 호출하거나 사용자 지정 작업 메뉴를 통해 호출하는 기능으로 vRealize Automation을 사용자 지정할 수 있습니다.

이 하위 디렉토리에는 IaaS 자격 증명, 끝점, 엔터프라이즈 그룹, 시스템 접두사 및 기타 엔티티를 관리하는 워크플로도 포함되어 있습니다.

표 3-18. 확장성

워크플로 이름	설명
vCO 사용자 지정 설치	사용자 지정된 상태 변경 워크플로와 메뉴 작업 워크플로를 포함하여 Orchestrator 사용자 지정을 설치합니다.
vCO 사용자 지정 제거	사용자 지정된 상태 변경 워크플로와 메뉴 작업 워크플로를 포함하여 Orchestrator 사용자 지정을 제거합니다.
IaaS 가상 시스템의 예약 변경	예약, 비즈니스 그룹, 관리되는 가상 시스템과 같은 특성을 변경합니다.
IaaS 가상 시스템 가져오기(더 이상 사용되지 않음)	Cloud Client를 사용하십시오. Cloud Client 다운로드 및 설명서는 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 에서 제공됩니다.
vCenter 가상 시스템 가져오기(더 이상 사용되지 않음)	Cloud Client를 사용하십시오. Cloud Client 다운로드 및 설명서는 <a href="https://developercenter.vmware.com/tool/cloudclient">https://developercenter.vmware.com/tool/cloudclient</a> 에서 제공됩니다.
가상 시스템 등록 취소(vRealize Automation 7.0에서 제거됨)	대체 워크플로가 제공되지 않습니다.
Blueprint 및 해당 가상 시스템에 메뉴 작업 할당(더 이상 사용되지 않음)	가상 시스템에서 메뉴 작업을 추가하거나 업데이트합니다. 사용되는 대체 워크플로에는 사용 권한에 리소스 작업 할당 및 복합 Blueprint 가져오기가 포함됩니다.
가상 시스템에 메뉴 작업 할당(더 이상 사용되지 않음)	해당 ID를 사용하여 vRealize Automation 모델 엔티티를 업데이트합니다. 사용되는 대체 워크플로에는 사용 권한에 리소스 작업 할당 및 복합 Blueprint 가져오기가 포함됩니다.
Blueprint 및 해당 가상 시스템에 상태 변경 워크플로 할당(더 이상 사용되지 않음)	vRealize Automation에서 이벤트 브로커 구독으로 대체되었습니다.
메뉴 작업 사용자 지정(vRealize Automation 7.0에서 제거됨)	대체 워크플로가 제공되지 않습니다.
Blueprint 및 해당 가상 시스템에서 메뉴 작업 제거(vRealize Automation 7.0에서 제거됨)	대체 워크플로가 제공되지 않습니다.
Blueprint 및 해당 가상 시스템에서 상태 변경 워크플로 제거	Blueprint 및 해당 가상 시스템에서 상태 변경 워크플로를 제거합니다.

## vRealize Automation IaaS 모델 엔티티 생성

워크플로를 실행하여 사용자에게 대한 가상 시스템 참조와 같은 간단하거나 복잡한 vRealize Automation IaaS 엔티티를 만들 수 있습니다.

### 프로시저

- 1 Orchestrator 클라이언트의 드롭다운 목록에서 **실행** 또는 **설계**를 선택합니다.
- 2 **워크플로** 보기를 클릭합니다.
- 3 **라이브러리 > vRealize Automation > 인프라 관리**를 확장합니다.
- 4 **IaaS 모델 엔티티 생성** 워크플로를 마우스 오른쪽 버튼으로 클릭하고 **워크플로 시작**을 선택합니다.
- 5 vRealize Automation 호스트를 선택합니다.
- 6 **모델 이름** 텍스트 상자에 모델의 이름을 입력합니다.

- 7 **엔티티 집합 이름** 텍스트 상자에 엔티티 집합의 이름을 입력합니다.  
스크립팅 또는 REST API를 사용하여 단순 속성, 복잡한 속성의 링크 및 HTTP 헤더 속성을 설정합니다.
- 8 **제출**을 클릭하여 워크플로를 실행합니다.

## vRealize Automation IaaS 모델 엔티티 읽기

워크플로를 실행하여 vRealize Automation IaaS 모델 엔티티를 읽을 수 있습니다.

### 프로시저

- 1 Orchestrator 클라이언트의 드롭다운 목록에서 **실행** 또는 **설계**를 선택합니다.
- 2 **워크플로** 보기를 클릭합니다.
- 3 **라이브러리 > vRealize Automation > 인프라 관리**를 확장합니다.
- 4 **IaaS 모델 엔티티 읽기**를 마우스 오른쪽 버튼으로 클릭하고 **워크플로 시작**을 선택합니다.
- 5 vRealize Automation 호스트를 선택합니다.
- 6 **모델 이름** 텍스트 상자에 모델의 이름을 입력합니다.
- 7 **엔티티 집합 이름** 텍스트 상자에 엔티티 집합의 이름을 입력합니다.  
스크립팅 또는 REST API를 사용하여 HTTP 헤더 속성을 설정합니다.
- 8 **제출**을 클릭하여 워크플로를 실행합니다.

## vRealize Automation 플러그인 요청 워크플로 사용

요청 워크플로를 사용하여 카탈로그 항목 및 리소스 작업을 요청하고 작업 항목을 완료하거나 취소할 수 있습니다.

작업 항목에는 사용자 입력 또는 작업이 필요합니다. 예를 들어 워크플로 상호 작용, 승인 작업, 회수 요청에 대한 응답 등이 필요합니다.

이러한 워크플로는 vRealize Orchestrator 클라이언트의 **워크플로** 보기에서 플러그인 라이브러리의 **요청** 하위 디렉토리를 통해 액세스할 수 있습니다.

워크플로	설명
작업 항목 취소	활성 작업 항목을 취소합니다. 이 워크플로는 시스템 관리자만 사용할 수 있습니다.
작업 항목 완료	제공된 사용자 입력에 따라 작업 항목을 완료합니다.
카탈로그 항목 요청	워크플로를 실행하는 사용자에게 대한 카탈로그 항목을 요청합니다. 복합 Blueprint를 요청하는 워크플로가 필요한 경우 프로비저닝 요청과 함께 카탈로그 요청 워크플로를 사용합니다.
사용자를 대신해 카탈로그 항목 요청	사용자를 대신해 카탈로그 항목에 대한 요청을 전송합니다. 대리자와 해당 사용자 모두가 권한이 있는 카탈로그 항목에 대해서만 이 워크플로를 사용할 수 있습니다.
프로비저닝 요청과 함께 카탈로그 요청	워크플로를 실행하는 사용자에게 대한 카탈로그 항목으로 복합 Blueprint를 요청합니다. 요청에 사용자 지정 입력을 제공하는 경우 워크플로를 사용자 지정해야 합니다. 복합 Blueprint에 이 워크플로를 사용합니다.
리소스 작업 요청	워크플로를 실행하는 사용자가 소유한 카탈로그 항목에 대한 리소스 작업을 요청합니다.
사용자를 대신해 리소스 작업 요청	사용자를 대신해 리소스 작업에 대한 요청을 전송합니다. 대리자와 해당 사용자 모두가 권한이 있는 리소스 작업에 대해서만 이 워크플로를 사용할 수 있습니다.



워크플로	설명
요청 템플릿을 사용하여 리소스 작업 요청	<p>복잡한 매개 변수가 포함된 리소스 작업을 요청합니다. 워크플로를 복제한 다음 작업에 맞게 사용자 지정하는 것이 가장 좋습니다. 이 워크플로를 사용하여 요청 양식에 표시하지 않으려는 숨겨진 매개 변수 또는 복잡한 매개 변수를 전달할 수 있습니다. 이 워크플로의 주된 응용 분야 중 하나는 IaaS 가상 시스템 재구성 작업을 사용자 지정하는 것입니다.</p> <p>가상 시스템에서 재구성 작업을 생성하려면, 워크플로 사본을 생성한 다음 스크립트를 수정해야 합니다. vRealize Orchestrator에 나타난 매개 변수를 구성하고 Cafe.Shim.VirtualMachine.Reconfigure.Requestor 매개 변수를 설정합니다. 이 매개 변수는 로깅에 사용되며 비어 있지 않아야 합니다. 다음 예를 참조하십시오.</p> <pre>var requestTemplate = vCACCAFERequestsHelper.getRequestForResourceAction(operation)  var jsonData = vCACCAFERequestsHelper.getResourceActionRequestData(requestTemplate); var json = JSON.parse(jsonData); //Change cpu example json.cpu = 2;  //This is a property needed for the Reconfigure IaaS operation: json["Cafe.Shim.VirtualMachine.Reconfigure.Requestor"] = 1; vCACCAFERequestsHelper.setResourceActionRequestData(requestTemplate, JSON.stringify(json));  request = System.getModule("com.vmware.library.vcaccafe.request").requestResourceActionWithRequestTemplate(operation, requestTemplate);</pre>
카탈로그 항목 요청 대기	카탈로그 항목 요청이 완료될 때까지 대기합니다.
리소스 작업 요청 대기	리소스 작업 요청이 완료될 때까지 대기합니다.
작업 항목 대기	작업 항목이 완료될 때까지 대기합니다.

## vRealize Automation 플러그인 샘플 워크플로 사용

샘플 워크플로를 예제로 사용하거나, 사용자 지정 워크플로를 생성하는 시작점으로 사용할 수 있습니다.

이러한 워크플로는 vRealize Orchestrator 클라이언트의 워크플로 보기에서 플러그인 라이브러리의 샘플 하위 디렉토리를 통해 확인할 수 있습니다.

워크플로 이름	설명
사용 권한 생성	권한 부여 클라이언트 및 사용 권한 서비스와 상호 작용하여 vRealize Automation에서 사용 권한을 생성하는 샘플 스크립트를 제공합니다.
테넌트 생성	기본 테넌트와 동일한 vRealize Automation 호스트 및 Active Directory 구성으로 테넌트를 생성합니다. 이 워크플로를 실행하려면 시스템 관리자 자격 증명으로 추가된 vRealize Automation 호스트를 선택합니다. 워크플로를 실행하기 전에 Active Directory 설정을 변경할 수 있습니다.
카탈로그 항목 나열	선택한 테넌트의 카탈로그 항목 목록을 반환합니다.
카탈로그 항목 프로 비저닝 요청을 JSON으로 인쇄	카탈로그 항목에 대한 기본 요청 양식을 검색하여 콘솔 로그에 JSON 형식으로 추가합니다. 이 데이터를 사용하여 프로 비저닝 요청과 함께 카탈로그 항목 요청 워크플로를 수정할 수 있습니다.

## vRealize Automation 플러그인 API 액세스

Orchestrator에서는 vRealize Automation 플러그인 API를 검색할 수 있으며 스크립팅된 요소에서 사용 가능한 JavaScript 개체에 대한 설명서를 볼 수 있는 API 탐색기를 제공합니다.

업데이트된 vRealize Automation API 설명서는

<https://www.vmware.com/support/pubs/vcac-pubs.html>을 참조하십시오.

### 프로시저

- 1 Orchestrator 클라이언트에 관리자로 로그인합니다.
- 2 **도구 > API 탐색기**를 선택합니다.
- 3 왼쪽 창에서 **vCAC** 및 **VCACCAFE** 모듈을 두 번 클릭하여 vRealize Automation 플러그인 API 개체의 계층형 목록을 확장합니다.

### 후속 작업

API 요소에서 코드를 복사하여 스크립팅 상자에 붙여 넣을 수 있습니다. API 스크립팅에 대한 자세한 내용은 VMware vRealize Orchestrator를 사용한 개발을 참조하십시오.

개발 모범 사례에 대한 자세한 내용은 [vRealize Orchestrator 설명서](#)를 참조하십시오.

# 예제 vRealize Automation 플러그인 스크립트

# 4

제공된 JavaScript 예제를 잘라내고, 붙여 넣고, 편집하여 vRealize Automation 작업을 자동화할 수 있는 사용자 지정 스크립트를 개발할 수 있습니다.

이 장에서는 다음 주제에 대해 설명합니다.

- “CRUD 인프라 관리 작업 예제 스크립트,” (27 페이지)
- “vRealize Automation 엔티티 찾기 예제 스크립트,” (31 페이지)
- “vRealize Automation 예제 스크립트를 통해 프로비저닝된 리소스 가져오기,” (32 페이지)
- “일반 작업 예제 스크립트,” (33 페이지)

## CRUD 인프라 관리 작업 예제 스크립트

JavaScript 예제를 잘라내고, 붙여 넣고, 편집하여 CRUD vRealize Automation 작업에 대한 스크립트를 작성할 수 있습니다.

vRealize Orchestrator의 스크립팅에 대한 자세한 내용은 VMware vRealize Orchestrator를 사용한 개발을 참조하십시오.

### 예: vRealize Automation 모델 엔티티 생성

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 모델 이름 및 엔티티 집합 이름을 정의합니다.
- 2 호스트 접두사의 속성을 정의합니다.
- 3 호스트 접두사 엔티티를 저장합니다.
- 4 프로비저닝 그룹의 속성을 정의합니다.
- 5 프로비저닝 그룹을 링크로 정의합니다.
- 6 프로비저닝 그룹 엔티티를 호스트 이름 접두사와 연결하여 저장합니다.

표 4-1. 입력 변수

변수	유형
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'HostNamePrefixes';
var links = null;
var headers = null;
//Create properties for prefix entity
```

```

var prefixInputProperties = {
    MachinePrefix:'test-prefix',
    NextMachineNo:1,
    MachineNumberLength:3
};
//Save the prefix
var prefixEntity = vCACEntityManager
    .createModelEntity(host.id, modelName, entitySetName, prefixInputProperties, links, headers);
entitySetName = 'ProvisioningGroups';
//Create properties for the provisioning group entity
inputProperties = {
    GroupName:'TestGroupName',
    GroupDescription:'This group was generated with a vCO workflow',
    AdministratorEmail:'test@test.com',
    AdContainer:'AD',
    IsTestGroup:false,
    Flags:2,
    GroupType:1};
//Add a reference to the newly created prefix entity
links = {
    HostNamePrefix:prefixEntity
};
//Save the provisioning group
var entity = vCACEntityManager.createModelEntity(host.id, modelName, entitySetName, inputProperties, links,
headers);

```

## 예: vRealize Automation 모델 엔티티 업데이트

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 제공된 엔티티에서 호스트 ID를 가져옵니다.
- 2 제공된 엔티티에서 모델 이름을 가져옵니다.
- 3 제공된 엔티티에서 엔티티 집합 이름을 가져옵니다.
- 4 제공된 엔티티에서 엔티티 ID를 가져옵니다.
- 5 업데이트할 속성 집합을 정의합니다.
- 6 엔티티를 업데이트하는 작업을 시작합니다.

**표 4-2.** 입력 변수

변수	유형
entity	vCAC:Entity
updatedDescription	문자열

```

var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
var entityIdString = entity.keyString;
var links = null;
var headers = null;
var updateProperties = new Properties();

```

```
updateProperties.put("UserNameDescription", updatedDescription);
//Update the user description
System.getModule("com.vmware.library.vcac")
    .updateVCACEntity(hostId, modelName, entitySetName, entityIdString, updateProperties, links, headers);
```

## 예: vRealize Automation 모델 엔티티 읽기

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 모델 이름 및 엔티티 집합 이름을 정의합니다.
- 2 속성 개체와 함께 Blueprint ID를 정의합니다.
- 3 엔티티를 읽습니다.

**표 4-3.** 입력 변수

변수	유형
host	vCAC:VcacHost
blueprintID	문자열

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var links = null;
var headers = null;
//Create properties for the prefix entity
var blueprintId = {
    VirtualMachineTemplateID:blueprintId,
};
//Read the blueprint
var entity = vCACEntityManager
    .readModelEntity(host.id, modelName, entitySetName, blueprintId, headers);
```

## 예: vRealize Automation 모델 엔티티 삭제

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 제공된 엔티티에서 호스트 ID를 가져옵니다.
- 2 제공된 엔티티에서 모델 이름을 가져옵니다.
- 3 제공된 엔티티에서 엔티티 집합 이름을 가져옵니다.
- 4 제공된 엔티티에서 엔티티 ID를 가져옵니다.
- 5 엔티티를 삭제하는 작업을 시작합니다.

**표 4-4.** 입력 변수

변수	유형
entity	vCAC:Entity

```
var hostId = entity.hostId;
var modelName = entity.modelName;
var entitySetName = entity.entitySetName;
var entityKeyString = entity.keyString;
var headers = null;
//Delete the entity
System.getModule("com.vmware.library.vcac")
    .deleteVCACEntity(hostId, modelName, entitySetName, entityKeyString, headers);
```

## 예: 사용자 지정 필터를 사용하여 vRealize Automation 엔티티 읽기

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 모델 이름 및 엔티티 집합 이름을 정의합니다.
- 2 엔티티를 필터링할 기준 속성을 정의합니다.
- 3 엔티티 목록을 읽습니다.

**표 4-5.** 입력 변수

변수	유형
host	vCAC:VcacHost
templateName	문자열

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachineTemplates';
var headers = null;
//Create properties for prefix entity
var properties = {
    VirtualMachineTemplateName: templateName,
};
//Read a list of entities
var entities = vCACEntityManager
    .readModelEntitiesByCustomFilter(host.id, modelName, entitySetName, properties, headers);
```

## 예: 시스템 쿼리를 사용하여 vRealize Automation 엔티티 읽기

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 모델 이름 및 엔티티 집합 이름을 정의합니다.
- 2 엔티티를 필터링할 기준 시스템 쿼리를 정의하고, 시스템 상태 및 구성 요소 플래그별로 필터링된 모든 가상 시스템의 상위 10개 결과를 선택합니다.
- 3 엔티티 목록을 읽습니다.

**표 4-6.** 입력 변수

변수	유형
host	vCAC:VcacHost

```
var modelName = 'ManagementModelEntities.svc';
var entitySetName = 'VirtualMachines';
var filter = "VirtualMachineState eq 'Off' and IsComponent eq true";
var orderBy = 'VirtualMachineName asc';
var top = 10; {
var skip = 0; ,
var headers = null;
var select = null;
var entities = vCACEntityManager
    readModelEntitiesBySystemQuery(host.id, modelName, entitySetName, filter, orderBy, select, top, skip,
headers);
```

## vRealize Automation 엔티티 찾기 예제 스크립트

JavaScript 예제를 잘라내고, 붙여 넣고, 편집하여 vCACCAFEEntitiesFinder 스크립팅 유틸리티 개체를 통해 vRealize Automation 엔티티를 찾는 스크립트를 작성할 수 있습니다.

vRealize Orchestrator의 스크립팅에 대한 자세한 내용은 VMware vRealize Orchestrator를 사용한 개발을 참조하십시오.

### 예: 이름별로 필터링된 카탈로그 리소스 찾기

표 4-7. 입력 변수

변수	유형
host	vCACCAFE:VcacHost

다음 예제 중 하나를 사용할 수 있습니다.

- 이 예제 스크립트는 이름 및 설명을 사용하여 name\_of\_the\_resource 쿼리와 일치하는 대상 호스트의 모든 카탈로그 리소스를 가져옵니다.

```
var items = vCACCAFEEntitiesFinder.findCatalogResources(host, "name_of_the_resource");
```

- 이 예제 스크립트는 다음 작업을 수행합니다.

- 소비자 리소스 서비스를 가져오고 vCACCAFEPAGE0dataRequest 개체의 인스턴스를 Pageable 매개 변수로 전달하는 get 메서드를 호출합니다.
- odata 쿼리를 name\_of\_the\_resource 문자열과 일치하는 name 특성의 단일 필터로 제공하여 vCACCAFEPAGE0dataRequest 개체를 생성합니다.

```
var service = host.createCatalogClient().getCatalogConsumerResourceService();
```

```
var filter = new Array();
filter[0] = vCACCAFEFILTERParam.equal("name", vCACCAFEFILTERParam.string("name_of_the_resource"));
var query = vCACCAFEOdataQuery.query().addFilter(filter);
```

```
var items = service.getResourcesList(new vCACCAFEPAGE0dataRequest(query));
```

### 예: 소유자별로 필터링된 카탈로그 리소스 찾기

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 소비자 리소스 서비스를 가져오고 vCACCAFEPAGE0dataRequest 개체의 인스턴스를 Pageable 매개 변수로 전달하는 get 메서드를 호출합니다.
- 2 odata 쿼리를 user@domain.com 문자열과 일치하는 owner/ref 특성의 단일 필터로 제공하여 vCACCAFEPAGE0dataRequest 개체를 생성합니다.

owners/ref 특성은 카탈로그 리소스의 필드와 내부 구조를 기반으로 하는 컴퍼지션입니다. vCACCAFECatalogResource 엔티티에는 vCACCAFECatalogPrincipal 엔티티의 컬렉션인 owners 특성이 있습니다. vCACCAFECatalogPrincipal 엔티티에는 사용자의 주체 ID에 대한 문자열 표현인 ref 속성이 있습니다.

```
var filter = new Array();
filter[0] = vCACCAFEFILTERParam.substringOf("owners/ref", vCACCAFEFILTERParam.string("user@domain.com"));
var query = vCACCAFEOdataQuery.query().addFilter(filter);
```

```
var items = service.getResourcesList(new vCACCAFEPAGE0dataRequest(query));
```

## 예: 이름 및 소유자별로 필터링된 카탈로그 리소스 찾기

이 예제 스크립트는 vCACCAFEMilterParam.and(array of conditions) 논리 연산자를 사용하여 위 두 예제의 OData 쿼리를 단일 조건으로 결합합니다.

```
var conditions = new Array();
conditions[0] = vCACCAFEMilterParam.equal("name", vCACCAFEMilterParam.string("name_of_the_resource_here"));
conditions[1] = vCACCAFEMilterParam.substringOf("owners/ref",
vCACCAFEMilterParam.string("user@domain.com"));
```

```
var filter = new Array();
filter[0] = vCACCAFEMilterParam.and(conditions);
var query = vCACCAFEOdataQuery.query().addFilter(filter);
```

```
var items = service.getResourcesList(new vCACCAFEPAGE0dataRequest(query));
```

vCACCAFEMilterParam.group(array of parameters), vCACCAFEMilterParam.not(parameter), vCACCAFEMilterParam.startsWith(id, string), vCACCAFEMilterParam.endsWith(id, string), vCACCAFEMilterParam.greaterThan(id, number), vCACCAFEMilterParam.lessThan(id, number) 등의 여러 논리 연산자를 사용하여 다른 조건을 정의할 수 있습니다.

## vRealize Automation 예제 스크립트를 통해 프로비저닝된 리소스 가져오기

JavaScript 예제를 잘라내고, 붙여 넣고, 편집하여 vRealize Automation 프로비저닝된 리소스의 실제 엔티티를 검색할 스크립트를 작성할 수 있습니다.

CatalogResource 유형은 vRealize Automation에서 프로비저닝된 리소스를 나타냅니다. 이 유형에는 다음 특성을 사용하여 카탈로그 리소스와 해당 공급자 간의 관계를 나타내는 ProviderBinding 유형의 특성이 있습니다.

- bindingId - 공급자에 고유한 엔티티의 식별자를 나타냅니다.
- providerRef - vRealize Automation 구성 요소 레지스트리에 등록된 서비스에 직접 해당하는 카탈로그 공급자를 식별합니다.

vRealize Orchestrator의 스크립팅에 대한 자세한 내용은 VMware vRealize Orchestrator를 사용한 개발을 참조하십시오.

## 예: vRealize Automation 카탈로그 리소스로 프로비저닝된 가상 시스템 가져오기

이 예제에서는 vRealize Automation 호스트와 해당 IaaS 호스트를 입력 매개 변수로 사용하고, 제공된 리소스 ID에 대해 해당 IaaS 가상 시스템을 반환합니다. 스크립팅 코드는 iaas-service 공급자를 통해 프로비저닝된 Virtual Machine 유형의 카탈로그 리소스만 가져옵니다.

표 4-8. 입력 변수

변수	유형
vcachost	vCACCAFE:VCACHost
iaasHost	vCAC:VCACHost

```
// Id of the catalog resource (or vCACCAFECatalogResource_instance.getId())
var resourceId = "c222629c-6f90-4458-8c92-8ece0ba06173";
```

```
var resource = vCACCAFEEntitiesFinder.getCatalogResource(vcachost, resourceId);
```



```

var resourceType = resource.getResourceTypeRef().getLabel();
System.log("resource type: " + resourceType);

var providerBinding = resource.getProviderBinding();

var bindingId = providerBinding.getBindingId();
System.log("provider binding id: " + bindingId);

var provider = providerBinding.getProviderRef();
System.log("provider id: " + provider.getId());
System.log("provider name: " + provider.getLabel());

if ((resourceType == "Virtual Machine") && (provider.getLabel() == "iaas-service")) {
    System.log("It is an IaaS VM!");

    // IaaS virtual machine
    var vm = Server.findForType("vCAC:VirtualMachine", bindingId);
    System.log("IaaS VM id: " + vm.virtualMachineId);
    System.log("IaaS VM name: " + vm.displayName);

    // IaaS Entity
    var entity = System.getModule("com.vmware.library.vcac").getVirtualMachineEntityFromId(iaasHost,
bindingId);
    System.log("IaaS entity id: " + entity.keyString);
}

```

## 일반 작업 예제 스크립트

JavaScript 예제를 잘라내고, 붙여 넣고, 편집하거나, 이를 학습용 샘플로 사용하여 일반 vRealize Automation 작업을 위한 사용자 고유의 스크립트를 개발할 수 있습니다.

vRealize Orchestrator의 스크립팅에 대한 자세한 내용은 VMware vRealize Orchestrator를 사용한 개발을 참조하십시오.

## 예: vRealize Automation 고급 서비스 Blueprint 생성

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 서비스 Blueprint를 빌드하는 데 사용되는 vRealize Orchestrator 워크플로를 설정합니다.
- 2 이 워크플로를 기반으로 서비스 Blueprint의 콘텐츠를 생성합니다.
- 3 서비스 Blueprint 엔티티를 생성합니다.
- 4 서비스 Blueprint를 게시합니다.

**표 4-9.** 입력 변수

변수	유형
host	vCACCAFE:VCACHost

```

//ID of the workflow used to create the service blueprint
var workflowId = "44e42047-2fa0-4e4a-ba0c-12086540b28b";

var name = "MyBlueprint"
var description = "Blueprint description";
var workflowClient = host.createAdvancedDesignerClient().getAdvancedDesignerWorkflowService();

```

```
//Generate a service blueprint based on the workflow ID
var blueprint = workflowClient.generateServiceBlueprintByWorkflowId(workflowId);
blueprint.setTenant(host.tenant);
blueprint.setName(name);
blueprint.setDescription(description);

//Create the service blueprint
var blueprintService = host.createAdvancedDesignerClient().getAdvancedDesignerServiceBlueprintService();
var uri = blueprintService.createServiceBlueprint(host.tenant , blueprint);

//Publish the service blueprint
var createdBlueprint = blueprintService.getServiceBlueprintByUri(uri);
blueprintService.updateServiceBlueprintStatus(host.tenant, createdBlueprint.getId(),
vCACCAFEDesignerPublishStatus.PUBLISHED);
```

## 예: vRealize Automation 승인 정책 생성

이 예제 스크립트는 다음 작업을 수행합니다.

- 1 승인 정책 유형을 가져옵니다.
- 2 승인이 필요한 사용자 및 그룹을 설정합니다.
- 3 승인 수준을 설정합니다.
- 4 사전 프로비저닝 승인 단계를 정의합니다.
- 5 사후 프로비저닝 승인 단계를 정의합니다.
- 6 이름, 설명 및 유형과 같은 승인 정책 사양을 정의합니다.
- 7 승인 정책을 생성합니다.
- 8 승인 정책을 게시합니다. 승인 정책이 게시되면 읽기 전용 상태가 됩니다.

**표 4-10.** 입력 변수

변수	유형
host	vCACCAFE:VCACHost

```
// Get the type of approval policy by ID
var typeService = host.createApprovalClient().getApprovalApprovalPolicyTypeService();
var type = typeService.getApprovalPolicyType("com.vmware.cafe.catalog.request");

// Set the user and group required to complete the approval
var user = new vCACCAFEApprovalPrincipal();
user.setValue("user@domain.com");
user.setType(vCACCAFEApprovalPrincipalType.USER);

var group = new vCACCAFEApprovalPrincipal();
group.setValue("group@domain.com");
group.setType(vCACCAFEApprovalPrincipalType.GROUP);

// Set the level of the approval
var level = new vCACCAFEApprovalLevel();
level.setName("IT Approval Level");
level.setDescription("IT Approval Level description");
level.setApprovalMode(vCACCAFEApprovalMode.ALL);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers", user);
```

```

System.getModule("com.vmware.library.vcaccafe.util").addElementToList(level, "getApprovers", group);
level.setLevelNumber(1);

// Set pre-provisioning phase type and the phase of the approval
var phase1Type = new vCACCAFEApprovalPhaseType();
phase1Type.setId("com.vmware.cafe.catalog.request.pre");
phase1Type.setName("Pre-Provisioning type");
phase1Type.setDescription("Pre-Provisioning type description");
phase1Type.setPhaseOrder(1);

var phase1 = new vCACCAFEPhase();
phase1.setName("Pre-Provisioning");
phase1.setDescription("Pre provisioning phase");
phase1.setPhasetype(phase1Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase1, "getLevels", level);

// Set post-provisioning phase type and the phase of the approval
var phase2Type = new vCACCAFEApprovalPhaseType();
phase2Type.setId("com.vmware.cafe.catalog.request.post");
phase2Type.setName("Post-Provisioning type");
phase2Type.setDescription("Post-Provisioning type description");
phase2Type.setPhaseOrder(1);

var phase2 = new vCACCAFEPhase();
phase2.setName("Post-Provisioning");
phase2.setDescription("Post provisioning phase");
phase2.setPhasetype(phase2Type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(phase2, "getLevels", level);

// Create the approval policy specifications
var spec = new vCACCAFEApprovalPolicy();
spec.setName("New Policy");
spec.setDescription("New Policy description");
spec.setPolicyType(type);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase1);
System.getModule("com.vmware.library.vcaccafe.util").addElementToList(spec, "getPhases", phase2);

// Create the approval policy
var approvalPolicyService = host.createApprovalClient().getApprovalApprovalPolicyService();
var approvalPolicy = approvalPolicyService.createPolicy(spec);

// Publish the approval policy
approvalPolicy.setState(vCACCAFEApprovalPolicyState.PUBLISHED);
approvalPolicy = approvalPolicyService.update(approvalPolicy);
System.log("New approval policy id: " + approvalPolicy.getId());

```



# 색인

## A

API 액세스 25

## C

CRUD 작업, vRealize Automation 13,  
14

## I

IaaS 호스트, 구성 11

## V

vCACCAFEEntitiesFinder 개체 사용 31

vRealize Automation, CRUD 작업 13,  
14

vRealize Automation 모델 엔티티  
읽기 24

추가 23

vRealize Automation 모델 엔티티 읽기 24

vRealize Automation 모델 엔티티 추가 23

vRealize Automation 플러그인  
구성 9

소개 7

vRealize Automation 호스트, 구성 10

vRealize Orchestrator 7

## ㄷ

대상 5

## 入

스크립팅 27, 33

스크립팅 가능한 작업 요소 27, 33

## ㅇ

예 32

예제 31

워크플로

CRUD 16, 21, 25

IaaS 21

구성 9

모델 엔티티 16, 21, 25

요청 워크플로 24

표준 워크플로 16, 21, 25

확장성 워크플로 21

워크플로 라이브러리 13, 27

인벤토리 15

## ㅋ

카탈로그 리소스 찾기 31

## ㅋ

프로비저닝된 가상 시스템 가져오기 32

## ㅎ

호스트

관리 9

구성 9

