

# 수명 주기 확장성

vRealize Automation 7.3



vmware®

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

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

Copyright © 2008 – 2017 VMware, Inc. 판권 소유. [저작권 및 상표 정보](#).

# 목차

<b>1</b>	<b>수명 주기 확장성</b>	<b>4</b>
	시스템 확장성 개요	4
	시스템 수명 주기 확장성	4
	수명 주기 확장성 시나리오 선택	6
	vRealize Orchestrator 를 사용하여 시스템 수명 주기 확장	7
	vRealize Orchestrator 검사 목록을 사용하여 시스템 수명 주기 확장	7
	시스템 확장성을 위해 vRealize Automation 플러그인 구성	8
	vRealize Orchestrator 를 사용하여 IaaS 워크플로 사용자 지정	11
	vRealize Automation 확장을 위해 워크플로 구독 구성	12
	vRealize Automation 에서 제공되는 이벤트 항목	13
	워크플로 구독 및 이벤트 브로커 용어	14
	차단 가능 및 회신 가능 이벤트 항목	15
	워크플로 구독에 대한 vRealize Orchestrator 워크플로 생성을 위한 모범 사례	16
	워크플로 구독 설정	17
	프로비저닝 및 수명 주기 워크플로 구독 사용	21
	승인 워크플로 구독 사용	40
	워크플로 구독 문제 해결	45
	vRealize Automation Designer 를 사용하여 시스템 수명 주기 확장	48
	vRealize Automation Designer 검사 목록을 사용하여 시스템 수명 주기 확장	48
	vRealize Automation Designer 설치 및 구성	49
	vRealize Automation Designer 를 사용하여 IaaS 워크플로 사용자 지정	53
	워크플로 및 분산 실행 관리	68
	기술을 사용하여 워크플로와 DEM 작업자 연결	69
	기술 및 DEM 작업자 사이의 연결 제거	70
	기술 및 워크플로 사이의 연결 제거	70
	기술 제거	70
	CloudUtil 명령 참조	71
	DEM 명령	71
	File 명령	72
	Operation 명령	76
	Skill 명령	77
	Workflow 명령	78
	가져오기 명령	81
	vRealize Automation 워크플로 작업 참조	83
	DynamicOps.Repository.Activities	83
	DynamicOps.Cdk.Activities	87

## 수명 주기 확장성

vRealize Orchestrator를 vRealize Automation과 함께 사용하면 IaaS 시스템의 수명 주기를 관리하는 방법을 확장할 수 있습니다.

vRealize Automation을 확장하려면 제공된 vRealize Orchestrator 워크플로를 사용하고 사용자 지정 워크플로를 생성해야 합니다.

이 장에서는 다음 주제에 대해 설명합니다.

- [시스템 확장성 개요](#)
- [vRealize Orchestrator를 사용하여 시스템 수명 주기 확장](#)
- [vRealize Automation 확장을 위해 워크플로 구독 구성](#)
- [vRealize Automation Designer를 사용하여 시스템 수명 주기 확장](#)
- [워크플로 및 분산 실행 관리](#)
- [CloudUtil 명령 참조](#)
- [vRealize Automation 워크플로 작업 참조](#)

### 시스템 확장성 개요

특히 미션 크리티컬 시스템과 같은 새 시스템을 프로비저닝하거나 서비스 해제할 때에는 일반적으로 DNS 서버, 로드 밸런서, CMDB, IP 주소 관리 및 기타 시스템을 비롯한 다양한 관리 시스템과의 상호 작용이 필요합니다.

### 시스템 수명 주기 확장성

워크플로 스텝이라고 하는 IaaS 상태 변경 워크플로를 활용하면 사전 지정된 여러 IaaS 수명 주기 단계에 사용자 지정 논리를 삽입할 수 있습니다. 워크플로 스텝을 사용하여 외부 관리 시스템과의 양방향 통합을 위해 vRealize Orchestrator에 호출할 수 있습니다.

---

**참고** 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. [vRealize Automation 확장을 위해 워크플로 구독 구성](#) 항목을 참조하십시오.

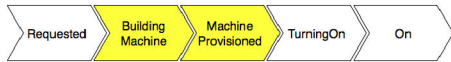
---

상태 변경 워크플로를 생성하면 IaaS 마스터 워크플로가 특정 상태로 전환되기 전에 워크플로 실행을 트리거할 수 있습니다. 예를 들어 외부 데이터베이스에 통합되어 시스템 수명 주기의 여러 단계에서 정보를 기록하도록 사용자 지정 워크플로를 생성할 수 있습니다.

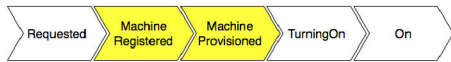
- 마스터 워크플로가 MachineProvisioned 상태로 전환되기 전에 실행되어 시스템 소유자, 승인자 등의 정보를 기록하는 사용자 지정 워크플로를 생성합니다.
- 시스템이 MachineDisposing 상태로 전환되기 전에 실행되어 시스템이 제거된 시간 및 마지막 데이터 수집 시의 리소스 활용도, 마지막 로그인 등의 데이터를 기록하는 사용자 지정 워크플로를 생성합니다.

마스터 워크플로 그림에서는 마스터 워크플로의 기본 상태를 보여 주며 IaaS 워크플로 스텝을 사용하여 사용자 지정할 수 있는 상태가 노란색으로 강조 표시되어 있습니다. 사용자 지정 가능한 상태 변경 워크플로 표에는 사용할 수 있는 워크플로 스텝, 마스터 워크플로 상태에서 해당 워크플로 스텝의 위치 및 시스템 수명 주기를 확장하기 위해 각 상태에서 사용할 수 있는 사용자 지정 논리의 예가 나와 있습니다.

**그림 1-1. 시스템 프로비저닝에 대한 마스터 워크플로 상태**



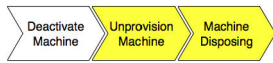
**그림 1-2. 시스템 가져오기에 대한 마스터 워크플로 상태**



**그림 1-3. 시스템 임대 만료에 대한 마스터 워크플로 상태**



**그림 1-4. 시스템 삭제에 대한 마스터 워크플로 상태**



**표 1-1. 사용자 지정 가능한 상태 변경 워크플로**

마스터 워크플로 상태	사용자 지정 가능한 워크플로 이름	확장성 예
BuildingMachine	WFStubBuildingMachine	하이퍼바이저에 생성할 시스템을 준비합니다. CMDB(구성 관리 데이터베이스) 레코드를 생성하고, 시스템에 IP 주소를 할당하도록 외부 시스템을 호출한 다음, 시스템을 삭제하는 동안 WFStubMachineDisposing을 사용하여 IP 주소를 풀에 반환합니다.
RegisterMachine	WFStubMachineRegistered	가져온 시스템을 애플리케이션 프로비저닝 도구에 추가하여 업데이트를 수신하고 규정 준수 검사를 수행합니다.
MachineProvisioned	WFStubMachineProvisioned	시스템이 하이퍼바이저에 있고, 게스트 에이전트 사용자 지정 같은 추가적인 사용자 지정 작업이 이 시점에 완료됩니다. 이 워크플로 스텝을 사용하면 DHCP IP 주소 및 스토리지 정보를 사용하여 CMDB(구성 관리 데이터베이스) 레코드를 업데이트할 수 있습니다. WFStubMachineProvisioned를 사용하여 수행하는 사용자 지정 작업은 일반적으로 WFStubUnprovisionMachine을 사용하여 되돌릴 수 있습니다.

표 1-1. 사용자 지정 가능한 상태 변경 워크플로 (계속)

마스터 워크플로 상태	사용자 지정 가능한 워크플로 이름	확장성 예
Expired	WFStubMachineExpired	만료된 시스템을 비용이 저렴한 스토리지로 이동하여 보관 비용을 줄이고, 스토리지 및 비용 변경 사항을 반영하도록 CMDB 기록과 청구 시스템을 업데이트합니다.
UnprovisionMachine	WFStubUnprovisionMachine	Active Directory 계정에서 시스템을 제거합니다. WFStubMachineProvisioned를 사용하여 수행하는 사용자 지정 작업은 일반적으로 WFStubUnprovisionMachine을 사용하여 되돌릴 수 있습니다.
Disposing	WFStubMachineDisposing	IP 주소를 풀에 반환합니다.

## 수명 주기 확장성 시나리오 선택

vRealize Orchestrator 또는 vRealize Automation Designer를 사용하여 시스템 수명 주기를 확장할 수 있습니다.

vRealize Automation Designer를 사용하여 vRealize Orchestrator에 호출하거나 vRealize Orchestrator를 직접 사용하여 시스템 수명 주기를 확장할 수 있습니다. 두 가지 접근 방법 모두 사용자 지정 vRealize Orchestrator 워크플로를 생성하여 IaaS 시스템 수명 주기의 미리 결정된 단계에 사용자 지정 논리를 넣은 다음 사용자 지정 워크플로를 상태 변경 워크플로 스텝에 삽입할 수 있습니다. 하지만 vRealize Orchestrator를 직접 사용 중인 경우에는 사용자 지정 상태 변경 논리를 특정 Blueprint로만 제한할 수 있으며, vRealize Automation Designer에 의해 워크플로의 실행을 특정 Distributed Execution Manager로만 제한할 수 있습니다.

**참고** 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. [vRealize Automation 확장을 위해 워크플로 구독 구성](#) 항목을 참조하십시오.

표 1-2. 수명 주기 확장성 시나리오 선택

시나리오	절차
사용자 지정 논리를 IaaS 시스템 수명 주기의 미리 결정된 단계에 넣고 해당 사용자 지정 논리를 특정 Blueprint에 적용합니다.	<a href="#">vRealize Orchestrator</a> 검사 목록을 사용하여 시스템 수명 주기 확장
사용자 지정 논리를 IaaS 시스템 수명 주기의 미리 결정된 단계에 넣고 해당 사용자 지정 논리를 모든 Blueprint에 전체적으로 적용합니다.	<a href="#">vRealize Automation Designer</a> 검사 목록을 사용하여 시스템 수명 주기 확장
vRealize Automation Designer에서 기술을 사용하여 워크플로의 실행을 특정 Distributed Execution Manager로 제한합니다. 기술은 워크플로와 DEM 작업자 인스턴스 모두에 적용할 수 있는 태그와 유사합니다.  예를 들어 클라우드 프로비저닝 워크플로를 Amazon URL에 대해 필수 네트워크 액세스 권한을 가진 호스트에서 실행되는 특정 DEM으로 제한할 수 있습니다.	<a href="#">기술을 사용하여 워크플로와 DEM 작업자 연결</a>

## vRealize Orchestrator 를 사용하여 시스템 수명 주기 확장

사용자 지정 vRealize Orchestrator 워크플로를 생성하여 사용자 지정 논리를 미리 결정된 IaaS 시스템 수명 주기 단계에 삽입한 다음 vRealize Orchestrator를 사용하여 사용자 지정 워크플로를 특정 Blueprint로 구축된 시스템의 수명 주기에 삽입할 수 있습니다.

## vRealize Orchestrator 검사 목록을 사용하여 시스템 수명 주기 확장

“vRealize Orchestrator 검사 목록을 사용하여 시스템 수명 주기 확장”은 IaaS 시스템 수명 주기를 사용자 지정하기 위해 vRealize Orchestrator를 설치하고 구성하는 데 필요한 단계에 대한 간략한 내용을 제공합니다.

**표 1-3. vRealize Orchestrator 검사 목록을 사용하여 시스템 수명 주기 확장**

작업	세부 정보
<input type="checkbox"/> vRealize Orchestrator용 vRealize Automation 호스트를 구성합니다.	<a href="#">vRealize Automation 호스트 추가</a>
<input type="checkbox"/> vRealize Orchestrator용 IaaS 호스트를 구성합니다.	<a href="#">IaaS 호스트 추가</a>
<input type="checkbox"/> IaaS 시스템 수명 주기 확장을 위해 vRealize Orchestrator 사용자 지정 항목을 설치합니다.	<a href="#">vRealize Orchestrator 사용자 지정 설치</a>
<input type="checkbox"/> vRealize Orchestrator 인스턴스에 대한 vRealize Automation 끝점을 생성합니다.	<a href="#">vRealize Orchestrator 끝점 생성</a>
<input type="checkbox"/> vRealize Automation 플러그인 라이브러리의 확장성 하위 디렉토리에 제공된 워크플로 템플릿을 사용하여 시스템 수명 주기 동안 실행할 사용자 지정 vRealize Orchestrator 워크플로를 생성합니다. 단일 래퍼 워크플로 아래에 중첩하는 경우라면 동일한 Blueprint에 대해 동일한 상태의 여러 워크플로를 실행할 수 있습니다.	vRealize Orchestrator를 사용한 워크플로 개발에 대한 자세한 내용은 vRealize Orchestrator 설명서를 참조하십시오. vRealize Automation 통합을 위한 vRealize Orchestrator 개발 교육은 VMware Education에서 사용할 수 있는 교육 과정과 VMware Learning에서 제공하는 지침 자료를 참조하십시오.
<input type="checkbox"/> 사용자 지정 워크플로를 IaaS 워크플로 스텝에 삽입하고 IaaS 워크플로 스텝 호출을 위해 Blueprint를 구성하는, 제공된 워크플로를 실행합니다.	<a href="#">Blueprint 및 해당 가상 시스템에 상태 변경 워크플로 할당</a>
<b>참고</b> 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. <a href="#">vRealize Automation 확장을 위해 워크플로 구독 구성</a> 항목을 참조하십시오.	

## 시스템 확장성을 위해 vRealize Automation 플러그인 구성

vRealize Automation 및 IaaS 호스트를 구성하고, 시스템 확장성을 위해 사용자 지정 항목을 설치하고, vRealize Orchestrator 인스턴스에 대한 vRealize Automation 끝점을 생성합니다.

### 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 > 구성을 확장합니다.
- 4 vRA 호스트 추가 워크플로를 마우스 오른쪽 버튼으로 클릭하고 워크플로 시작을 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	인증 사용자 이름 및 인증 암호 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 제출을 클릭합니다.

#### 후속 작업

vRealize Automation 인프라 관리 호스트를 추가합니다.



## laaS 호스트 추가

워크플로를 실행하여 vRealize Automation 호스트의 laaS 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 **laaS 호스트 추가**를 마우스 오른쪽 버튼으로 클릭하고 **워크플로 시작**을 선택합니다.
- 5 **vCAC 호스트** 드롭다운 메뉴에서 laaS 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
SSO	vCenter Single Sign-On을 사용하려면 이 옵션을 선택합니다.
NTLM	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 Orchestrator 사용자 지정 설치

사용자 지정된 상태 변경 워크플로 스텝 및 Orchestrator 메뉴 작업 워크플로를 설치하는 워크플로를 실행할 수 있습니다.

**참고** 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. [vRealize Automation 확장을 위해 워크플로 구독 구성](#) 항목을 참조하십시오.

### 프로시저

- 1 Orchestrator 클라이언트의 드롭다운 목록에서 **실행** 또는 **설계**를 선택합니다.
- 2 워크플로 보기를 클릭합니다.
- 3 라이브러리 > vCloud Automation Center > 인프라 관리 > 확장성 > 설치를 선택합니다.
- 4 vCO 사용자 지정 설치 워크플로를 마우스 오른쪽 버튼으로 클릭하고 워크플로 시작을 선택합니다.
- 5 IaaS 호스트를 선택합니다.
- 6 **다음**을 클릭합니다.
- 7 설치할 상태 변경 워크플로 스텝을 하나 이상 선택하여, 사용자 지정 논리를 추가하려는 수명 주기 단계를 선택합니다.
- 8 **제출**을 클릭합니다.

## vRealize Orchestrator 끝점 생성

vRealize Orchestrator 끝점을 생성하여 vRealize Orchestrator 서버에 연결할 수 있습니다.

서로 다른 vRealize Orchestrator 서버에 연결하도록 여러 끝점을 구성할 수 있지만 각 끝점에 대해 우선 순위를 구성해야 합니다.

vRealize Orchestrator 워크플로를 실행할 때 vRealize Automation은 우선 순위가 가장 높은 vRealize Orchestrator 끝점을 먼저 시도합니다. 이 끝점에 연결할 수 없으면

vRealize Orchestrator 서버가 워크플로를 실행할 수 있을 때까지 우선 순위가 다음으로 높은 끝점을 계속 시도합니다.

### 필수 조건

- IaaS 관리자로 vRealize Automation 콘솔에 로그인합니다.

### 프로시저

- 1 인프라 > 끝점 > 끝점을 선택합니다.

2 새로 만들기 > 오케스트레이션 > vRealize Orchestrator를 선택합니다.

3 이름을 입력하고 원하는 경우 설명을 입력합니다.

4 vRealize Orchestrator 서버의 정규화된 이름 또는 IP 주소와 vRealize Orchestrator 포트 번호가 포함된 URL을 입력합니다.

전송 프로토콜은 HTTPS여야 합니다. 지정된 포트가 없으면 기본 포트 443이 사용됩니다.

vRealize Automation 장치에 포함된 기본 vRealize Orchestrator 인스턴스를 사용하려면 `https://vrealize-automation-appliance-hostname:443/vco`를 입력합니다.

5 사용자 이름 및 암호 텍스트 상자에 vRealize Orchestrator 자격 증명을 입력하여 vRealize Orchestrator 끝점에 연결합니다.

사용하는 자격 증명에는 IaaS에서 호출할 모든 vRealize Orchestrator 워크플로에 대한 실행 권한이 있어야 합니다.

vRealize Automation 장치에 포함된 기본 vRealize Orchestrator 인스턴스를 사용하려는 경우 사용자 이름은 `administrator@vsphere.local`이고 암호는 SSO 구성 시 지정한 관리자 암호입니다.

6 우선 순위 텍스트 상자에 1보다 크거나 같은 정수를 입력합니다.

값이 낮을수록 우선 순위가 높습니다.

7 (선택 사항) 속성을 클릭하고 제공된 사용자 지정 속성, 속성 그룹 또는 끝점에 대한 사용자 고유의 속성 정의를 추가합니다.

8 확인을 클릭합니다.

## vRealize Orchestrator 를 사용하여 IaaS 워크플로 사용자 지정

vRealize Orchestrator에서 단일 워크플로를 사용하여 사용자 지정 논리를 IaaS 워크플로 스텝에 넣고 사용자 지정된 수명 주기를 시스템 Blueprint에 할당합니다.

---

**참고** 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. [vRealize Automation 확장을 위해 워크플로 구독 구성](#) 항목을 참조하십시오.

---

문자열 입력을 수락하려면 사용자 지정 vRealize Orchestrator 워크플로를 설계해야 합니다. 사용자 지정 워크플로에 복합 데이터 유형이 필요한 경우 이 복합 값을 조회하고 이를 문자열로 변환하는 래퍼 워크플로를 생성합니다. 예를 들어 래핑 워크플로는 **라이브러리 > vRealize Automation > 인프라 > 확장성**에 제공된 샘플 워크플로 템플릿을 참조하십시오.

## Blueprint 및 해당 가상 시스템에 상태 변경 워크플로 할당

사용자 지정 워크플로를 상태 변경 워크플로 스텝과 연결하고 워크플로를 Blueprint에 할당하여 마스터 시스템의 특정 단계에서 실행되도록 사용자 지정 vRealize Orchestrator 워크플로를 구성합니다.

**참고** 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. [vRealize Automation 확장을 위해 워크플로 구독 구성](#) 항목을 참조하십시오.

### 필수 조건

vRealize Automation 플러그인 라이브러리의 확장성 하위 디렉토리에 제공된 워크플로 템플릿을 사용하여 시스템 수명 주기 동안 실행할 사용자 지정 워크플로를 생성합니다.

### 프로시저

- 1 Orchestrator 클라이언트의 드롭다운 목록에서 **실행** 또는 **설계**를 선택합니다.
- 2 **워크플로** 보기를 클릭합니다.
- 3 **라이브러리 > vRealize Automation > 인프라 > 확장성**을 선택합니다.
- 4 **Blueprint 및 해당 가상 시스템에 상태 변경 워크플로 할당** 워크플로를 마우스 오른쪽 버튼으로 클릭하고 **워크플로 시작**을 선택합니다.
- 5 **사용할 vCAC 워크플로 스텝** 드롭다운 메뉴에서 스텝을 선택하여 워크플로를 실행할 수명 주기 단계를 선택합니다.
- 6 IaaS 호스트를 선택합니다.
- 7 **다음**을 클릭합니다.
- 8 워크플로를 할당하려는 Blueprint를 선택합니다.
- 9 이러한 워크플로를 이 Blueprint로 프로비저닝된 기존 시스템에 적용할지 여부를 선택합니다.
- 10 시스템 수명 주기 동안 실행할 워크플로를 선택합니다.
- 11 시스템에 사용자 지정 속성으로 추가할 워크플로 입력 값을 구성합니다.
  - a vCO 워크플로 입력을 Blueprint 속성으로 추가합니다.
  - b 마지막 vCO 워크플로 실행 입력 값을 Blueprint 속성으로 추가합니다.
- 12 **제출**을 클릭합니다.

## vRealize Automation 확장을 위해 워크플로 구독 구성

이벤트 브로커 서비스를 사용하는 워크플로 구독을 생성하여 vRealize Automation의 이벤트 메시지에 대해 등록된 서비스를 모니터링하고 구독의 조건이 충족될 때 지정된 vRealize Orchestrator 워크플로를 실행합니다. 구독을 구성하려면 이벤트 항목, 트리거 조건 그리고 트리거 시 실행되는 워크플로를 지정합니다.

테넌트 관리자는 해당 테넌트와 관련된 워크플로 구독을 생성하고 관리할 수 있습니다.

시스템 관리자는 시스템 워크플로 구독을 생성하고 관리할 수 있습니다. 생성된 시스템 워크플로 구독은 모든 테넌트의 이벤트 및 모든 시스템 이벤트에 대해 활성 상태가 됩니다.

## vRealize Automation 에서 제공되는 이벤트 항목

이벤트 항목은 다른 서비스에 의해 이벤트 브로커 서비스로 전송되는 이벤트 메시지의 유형을 설명합니다. 이벤트 항목을 선택하고 항목을 기반으로 워크플로 구독을 구성합니다.

**표 1-4. 이벤트 항목**

이벤트 항목 이름	설명	서비스
Blueprint 구성 요소 완료됨	복합 Blueprint의 일부인 Blueprint 구성 요소가 프로비저닝을 완료합니다. 구성 요소는 복합 Blueprint의 일부인 임의의 Blueprint입니다.	composition-service
Blueprint 구성 요소 요청됨	복합 Blueprint의 일부인 Blueprint 구성 요소가 요청되었습니다. 구성 요소는 복합 Blueprint의 일부인 임의의 Blueprint입니다.	composition-service
Blueprint 구성	Blueprint가 생성, 업데이트 또는 삭제됩니다.	composition-service
비즈니스 그룹 구성	비즈니스 그룹이 생성, 업데이트 또는 삭제됩니다.	identity
카탈로그 항목 요청 완료됨	복합 Blueprint가 프로비저닝을 완료했습니다. 이 이벤트 항목에는 모든 Blueprint 구성 요소가 포함됩니다. 독립형 XaaS Blueprint는 포함되지 않습니다.	composition-service
카탈로그 항목 요청됨	복합 Blueprint가 요청되었습니다. 이 이벤트 항목에는 XaaS Blueprint가 포함되지 않습니다.	composition-service
구성 요소 작업 완료됨	배포 작업이 요청되었을 때 배포된 Blueprint 구성 요소에서 작업이 실행되었습니다.	composition-service
구성 요소 작업 요청됨	배포 작업이 요청되었을 때 배포된 Blueprint 구성 요소에서 실행될 작업이 요청되었습니다.	composition-service
배포 작업 완료됨	배포된 Blueprint에서의 작업이 실행을 완료했습니다. 여기에는 모든 구성 요소 작업 실행이 포함됩니다.	composition-service
배포 작업 요청됨	배포된 Blueprint에서의 작업이 요청되었습니다.	composition-service
EventLog 기본 이벤트	표준 항목이 이벤트 로그에 추가됩니다. 로그 항목은 구독자에게 배포되지 않습니다.	eventlog-service

표 1-4. 이벤트 항목 (계속)

이벤트 항목 이름	설명	서비스
IPAM IP 수명 주기 이벤트 완료	IP 할당 또는 할당 해제 요청이 완료되었습니다.	ipam-service
시스템 수명 주기	제공된 IaaS 작업이 프로비저닝된 시스템에서 실행됩니다.	iaas-service
시스템 프로비저닝	IaaS 시스템이 프로비저닝되는 중입니다.	iaas-service
오케스트레이션 서버 구성	vRealize Orchestrator 서버 구성이 생성, 업데이트, 삭제 또는 서로 다른 기본 인스턴스를 사용하도록 수정됩니다.	o11n-gateway-service
오케스트레이션 서버 구성(XaaS) - 사용되지 않음	vRealize Orchestrator 서버 구성이 생성, 업데이트, 삭제 또는 서로 다른 기본 인스턴스를 사용하도록 수정됩니다.	advanced-designer-service
사후 승인	사후 승인 정책 수준은 이벤트 구독 옵션을 사용하도록 구성됩니다.	approval-service
사전 승인	사전 승인 정책 수준은 이벤트 구독 옵션을 사용하도록 구성됩니다.	approval-service
리소스 회수 완료 이벤트	리소스 리스가 완료되고 리소스가 회수됩니다.	management-service

## 워크플로 구독 및 이벤트 브로커 용어

워크플로 구독 및 이벤트 브로커 서비스를 사용할 때 구독과 이벤트 브로커 서비스에 한정된 몇 가지 용어를 확인할 수 있습니다.

표 1-5. 워크플로 구독 및 이벤트 브로커 용어

용어	설명
이벤트 항목	동일한 논리적 의도와 동일한 구조를 갖는 일련의 이벤트를 설명합니다. 모든 이벤트가 이벤트 항목의 인스턴스입니다.
이벤트	생산자 또는 생산자가 관리하는 엔티티의 상태 변경을 나타냅니다. 이벤트는 이벤트 발생에 대한 정보를 기록하는 엔티티입니다.
메시지	다양한 서비스 및 구성 요소 간 이벤트에 대한 정보를 전송합니다. 예를 들어 생산자에서 이벤트 브로커 서비스로 또는 이벤트 브로커 서비스에서 구독자로 정보를 전송합니다.
이벤트 브로커 서비스	생산자가 구독 소비자에게 게시하는 메시지를 디스패치하는 서비스입니다.
페이로드	이벤트 데이터입니다.
구독	이벤트 항목을 구독하고 알림을 트리거하는 조건을 정의하여 구독자가 이벤트 관련 알림을 받는 것에 관심이 있음을 나타냅니다.
구독자	구독 정의를 기반으로 이벤트 브로커 서비스에 게시된 이벤트를 소비합니다. 구독자는 소비자라고도 합니다.

표 1-5. 워크플로 구독 및 이벤트 브로커 용어 (계속)

용어	설명
제공자	이벤트 브로커 서비스의 이벤트 항목을 등록합니다.
생산자	이벤트 브로커 서비스에 이벤트를 게시합니다.
시스템 관리자	API 또는 vRealize Automation 플러그인을 사용하여 테넌트 워크플로 구독 및 시스템 워크플로 구독을 생성하고, 읽고, 업데이트하고, 삭제할 수 있는 권한을 가진 사용자입니다. vRealize Automation에는 시스템 관리자용 사용자 인터페이스가 포함되지 않습니다.
테넌트 관리자	해당 테넌트에 대해 테넌트 워크플로 구독을 생성하고, 읽고, 업데이트하고, 삭제할 수 있는 권한을 가진 사용자입니다.
워크플로 구독	vRealize Orchestrator 워크플로를 트리거하는 이벤트 항목 및 조건을 지정합니다.
시스템 워크플로 구독	시스템 이벤트 및 모든 테넌트의 이벤트에 반응하는 특수한 워크플로 구독입니다.
테넌트 워크플로 구독	동일한 테넌트의 이벤트에 대해 vRealize Orchestrator 워크플로를 트리거하는 조건을 지정하는 특수한 워크플로 구독입니다.

## 차단 가능 및 회신 가능 이벤트 항목

이벤트 항목에서 차단 가능 및 회신 가능 이벤트를 지원할 수 있습니다. 워크플로 구독의 동작은 항목이 이러한 이벤트 유형을 지원하는지 여부와 워크플로 구독을 구성하는 방법에 따라 다릅니다.

### 차단 불가능 이벤트 항목

차단 불가능 이벤트 항목은 비차단 구독만 생성할 수 있도록 허용합니다. 비차단 구독은 비동기식으로 트리거되므로 구독이 트리거되는 순서 또는 vRealize Orchestrator 워크플로가 실행되는 순서에 의존할 수 없습니다. 비차단 구독은 항목이 회신 가능한 경우 응답만 반환합니다.

### 차단 가능 이벤트 항목

일부 이벤트 항목은 차단을 지원합니다. 워크플로 구독이 차단으로 표시되는 경우 첫 번째 워크플로가 완료될 때까지 일치하는 조건을 가진 다른 워크플로 구독에서 구성된 조건을 충족하는 모든 메시지를 수신하지 않습니다. 동일한 이벤트 항목에 대해 여러 개의 차단 워크플로 구독이 있는 경우 구독의 우선 순위를 지정합니다.

차단 구독은 우선 순위에 따라 실행됩니다. 가장 높은 우선 순위 값은 0입니다. 우선 순위 수준이 같은 동일한 이벤트 항목에 대해 둘 이상의 차단 구독이 있는 경우 구독은 이름을 기반으로 알파벳 순서로 실행됩니다. 모든 차단 구독이 처리되고 나면 메시지가 모든 비차단 구독에 동시 전송됩니다. 차단 워크플로 구독은 동기식으로 실행되기 때문에 후속 워크플로 구독에 대한 알림이 표시되면 변경된 이벤트 페이로드에 갱신된 이벤트가 포함됩니다.

선택된 워크플로 및 목표에 따라 하나 이상의 워크플로 구독에 차단을 적용합니다.

예를 들어 두 번째 워크플로가 첫 번째 워크플로의 결과에 따라 달라지는 두 개의 프로비저닝 워크플로 구독이 있다고 가정하면 첫 번째 구독은 프로비저닝 중에 속성을 변경하고 두 번째 구독은 파일 시스템에 새 속성(아마도 시스템 이름)을 기록합니다. ChangeProperty 구독은 0으로 우선 순위가 지정되고 RecordProperty는 ChangeProperty 구독의 결과를 사용하므로 1로 우선 순위가 지정됩니다. 시스템이 프로비저닝되면 ChangeProperty 구독이 실행을 시작합니다. RecordProperty 구독 조건은 사후 프로비저닝 조건을 기반으로 하기 때문에 메시지가 RecordProperty 구독을 트리거합니다. 하지만 ChangeProperty 워크플로는 차단 워크플로이므로 완료되기 전까지 메시지가 수신되지 않습니다. 이름이 변경되고 첫 번째 워크플로가 완료되면 두 번째 워크플로가 실행되면서 파일 시스템에 이름을 기록합니다.

차단을 지원하는 이벤트 항목인 경우에도 워크플로 구독에 종속된 후속 워크플로가 없는 경우 비차단 워크플로 구독을 생성할 수 있습니다. 워크플로 구독이 트리거되고 vRealize Automation 또는 외부 시스템의 추가 상호 작용 없이 vRealize Orchestrator 워크플로를 실행합니다.

## 회신 가능 이벤트 항목

일부 이벤트 항목은 구독 서비스의 회신을 지원합니다. 회신 가능 이벤트 항목을 등록한 서비스는 보통 시스템 또는 사용자와의 상호 작용의 결과로서 워크플로 출력을 제공하는 회신 이벤트를 승인할 수 있습니다. 회신 출력 매개 변수는 원래 회신 가능 이벤트를 게시한 vRealize Automation 서비스가 처리할 수 있도록 회신 스키마에 정의된 조건을 충족해야 합니다. 예를 들어 사전 승인 및 사후 승인 워크플로 구독은 회신 가능합니다. 승인 요청을 외부 시스템으로 전송하는 워크플로를 생성하는 경우 승인 또는 거부된 회신이 vRealize Automation에 의해 처리되고 카탈로그 항목이 프로비저닝되거나 사용자에게 요청이 거부되었다는 알림이 표시됩니다.

회신은 vRealize Orchestrator 워크플로의 출력일 수 있고 워크플로가 시간 초과되거나 실패하는 경우에는 오류일 수 있습니다. 회신이 워크플로 출력 매개 변수로부터 온 것인 경우 회신은 올바른 회신 스키마 형식이어야 합니다.

## 워크플로 구독에 대한 vRealize Orchestrator 워크플로 생성을 위한 모범 사례

워크플로 구독은 특정 항목 스키마를 기반으로 합니다. 구독에서 vRealize Orchestrator 워크플로를 시작할 수 있으려면 이벤트 데이터를 사용하도록 올바른 입력 매개 변수로 워크플로를 구성해야 합니다.

### 워크플로 입력 매개 변수

생성하는 사용자 지정 워크플로에는 모든 매개 변수 또는 페이로드의 모든 데이터를 사용하는 단일 매개 변수가 포함될 수 있습니다.

- 개별 매개 변수를 포함하려면 하나 이상의 매개 변수를 구성합니다. 이름 및 유형이 스키마에 지정된 이름 및 유형과 일치하는지 확인합니다.
- 단일 매개 변수를 사용하려면 유형이 Properties인 하나의 매개 변수를 구성합니다. 유용한 아무 이름이나 제공할 수 있습니다. 예를 들어 매개 변수 이름으로 payload를 사용할 수 있습니다.



## 워크플로 출력 매개 변수

생성하는 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 스키마와 일치해야 합니다.

## 워크플로 구독 설정

이 구독 옵션은 vRealize Automation의 이벤트 메시지를 기반으로 워크플로가 실행되는 시기를 결정합니다. 옵션을 사용하여 구독을 관리합니다.

구독은 지정된 이벤트 항목에 대한 이벤트를 구독하고 정의된 조건과 일치하는 항목에 대한 이벤트를 수신할 때 워크플로를 실행하려는 사용자의 의도를 나타냅니다.

워크플로 구독을 생성하려면 테넌트 관리자여야 합니다. 모든 워크플로 구독은 테넌트와 관련되어 있습니다.

워크플로 구독을 관리하려면 **관리 > 이벤트 > 구독**을 선택합니다.

**표 1-6. 워크플로 구독 옵션**

옵션	설명
새로 만들기	새 구독을 생성합니다.
편집	선택된 구독을 수정합니다. 구독이 게시되면 저장된 변경 내용은 즉시 활성화됩니다. 게시되거나 게시 취소된 구독에 대한 차단 옵션을 수정하거나 이벤트 항목을 편집할 수 없습니다.
게시	구독을 활성화합니다. 이벤트 브로커 서비스의 이벤트가 처리되고 구독 조건이 평가됩니다. 구성된 조건이 True인 경우 워크플로가 트리거됩니다.
게시 취소	구독을 초안 상태로 반환합니다. 구독이 환경에서 더 이상 활성 상태가 아니며 더 이상 이벤트를 수신하지 않습니다. 구독을 다시 게시하면 구독이 새 이벤트를 수신하기 시작합니다. 지난 이벤트는 수신되지 않습니다.
삭제	선택된 구독을 삭제합니다.

### ■ 워크플로 구독 이벤트 항목 탭 설정

이벤트 항목은 vRealize Automation에 제공된 이벤트의 클래스입니다. 구독을 정의할 이벤트 항목을 선택합니다.

### ■ 워크플로 구독 조건 탭 설정

구독에 대해 구성하는 조건은 이벤트 데이터를 기반으로 실행하도록 워크플로를 트리거할지 결정합니다.

### ■ 워크플로 구독 워크플로 탭 설정

선택한 vRealize Orchestrator 워크플로가 구독 조건이 True로 평가될 때 실행됩니다.

## ■ 워크플로 구독 세부 정보 탭 설정

구독 세부 정보는 구독이 처리되는 방법을 결정합니다.

## 워크플로 구독 이벤트 항목 탭 설정

이벤트 항목은 vRealize Automation에 제공된 이벤트의 클래스입니다. 구독을 정의할 이벤트 항목을 선택합니다.

이 페이지는 정보를 제공하기 위한 것입니다. 어떤 값도 제공할 필요가 없습니다.

워크플로 구독에 대한 이벤트 항목을 선택하려면 **관리 > 이벤트 > 구독**을 선택합니다. **새로 만들기**를 클릭하고 이벤트 항목을 선택합니다.

**표 1-7. 이벤트 항목 옵션**

옵션	설명
항목 ID	이벤트 항목 식별자입니다.
이름	이벤트 항목의 이름입니다.
설명	이벤트 항목에 대한 설명입니다.
게시자	이 이벤트 항목을 등록할 서비스의 이름입니다.
차단 가능	이 이벤트 항목에 대해 차단 구독을 생성할 수 있는지 여부를 나타냅니다. 차단 구독은 동일한 이벤트에 대한 두 번째 워크플로의 결과가 첫 번째 워크플로의 결과에 따라 달라질 때 이벤트의 페이로드를 변경하거나 사용자 지정 논리를 동기식으로 실행하는데 사용됩니다.
회신 가능	이벤트 항목 구독에서 이벤트를 처음 생산한 서비스에 회신 이벤트를 게시할 수 있는지 여부를 나타냅니다. 값이 예인 경우 워크플로가 완료되면 원래 이벤트를 게시한 서비스로 회신이 전송됩니다. 회신에는 vRealize Orchestrator 워크플로의 출력 및 모든 오류 세부 정보가 포함됩니다.
스키마	이벤트 페이로드의 구조를 설명합니다. 스키마를 사용하여 페이로드 정보를 사용할 수 있는 워크플로를 생성할 수 있습니다.

## 워크플로 구독 조건 탭 설정

구독에 대해 구성하는 조건은 이벤트 데이터를 기반으로 실행하도록 워크플로를 트리거할지 결정합니다.

**조건에 따라 실행**을 선택하는 경우 사용 가능한 옵션에는 다음이 포함될 수 있습니다.

- 데이터. 선택된 이벤트 항목과 관련된 이벤트 메시지의 정보입니다. 예를 들어 시스템 수명 주기 이벤트 항목에 대한 조건을 생성하는 경우 데이터 필드는 Blueprint 및 시스템과 관련됩니다. 승인 전 이벤트 항목을 선택하는 경우 데이터 필드는 승인 정책과 관련됩니다.

또한 트리 위의 텍스트 상자에 경로를 입력하여 스키마에 포함되지 않은 필드에 대한 조건을 추가할 수도 있습니다. **\${PATH}** 형식을 사용합니다. PATH는 스키마의 경로입니다. ~를 사용하여 노드를 구분합니다. 예를 들어 **\${data~machine~properties~SomeCustomProperty}**와 같이 입력합니다.

- 핵심 이벤트 메시지 값. 이벤트 메시지에 대한 일반 정보입니다. 예를 들어 이벤트 유형, 타임 스탬프 또는 사용자 이름과 같은 정보가 포함됩니다.

워크플로 구독에 대한 이벤트 항목을 선택하려면 **관리 > 이벤트 > 구독**을 선택합니다. **새로 만들기**를 클릭합니다.

**표 1-8. 조건 탭 옵션**

옵션	설명
모든 이벤트에 대해 실행	이 이벤트 항목에 대한 메시지가 수신되면 선택된 워크플로가 실행됩니다.
조건에 따라 실행	<p>이벤트 메시지가 감지되고 이벤트가 구성된 조건을 충족할 때 선택된 워크플로가 실행됩니다.</p> <p>이 옵션을 선택하는 경우 이 구독에 대해 선택된 워크플로를 트리거하기 위해 이벤트 데이터를 기반으로 조건을 지정해야 합니다.</p> <ul style="list-style-type: none"> <li>■ <b>단일 조건.</b> 구성된 절이 True일 때 워크플로가 트리거됩니다.</li> <li>■ <b>다음 중 모두.</b> 모든 절이 True이고 최소 두 개의 조건을 제공했을 때 워크플로가 트리거됩니다.</li> <li>■ <b>다음 중 일부.</b> 하나 이상의 절이 True이고 최소 두 개의 조건을 제공했을 때 워크플로가 트리거됩니다.</li> <li>■ <b>다음 제외.</b> True인 절이 없을 때 워크플로가 트리거됩니다.</li> </ul> <p>상수 값을 기반으로 조건을 생성하는 경우 해당 값은 대/소문자를 구분하지 않는 상태로 처리됩니다. 예를 들어 조건이 UNIX가 포함된 Blueprint 이름이지만 Blueprint가 이름에 Unix를 사용하는 경우 조건은 계속해서 올바르게 처리됩니다.</p> <p>Blueprint 이름과 일치하도록 조건 이름을 변경하려면 먼저 해당 값을 동일한 문자열이 포함되지 않은 특정 값으로 변경합니다. 예를 들어 조건 UNIX를 편집하려면 해당 값을 xxxx로 변경하고 저장한 다음 xxxx를 Unix로 변경하고 저장합니다.</p>

## 워크플로 구독 워크플로 탭 설정

선택한 vRealize Orchestrator 워크플로가 구독 조건이 True로 평가될 때 실행됩니다.

실행하려는 워크플로가 **관리 > vRO 구성 > 서버 구성**에 지정된 vRealize Orchestrator에 이미 존재합니다.

워크플로 구독에 대한 워크플로를 선택하려면 **관리 > 이벤트 > 구독**을 선택합니다. **새로 만들기**를 클릭합니다.

**표 1-9. 워크플로 탭 옵션**

옵션	설명
워크플로 선택	워크플로로 이동합니다.
선택한 워크플로	실행하려는 항목임을 확인할 수 있도록 입력 및 출력 매개 변수를 포함하여 워크플로에 대한 정보를 표시합니다.

## 워크플로 구독 세부 정보 탭 설정

구독 세부 정보는 구독이 처리되는 방법을 결정합니다.

워크플로 구독을 관리하려면 **관리 > 이벤트 > 구독**을 선택합니다. **새로 만들기**를 클릭합니다.

옵션은 이벤트 항목의 유형에 따라 달라집니다.

**표 1-10. 세부 정보 탭 옵션**

옵션	설명
이름	기본적으로, 표시된 이름은 선택된 워크플로의 이름입니다. 이 이름이 구독 목록에 표시됩니다. 이름은 테넌트에서 고유해야 합니다.
우선 순위	차단 구독이 실행되는 순서입니다. 0이 가장 높은 우선 순위입니다. 이벤트 항목에 우선 순위가 동일한 여러 개의 차단 워크플로 구독이 있는 경우 구독은 구독 이름을 기준으로 알파벳 순서로 처리됩니다. 이 옵션은 차단 워크플로 구독에서만 사용할 수 있습니다.
시간 제한(분)	워크플로가 실패로 간주되기 전에 완료되어야 하는 시간(분)을 지정합니다. 워크플로가 지정된 시간 내에 완료되지 못하면 워크플로가 취소되고 메시지가 우선 순위 목록의 다음 구독에 전송됩니다. 값을 지정하지 않는 경우 시간 제한은 무제한입니다. 차단 또는 회신 가능 이벤트에 대해 회신이 필요한 서비스에는 고유한 기본 시간 제한 값이 있을 수 있습니다. 예를 들어 IaaS 프로비저닝 및 수명 주기 이벤트 항목은 30분에서 시간 제한이 발생합니다. 이 값은 IaaS 서버에 구성됩니다. 승인 항목의 기본값은 24시간입니다. 이 값은 시스템에 구성됩니다.

표 1-10. 세부 정보 탭 옵션 (계속)

옵션	설명
설명	기본적으로 표시되는 설명은 워크플로 설명입니다.
차단	<p>워크플로에서 회신을 기다리는 동안 동일한 이벤트 항목에 대해 후속 워크플로가 이벤트 메시지를 수신하지 못하도록 차단할 수 있는지 여부를 결정합니다.</p> <p>차단을 사용하도록 설정된 구독은 우선 순위를 기반으로 동일한 이벤트 항목에 대해 차단으로 구성되지 않은 구독 앞에서 메시지를 수신합니다. 워크플로가 완료되면 메시지가 우선 순위의 다음 차단 구독에 전송됩니다. 모든 차단 구독이 처리되고 나면 메시지가 모든 비차단 구독에 동시 전송됩니다.</p> <p>이벤트 항목이 차단 가능한 경우에만 차단 옵션을 사용할 수 있습니다. 이 정보는 [이벤트 항목] 탭에 제공됩니다.</p> <p>차단 적격성은 [이벤트 항목] 탭에 표시되어 있습니다.</p> <ul style="list-style-type: none"> <li>■ 확인란을 선택하지 않으면 이벤트 브로커가 후속 워크플로를 차단하지 않습니다.</li> <li>■ 확인란을 선택하면 이벤트 브로커가 구성된 조건을 기반으로 어떤 워크플로 구독이 이 이벤트에 적합한지 계산하고 우선 순위에 따라 워크플로를 실행합니다. 이벤트 브로커는 다음 워크플로를 실행하기 전에 각 워크플로의 응답을 기다립니다. 현재 워크플로의 실행 이후에 변경된 모든 매개 변수는 대기열에 있는 다음 워크플로에 전달됩니다.</li> </ul> <p>응답을 기다리는 동안 사용 중인 시스템이 응답할 때까지 다른 워크플로는 이벤트에 대한 알림을 받지 못합니다.</p> <p>워크플로 구독이 게시된 후에는 이 옵션을 수정할 수 없습니다.</p>

## 프로비저닝 및 수명 주기 워크플로 구독 사용

vRealize Orchestrator를 사용하여 IaaS 시스템의 관리를 확장하도록 프로비저닝 및 수명 주기 워크플로 구독을 생성합니다. 프로비저닝 구독은 프로비저닝 프로세스 동안 수행할 수 있는 작업을 확장합니다. 수명 주기 구독은 사용자가 프로비저닝된 항목을 관리할 때 수행할 수 있는 작업을 확장합니다.

### IaaS 서비스 통합

IaaS 서비스에서 생성하는 메시지를 기반으로 사용자 지정 vRealize Orchestrator 워크플로를 실행하는 프로비저닝 또는 수명 주기 이벤트 항목 기반 워크플로 구독을 생성합니다.

vRealize Automation에는 IaaS 통합에 사용할 수 있는 두 개의 이벤트 항목이 포함됩니다.

- 시스템 프로비저닝. IaaS 시스템의 프로비저닝 및 삭제 중에 워크플로를 실행하는 워크플로 구독을 생성합니다.
- 시스템 수명 주기. 소유 사용자가 프로비저닝된 시스템에서 실행하는 관리 작업과 관련된 워크플로를 실행하는 워크플로 구독을 생성합니다.

### 프로비저닝 및 수명 주기 워크플로에 대해 vRealize Orchestrator 워크플로 구성

IaaS 서비스 메시지를 지원하려면 vRealize Orchestrator 워크플로를 구성해야 합니다.

## 프로비저닝 및 수명 주기 이벤트 항목 스키마

시스템 프로비저닝 및 시스템 수명 주기 이벤트 항목은 동일한 수명 주기 스키마를 사용합니다. 차이점은 트리거 상태에 있습니다. 시스템 프로비저닝은 프로비저닝 상태 및 이벤트를 기반으로 메시지를 수신하고 시스템 수명 주기는 활성 상태 및 이벤트를 기반으로 메시지를 수신합니다. 일부 프로비저닝 상태에는 BuildingMachine 및 Disposing이 포함됩니다. 일부 수명 주기 상태에는 InstallTools 및 Off가 포함됩니다.

이벤트 메시지는 이벤트 데이터 페이로드입니다. 다음은 이벤트 데이터 페이로드의 구조입니다.

```
{
  machine : {
    id          : STRING,      /* IaaS machine ID */
    name         : STRING,      /* machine name */
    externalReference : STRING, /* machine ID on the hypervisor */
    owner        : STRING,      /* machine owner */
    type         : INTEGER,     /* machine type: 0 - virtual machine; 1 - physical machine; 2 - cloud machine */
    properties    : Properties  /* machine properties, see notes below how to expose virtual machine properties */
  },
  blueprintName : STRING,      /* blueprint name */
  componentId   : STRING,      /* component id */
  componentTypeId : STRING,    /* component type id */
  endpointId    : STRING,      /* endpoint id */
  requestId     : STRING,      /* request id */
  lifecycleState : {                               /* see Life Cycle State Definitions */
    state : STRING,
    phase : STRING,
    event : STRING
  },
  virtualMachineEvent : STRING, /* fire an event on that machine - only processed by Manager Service as consumer */
  workflowNextState   : STRING, /* force the workflow to a specific state - only processed by Manager Service as consumer */
  virtualMachineAddOrUpdateProperties : Properties, /* properties on the machine to add/update - only processed by Manager Service as consumer */
  virtualMachineDeleteProperties     : Properties /* properties to remove from the machine - only processed by Manager Service as consumer */
}
```

vRealize Orchestrator 매개 변수는 이름 및 유형별로 이벤트의 페이로드에 매핑됩니다.

virtualMachineEvent 및 workflowNextState를 출력 매개 변수로 사용할 때, 제공하는 값은 이벤트를 트리거하고 현재 vRealize Orchestrator 워크플로를 시작한 워크플로의 상태 또는 이벤트를 나타내야 합니다. 가능한 수명 주기 상태 및 이벤트를 검토하려면 [VMPS 마스터 워크플로 수명 주기 상태 및 시스템 유형별 프로비저닝 수명 주기 상태](#) 항목을 참조하십시오.

## 확장성 사용자 지정 속성 사용

가상 시스템 사용자 지정 속성은 수명 주기 상태에 대해 확장성 사용자 지정 속성으로 지정된 경우가 아니면 이벤트 페이로드에 포함되지 않습니다. 이러한 속성을 IaaS 끝점, 예약, Blueprint, 요청 및 사용자 지정 속성을 지원하는 기타 개체에 추가할 수 있습니다.

개체에 추가하는 사용자 지정 속성의 형식은 `Extensibility.Lifecycle.Properties.{workflowName}. {stateName}`입니다.

예를 들어 가상 시스템 상태가 `BuildingMachine`일 때 숨겨진 속성과 “Virtual”로 시작하는 모든 속성을 포함시키려면 `Blueprint`의 시스템에 사용자 지정 속성을 추가합니다. 이 예의 사용자 지정 속성 이름은 `Extensibility.Lifecycle.Properties.VMPSMasterWorkflow32.BuildingMachine`이고, 값은 `__*` 및 `Virtual*`입니다(점표로 구분됨).

두 마디의 밑줄(`__*`)에는 숨겨진 속성이 포함됩니다. `Virtual*` 값에는 ‘virtual’로 시작하는 모든 속성이 포함됩니다. 별표(`*`)는 와일드카드로, 단독 값으로 사용할 수 있지만 이런 식으로 와일드카드를 사용하면 대량의 데이터가 전송됩니다.

사용자 지정 속성이 포함된 나중에 트리거되는 여러 개의 워크플로 구독이 있는 경우 페이로드 확인에서 사용자 지정 속성이 유지되도록 워크플로에 적절한 항목을 포함시켜야 합니다.

**표 1-11. 사용자 지정 속성을 유지하기 위한 작업 항목**

상태	작업 항목
추가되거나 업데이트된 사용자 지정 속성	<pre>virtualMachineAddOrUpdateProperties = payload.virtualMachineAddOrUpdateProperties    new Properties();</pre>
삭제된 사용자 지정 속성	<pre>virtualMachineDeleteProperties = payload.virtualMachineDeleteProperties    new Properties();</pre>

## 수명 주기 또는 프로비저닝 스키마를 기반으로 vRealize Orchestrator 워크플로 생성

생성하는 사용자 지정 워크플로에는 `Properties` 유형을 가진 `payload`인 입력 매개 변수가 있어야 합니다. 워크플로가 vRealize Orchestrator에서 실행될 때 프로비저닝 또는 수명 주기 이벤트 데이터 페이로드가 이 매개 변수에 위치합니다. 또한 이벤트의 페이로드에 있는 필드의 이름 및 유형과 일치하는 개별 입력 매개 변수를 포함할 수도 있습니다.

## 워크플로 구독 수명 주기 상태 정의

수명 주기 상태를 기반으로 워크플로 구독 조건을 구성하는 경우 다음과 같은 정의가 값을 식별하는 데 도움이 될 수 있습니다.

각 메시지는 `laaS` 시스템 상태 변경에 기반한 `lifecycleState` 요소가 포함되어 있습니다.

메시지에서 요소에는 다음과 같은 구조가 있습니다.

```
lifecycleState : {
  state : STRING,
  phase : STRING,
  event : STRING
}
```

표 1-12. LifecycleState 요소

속성	설명	형식 및 값	예
state	워크플로 이름 및 상태 이름이 포함되어 있습니다.	{workflowName}.{stateName}	<ul style="list-style-type: none"> <li>■ VMPSMasterWorkflow32.Requested</li> <li>■ VMPSMasterWorkflow32.MachineActivated</li> <li>■ BasicVmWorkflow.BuildComplete</li> </ul>
phase	메시지를 트리거한 단계가 포함되어 있습니다.	PRE, POST, EVENT	<ul style="list-style-type: none"> <li>■ PRE. 이 상태에 진입하면 이벤트가 게시됩니다.</li> <li>■ POST. 이 상태를 종료하면 이벤트가 게시됩니다.</li> <li>■ EVENT. 이 상태에서 IaaS 이벤트가 수신되면 이벤트가 게시됩니다.</li> </ul>
event	이벤트가 포함되어 있습니다. 이 속성은 선택 사항이며 단계가 EVENT인 경우에만 존재합니다.	{workflowName}. {stateName}.EVENT. {eventName}	<ul style="list-style-type: none"> <li>■ VMPSMasterWorkflow32.Requested.EVENT.OnProvisionMachine</li> <li>■ VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnBuildSuccess</li> <li>■ BasicVmWorkflow.CreatingMachine.EVENT.OnCreatingMachineComplete</li> </ul>

### VMPS 마스터 워크플로 수명 주기 상태

VMPS 마스터 워크플로 수명 주기 상태는 요청에서 소멸까지의 IaaS 가상 시스템 수명 주기를 나타냅니다. 수명 주기 상태 이벤트와 수명 주기 상태 이름을 기반으로 트리거 조건을 생성할 때 VMPS 마스터 워크플로 상태 및 이벤트를 사용할 수 있습니다.

각 가상 시스템은 4개의 기본 단계를 거칩니다.

- 요청. 승인이 포함됩니다.
- 프로비저닝. 생성, 복제, kickstart 또는 WIM과 같은 서로 다른 프로비저닝 유형이 포함됩니다.
- 관리. 전원 켜기, 전원 끄기, 스냅샷과 같은 작업이 포함됩니다.
- 제거. 시스템 비활성화, 프로비저닝 취소 및 삭제가 포함됩니다.

이러한 기본 단계는 마스터 워크플로에 포함되어 있습니다. 다음 이벤트 항목에 대한 조건을 생성할 때 VMPSMasterWorkflow32 상태를 사용할 수 있습니다.

- 시스템 수명 주기
- 시스템 프로비저닝

글로벌 이벤트 상태는 VMPS 마스터 워크플로에 의해 이벤트 브로커로 전송되는 메시지입니다. 글로벌 이벤트는 언제든지 트리거할 수 있습니다.



표 1-13. 글로벌 이벤트

상태(항목)	이벤트 [트리거 문자열](항목)
글로벌	<ul style="list-style-type: none"> <li>onBuildFailure [BuildFailure] (Provision)</li> <li>OnBuildSuccess [BuildSuccess] (Provision)</li> <li>OnFinalizeMachine [Destroy] (Provision)</li> <li>OnForceUnregisterEvent [ForceUnregister] (Provision)</li> <li>ReconfigureVM.Pending [ReconfigureVM.Pending](활성)</li> <li>ReconfigureVM.ExecutionUpdated [ReconfigureVM.ExecutionUpdated](활성)</li> <li>ReconfigureVM.RetryRequestMade [ReconfigureVM.RetryRequestMade](활성)</li> <li>ReconfigureVM.Failed [ReconfigureVM.Failed](활성)</li> <li>ReconfigureVM.Successful [ReconfigureVM.Successful](활성)</li> <li>ReconfigureVM.Complete [ReconfigureVM.Complete](활성)</li> <li>ReconfigureVM.Canceled [ReconfigureVM.Canceled](활성)</li> </ul>

활성 글로벌 상태는 프로비저닝된 시스템에서 실행할 수 있는 작업입니다.

표 1-14. 활성 이벤트

상태	이벤트 [트리거 문자열](항목)
Active	<ul style="list-style-type: none"> <li>OnExpireLease [Expire] (Active)</li> <li>OnForceExpire [ForceExpire] (Active)</li> <li>onReprovision [Reprovision] (Active)</li> <li>onResetBuildSuccess [ResetBuildSuccess] (Active)</li> </ul>

마스터 워크플로에서 시스템 프로비저닝 수명 주기 동안 프로비저닝 이벤트가 발생합니다. 활성 이벤트는 프로비저닝된 시스템에서 실행할 수 있는 작업입니다. 마스터 워크플로에 대한 그림은 [VMPS 마스터 워크플로의 예](#) 항목을 참조하십시오.

각 시스템 유형에는 고유한 프로비저닝 워크플로가 있습니다. 개별 시스템 유형에 대한 정보는 [시스템 유형별 프로비저닝 수명 주기 상태](#) 항목을 참조하십시오.

표 1-15. VMPSMasterWorkflow32 상태 및 이벤트

상태(항목)	이벤트 [트리거 문자열](항목)
BuildingMachine	<ul style="list-style-type: none"> <li>Pre(Provision)</li> <li>Post(Provision)</li> </ul>
DeactivateMachine	<ul style="list-style-type: none"> <li>Pre(Provision)</li> <li>Post(Provision)</li> </ul>

표 1-15. VMPSMasterWorkflow32 상태 및 이벤트 (계속)

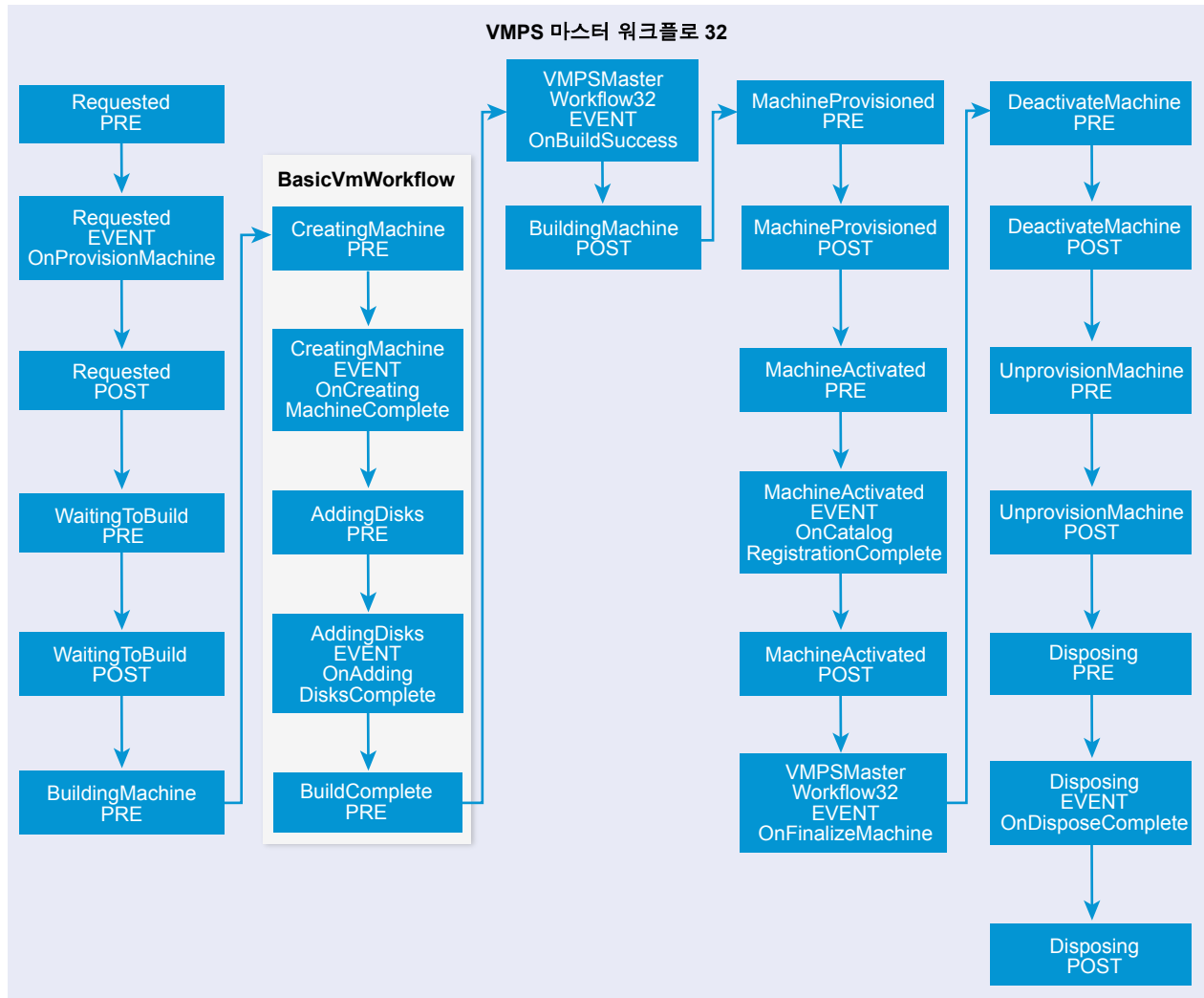
상태(항목)	이벤트 [트리거 문자열](항목)
삭제 <ul style="list-style-type: none"> <li>Pre(Provision)</li> <li>Post(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnDisposeComplete(Provision)</li> <li>OnDisposeTimeout(Provision)</li> <li>OnUnregisterMachine [Unregister] (Provision)</li> </ul>
Expired <ul style="list-style-type: none"> <li>Pre(Active)</li> <li>Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>OnActiveExpiredMachine [ActivateExpiredMachine] (Active)</li> <li>TurnOffFromExpired [TurnOffExpiredMachine] (Active)</li> </ul>
InstallTools <ul style="list-style-type: none"> <li>Pre(Active)</li> <li>Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>InstallToolsComplete(Active)</li> <li>TimeoutInstallTools(Active)</li> </ul>
Leased	<ul style="list-style-type: none"> <li>OnChangeLease [Change Lease] (Active)</li> <li>OnUpdateDescription [UpdateDescription] (Active)</li> <li>OnUpdateOwner [UpdateOwner] (Active)</li> </ul>
MachineActivated <ul style="list-style-type: none"> <li>Pre(Provision)</li> <li>Post(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnCatalogRegistrationComplete [CatalogRegistrationComplete] (Provision)</li> </ul>
MachineProvisioned <ul style="list-style-type: none"> <li>Pre(Provision)</li> <li>Post(Provision)</li> </ul>	
Off <ul style="list-style-type: none"> <li>Pre(Active)</li> <li>Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>OnForceOn [ForceOn] (Active)</li> <li>OnResetOff [Turn Off] (Active)</li> <li>OnTurnOn [Turn On] (Active)</li> </ul>
On <ul style="list-style-type: none"> <li>Pre(Active)</li> <li>Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>OnForceOff [ForceOff] (Active)</li> <li>onInstallTools [InstallTools] (Active)</li> <li>OnReboot [Reboot] (Active)</li> <li>OnReset [Reset] (Active)</li> <li>OnResetOn [Turn On] (Active)</li> <li>OnShutdown [Shutdown] (Active)</li> <li>OnSuspend [Suspend] (Active)</li> <li>OnTurnOff [Turn Off] (Active)</li> </ul>
Rebooting <ul style="list-style-type: none"> <li>Pre(Active)</li> <li>Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>OnRebootComplete(Active)</li> <li>TimeoutFromReboot(Active)</li> </ul>
RegisterMachine <ul style="list-style-type: none"> <li>Pre(Provision)</li> <li>Post(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>onRegisterComplete(Provision)</li> <li>RegisterTimeout(Provision)</li> </ul>
Requested <ul style="list-style-type: none"> <li>Pre(Provision)</li> <li>Post(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnProvisionMachine [Provision] (Provision)</li> </ul>

표 1-15. VMPSMasterWorkflow32 상태 및 이벤트 (계속)

상태(항목)	이벤트 [트리거 문자열](항목)
Resetting <ul style="list-style-type: none"> <li>■ Pre(Active)</li> <li>■ Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>■ OnResetComplete(Active)</li> <li>■ TimeoutFromReset(Active)</li> </ul>
ShuttingDown <ul style="list-style-type: none"> <li>■ Pre(Active)</li> <li>■ Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>■ OnShutdownComplete(Active)</li> <li>■ TimeoutFromShutdown(Active)</li> </ul>
Suspending <ul style="list-style-type: none"> <li>■ Pre(Active)</li> <li>■ Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>■ OnSuspendComplete(Active)</li> <li>■ TimeoutFromSuspend(Active)</li> </ul>
TurningOff <ul style="list-style-type: none"> <li>■ Pre(Active)</li> <li>■ Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>■ OnTurningOffComplete(Active)</li> <li>■ TimeoutFromPowerOff(Active)</li> </ul>
TurningOn <ul style="list-style-type: none"> <li>■ Pre(Active)</li> <li>■ Post(Active)</li> </ul>	<ul style="list-style-type: none"> <li>■ OnTurningOnComplete(Active)</li> <li>■ TimeoutPowerOn(Active)</li> </ul>
UnprovisionMachine <ul style="list-style-type: none"> <li>■ Pre(Provision)</li> <li>■ Post(Provision)</li> </ul>	
WaitingToBuild <ul style="list-style-type: none"> <li>■ Pre(Provision)</li> <li>■ Post(Provision)</li> </ul>	

### VMPS 마스터 워크플로의 예

VMPS 워크플로는 다른 프로비저닝 워크플로가 포함되어 있는 마스터입니다. 이 예에는 가상 시스템의 수명 주기를 설명하는 기본 VM 워크플로가 포함되어 있습니다. 이것이 사용자 환경의 특정 워크플로를 나타내는 것은 아닙니다.



## 시스템 유형별 프로비저닝 수명 주기 상태

시스템 유형별 수명 주기 상태는 특정 가상 시스템 유형과 관련되어 있습니다. 마스터 워크플로 외에, 워크플로 구독에 대한 트리거 조건을 생성할 때 프로비저닝 워크플로 상태 및 이벤트를 사용할 수 있습니다.

## BladeLogic 베어메탈

상태(항목)	이벤트(항목)
BuildFinished	
<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>	
CreatingMachine	
<ul style="list-style-type: none"> <li>■ Pre(Provision)</li> </ul>	

## Opsware 베어메탈

상태(항목)	이벤트(항목)
BuildFinished	
■ Pre(Provision)	
OpswareRegister	■ OnOpswareRegister(Provision)
■ Pre(Provision)	

## 클라우드 프로비저닝 워크플로

상태(항목)	이벤트(항목)
BuildComplete	
■ Pre(Provision)	
CloudProvisioning	■ OnCloudProvisioningTimeout(Provision)
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	

## App 서비스 프로비저닝 워크플로

상태(항목)	이벤트(항목)
AppServiceProvisioning	■ OnAppServiceProvisioningTimeout(Provision)
■ Pre(Provision)	
BuildComplete	
■ Pre(Provision)	
FailedProvisioning	
■ Pre(Provision)	

## 기본 VM 워크플로

상태(항목)	이벤트(항목)
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	

## Opware 가상

상태(항목)	이벤트(항목)
AddingDisks ■ Pre(Provision)	■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision)
BuildFinished ■ Pre(Provision)	
CreatingVM ■ Pre(Provision)	■ OnCreateVMComplete(Provision) ■ OnCreateVMTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
OpwareRegister ■ Pre(Provision)	■ OnOpwareRegister(Provision)

## 클라우드 Linux Kickstart 워크플로

상태(항목)	이벤트(항목)
BuildComplete ■ Pre(Provision)	
CreatingMachine ■ Pre(Provision)	■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision)
CustomizeOS ■ Pre(Provision)	■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
InstallingOS ■ Pre(Provision)	■ OnInstallingOSComplete(Provision) ■ OnInstallingOSTimeout(Provision)

## 복제 워크플로

상태(항목)	이벤트(항목)
BuildComplete ■ Pre(Provision)	
CloneMachine ■ Pre(Provision)	■ OnCloneMachineComplete(Provision) ■ OnCloneMachineTimeout(Provision)
CustomizeMachine ■ Pre(Provision)	■ OnCustomizeMachineComplete(Provision) ■ OnCustomizeMachineTimeout(Provision)
CustomizeOS	■ OnCustomizeOS(Provision) ■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision)

상태(항목)	이벤트(항목)
EjectCD ■ Pre(Provision)	■ OnEjectCDComplete(Provision) ■ OnEjectCDTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
FinalizeProvisioning ■ Pre(Provision)	■ OnFinalizeComplete(Provision) ■ OnFinalizeTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
InstallSoftware ■ Pre(Provision)	■ OnInstallSoftwareComplete(Provision) ■ OnInstallSoftwareTimeout(Provision)
MountCD ■ Pre(Provision)	■ OnMountCDComplete(Provision) ■ OnMountCDTimeout(Provision)
PostInstallSoftwareChecks ■ Pre(Provision)	
PrepareInstallSoftware ■ Pre(Provision)	

## 클라우드 WIM 이미지 워크플로

상태(항목)	이벤트(항목)
BuildComplete ■ Pre(Provision)	
CreatingMachine ■ Pre(Provision)	■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
InstallIOS ■ Pre(Provision)	■ onInstallIOSComplete(Provision) ■ OnInstallIOSTimeout(Provision)
재부팅 ■ Pre(Provision)	■ OnRebootComplete(Provision) ■ OnRebootTimeout(Provision)
SetupOS ■ Pre(Provision)	■ OnSetupOSComplete(Provision) ■ OnSetupOSTimeout(Provision)

## 외부 프로비저닝 워크플로

상태(항목)	이벤트(항목)
AddingDisks ■ Pre(Provision)	■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision)
BuildComplete ■ Pre(Provision)	
CreatingMachine ■ Pre(Provision)	■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision)
EpiRegister ■ Pre(Provision)	■ OnEpiRegisterComplete(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)

## Linux Kickstart 워크플로

상태(항목)	이벤트(항목)
AddingDisks ■ Pre(Provision)	■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision)
BuildComplete ■ Pre(Provision)	
CreatingMachine ■ Pre(Provision)	■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision)
CustomizeOS ■ Pre(Provision)	■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision)
EjectingCD ■ Pre(Provision)	■ OnEjectingCDComplete(Provision) ■ OnEjectingCDTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision)
InstallingOS ■ Pre(Provision)	■ OnInstallingOSComplete(Provision) ■ OnInstallingOSTimeout(Provision)



## 물리적 프로비저닝 워크플로

상태(항목)	이벤트(항목)
FailedProvisioning ■ Pre(Provision)	
FinalizeProvisioning ■ Pre(Provision)	■ OnFinalizeProvisioningTimeout(Provision)
InitializeProvisioning ■ Pre(Provision)	■ OnInitializeProvisioningTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallIOS ■ Pre(Provision)	■ OnInstallIOSComplete(Provision) ■ OnInstallIOSTimeout(Provision)
재부팅 ■ Pre(Provision)	■ OnRebootComplete(Provision) ■ OnRebootTimeout(Provision)
SetupOS ■ Pre(Provision)	■ OnSetupOSComplete(Provision) ■ OnSetupOSTimeout(Provision)

## 물리적 PXE 프로비저닝 워크플로

상태(항목)	이벤트(항목)
CheckHardwareType ■ Pre(Provision)	
CleanPxe ■ Pre(Provision)	■ OnCleanPxeTimeout(Provision)
FailedProvisioning ■ Pre(Provision)	
FinalizeProvisioning ■ Pre(Provision)	■ OnFinalizeProvisioningTimeout(Provision)
InitializeProvisioning ■ Pre(Provision)	■ OnInitializeProvisioningTimeout(Provision)
InitialPowerOn ■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallIOS ■ Pre(Provision)	■ OnInstallIOSComplete(Provision) ■ OnInstallIOSTimeout(Provision)
재부팅 ■ Pre(Provision)	■ OnRebootComplete(Provision) ■ OnRebootTimeout(Provision)
SetupOS ■ Pre(Provision)	■ OnSetupOSComplete(Provision) ■ OnSetupOSTimeout(Provision)
SetupPxe ■ Pre(Provision)	■ OnSetupPxeTimeout(Provision)

## 물리적 SCCM 프로비저닝 워크플로

상태(항목)	이벤트(항목)
CheckHardwareType ■ Pre(Provision)	
완료 ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnCompleteProvisioningComplete(Provision)</li> <li>■ OnCompleteProvisioningTimeout(Provision)</li> </ul>
FailedProvisioning ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnFailedProvisioningTimeout(Provision)</li> </ul>
FinalizeProvisioning ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnFinalizeProvisioningTimeout(Provision)</li> </ul>
InitializeProvisioning ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnInitializeProvisioningTimeout(Provision)</li> </ul>
InitialPowerOn ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnInitialPowerOnTimeout(Provision)</li> </ul>
SccmRegistration ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnSccmRegistrationTimeout(Provision)</li> </ul>

## 물리적 SCCM PXE 프로비저닝 워크플로

상태(항목)	이벤트(항목)
CheckHardwareType ■ Pre(Provision)	
CleanPxe ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnCleanPxeTimeout(Provision)</li> </ul>
완료 ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnCompleteProvisioningComplete(Provision)</li> <li>■ OnCompleteProvisioningTimeout(Provision)</li> </ul>
삭제 ■ Pre(Provision)	
FailedProvisioning ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnFailedProvisioningTimeout(Provision)</li> </ul>
FinalizeProvisioning ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnFinalizeProvisioningTimeout(Provision)</li> </ul>
InitializeProvisioning ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnInitializeProvisioningTimeout(Provision)</li> </ul>
InitialPowerOn ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnInitialPowerOnTimeout(Provision)</li> </ul>
SccmRegistration ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnSccmRegistrationTimeout(Provision)</li> </ul>
SetupPxe ■ Pre(Provision)	<ul style="list-style-type: none"> <li>■ OnSetupPxeTimeout(Provision)</li> </ul>

## vApp 복제 워크플로

상태(항목)	이벤트 [트리거 문자열](항목)
글로벌	<ul style="list-style-type: none"> <li>OnFailProvisioning [FailProvisioning] (Provision)</li> <li>OnMasterProvisioned [MasterProvisioned] (Provision)</li> </ul>
BuildComplete <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	
CloneMachine <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnCloneMachineComplete(Provision)</li> <li>OnCloneMachineTimeout(Provision)</li> </ul>
CustomizeMachine <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnCustomizeMachineComplete(Provision)</li> <li>OnCustomizeMachineTimeout(Provision)</li> </ul>
CustomizeOS	<ul style="list-style-type: none"> <li>OnCustomizeOS(Provision)</li> <li>OnCustomizeOSComplete(Provision)</li> <li>OnCustomizeOSTimeout(Provision)</li> </ul>
FailedProvisioning <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	
FinalizeProvisioning <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnFinalizeComplete(Provision)</li> <li>OnFinalizeTimeout(Provision)</li> </ul>
InitialPowerOn <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnInitialPowerOnComplete(Provision)</li> <li>OnInitialPowerOnTimeout(Provision)</li> </ul>
WaitingForMaster <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnWaitingForMasterTimeout(Provision)</li> </ul>

## 가상 SCCM 프로비저닝 워크플로

상태(항목)	이벤트(항목)
AddingDisks <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnAddingDisksComplete(Provision)</li> <li>OnAddingDisksTimeout(Provision)</li> </ul>
BuildComplete <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	
CreatingMachine <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>CreatingMachineComplete(Provision)</li> <li>OnCreatingMachineTimeout(Provision)</li> </ul>
삭제 <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	
EjectingCD <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnEjectingCDComplete(Provision)</li> <li>OnEjectingCDTimeout(Provision)</li> </ul>
FailedProvisioning <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	
InitialPowerOn <ul style="list-style-type: none"> <li>Pre(Provision)</li> </ul>	<ul style="list-style-type: none"> <li>OnInitialPowerOnComplete(Provision)</li> <li>OnPowerOnTimeout(Provision)</li> </ul>

상태(항목)	이벤트(항목)
InstallingOS	■ OnInstallingOSComplete(Provision)
■ Pre(Provision)	■ OnInstallingOSTimeout(Provision)
SccmRegistration	■ OnSccmRegistrationTimeout(Provision)
■ Pre(Provision)	

## WIM 이미지 워크플로

상태(항목)	이벤트(항목)
AddingDisks	■ OnAddingDisksComplete(Provision)
■ Pre(Provision)	■ OnAddingDisksTimeout(Provision)
BuildComplete	
■ Pre(Provision)	
CreatingMachine	■ OnCreatingMachineComplete(Provision)
■ Pre(Provision)	■ OnCreatingMachineTimeout(Provision)
EjectingCD	■ OnEjectingCDComplete(Provision)
■ Pre(Provision)	■ OnEjectingCDTimeout(Provision)
FailedProvisioning	
■ Pre(Provision)	
InitialPowerOn	■ OnInitialPowerOnComplete(Provision)
■ Pre(Provision)	■ OnInitialPowerOnTimeout(Provision)
InstallIOS	■ onInstallIOSComplete(Provision)
■ Pre(Provision)	■ OnInstallIOSTimeout(Provision)
재부팅	■ OnRebootComplete(Provision)
■ Pre(Provision)	■ OnRebootTimeout(Provision)
SetupOS	■ OnSetupOSComplete(Provision)
■ Pre(Provision)	■ OnSetupOSTimeout(Provision)

## 상태 및 이벤트에 대한 시간 제한 값 구성

모든 상태 및 이벤트에 대한 기본 시간 제한 값은 30분으로, vRealize Automation 글로벌 설정에 구성되어 있습니다. 일부 워크플로를 성공적으로 실행하는 데에는 시간이 더 걸릴 수 있습니다. 환경에 서로 다른 워크플로를 수용하기 위해 개별 워크플로 또는 상태에 대해 시간 제한 재정의 값을 추가할 수 있습니다.

기본 시간 제한 값을 수정하려면 **인프라 > 관리 > 글로벌 설정**을 선택하고 **확장성 수명 주기 메시지 시간 제한** 값을 편집합니다. 글로벌 설정을 변경하면 Manager Service를 다시 시작해야 합니다.

개별 시간 제한 값을 구성하려면 IaaS 서버에 있는 ManagerService.exe.config 파일의 appSetting 섹션에 워크플로 또는 이벤트 속성을 추가합니다. 이 파일은 일반적으로 %System-Drive%\Program Files x86\VMware\WvCACWServer 디렉토리에 있습니다. 파일을 편집하기 전에 항상 파일의 복사본을 만들어야 합니다. 개별 설정을 변경하면 Manager Service를 다시 시작해야 합니다.

키의 기본 형식은 다음 예와 유사합니다.

- 워크플로의 경우. `Extensibility.{workflow}.Timeout`
- 이벤트의 경우. `Extensibility.{workflow}.{state}.EVENT.{event}.Timeout`
- 상태의 경우. `Extensibility.{workflow}.{state}.(PRE/POST).Timeout`

appSetting 섹션에 키를 추가할 때 다음을 예로 사용하십시오. 시간 제한 값 형식은 D.HH:mm:ss.ms입니다. D는 일이고 ms는 밀리초입니다. 일과 밀리초는 선택 사항입니다. 시간, 분 및 초는 필수입니다.

- 전체 BasicVmWorkflow 워크플로에 대한 시간 제한을 30분으로 설정하려면 `<add key="Extensibility.BasicVmWorkflow.Timeout" value="00:30:00"/>`을 추가합니다.
- VMPSMasterWorkflow32의 OnFinalizeMachine 글로벌 이벤트에 대한 시간 제한을 2시간으로 설정하려면 `<add key="Extensibility.VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnFinalizeMachine.Timeout" value="02:00:00"/>`을 추가합니다.
- VMPSMasterWorkflow32의 사전 요청 상태에 대한 시간 제한을 2일로 설정하려면 `<add key="Extensibility.VMPSMasterWorkflow32.Requested.PRE.Timeout" value="2.00:00:00"/>`을 추가합니다.

## 상태 및 이벤트에 대한 오류 동작 구성

워크플로 구독 시간 제한 및 오류 처리에는 기본 동작이 있습니다. 사용자 환경의 시스템에 대한 동작을 사용자 지정할 수 있습니다.

IaaS는 이벤트 브로커 서비스의 이벤트 시간 제한 및 오류를 처리합니다.

각 상태 전환 시에 SendEBSMessage가 이벤트 브로커 서비스에 이벤트를 전송하고 회신을 기다립니다. 기본적으로, 이벤트 브로커 서비스에서 시간 제한 또는 오류를 보고하면 해당 내용이 기록되고 워크플로가 재개됩니다.

마스터 워크플로의 다음 상태에서 시간 제한 또는 오류가 발생하면 워크플로가 재개되는 대신 워크플로가 강제로 오류 상태로 설정됩니다.

**표 1-16. 워크플로가 재개되지 않는 예외 상황**

오류가 발생하는 상태	오류 상태
PRE MachineProvisioned	UnprovisionMachine
PRE BuildingMachine	Disposing
PRE RegisterMachine	Finalized

시간 제한 또는 오류 동작을 사용자 지정하려는 경우, 이벤트를 트리거하거나 상태 변경을 강제하려는 모든 이벤트 또는 상태에 대해 사용자 지정 속성을 시스템에 추가할 수 있습니다. 사용자 지정 속성을 구성하려면 다음 예를 사용하십시오.

- `Extensibility.Lifecycle.Error.Event.{Workflow}.{State}`. 속성 값은 시간 제한 또는 오류 발생 시 워크플로에서 트리거될 이벤트의 이름입니다.

- Extensibility.Lifecycle.Error.State.{Workflow}.State}. 속성 값은 시간 제한 또는 오류 발생 시 워크플로에서 전환을 강제할 상태의 이름입니다.

## 시나리오: 가상 시스템의 사후 프로비저닝 스냅샷 생성

테넌트 관리자로서, 서비스 카탈로그 사용자가 가상 시스템의 사후 프로비저닝 스냅샷을 생성하여 새 시스템을 요청하는 대신 새로 설치된 시스템으로 되돌리도록 지정합니다.

### 프로시저

#### 1 시나리오: 사후 프로비저닝 스냅샷 작업에 대한 vRealize Orchestrator 워크플로 생성

필수 입력 매개 변수를 수락하는 vRealize Orchestrator 워크플로를 생성합니다. 사후 프로비저닝 목표를 달성하도록 워크플로를 설계합니다.

#### 2 시나리오: 사후 프로비저닝 스냅샷 워크플로 구독 생성

테넌트 관리자로서, 가상 시스템 생성 후 각 가상 시스템의 스냅샷을 생성할 수 있습니다. 시스템 프로비저닝 이벤트 항목을 기반으로 워크플로 구독을 구성하고 게시를 통해 활성화합니다.

### 시나리오: 사후 프로비저닝 스냅샷 작업에 대한 vRealize Orchestrator 워크플로 생성

필수 입력 매개 변수를 수락하는 vRealize Orchestrator 워크플로를 생성합니다. 사후 프로비저닝 목표를 달성하도록 워크플로를 설계합니다.

vRealize Orchestrator 폴더 및 워크플로 생성에 대한 자세한 내용은 VMware vRealize Orchestrator를 사용한 개발을 참조하십시오.

### 필수 조건

vRealize Automation에 대해 구성된 인스턴스인 vRealize Orchestrator에 워크플로를 생성할 수 있도록 허용하는 권한과 함께 로그인합니다.

### 프로시저

#### 1 워크플로 라이브러리의 워크플로 구독 워크플로를 위한 폴더를 생성합니다.

#### 2 새 워크플로를 생성합니다.

이 시나리오의 경우 워크플로 이름을 Automation Post-Provisioning Snapshot으로 지정합니다.

#### 3 다음 입력 매개 변수를 추가합니다.

이름	유형
payload	Properties

#### 4 입력 매개 변수를 수락하고 가상 시스템 스냅샷을 생성하는 스크립팅 가능한 작업을 추가합니다.

#### 5 워크플로를 저장합니다.

### 후속 작업

Automation Post-Provisioning Snapshot 워크플로를 실행하는 워크플로 구독을 생성합니다.

[시나리오: 사후 프로비저닝 스냅샷 워크플로 구독 생성](#)

## 시나리오: 사후 프로비저닝 스냅샷 워크플로 구독 생성

테넌트 관리자로서, 가상 시스템 생성 후 각 가상 시스템의 스냅샷을 생성할 수 있습니다. 시스템 프로비저닝 이벤트 항목을 기반으로 워크플로 구독을 구성하고 계시를 통해 활성화합니다.

가상 시스템이 프로비저닝되고 감지된 이벤트 메시지가 활성화된 상태일 때 스냅샷 생성 워크플로를 실행하도록 워크플로 구독을 구성합니다.

### 필수 조건

- 테넌트 관리자 vRealize Automation 콘솔에 로그인합니다.
- vCenter Server 플러그인을 vRealize Orchestrator 끝점으로 구성합니다. vRealize Automation 구성 항목을 참조하십시오.
- vSphere 가상 시스템 Blueprint가 있는지 확인합니다.
- 가상 시스템의 스냅샷을 생성하는 vRealize Orchestrator 워크플로가 있는지 확인합니다. vRealize Automation 플러그인에서 제공하는 스냅샷 생성 워크플로는 사용할 수 없습니다. 제공되는 스냅샷 워크플로는 XaaS 통합에만 적용됩니다. [프로비저닝 및 수명 주기 워크플로에 대해 vRealize Orchestrator 워크플로 구성](#) 항목을 참조하십시오.

### 프로시저

- 1 **관리 > 이벤트 > 구독**을 선택합니다.
- 2 **새로 만들기** 아이콘(+)을 클릭합니다.
- 3 **시스템 프로비저닝**을 선택합니다.
- 4 **다음**을 클릭합니다.
- 5 [조건] 탭에서 트리거 조건을 구성합니다.
  - a **조건에 따라 실행**을 선택합니다.
  - b **절** 드롭다운 메뉴에서 **다음 중 모두**를 선택합니다.
  - c 다음 조건을 구성합니다.

속성	연산자	값
데이터 > 시스템 > 시스템 유형	같음	상수 > 가상 시스템
데이터 > 수명 주기 상태 > 수명 주기 상태 이름	같음	상수 > VMPSMasterWorkflow32.MachineActivated
데이터 > 수명 주기 상태 > 상태 단계	같음	상수 > POST

- d **다음**을 클릭합니다.
- 6 [워크플로] 탭에서 Orchestrator 트리를 찾아보고 **자동화 사후 프로비저닝 스냅샷** 워크플로를 선택합니다.
- 7 **다음**을 클릭합니다.

## 8 [세부 정보] 탭에서 이름과 설명을 입력합니다.

이 시나리오에서는 이름으로 **Post-Provisioning Virtual Machine Snapshot**을 입력하고 설명으로 **Create a snapshot when new virtual machine is provisioned and activated**를 입력합니다.

## 9 마침을 클릭합니다.

## 10 Post-Provisioning Virtual Machine Snapshot 행을 선택하고 게시를 클릭합니다.

워크플로 구독이 활성화되었으며 이벤트 메시지에서 요청된 가상 시스템이 프로비저닝 및 활성화되었음을 표시하면 스냅샷 워크플로가 트리거됩니다.

### 후속 작업

워크플로 구독을 테스트하려면 서비스 카탈로그에서 가상 시스템을 요청합니다. 요청이 프로비저닝 성공을 표시하면 스냅샷이 생성되었는지 확인합니다.

## 승인 워크플로 구독 사용

처리를 위해 승인 요청을 외부 시스템에 보낼 수 있도록 사전 승인 및 사후 승인 워크플로 구독을 생성합니다. 그러면 vRealize Automation에서 승인 또는 거부된 응답을 처리합니다.

### 승인 서비스 통합

vRealize Automation 외부의 시스템에서 승인 요청을 처리하는 사용자 지정 vRealize Orchestrator 워크플로를 실행하는 사전 승인 또는 사후 승인 워크플로 구독을 생성합니다.

승인 정책 승인 수준에서, 승인자로 **이벤트 구독 사용**을 선택할 수 있습니다. 이 수준은 승인 정책에서 몇 가지 중 하나가 될 수 있습니다. 서비스 카탈로그 사용자가 **이벤트 구독 사용** 승인자를 포함하고 승인 정책이 적용되는 항목을 요청할 때 승인 서비스에서는 다음 결과와 함께 이벤트 브로커 서비스에 메시지를 보냅니다.

- 일치 조건을 가진 워크플로 구독을 게시한 경우 vRealize Orchestrator에서 승인 또는 거부 워크플로를 실행합니다.
- 워크플로 구독을 게시했지만 조건이 일치하지 않거나, 워크플로 구독을 취소했거나, 게시된 구독이 없는 경우 승인 수준이 승인되고 승인 프로세스가 다음 승인 수준으로 이동합니다.

승인 워크플로 구독이 승인 서비스로부터 메시지를 수신하고 이 메시지를 승인 구독에 대해 구성된 조건과 비교합니다. 조건과 일치하는 메시지를 찾은 경우 선택된 vRealize Orchestrator 워크플로가 실행을 시작합니다. 게시된 이벤트 데이터가 입력으로 워크플로에 전달되고 워크플로에 지정된 방법으로 처리됩니다. 워크플로의 결과가 vRealize Automation으로 반환되고 요청이 처리됩니다. 승인된 경우 다음 승인 수준이 평가됩니다. 거부된 경우 요청이 거부됩니다. 승인 서비스가 승인 서비스의 기본 시간 제한인 24시간 내에 회신을 수신하지 못하면 요청이 거부됨으로 처리됩니다.

## 승인 이벤트 항목에 대한 vRealize Orchestrator 워크플로 구성

승인 메시지를 지원하고 vRealize Automation에서 처리할 수 있는 올바른 형식의 정보로 회신하려면 사용자 지정 vRealize Orchestrator 워크플로를 구성해야 합니다.



## 승인 이벤트 항목 스키마

사전 승인 및 사후 승인 이벤트 메시지 스키마에는 필드 이름과 값, 요청에 포함된 정보 그리고 요청의 소스에 대한 정보가 포함됩니다.

다음은 이벤트 데이터 페이로드의 구조입니다.

```
{
  fieldNames : Properties,          // Property names

  fieldValues : Properties,         // Property values

  // Information about the request
  requestInfo : {
    requestRef : STRING,            // Identifier for the source request
    itemName : STRING,              // Name of the requested item
    itemDescription : STRING,        // Description of the requested item
    reason : STRING,                // Justification provided by the user specifying why the request is
    required
      description : STRING,          // Description entered by the user specifying the purpose of the request
      approvalLevel:ExternalReference, // Approval level ID. This is a searchable field
      approvalLevelName : STRING,    // Approval level name
      createDate : DATE_TIME,        // Time the approval request is created
      requestedFor : STRING,          // Principal id of the user for whom the source request is initiated
      subtenantId : STRING,           // Business group id
      requestedBy : STRING            // Principal id of the user who actually submits the request
    },

    // Information about the source of the request
    sourceInfo : {
      externalInstanceId : STRING,    // Identifier of the source object, as defined by the initiator service
      serviceId : STRING,             // Identifier of the service which initiated the approval
      externalClassId : STRING        // Identifier of the class to which the source object belongs
    }
  }
}
```

속성 이름과 속성 값은 승인 정책 수준에서 구성하는 사용자 지정 속성 또는 시스템 속성이 될 수 있습니다. 이러한 속성은 승인 정책에서 구성되어 승인자가 승인 프로세스 중에 값을 변경할 수 있도록 허용합니다. 예를 들어 CPU가 포함되는 경우 승인자는 승인 요청 양식에서 CPU의 수를 줄일 수 있습니다.

회신 이벤트 데이터 페이로드는 워크플로에 의해 vRealize Automation으로 반환되는 정보입니다. 회신 페이로드의 콘텐츠는 요청을 승인할지 아니면 거부할지 결정합니다.

```
{
  approved : BOOLEAN,

  // Property values
  fieldValues : Properties
}
```

회신 이벤트 페이로드에서 승인된 매개 변수는 승인된 요청의 경우 True, 거부된 요청의 경우 False 입니다. 속성 값은 vRealize Orchestrator 워크플로에 의해 수정되고, vRealize Automation으로 반환되고, 승인 프로세스에 포함된 사용자 지정 또는 시스템 속성입니다.

businessJustification에 대한 하나의 출력 매개 변수와 함께 vRealize Orchestrator 워크플로를 구성해야 합니다. 이 매개 변수를 사용하여 외부 시스템의 승인자가 제공한 설명을 vRealize Automation 승인 프로세스에 전달할 수 있습니다. 이러한 설명은 승인 또는 거부에 대한 것일 수 있습니다.

### 승인 스키마를 기반으로 vRealize Orchestrator 워크플로 생성

생성하는 사용자 지정 승인 워크플로에는 유용한 이름을 가진, Properties 유형으로 구성된 입력 매개 변수가 있어야 합니다. 승인 이벤트 데이터 페이로드는 워크플로 구독이 실행되도록 트리거될 때 이 매개 변수에 위치합니다.

다시 vRealize Automation으로 전송되는 워크플로의 출력 매개 변수는 approved : Boolean 및 fieldValues : Properties입니다. 반환된 approved : Boolean 매개 변수는 승인 수준을 승인할지 아니면 거부할지 결정합니다. fieldValues : Properties 매개 변수에는 외부 시스템에서 수정된 값이 포함되어 있습니다.

### 시나리오: 승인을 위해 외부 시스템에 소프트웨어 요청 전송

테넌트 관리자로서, 서비스 카탈로그 사용자가 소프트웨어가 포함된 시스템을 요청할 때 vRealize Automation 외부의 사용자가 소프트웨어 구성 요소를 승인하도록 구성합니다. 정의된 조건과 일치하는 사전 승인 메시지를 수신할 때 실행되도록 구성된 모든 소프트웨어 프로비저닝 및 워크플로 구독에 대해 승인을 필요로 하는 승인 정책을 구성합니다.

#### 프로시저

##### 1 시나리오: 승인 워크플로 구독에 대한 vRealize Orchestrator 워크플로 생성

vRealize Automation의 필수 승인 입력 매개 변수를 수락하고 필요한 출력 매개 변수를 반환하여 승인 프로세스를 완료하는 vRealize Orchestrator 워크플로를 생성합니다.

##### 2 시나리오: 외부 승인을 위한 승인 정책 생성

테넌트 관리자로서, 승인 서비스에 의해 게시되는 이벤트 메시지를 생성하는 승인 정책을 생성합니다. 이벤트 메시지와 일치하는 조건을 가진 워크플로 구독을 구성한 경우 vRealize Orchestrator가 선택된 워크플로를 실행합니다.

##### 3 시나리오: 사전 승인 워크플로 구독 생성

테넌트 관리자로서, 서비스 카탈로그 요청에서 구성된 조건과 일치하는 승인 요청을 생성할 때 vRealize Orchestrator 워크플로를 실행하는 사전 승인 워크플로 구독을 생성합니다.

### 시나리오: 승인 워크플로 구독에 대한 vRealize Orchestrator 워크플로 생성

vRealize Automation의 필수 승인 입력 매개 변수를 수락하고 필요한 출력 매개 변수를 반환하여 승인 프로세스를 완료하는 vRealize Orchestrator 워크플로를 생성합니다.

승인 목표를 달성하도록 워크플로를 설계해야 합니다. vRealize Orchestrator 폴더 및 워크플로 생성에 대한 자세한 내용은 VMware vRealize Orchestrator를 사용한 개발을 참조하십시오.

**필수 조건**

vRealize Automation에 대해 구성된 인스턴스인 vRealize Orchestrator에 워크플로를 생성할 수 있도록 허용하는 권한과 함께 로그인합니다.

**프로시저**

- 1 워크플로 라이브러리의 워크플로 구독 워크플로를 위한 폴더를 생성합니다.
- 2 새 워크플로를 생성합니다.

이 시나리오의 경우 워크플로 이름을 **Automation Approval Request**로 지정합니다.

- a 다음 입력 매개 변수를 추가합니다.

이름	유형
input	Properties

- b 다음 출력 매개 변수를 추가합니다.

이름	유형
approved	boolean
fieldValues	Properties

- 3 입력 및 출력 매개 변수를 처리하는 스크립팅 가능한 작업을 생성합니다.
- 4 워크플로를 저장합니다.

**후속 작업**

승인자로서 워크플로 구독을 사용하는 승인 정책을 생성합니다. [시나리오: 외부 승인을 위한 승인 정책 생성](#)

**시나리오: 외부 승인을 위한 승인 정책 생성**

테넌트 관리자로서, 승인 서비스에 의해 게시되는 이벤트 메시지를 생성하는 승인 정책을 생성합니다. 이벤트 메시지와 일치하는 조건을 가진 워크플로 구독을 구성한 경우 vRealize Orchestrator가 선택된 워크플로를 실행합니다.

**필수 조건**

- 테넌트 관리자 또는 승인 관리자로 vRealize Automation 콘솔에 로그인합니다.

**프로시저**

- 1 **관리 > 승인 정책**을 선택합니다.
- 2 소프트웨어 구성 요소를 위한 승인 정책을 생성합니다.
  - a **새로 만들기** 아이콘(+ )을 클릭합니다.
  - b **승인 정책 유형 선택**을 선택합니다.
  - c 목록에서 **서비스 카탈로그 - 카탈로그 항목 요청 - 소프트웨어 구성 요소**를 선택합니다.

- d **확인**을 클릭합니다.
- e 다음 옵션을 구성합니다.

옵션	구성
이름	Software external approval을 입력합니다.
설명	Approval request sent to external approval system을 입력합니다.
상태	Active를 선택합니다.

- 3 **사전 승인** 탭에서 **추가** 아이콘(+)을 클릭합니다.
- 4 트리거 조건 및 승인 작업을 사용하여 **수준 정보** 탭을 구성합니다.
  - a **이름** 텍스트 상자에 External level for software를 입력합니다.
  - b **설명** 텍스트 상자에 Software approval request sent to external approval system을 입력합니다.
  - c **항상 필요**를 선택합니다.
  - d **이벤트 구독 사용**을 선택합니다.
- 5 **확인**을 클릭합니다.

#### 후속 작업

- 구성된 승인 수준을 기반으로 이벤트 메시지를 수신하는 사전 승인 워크플로 구독을 생성합니다.  
[시나리오: 사전 승인 워크플로 구독 생성](#) 항목을 참조하십시오.
- 사용 권한의 소프트웨어 구성 요소에 승인 정책을 적용합니다. vRealize Automation 구성의 내용을 참조하십시오.

#### 시나리오: 사전 승인 워크플로 구독 생성

테넌트 관리자로서, 서비스 카탈로그 요청에서 구성된 조건과 일치하는 승인 요청을 생성할 때 vRealize Orchestrator 워크플로를 실행하는 사전 승인 워크플로 구독을 생성합니다.

#### 필수 조건

- **테넌트 관리자**로 vRealize Automation 콘솔에 로그인합니다.
- '소프트웨어에 대한 외부 수준'이라는 이름의 승인 정책 수준을 구성합니다. [시나리오: 외부 승인을 위한 승인 정책 생성](#) 항목을 참조하십시오.
- 외부 시스템으로 요청을 전송하는 사용자 지정 vRealize Orchestrator 워크플로를 생성합니다. 이 시나리오에서는 자동 승인 요청 워크플로를 사용합니다.

#### 프로시저

- 1 **관리 > 이벤트 > 구독**을 선택합니다.
- 2 **새로 만들기** 아이콘(+)을 클릭합니다.
- 3 **사전 승인**을 클릭합니다.

4 다음을 클릭합니다.

5 조건 탭에서 트리거 조건을 구성합니다.

- a 조건에 따라 실행을 선택합니다.
- b 질 드롭다운 메뉴에서 다음 조건을 구성합니다.

속성	연산자	값
데이터 > 요청에 대한 정보 > 승인 수준 이름	같음	소프트웨어에 대한 외부 수준

c 다음을 클릭합니다.

6 [워크플로] 탭에서 Orchestrator 트리를 찾아보고 **자동화 승인 요청** 워크플로를 선택합니다.

7 다음을 클릭합니다.

8 [세부 정보] 탭에서 이름과 설명을 입력합니다.

이 시나리오에서는 이름으로 **Software External**을 입력하고 설명으로 **Sends approval request to external system**을 입력합니다.

9 시간 제한(분) 텍스트 상자에 120을 입력합니다.

지정하는 구독 워크플로 시간 제한 시간은 대상 시스템에 따라 다릅니다. vRealize Automation에서 지정된 시간 내에 대상 시스템의 회신을 처리하지 않으면 요청이 자동으로 거부됩니다.

값을 지정하지 않는 경우 기본 시간 제한은 24시간입니다.

10 마침을 클릭합니다.

11 Software External 행을 선택하고 **게시**를 클릭합니다.

Software External 사전 승인 이벤트 구독이 이제 사전 승인 이벤트 메시지를 수신합니다.

#### 후속 작업

- 활성 사용 권한의 소프트웨어 구성 요소에 승인 정책을 적용한 경우 서비스 카탈로그의 항목을 요청하고 승인 정책과 워크플로 구독이 설계한 대로 작동하는지 확인합니다.

## 워크플로 구독 문제 해결

워크플로 구독 문제 해결에는 일부 일반 문제가 포함됩니다. 다양한 로그를 검토해야 할 수 있습니다.

#### ■ 시작되지 않는 vRealize Orchestrator 워크플로 문제 해결

이벤트 메시지가 수신되면 사용자 지정 워크플로를 실행하도록 워크플로 구독을 구성하지만 워크플로가 실행되지 않습니다.

#### ■ 시간이 너무 많이 걸리는 프로비저닝 요청 문제 해결

IaaS 시스템이 프로비저닝에 10시간 이상 걸립니다.

#### ■ 승인 요청에 대해 실행되지 않는 vRealize Orchestrator 워크플로 문제 해결

vRealize Orchestrator 워크플로를 실행하도록 사전 승인 또는 사후 승인 워크플로 구독을 구성했습니다. 서비스 카탈로그에서 정의된 조건과 일치하는 시스템이 요청될 때 워크플로가 실행되지 않습니다.

#### ■ 승인되어야 하는 거부된 승인 요청 문제 해결

지정된 vRealize Orchestrator 워크플로를 실행하는 사전 승인 또는 사후 승인 워크플로 구독을 구성하지만 승인되어야 하는 요청이 거부됩니다.

#### ■ 거부된 승인 요청 문제 해결

지정된 vRealize Orchestrator 워크플로를 실행하는 사전 승인 또는 사후 승인 워크플로 구독을 구성하지만 요청이 예기치 않게 거부됩니다.

### 시작되지 않는 vRealize Orchestrator 워크플로 문제 해결

이벤트 메시지가 수신되면 사용자 지정 워크플로를 실행하도록 워크플로 구독을 구성하지만 워크플로가 실행되지 않습니다.

#### 해결 방법

- 1 워크플로 구독을 게시했는지 확인합니다.
- 2 워크플로 구독 조건이 올바르게 구성되었는지 확인합니다.
- 3 vRealize Orchestrator 서버에 지정된 워크플로가 있는지 확인합니다.

### 시간이 너무 많이 걸리는 프로비저닝 요청 문제 해결

IaaS 시스템이 프로비저닝에 10시간 이상 걸립니다.

#### 원인

프로비저닝 상태에서 트리거할 워크플로 구독을 구성한 경우 환경에서 두 개의 Manager Service 인스턴스가 실행 중일 수 있습니다.

#### 해결 방법

- ◆ 하나의 Manager Service 인스턴스만 활성화 상태인지 확인합니다. 두 개 모두 활성화인 경우 로그에서 이와 같은 오류를 확인할 수도 있습니다.

두 개 이상 활성화인 경우 다음 오류 메시지와 유사한 오류가 Manager Service 로그에 나타납니다.

```
[EventBrokerService] Failed resuming workflow b6e9276a-f20f-40f1-99ad-6d9524560cc2 on queue 3679fa71-ac2a-42d5-8626-f98ea096f0d3. System.Workflow.Runtime.QueueException: Event Queue operation failed with MessageQueueErrorCode QueueNotFound for queue '3679fa71-ac2a-42d5-8626-f98ea096f0d3'. at System.Workflow.Runtime.WorkflowQueueingService.EnqueueEvent(IComparable queueName, Object item) at System.Workflow.Runtime.WorkflowExecutor.EnqueueItem(IComparable queueName, Object item, IPendingWork pendingWork, Object workItem) at System.Workflow.Runtime.WorkflowInstance.EnqueueItem(IComparable queueName, Object item, IPendingWork pendingWork, Object workItem) at DynamicOps.VMPS.Service.Workflow.Services.EventBrokerService.OnMessage(EventObject obj) [UTC:2015-11-14 07:14:25 Local:2015-11-13 23:14:25] [Error]: Thread-Id="15" - context="HKBsp6Tt" token="JeuTG7ru" [EventBrokerClient] Invoking subscription callback failed: Event Queue operation failed with MessageQueueErrorCode QueueNotFound for queue '3679fa71-ac2a-42d5-8626-f98ea096f0d3'.
```

## 승인 요청에 대해 실행되지 않는 vRealize Orchestrator 워크플로 문제 해결

vRealize Orchestrator 워크플로를 실행하도록 사전 승인 또는 사후 승인 워크플로 구독을 구성했습니다. 서비스 카탈로그에서 정의된 조건과 일치하는 시스템이 요청될 때 워크플로가 실행되지 않습니다.

### 원인

승인에 대한 워크플로 구독을 성공적으로 실행하려면 모든 구성 요소가 올바르게 구성되어 있는지 확인해야 합니다.

### 해결 방법

- 1 승인 정책이 활성화고 정책에서 승인 수준에 대해 **이벤트 구독 사용**을 선택했는지 확인합니다.
- 2 사용 권한에 승인 정책이 올바르게 적용되었는지 확인합니다.
- 3 워크플로 구독이 올바르게 구성 및 게시되었는지 확인합니다.
- 4 이벤트 로그에서 승인 관련 메시지를 검토합니다.

## 승인되어야 하는 거부된 승인 요청 문제 해결

지정된 vRealize Orchestrator 워크플로를 실행하는 사전 승인 또는 사후 승인 워크플로 구독을 구성하지만 승인되어야 하는 요청이 거부됩니다.

### 해결 방법

- 1 vRealize Orchestrator에서 워크플로를 검토합니다.
  - a vRealize Orchestrator에 관리자 권한으로 로그인합니다.
  - b 워크플로가 오류 없이 실행되었는지 확인합니다.
  - c approval 및 fieldValues 매개 변수에 대해 예상 값이 반환되었는지 확인합니다.
- 2 vRealize Automation에서 요청을 검토합니다.
  - a vRealize Automation에 거부된 항목을 요청한 사용자로 로그인합니다.
  - b **요청** 탭을 클릭합니다.
  - c 거부된 요청을 엽니다.
  - d **승인 상태**를 클릭하고 [근거] 열에서 자세한 내용을 검토합니다.  
오류가 발생한 경우 오류에 대한 정보가 근거 데이터로 표시됩니다.

## 거부된 승인 요청 문제 해결

지정된 vRealize Orchestrator 워크플로를 실행하는 사전 승인 또는 사후 승인 워크플로 구독을 구성하지만 요청이 예기치 않게 거부됩니다.

### 문제점

이 외부 승인 수준 이전의 모든 승인 수준이 승인되었으며 이 수준이 승인되었어야 하지만 거부됨으로 처리되었습니다.

## 원인

하나의 가능한 원인은 vRealize Orchestrator가 워크플로를 실행하려고 했을 때의 내부 오류입니다. 예를 들어 워크플로가 누락되었거나 vRealize Orchestrator 서버가 실행 중이지 않습니다.

## 해결 방법

- 1 관리 > 이벤트 > 이벤트 로그를 선택합니다.
- 2 로그에서 승인 관련 메시지를 검토합니다.

## vRealize Automation Designer 를 사용하여 시스템 수명 주기 확장

상태 변경 워크플로 스텝을 직접 편집할 수 있도록 vRealize Automation Designer를 사용하여 사용자 지정 논리를 미리 결정된 IaaS 시스템 수명 주기 단계에 삽입하고 필요한 경우 사용자 지정 vRealize Orchestrator 워크플로에 호출할 수 있습니다.

**참고** 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. [vRealize Automation 확장을 위해 워크플로 구독 구성](#) 항목을 참조하십시오.

## vRealize Automation Designer 검사 목록을 사용하여 시스템 수명 주기 확장

“vRealize Automation Designer 검사 목록을 사용하여 시스템 수명 주기 확장”은 IaaS 시스템 수명 주기를 사용자 지정하기 위해 vRealize Automation Designer를 설치하고 구성하는 데 필요한 단계에 대한 간략한 내용을 제공합니다.

**표 1-17. vRealize Automation Designer 검사 목록을 사용하여 시스템 수명 주기 확장**

작업	세부 정보
<input type="checkbox"/> vRealize Automation Designer를 다운로드 하고 설치합니다.	<a href="#">vRealize Automation Designer 설치</a>
<input type="checkbox"/> vRealize Orchestrator 인스턴스에 대한 vRealize Automation 끝점을 생성합니다.	<a href="#">vRealize Orchestrator 끝점 생성</a>
<input type="checkbox"/> vRealize Orchestrator 끝점과 시스템 Blueprint를 연결합니다.	<a href="#">vRealize Orchestrator 끝점을 Blueprint와 연결</a>



**표 1-17. vRealize Automation Designer 검사 목록을 사용하여 시스템 수명 주기 확장 (계속)**

작업	세부 정보
<input type="checkbox"/> vRealize Automation Designer 작업을 사용하여 IaaS 워크플로 스텝을 사용자 지정합니다.	<a href="#">IaaS 워크플로 사용자 지정</a>
<b>참고</b> 워크플로 스텝은 이벤트 브로커 워크플로 구독으로 대체되었습니다. 워크플로 스텝은 아직 사용 가능하고 지원되지만 vRealize Automation의 나중 버전에서는 제거될 예정입니다. 미래 제품의 호환성을 보장하기 위해서는 워크플로 구독을 사용하여 상태 변경에 따라 사용자 지정 워크플로를 실행해야 합니다. <a href="#">vRealize Automation 확장을 위해 워크플로 구독 구성</a> 항목을 참조하십시오.	
필요한 경우, vRealize Orchestrator 워크플로 작업을 사용하여 사용자 지정 vRealize Orchestrator 워크플로에 호출할 수 있습니다.	
<input type="checkbox"/> 사용자 지정 상태 변경 워크플로를 생성한 후 테넌트 관리자 또는 비즈니스 그룹 관리자는 사용자 지정 속성을 추가하여 특정 Blueprint에 대해 이 워크플로를 사용하도록 설정해야 합니다.	<a href="#">상태 변경 워크플로를 호출하도록 Blueprint 구성</a>

## vRealize Automation Designer 설치 및 구성

vRealize Automation Designer는 Windows 시스템에 설치하여 원격 Model Manager 인스턴스와 통신하도록 구성할 수 있습니다. IaaS 워크플로를 사용하여 vRealize Orchestrator 워크플로를 호출하는 경우에는 IaaS에도 vRealize Orchestrator 인스턴스를 구성해야 합니다.

### vRealize Automation Designer 설치

vRealize Automation Designer는 Windows 시스템에 설치하여 원격 Model Manager 인스턴스와 통신하도록 구성할 수 있습니다.

### vRealize Automation Designer 사전 요구 사항

vRealize Automation Designer는 일반적으로 서버가 아니라 개발 시스템에 설치됩니다.

#### 지원되는 운영 체제

vRealize Automation Designer에 대해 지원하는 운영 체제는 VMware vRealize Automation 설명서 페이지의 vRealize Automation 지원 매트릭스에 나열되어 있습니다.

#### 시스템 구성 요구 사항

이 정보에 대한 잠재적인 업데이트 사항은 사용 중인 vRealize Automation 버전의 vRealize Automation 지원 매트릭스를 참조하십시오.

- .NET Framework 4.5가 설치되어 있어야 합니다.

- vRealize Automation Designer 호스트가 네트워크를 통해 IaaS 웹 사이트 구성 요소(특히 Model Manager Web 구성 요소)에 액세스할 수 있어야 합니다.
- Model Manager가 원격에 설치되어 있는 경우에는 Model Manager Web 구성 요소에 사용되는 인증서를 vRealize Automation Designer 호스트가 신뢰해야 합니다.

### vRealize Automation Designer 설치 관리자 다운로드

vRealize Automation 장치에서 vRealize Automation Designer 설치 관리자를 다운로드할 수 있습니다.

#### 필수 조건

- Windows 시스템에 로컬 관리자로 로그인합니다.
- Internet Explorer를 사용하는 경우 보안 강화 구성이 사용되지 않도록 설정되었는지 확인합니다. `res://iesetup.dll/SoftAdmin.htm`을 참조하십시오.

#### 프로시저

- 1 브라우저를 엽니다.
- 2 (`https://vra-va-hostname.domain.name:5480/installer/`)의 호스트 이름을 사용하여 Windows 설치 관리자 다운로드 페이지로 이동합니다.
- 3 **vRealize Automation Designer**를 클릭합니다.
- 4 메시지가 표시되면 설치 관리자를 저장합니다.

#### 후속 작업

[vRealize Automation Designer 설치](#).

### vRealize Automation Designer 설치

vRealize Automation Designer 설치 관리자는 Windows 설치 마법사로 패키징됩니다.

#### 필수 조건

[vRealize Automation Designer 설치 관리자 다운로드](#).

#### 프로시저

- 1 설치 관리자를 다운로드한 디렉토리로 이동합니다.
- 2 DesignCenter-Setup.exe를 마우스 오른쪽 버튼으로 클릭하고 **관리자 권한으로 실행**을 선택합니다.
- 3 시작 페이지에서 **다음**을 클릭합니다.
- 4 라이선스 계약을 읽고 **라이선스 계약 내용에 동의함**을 선택한 후 **다음**을 클릭합니다.
- 5 **사용자 지정 설치** 페이지에서 **다음**을 클릭합니다.
- 6 Model Manager Web 인스턴스의 정규화된 도메인 이름과 포트를 `hostname:port` 형식으로 지정합니다.

기본 포트는 443입니다.

7 Model Manager 서비스 사용자 자격 증명을 지정합니다.

8 다음을 클릭합니다.

설치 관리자가 Model Manager에 액세스를 시도하여 Model Manager 호스트와 자격 증명의 조합을 확인합니다. 오류가 반환될 경우, Model Manager 호스트와 자격 증명의 올바른 조합을 제공해야만 계속할 수 있습니다.

9 설치를 클릭합니다.

10 마침을 클릭합니다.

### 후속 작업

설치 디렉토리로 이동하여 Windows 시작 메뉴에서 vRealize Automation Designer를 시작할 수 있습니다.

## vRealize Orchestrator 끝점 구성

vRealize Orchestrator 워크플로를 호출하기 위해 vRealize Automation 워크플로를 사용 중인 경우 vRealize Orchestrator 인스턴스 또는 서버를 끝점으로 구성해야 합니다.

vRealize Orchestrator 끝점을 시스템 Blueprint와 연결하여 해당 Blueprint에서 프로비저닝된 시스템에 대한 모든 vRealize Orchestrator 워크플로가 해당 끝점을 사용하여 실행되도록 할 수 있습니다.

vRealize Automation에는 포함된 vRealize Orchestrator 인스턴스가 기본적으로 포함되어 있습니다. 테스트 환경에서 vRealize Automation 워크플로를 실행하거나 개념 검증을 생성하기 위한 vRealize Orchestrator 끝점으로 이 인스턴스를 사용하는 것이 좋습니다.

외부 vRealize Orchestrator 서버에 플러그인을 설치할 수도 있습니다.

### vCenter Orchestrator 통합 사전 요구 사항

vRealize Automation 워크플로를 사용하여 VC:VirtualMachine 유형의 입력 또는 출력 매개 변수를 가진 vRealize Orchestrator 워크플로를 실행하는 경우, vRealize Orchestrator와 IaaS 간에 가상 시스템 유형을 변환하는 vRealize Orchestrator 워크플로가 있는지 확인해야 합니다.

기본적으로 vRealize Orchestrator 5.5 이상에서는 필요한 워크플로가 vCenter 플러그인의 일부로 포함됩니다.

vRealize Orchestrator 5.1을 사용하는 경우에는 vRealize Orchestrator용 vRealize Automation 통합 패키지를 설치해야 합니다. vRealize Orchestrator 커뮤니티 사이트 (<http://communities.vmware.com/community/vmtn/server/vcenter/orchestrator>)에서 com.vmware.library.vcenter.vcac-integration.package 패키지를 다운로드하십시오. 그런 후 IaaS에서 끝점으로 설정하는 각 vRealize Orchestrator 서버에 패키지를 가져옵니다.

vRealize Orchestrator에 패키지를 가져오는 데 대한 자세한 내용은 vRealize Orchestrator 설명서를 참조하십시오.

### vRealize Orchestrator 끝점 생성

vRealize Orchestrator 끝점을 생성하여 vRealize Orchestrator 서버에 연결할 수 있습니다.

서로 다른 vRealize Orchestrator 서버에 연결하도록 여러 끝점을 구성할 수 있지만 각 끝점에 대해 우선 순위를 구성해야 합니다.

vRealize Orchestrator 워크플로를 실행할 때 vRealize Automation은 우선 순위가 가장 높은 vRealize Orchestrator 끝점을 먼저 시도합니다. 이 끝점에 연결할 수 없으면 vRealize Orchestrator 서버가 워크플로를 실행할 수 있을 때까지 우선 순위가 다음으로 높은 끝점을 계속 시도합니다.

### 필수 조건

- **laaS 관리자**로 vRealize Automation 콘솔에 로그인합니다.

### 프로시저

- 1 **인프라 > 끝점 > 끝점**을 선택합니다.
- 2 **새로 만들기 > 오케스트레이션 > vRealize Orchestrator**를 선택합니다.
- 3 이름을 입력하고 원하는 경우 설명을 입력합니다.
- 4 vRealize Orchestrator 서버의 정규화된 이름 또는 IP 주소와 vRealize Orchestrator 포트 번호가 포함된 URL을 입력합니다.  
전송 프로토콜은 HTTPS여야 합니다. 지정된 포트가 없으면 기본 포트 443이 사용됩니다.  
vRealize Automation 장치에 포함된 기본 vRealize Orchestrator 인스턴스를 사용하려면 `https://vrealize-automation-appliance-hostname:443/vco`를 입력합니다.
- 5 **사용자 이름** 및 **암호** 텍스트 상자에 vRealize Orchestrator 자격 증명을 입력하여 vRealize Orchestrator 끝점에 연결합니다.  
사용하는 자격 증명에는 laaS에서 호출할 모든 vRealize Orchestrator 워크플로에 대한 실행 권한이 있어야 합니다.  
vRealize Automation 장치에 포함된 기본 vRealize Orchestrator 인스턴스를 사용하려는 경우 사용자 이름은 `administrator@vsphere.local`이고 암호는 SSO 구성 시 지정한 관리자 암호입니다.
- 6 **우선 순위** 텍스트 상자에 1보다 크거나 같은 정수를 입력합니다.  
값이 낮을수록 우선 순위가 높습니다.
- 7 (선택 사항) **속성**을 클릭하고 제공된 사용자 지정 속성, 속성 그룹 또는 끝점에 대한 사용자 고유의 속성 정의를 추가합니다.
- 8 **확인**을 클릭합니다.

### vRealize Orchestrator 끝점을 Blueprint와 연결

Blueprint에서 사용할 특정 vRealize Orchestrator 끝점을 지정할 수 있습니다.

이 Blueprint에서 프로비저닝된 모든 시스템에 대해 laaS에서 vRealize Orchestrator 워크플로를 실행할 때 laaS는 항상 연결된 끝점을 사용합니다. 끝점에 연결할 수 없는 경우 워크플로는 실패합니다.

## 필수 조건

vRealize Automation 콘솔에 **인프라 설계자**로 로그인합니다.

## 프로시저

1 **설계 > Blueprint**를 선택합니다.

2 새 Blueprint를 생성하거나 기존 Blueprint를 편집합니다.

기존 Blueprint를 편집하는 경우 지정하는 vRealize Orchestrator 끝점은 업데이트된 Blueprint에서 프로비저닝된 새 시스템에만 적용됩니다. 이 속성을 시스템에 수동으로 추가하는 경우가 아니면 Blueprint에서 프로비저닝된 기존 시스템은 우선 순위가 가장 높은 끝점을 계속 사용합니다.

3 **속성** 탭을 클릭합니다.

a **새 속성**을 클릭합니다.

b **이름** 텍스트 상자에 `VMware.VCenterOrchestrator.EndpointName`을 입력합니다.

속성 이름은 대/소문자를 구분합니다.

c **값** 텍스트 상자에 vRealize Orchestrator 끝점의 이름을 입력합니다.

d **저장** 아이콘(🟢)을 클릭합니다.

4 **확인**을 클릭합니다.

## vRealize Automation Designer 를 사용하여 IaaS 워크플로 사용자 지정

VMware에서는 vRealize Automation Designer를 사용하여 사용자 지정할 수 있는 다수의 워크플로를 제공합니다. 여기에는 상태 변경 워크플로와 메뉴 작업 워크플로가 포함됩니다.

IaaS 워크플로는 NET Framework 4의 일부인 Microsoft Windows Workflow Foundation 4를 사용하여 생성됩니다. Windows Workflow Foundation과 워크플로 생성에 대한 자세한 내용은 Microsoft 설명서를 참조하십시오. vRealize Automation도 vRealize Orchestrator 워크플로 실행 및 모니터링을 위한 몇 개의 vRealize Automation Designer 작업을 제공합니다.

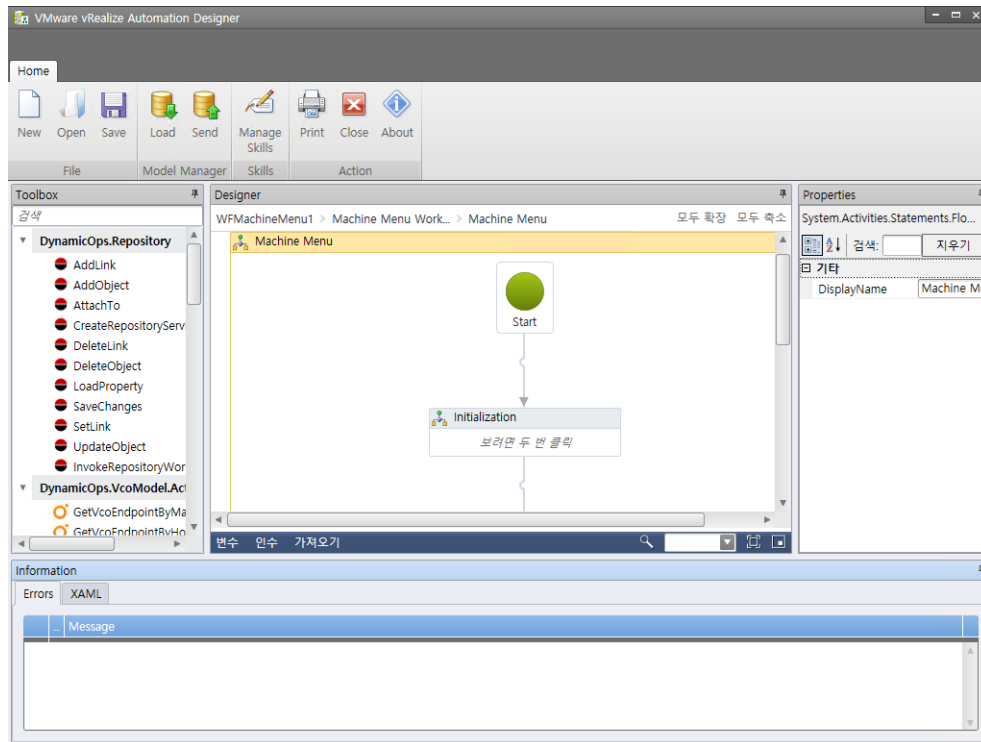
VMware에서 제공하는 사용자 지정 가능한 워크플로 템플릿은 초기화, 사용자 지정 논리 및 완료에 대한 별도의 시퀀스와 함께 워크플로 구성을 위한 모범 사례를 보여줍니다. 오류 처리를 위해 전체 워크플로가 TryCatch 블록에 래핑되어 있습니다. 확인되지 않은 또는 다시 발생한 예외는 워크플로를 실행하는 Distributed Execution Manager에 의해 기록됩니다.

사용자 지정 IaaS 워크플로를 생성한 후 Blueprint 작성자는 특정 Blueprint에 대해 워크플로를 사용할 수 있도록 설정해야 합니다.

## vRealize Automation Designer 콘솔

vRealize Automation Designer 콘솔은 IaaS 워크플로를 사용자 지정할 수 있는 시각적인 워크플로 편집기를 제공합니다.

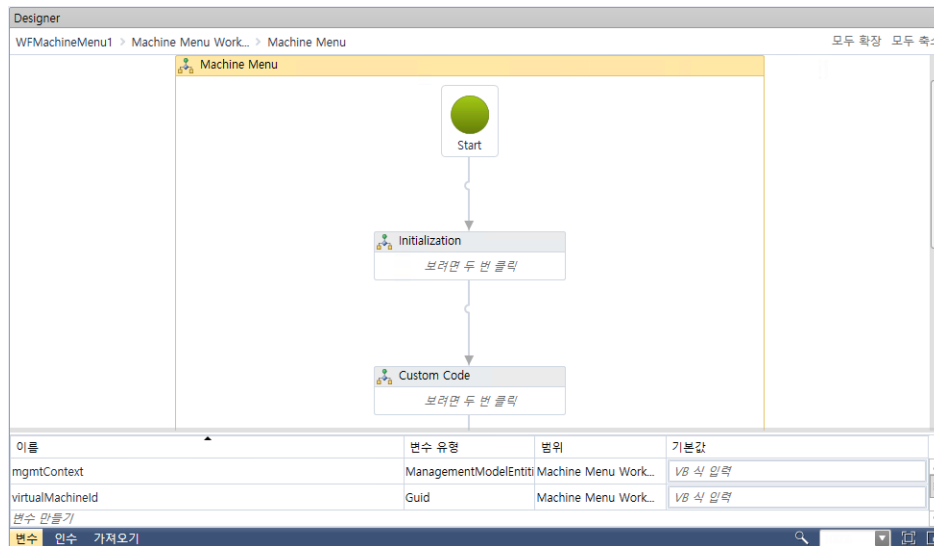
vRealize Automation Designer 콘솔을 시작하기 위해서는 vRealize Automation Designer 호스트(일반적으로 개발 시스템)에서 로컬 관리자 권한을 가지고 있어야 합니다.



왼쪽의 도구 상자 창에서는 vRealize Automation 워크플로 작업 라이브러리에 액세스할 수 있습니다. 도구 상자에서 [Designer] 창으로 작업을 끌어와 워크플로에 추가할 수 있습니다. [속성] 창에는 [Designer] 창에 현재 선택되어 있는 작업의 구성 가능한 속성이 표시됩니다. 이 인터페이스는 Visual Studio의 워크플로 디자이너와 매우 유사합니다.

[Designer] 창 아래쪽의 세부 정보 탭에서는 선택된 작업의 범위 내에서 변수를 표시 및 편집하거나, 선택된 작업에 대해 인수를 표시 및 편집할 수 있습니다.

**참고** 변수와 인수 둘 다 Visual Basic 식으로 지정됩니다. 그러나 변수 이름은 대/소문자를 구분하지 않는 반면 인수 이름은 대/소문자를 구분합니다. IaaS 워크플로 작업의 올바른 인수에 대한 자세한 내용은 [vRealize Automation 워크플로 작업 참조](#)를 참조하십시오.



[가져오기] 탭에는 가져온 네임스페이스가 표시되며 여기에서 워크플로에 추가할 엔티티 유형을 선택할 수 있습니다.

콘솔 아래쪽에 있는 축소 가능한 정보 창에는 구성 작업 중 발생한 오류가 표시되며 여기에서 워크플로의 XAML 표현을 액세스할 수 있습니다.

## IaaS 워크플로 유형

vRealize Automation Designer를 사용하여 상태 변경 워크플로와 메뉴 작업 워크플로의 두 가지 워크플로 유형을 사용자 지정할 수 있습니다.

- 상태 변경 워크플로는 상태 간에 마스터 워크플로가 전환될 때 실행됩니다. 예를 들어 새로운 시스템에 대한 프로비저닝 프로세스 중 특정 단계에서 실행될 수 있습니다.
- 메뉴 작업 워크플로는 사용자가 서비스 카탈로그의 [작업] 메뉴에서 옵션을 선택하거나 [인프라] 탭의 [시스템] 메뉴에서 옵션을 선택할 때 실행됩니다.

### 상태 변경 워크플로

상태 변경 워크플로를 생성하면 IaaS 마스터 워크플로가 특정 상태로 전환되기 전에 워크플로 실행을 트리거할 수 있습니다.

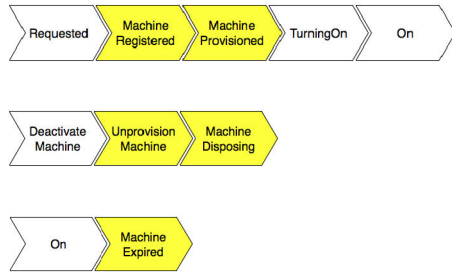
예를 들어 외부 데이터베이스에 통합되어 시스템 수명 주기의 여러 단계에서 정보를 기록하도록 사용자 지정 워크플로를 생성할 수 있습니다.

- 마스터 워크플로가 MachineProvisioned 상태로 전환되기 전에 실행되어 시스템 소유자, 승인자 등의 정보를 기록하는 사용자 지정 워크플로를 생성합니다.
- 시스템이 MachineDisposing 상태로 전환되기 전에 실행되어 시스템이 제거된 시간 및 마지막 데이터 수집 시의 리소스 활용도, 마지막 로그인 등의 데이터를 기록하는 사용자 지정 워크플로를 생성합니다.

다음 그림에서는 마스터 워크플로의 기본 상태를 보여 줍니다.







vRealize Automation Designer에서는 이러한 각 상태에 대해 사용자 지정 가능한 워크플로를 제공합니다.

**표 1-18. 사용자 지정 가능한 상태 변경 워크플로**

마스터 워크플로 상태	사용자 지정 가능한 워크플로 이름
BuildingMachine	WFStubBuildingMachine
Disposing	WFStubMachineDisposing
Expired	WFStubMachineExpired
MachineProvisioned	WFStubMachineProvisioned
RegisterMachine	WFStubMachineRegistered
UnprovisionMachine	WFStubUnprovisionMachine

### 상태 변경 워크플로 구성 개요

vRealize Automation Designer를 사용하여 상태 변경 워크플로를 사용자 지정할 수 있습니다. 그런 다음 Blueprint 작성자가 특정 Blueprint에 대해 이를 사용하도록 설정할 수 있습니다.

다음은 상태 변경 워크플로를 사용하도록 설정하는 데 필요한 단계에 대한 개괄적인 내용입니다.

- 1 워크플로 개발자가 vRealize Automation Designer를 사용하여 상태 변경 워크플로 템플릿 중 하나를 사용자 지정합니다. [IaaS 워크플로 사용자 지정](#) 항목을 참조하십시오.

모든 IaaS 워크플로에서 vRealize Orchestrator 워크플로를 호출할 수 있습니다. 자세한 내용은 [vRealize Orchestrator 워크플로 작업 사용](#)을 참조하십시오.

- 2 테넌트 관리자 또는 비즈니스 그룹 관리자가 특정 Blueprint에서 프로비저닝된 시스템에 대해 사용자 지정된 워크플로를 호출하도록 해당 Blueprint를 구성합니다. [상태 변경 워크플로를 호출하도록 Blueprint 구성](#) 항목을 참조하십시오.

### 메뉴 작업 워크플로

메뉴 작업 워크플로는 사용자가 서비스 카탈로그의 [작업] 메뉴 또는 [인프라] 탭의 시스템 메뉴에서 옵션을 선택하면 실행됩니다.

예를 들어 사용자가 시스템 메뉴에서 [지원 문제 등록]을 선택하여 시스템과 관련된 지원 티켓을 생성하도록 하는 사용자 지정 워크플로를 생성할 수 있습니다.

vRealize Automation Designer에는 메뉴 작업 워크플로를 사용자 지정할 수 있는 템플릿이 제공됩니다.



메뉴 작업 워크플로에는 워크플로 정의와 함께 작업 구성 파일도 필요합니다. 작업 구성 파일은 표시 텍스트, 해당 옵션에 액세스할 수 있는 역할, 작업을 사용할 수 있는 시스템 상태 등 사용자 지정 메뉴 옵션의 측면을 정의합니다.

**참고** XaaS 설계자는 XaaS를 사용하여 모든 카탈로그 항목에 대해 사용자 지정 작업을 정의할 수 있습니다. vSphere 또는 vCloud Director를 사용하여 프로비저닝된 IaaS 시스템 이외의 IaaS 시스템에 대한 사용자 지정 작업을 생성하려면 vRealize Automation 6.1 이상이 필요합니다.

### 메뉴 작업 워크플로 구성 개요

vRealize Automation Designer 및 CloudUtil 명령줄 유틸리티를 사용하여 메뉴 작업 워크플로를 사용자 지정할 수 있습니다. 그런 다음 Blueprint 작성자가 특정 Blueprint에 대해 이를 사용하도록 설정할 수 있습니다.

다음은 메뉴 작업 워크플로를 사용하도록 설정하는 데 필요한 단계에 대한 개괄적인 내용입니다.

- 1 워크플로 개발자가 vRealize Automation Designer를 사용하여 메뉴 작업 워크플로 템플릿 중 하나를 사용자 지정합니다. [IaaS 워크플로 사용자 지정](#) 항목을 참조하십시오.

모든 IaaS 워크플로에서 vRealize Orchestrator 워크플로를 호출할 수 있습니다. 자세한 내용은 [vRealize Orchestrator 워크플로 작업 사용](#)을 참조하십시오.

- 2 워크플로 개발자가 Model Manager에서 메뉴 작업을 구성합니다. [메뉴 작업 구성](#) 항목을 참조하십시오.
- 3 워크플로 개발자가 새 메뉴 작업을 서비스 카탈로그에 등록합니다. [서비스 카탈로그에 새 메뉴 작업 등록](#) 항목을 참조하십시오.
- 4 테넌트 관리자 또는 비즈니스 그룹 관리자가 특정 Blueprint에서 프로비저닝된 시스템에 대해 메뉴 작업을 사용하도록 해당 Blueprint를 구성합니다. [메뉴 작업 워크플로를 사용하도록 Blueprint 구성](#) 항목을 참조하십시오.

메뉴 작업이 서비스 카탈로그에서 사용하기 위한 것이라면 사용자에게 이에 대한 사용 권한도 부여해야 합니다. 자세한 내용은 테넌트 관리 항목을 참조하십시오.

### IaaS 워크플로 사용자 지정

vRealize Automation Designer를 사용하여 사용자 지정 가능한 워크플로를 편집하고 Model Manager에서 워크플로를 업데이트할 수 있습니다.

#### 필수 조건

vRealize Automation Designer를 시작합니다.

#### 프로시저

- 1 **로드**를 클릭합니다.

## 2 사용자 지정하려는 워크플로를 선택합니다.

옵션	설명
<b>WFMachineMenuN</b>	사용자 지정 가능한 메뉴 작업 워크플로
<b>WFStubBuildingMachine</b>	시스템이 BuildingMachine 상태에 들어가기 전에 실행되는 사용자 지정 가능한 상태 변경 워크플로
<b>WFStubMachineDisposing</b>	시스템이 Disposing 상태에 들어가기 전에 실행되는 사용자 지정 가능한 상태 변경 워크플로
<b>WFStubMachineExpired</b>	시스템이 Expired 상태에 들어가기 전에 실행되는 사용자 지정 가능한 상태 변경 워크플로
<b>WFStubMachineProvisioned</b>	시스템이 MachineProvisioned 상태에 들어가기 전에 실행되는 사용자 지정 가능한 상태 변경 워크플로
<b>WFStubMachineRegistered</b>	시스템이 RegisterMachine 상태에 들어가기 전에 실행되는 사용자 지정 가능한 상태 변경 워크플로
<b>WFStubUnprovisionMachine</b>	시스템이 UnprovisionMachine 상태에 들어가기 전에 실행되는 사용자 지정 가능한 상태 변경 워크플로

## 3 확인을 클릭합니다.

워크플로는 [Designer] 창에 표시됩니다.

## 4 작업을 도구 상자에서 [Designer] 창으로 끈 다음 해당 인수를 구성하여 워크플로를 사용자 지정합니다.

## 5 워크플로 편집이 완료되면 보내기를 클릭하여 Model Manager에서 워크플로를 업데이트합니다.

워크플로가 저장되고 다음에 워크플로를 로드할 때 새로운 개정 버전으로 목록에 표시됩니다. 언제나라도 이전 버전의 워크플로에 액세스할 수 있습니다. [워크플로의 이전 개정으로 되돌리기](#) 항목을 참조하십시오.

## vRealize Orchestrator 워크플로 작업 사용

vRealize Automation Designer 작업을 사용하면 vRealize Orchestrator 워크플로를 동기식 또는 비동기식으로 호출할 수 있습니다.

vRealize Orchestrator 끝점은 다음 방법 중 하나로 지정할 수 있습니다.

- VirtualMachineId는 가상 시스템 ID를 나타내는 변수의 이름입니다. 이 ID를 가진 가상 시스템이 선택되고 특정 가상 시스템에 대해 VMware.VCenterOrchestrator.EndpointName 사용자 지정 속성에서 검색되는 값이 vRealize Orchestrator 끝점 이름으로 사용됩니다.
- GetVcoEndpointByManagementEndpoint는 지정된 ManagementEndpoint 개체의 사용자 지정 속성 값을 반환합니다. CustomPropertyName을 지정하지 않으면 VMware.VCenterOrchestrator.EndpointName 속성의 값이 사용됩니다.
- GetVcoEndpointByHost는 지정된 호스트의 사용자 지정 속성 값을 반환합니다. CustomPropertyName을 지정하지 않으면 VMware.VCenterOrchestrator.EndpointName 속성의 값이 사용됩니다.

## 동기식 실행

InvokeVcoWorkflow 작업은 vRealize Orchestrator 워크플로를 호출하고, vRealize Orchestrator 워크플로가 완료될 때까지 해당되는 상위 IaaS 워크플로의 추가적인 실행을 차단합니다. 이 작업은 vRealize Orchestrator 워크플로에 대한 출력 매개 변수를 반환합니다.

동기식 실행에서는 다음의 속성도 함께 지원됩니다.

- WorkflowTimeout은 초 단위의 시간 제한 값입니다. vRealize Orchestrator 워크플로가 지정된 시간에 완료되지 않으면 응답이 반환될 때까지 워크플로를 차단하는 대신 예외가 생성됩니다. 값을 정의하지 않거나 0 값을 제공하면 시간 제한이 활성화되지 않습니다.  
VMware.VCenterOrchestrator.PollingInterval 사용자 지정 속성에서 값을 지정하여 끝점에 대해 폴링 시간을 수정하는 경우를 제외하고 이 기간 동안 워크플로 상태가 10초 간격으로 확인됩니다.

## 비동기식 실행

InvokeVcoWorkflowAsync 작업은 “발생 후 삭제” 작업으로, vRealize Orchestrator 워크플로를 호출한 후 vRealize Orchestrator 워크플로가 완료될 때까지 기다리지 않고 IaaS 워크플로의 작업을 계속해서 실행합니다.

이 작업은 워크플로를 모니터링하는 데 사용할 수 있는 고유한 워크플로 실행 토큰을 반환하거나, vRealize Orchestrator 서버에 대한 REST API 호출이 실패한 경우(예: 서버에 연결할 수 없는 경우)에 오류를 반환합니다.

이 작업에는 두 가지 추가적인 작업을 사용할 수 있습니다.

- GetVcoWorkflowExecutionStatus를 사용하면 vRealize Orchestrator 워크플로에 대해 해당 상태를 폴링할 수 있습니다.
- WaitForVcoWorkflowCompletion을 사용하면 vRealize Orchestrator 워크플로가 완료되거나 시간이 초과될 때까지 IaaS 워크플로의 추가적인 실행을 차단할 수 있습니다. 이 작업은 비동기식으로 실행하는 vRealize Orchestrator 워크플로의 결과를 검색하기 위해 사용할 수 있습니다.

## vRealize Orchestrator 워크플로 호출

InvokeVcoWorkflow 또는 InvokeVcoWorkflowAsync 작업을 사용하여 IaaS 워크플로에서 vRealize Orchestrator 워크플로를 호출할 수 있습니다.

일부 vRealize Orchestrator 워크플로는 실행 중 사용자 상호 작용이 필요합니다. 이러한 워크플로의 경우 사용자 안내 메시지가 vRealize Automation 콘솔이 아닌 vRealize Orchestrator 클라이언트에 나타나므로 워크플로가 입력을 대기하는 vRealize Automation의 최종 사용자는 명확하게 알 수 없습니다.

사용자 입력을 차단하는 워크플로를 피하려면 IaaS 워크플로의 사용자 상호 작용이 필요한 vRealize Orchestrator 워크플로를 호출하지 마십시오.

## 프로시저

- 1 vRealize Automation Designer에서 워크플로를 열고 vRealize Orchestrator 워크플로를 호출하려는 컨텍스트로 이동합니다.
- 2 InvokeVcoWorkflow 또는 InvokeVcoWorkflowAsync 작업을 [Designer] 창으로 끕니다.

### 3 실행할 vCenter Orchestrator 워크플로를 선택합니다.

- a [일반] 아래에서 워크플로 옆의 말줄임표를 클릭합니다.
- b [vCO 워크플로 찾아보기] 대화상자에서 워크플로를 선택합니다.
- c **확인**을 클릭합니다.

입력 및 출력 섹션에 선택된 워크플로의 입력 및 출력 매개 변수가 표시됩니다.

### 4 [속성] 창에서 다음 대상 매개 변수 중 하나를 지정합니다.

- VirtualMachineId는 가상 시스템 ID를 나타내는 변수의 이름입니다. 이 ID를 가진 가상 시스템이 선택되고 특정 가상 시스템에 대해 VMware.VCenterOrchestrator.EndpointName 사용자 지정 속성에서 검색되는 값이 vRealize Orchestrator 끝점 이름으로 사용됩니다.
- VcoEndpointName은 워크플로를 실행하는 데 사용되는 끝점 이름입니다. 지정하는 경우 이 값은 vRealize Orchestrator 끝점을 선택할 때 VirtualMachineId 값을 재정의합니다.
- WorkflowTimeout은 초 단위의 시간 제한 값입니다. vRealize Orchestrator 워크플로가 지정된 시간에 완료되지 않으면 응답이 반환될 때까지 워크플로를 차단하는 대신 예외가 생성됩니다. 값을 정의하지 않거나 0 값을 제공하면 시간 제한이 활성화되지 않습니다. VMware.VCenterOrchestrator.PollingInterval 사용자 지정 속성에서 값을 지정하여 끝점에 대해 폴링 시간을 수정하는 경우를 제외하고 이 기간 동안 워크플로 상태가 10초 간격으로 확인됩니다.

### 5 vRealize Orchestrator 워크플로에 대한 매개 변수를 지정합니다.

- [Designer] 창의 작업에 값을 입력합니다.
- [속성] 창에서 **InputParameters** 또는 **OutputParameters** 옆의 말줄임표를 클릭하여 [매개 변수] 대화상자를 엽니다. 이 대화상자는 각 매개 변수의 IaaS 유형을 표시합니다. 매개 변수 유형이 굵게 나타나면 해당 매개 변수는 필수입니다.

매개 변수의 텍스트 상자를 가리키면 vRealize Orchestrator 유형을 나타내는 도구 설명을 볼 수 있습니다.

InvokeVcoWorkflowAsync 작업을 사용 중인 경우 vRealize Orchestrator 워크플로의 출력 매개 변수가 정보 제공 목적으로 해당 유형과 함께 표시되지만 이 작업의 매개 변수에 대해 식을 지정할 수 없습니다.

#### 후속 작업

비동기식으로 실행하는 워크플로의 결과를 검색하려면 WaitForVcoWorkflowCompletion 작업을 사용합니다.

#### vRealize Orchestrator 워크플로의 상태 가져오기

GetVcoWorkflowExecutionStatus 작업을 사용하여 InvokeVcoWorkflowAsync 작업으로 호출된 vRealize Orchestrator 워크플로의 상태를 확인할 수 있습니다.

#### 필수 조건

InvokeVcoWorkflowAsync 작업을 사용하여 **vRealize Orchestrator 워크플로 호출**.

**프로시저**

- 1 vRealize Automation Designer에서, InvokeVcoWorkflowAsync 작업을 사용한 워크플로를 엽니다.
- 2 vRealize Orchestrator 워크플로의 상태를 확인하려는 컨텍스트로 이동합니다.
- 3 GetVcoWorkflowExecutionStatus 작업을 [Designer] 창으로 끕니다.
- 4 [속성] 창에서, 가상 시스템 ID를 나타내는 변수의 이름을 VirtualMachineId에 지정합니다.  
사용자 지정 가능한 워크플로에는 기본 이름이 virtualMachineId인 변수가 포함되어 있습니다. 이 이름은 초기화 중에 설정됩니다.
- 5 DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken 유형의 변수를 생성합니다.
- 6 토큰 변수의 이름을 InvokeVcoWorkflowAsync 작업의 executionToken 출력 매개 변수로 지정합니다.
- 7 동일한 변수 이름을 GetVcoWorkflowExecutionStatus 작업의 WorkflowExecutionToken 속성으로 지정합니다.
- 8 문자열 유형의 변수를 생성합니다.
- 9 문자열 변수의 이름을 GetVcoWorkflowExecutionStatus 작업의 VcoWorkflowExecutionStatus 속성으로 지정합니다.

워크플로가 실행될 때 VcoWorkflowExecutionStatus 변수의 값이 vRealize Orchestrator 워크플로의 상태로 설정됩니다.

**vRealize Orchestrator 워크플로의 결과 가져오기**

vRealize Orchestrator 워크플로를 비동기식으로 호출하고 나중에 완료된 워크플로의 결과를 가져오려는 경우 WaitForVcoWorkflowCompletion 작업을 사용할 수 있습니다.

WaitForVcoWorkflowCompletion 작업은 vRealize Orchestrator 워크플로가 완료될 때까지 또는 시간 제한에 도달할 때까지 IaaS 워크플로의 실행을 차단합니다. 이 작업은 vRealize Orchestrator 워크플로가 성공적으로 완료되면 해당 결과를 반환하고, 실패하면 오류를 반환하며, 워크플로 시간 제한이 발생하면 Null을 반환합니다.

**필수 조건**

InvokeVcoWorkflowAsync 작업을 사용하여 [vRealize Orchestrator 워크플로 호출](#).

**프로시저**

- 1 vRealize Automation Designer에서, InvokeVcoWorkflowAsync 작업을 사용한 워크플로를 엽니다.
- 2 vRealize Orchestrator 워크플로의 결과를 검색하려는 컨텍스트로 이동합니다.
- 3 WaitForVcoWorkflowCompletion 작업을 [Designer] 창으로 끕니다.
- 4 [속성] 창에서, 가상 시스템 ID를 나타내는 변수의 이름을 VirtualMachineId에 지정합니다.  
사용자 지정 가능한 워크플로에는 기본 이름이 virtualMachineId인 변수가 포함되어 있습니다. 이 이름은 초기화 중에 설정됩니다.

- 5 `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken` 유형의 변수를 생성합니다.
- 6 `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken` 유형의 변수를 생성합니다.
- 7 토큰 변수의 이름을 `InvokeVcoWorkflowAsync` 작업의 `executionToken` 출력 매개 변수로 지정합니다.
- 8 동일한 변수 이름을 `WaitForVcoWorkflowCompletion` 작업의 `WorkflowExecutionToken` 속성으로 지정합니다.
- 9 `vRealize Orchestrator` 워크플로의 출력을 검색합니다.
  - a `DynamicOps.VcoModel.Common.VcoWorkflowExecutionResult` 유형의 변수를 생성합니다.
  - b 결과 변수의 이름을 `WaitForVcoWorkflowCompletion` 작업의 `WorkflowOutput` 속성으로 지정합니다.

워크플로가 실행될 때 변수의 값이 `vRealize Orchestrator` 워크플로(있는 경우)의 결과로 설정됩니다.

### vRealize Orchestrator 및 IaaS 개체 유형

`InvokeVcoWorkflow` 또는 `InvokeVcoWorkflowAsync` 작업을 `vRealize Automation Designer`에서 사용하는 경우, 작업의 입력 및 출력 속성은 선택된 `vRealize Orchestrator` 워크플로의 매개 변수에 기반하여 자동으로 채워집니다.

`vRealize Orchestrator` 기본 유형은 다음의 IaaS 유형으로 변환됩니다.

**표 1-19. vRealize Orchestrator 및 IaaS 개체 유형**

vRealize Orchestrator 유형	IaaS 유형
string	string
boolean	bool
number	decimal
SecureString	string
Text	string
Array/T	Array<T>
Properties	Dictionary<string,object>
Date	DateTime
VC:VirtualMachine	VirtualMachine

**참고** `vRealize Orchestrator 5.1`을 사용하는 경우에는 `vRealize Automation` 통합 패키지가 설치되어 있어야 `VC:VirtualMachine` 개체 유형을 `VirtualMachine`으로 변환할 수 있습니다.

다른 모든 `vRealize Orchestrator` 유형은 IaaS 유형 `VcoSdkObject`로 변환됩니다.

### 상태 변경 워크플로를 호출하도록 Blueprint 구성

사용자 지정 상태 변경 워크플로를 생성한 후 테넌트 관리자 또는 비즈니스 그룹 관리자는 사용자 지정 속성을 추가하여 특정 Blueprint에 대해 이 워크플로를 사용하도록 설정해야 합니다.

각 상태 변경 워크플로는 특정 사용자 지정 속성과 연결되어 있습니다. 시스템이 해당 상태 변경 워크플로와 함께 특정 상태에 들어가면 IaaS에서 시스템에 해당 사용자 지정 속성이 있는지 확인합니다. 있는 경우 연결된 워크플로가 실행됩니다. 예를 들어 시스템에 사용자 지정 속성 ExternalWFStubs.MachineProvisioned가 있는 경우 마스터 워크플로가 MachineProvisioned 상태에 들어가기 전에 WFStubMachineProvisioned 워크플로가 실행됩니다.

사용자 지정 속성은 여러 소스의 시스템에 적용될 수 있지만 일반적으로 상태 변경 워크플로의 속성이 Blueprint에 지정되어 해당 Blueprint에서 프로비저닝된 모든 시스템에 대해 워크플로가 사용되도록 설정됩니다.

## 필수 조건

테넌트 관리자 또는 비즈니스 그룹 관리자로 vRealize Automation 콘솔에 로그인합니다.

## 프로시저

- 1 **설계 > Blueprint**를 선택합니다.
- 2 Blueprint의 이름을 가리키고 **편집**을 클릭합니다.
- 3 **속성** 탭을 클릭합니다.
- 4 **새 속성**을 클릭합니다.
- 5 사용하도록 설정하려는 워크플로와 연결된 사용자 지정 속성의 이름을 **이름** 텍스트 상자에 입력합니다.

사용자 지정 가능한 워크플로 이름	연결된 속성 이름
WFStubMachineProvisioned	ExternalWFStubs.MachineProvisioned
WFStubBuildingMachine	ExternalWFStubs.BuildingMachine
WFStubMachineDisposing	ExternalWFStubs.MachineDisposing
WFStubUnprovisionMachine	ExternalWFStubs.UnprovisionMachine
WFStubMachineRegistered	ExternalWFStubs.MachineRegistered
WFStubMachineExpired	ExternalWFStubs.MachineExpired

- 6 **값** 텍스트 상자는 비워 둡니다.  
워크플로는 특정 값이 아닌 속성이 있는지 여부에 따라 달라집니다.
- 7 **저장** 아이콘(📁)을 클릭합니다.
- 8 **확인**을 클릭합니다.

이제 워크플로가 이 Blueprint에서 프로비저닝된 새 시스템에 대해 사용되도록 설정되었습니다.

## 메뉴 작업 워크플로 구성

메뉴 작업 워크플로를 사용자 지정한 후에 이를 vRealize Automation 콘솔에서 사용자가 사용할 수 있으려면 먼저 추가 구성이 필요합니다.



## 메뉴 작업 구성

메뉴 작업을 구성하려면 작업 구성 파일을 생성하고 이 파일을 Model Manager에 설치합니다.

### 프로시저

#### 1 작업 구성 파일 생성

메뉴 작업 워크플로에는 작업 구성 파일이 필요합니다. 작업 구성 파일은 표시 텍스트, 옵션에 액세스할 수 있는 역할, 옵션을 사용할 수 있는 시스템 상태와 같은 vRealize Automation 콘솔의 사용자 지정 메뉴 옵션의 측면을 지정합니다.

#### 2 Model Manager에서 작업 설치

CloudUtil 명령줄 유틸리티를 사용하여 Model Manager에서 작업을 설치합니다.

### 후속 작업

메뉴 작업이 서비스 카탈로그에서 사용하기 위한 것이라면 이에 대한 사용 권한이 사용자에게 부여되도록 메뉴 작업을 서비스 카탈로그에 등록해야 합니다. [서비스 카탈로그에 새 메뉴 작업 등록](#).

## 작업 구성 파일 생성

메뉴 작업 워크플로에는 작업 구성 파일이 필요합니다. 작업 구성 파일은 표시 텍스트, 옵션에 액세스할 수 있는 역할, 옵션을 사용할 수 있는 시스템 상태와 같은 vRealize Automation 콘솔의 사용자 지정 메뉴 옵션의 측면을 지정합니다.

### 프로시저

#### 1 새 XML 파일을 생성합니다.

```
<?xml version="1.0" encoding="utf-8"?>
```

#### 2 루트 요소 customOperations를 생성합니다.

```
<customOperations xmlns="http://www.dynamicsops.com/schemas/2009/OperationConfig/">
</customOperations>
```

요소에서 XML 네임스페이스 `http://www.dynamicsops.com/schemas/2009/OperationConfig/`를 지정해야 합니다.

#### 3 정의하려는 각 작업에 대해 customOperations 내에 operation 요소를 추가합니다.

```
<operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
</operation>
```

operation 요소는 다음과 같은 특성을 가집니다.

특성	설명
<b>name</b>	이 작업이 실행하는 워크플로의 이름입니다.
<b>displayName</b>	시스템 메뉴의 옵션에 대한 설명 레이블입니다.



#### 4 메뉴 작업에 대한 액세스 권한을 부여할 역할을 지정합니다.

##### a authorizedTasks 요소를 추가합니다.

```
<operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
  <authorizedTasks>
  </authorizedTasks>
</operation>
```

##### b 작업에 대한 액세스 권한을 부여할 각 역할에 대해 task 요소를 추가합니다. 예는 다음과 같습니다.

```
<authorizedTasks>
  <task>VRM User Custom Event</task>
  <task>VRM Support Custom Event</task>
  <task>Group Administrator Custom Event</task>
  <task>Enterprise Administrator Custom Event</task>
  <task>VRM Administrator Custom Event</task>
</authorizedTasks>
```

task 요소의 올바른 콘텐츠는 다음과 같습니다.

요소 콘텐츠	설명
<b>VRM User Custom Event</b>	모든 사용자에게 작업에 대한 액세스 권한을 부여합니다.
<b>VRM Support Custom Event</b>	지원 사용자에게 작업에 대한 액세스 권한을 부여합니다.
<b>Group Administrator Custom Event</b>	비즈니스 그룹 관리자에게 작업에 대한 액세스 권한을 부여합니다.
<b>Enterprise Administrator Custom Event</b>	패브릭 관리자에게 작업에 대한 액세스 권한을 부여합니다.
<b>VRM Administrator Custom Event</b>	IaaS 관리자에게만 작업에 대한 액세스 권한을 부여합니다.

## 5 (선택 사항) 작업을 사용할 수 있는 시스템 상태를 지정합니다.

### a machineStates 요소를 추가합니다.

```
<operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
  <machineStates>
  </machineStates>
</operation>
```

### b 작업을 사용할 수 있는 각 상태에 대해 state 요소를 추가합니다.

```
<machineStates>
  <state>On</state>
  <state>Off</state>
</machineStates>
```

값은 가능한 임의의 시스템 상태 중 하나가 될 수 있습니다. 전체 시스템 상태 목록은 가상 플랫폼을 위한 IaaS 구성, 물리적 시스템의 IaaS 구성 또는 클라우드 플랫폼을 위한 IaaS 구성 항목을 참조하십시오.

요소를 생략하면 모든 시스템 상태에 대해 작업을 사용할 수 있습니다.

다음은 완전한 작업 구성 파일의 예입니다.

```
<?xml version="1.0" encoding="utf-8" ?>
<customOperations xmlns="http://www.dynamicops.com/schemas/2009/OperationConfig/">
  <operation name="WFMachineMenu1" displayName="Execute Machine Menu task">
    <authorizedTasks>
      <task>VRM User Custom Event</task>
      <task>VRM Support Custom Event</task>
      <task>Group Administrator Custom Event</task>
      <task>Enterprise Administrator Custom Event</task>
      <task>VRM Administrator Custom Event</task>
    </authorizedTasks>
    <machineStates>
      <state>On</state>
      <state>Off</state>
    </machineStates>
  </operation>
</customOperations>
```

## Model Manager에서 작업 설치

CloudUtil 명령줄 유틸리티를 사용하여 Model Manager에서 작업을 설치합니다.

### 필수 조건

작업 구성 파일 생성.

### 프로시저

#### 1 관리자 권한으로 명령 프롬프트를 엽니다.

## 2 다음 인수와 함께 CloudUtil.exe 명령을 실행합니다.

- `CloudUtil.exe Operation-Create -c <path to operation definition file>`
- 필요한 경우 Model Manager 호스트를 지정하고 오류 발생 시 스택 추적을 요청할 수 있습니다.

```
CloudUtil.exe Operation-Create -c <path to operation definition file>
--repository <Model Manager Root URI> -v
```

### 후속 작업

메뉴 작업이 서비스 카탈로그에서 사용하기 위한 것이라면 이에 대한 사용 권한이 사용자에게 부여되도록 메뉴 작업을 서비스 카탈로그에 등록해야 합니다. [서비스 카탈로그에 새 메뉴 작업 등록](#).

### 서비스 카탈로그에 새 메뉴 작업 등록

새 메뉴 작업을 설치한 후 워크플로 개발자는 사용자에게 해당하는 권한을 부여할 수 있게 새 작업 메뉴를 서비스 카탈로그에 등록해야 합니다.

### 필수 조건

- [메뉴 작업 구성](#).
- 관리자 권한을 가진 로컬 사용자로 IaaS Model Manager 호스트에 로그인합니다.

### 프로시저

- 1 관리자 권한으로 명령 프롬프트를 엽니다.
- 2 IaaS 루트 설치 디렉토리로 이동합니다.  
일반 설치의 경우 이 디렉토리는 C:\Program Files (x86)\VMware\WvCAC입니다.
- 3 Server\WModel Manager Data\Cafe로 이동합니다.
- 4 다음 명령을 실행합니다.

```
Vcac-Config.exe RegisterCatalogTypes -v
```

### 후속 작업

새 작업을 사용자가 서비스 카탈로그에서 사용할 수 있으려면 먼저 테넌트 관리자나 비즈니스 그룹 관리자가 새 작업에 대한 사용 권한을 부여해야 합니다. 자세한 내용은 테넌트 관리 항목을 참조하십시오.

### 메뉴 작업 워크플로를 사용하도록 Blueprint 구성

특정 Blueprint에 대한 보안 구성을 업데이트하여 해당 Blueprint에서 프로비저닝된 시스템에 대한 메뉴 작업 워크플로를 사용하도록 설정합니다.

### 필수 조건

테넌트 관리자 또는 비즈니스 그룹 관리자로 vRealize Automation 콘솔에 로그인합니다.

## 프로시저

- 1 **설계 > Blueprint**를 선택합니다.
- 2 Blueprint의 이름을 가리키고 **편집**을 클릭합니다.
- 3 **작업** 탭을 클릭합니다.
- 4 사용하도록 설정하려는 작업의 확인란을 선택합니다.
- 5 **확인**을 클릭합니다.

이제 이 Blueprint에서 프로비저닝된 시스템에 대해 메뉴 작업이 사용되도록 설정되었고 작업 구성 파일에 지정된 모든 사용자 역할에서 이 메뉴 작업을 사용할 수 있습니다.

## 후속 작업

메뉴 작업이 서비스 카탈로그에서 사용하기 위한 것이라면 사용자에게 이에 대한 사용 권한도 부여해야 합니다. 자세한 내용은 테넌트 관리 항목을 참조하십시오.

## 워크플로의 이전 개정으로 되돌리기

**워크플로 로드** 대화상자에는 Model Manager의 워크플로에 대한 모든 개정이 표시되기 때문에 워크플로의 전체 버전 기록을 액세스할 수 있습니다.

워크플로를 Model Manager에 보낼 때마다 개정 및 타임 스탬프가 업데이트됩니다.

## 필수 조건

vRealize Automation Designer 콘솔을 시작합니다.

## 프로시저

- 1 **로드**를 클릭합니다.
- 2 되돌리려는 워크플로 개정을 선택합니다.  
VMware에서 제공하는 원래 워크플로는 개정 0입니다.
- 3 **확인**을 클릭합니다.
- 4 **보내기**를 클릭하여 Model Manager에서 워크플로를 업데이트합니다.

Model Manager에서 이전 개정이 최신 개정이 됩니다. 예를 들어 워크플로에 대해 개정 1과 2를 생성한 후 개정 0을 로드하고 저장하면 이제 개정 0과 개정 3이 동일하며, 워크플로가 VMware에서 제공한 버전으로 되돌려집니다.

## 워크플로 및 분산 실행 관리

기술을 사용하여 특정 Distributed Execution Manager에서만 가능하도록 워크플로 실행을 제한할 수 있습니다.

기술은 워크플로와 DEM 작업자 인스턴스 둘 모두에 적용할 수 있는 태그와 유사합니다. 워크플로에 기술이 연결되어 있지 않으면 모든 DEM 작업자가 해당 워크플로를 실행할 수 있습니다. 워크플로에 기술이 하나 이상 연결되어 있으면 동일한 모든 기술에 연결된 DEM 작업자만 해당 워크플로를 실행할 수 있습니다.

기술은 특정 워크플로를 실행하기 위해 특정 사전 요구 사항을 가진 호스트에 DEM이 설치되어 있어야 하는 경우에 유용합니다. 예를 들어 클라우드 프로비저닝 워크플로를 Amazon URL에 대해 필수 네트워크 액세스 권한을 가진 호스트에서 실행되는 특정 DEM으로 제한할 수 있습니다.

기술은 워크플로를 특정 데이터 센터 위치에 연결하는 데도 사용될 수 있습니다. 예를 들어 DEM 하나를 보스턴 데이터 센터에 설치하고 다른 DEM을 런던 데이터 센터에 설치한 후 기술을 사용하여 특정 작업을 두 곳의 데이터 센터 중 하나에 지시할 수 있습니다.



## 기술을 사용하여 워크플로와 DEM 작업자 연결

기술을 Model Manager에 추가한 다음 기술을 하나 이상의 워크플로 및 DEM 작업자와 연결하는 방식으로 워크플로를 특정 DEM 작업자 또는 작업자 인스턴스 집합에 연결합니다.

### 필수 조건

vRealize Automation Designer 콘솔을 시작합니다.

### 프로시저

- 1 리본에서 **기술 관리**를 클릭합니다.
- 2 **기술 관리** 대화상자의 왼쪽 상단에 위치한 텍스트 필드에 새 기술의 이름을 입력하고 [추가] 버튼을 클릭합니다.  
기술 이름은 고유해야 합니다. 새 기술 이름이 기존 기술 이름과 일치하는 경우 [추가] 버튼을 사용할 수 없습니다.
- 3 왼쪽의 목록에서 기술 이름을 선택합니다.
- 4 기술을 하나 이상의 DEM 작업자와 연결합니다.
  - a Distributed Execution Manager 옆의 **추가** 아이콘()을 클릭합니다.
  - b **DEM 선택** 대화상자에서 하나 이상의 DEM 작업자 인스턴스를 선택합니다.
  - c **확인**을 클릭합니다.
- 5 기술을 하나 이상의 워크플로와 연결합니다.
  - a 워크플로 옆의 **추가** 아이콘()을 클릭합니다.
  - b **워크플로 선택** 대화상자에서 하나 이상의 워크플로를 선택합니다.
  - c **확인**을 클릭합니다.

이 기술과 연결된 워크플로는 이 기술과 연결된 DEM 작업자만 실행할 수 있습니다.
- 6 기술을 추가하고 기술을 DEM 작업자 및 워크플로와 연결했으면 **확인**을 클릭하여 **기술 관리** 대화상자를 닫고 Model Manager에 대한 변경 내용을 저장합니다.


## 기술 및 DEM 작업자 사이의 연결 제거

기술과 DEM 작업자 사이의 연결을 제거하면 해당 작업자 인스턴스는 기술에 연결된 워크플로를 더 이상 실행할 수 없습니다.

### 필수 조건

vRealize Automation Designer 콘솔을 시작합니다.

### 프로시저

- 1 리본에서 **기술 관리**를 클릭합니다.
- 2 **기술 관리** 대화상자의 왼쪽 목록에서 기술 이름을 선택합니다.
- 3 Distributed Execution Manager 목록에서 DEM 작업자 인스턴스 이름을 하나 이상 선택하고 **제거** 아이콘()을 클릭합니다.
- 4 **확인**을 클릭하여 **기술 관리** 대화상자를 닫고 Model Manager에 변경 내용을 저장합니다.


## 기술 및 워크플로 사이의 연결 제거

기술과 워크플로 사이의 연결을 제거하면 해당 워크플로는 동일한 기술에 연결된 DEM 작업자에 더 이상 제한되지 않습니다.

### 필수 조건

vRealize Automation Designer 콘솔을 시작합니다.

### 프로시저

- 1 리본에서 **기술 관리**를 클릭합니다.
- 2 **기술 관리** 대화상자의 왼쪽 목록에서 기술 이름을 선택합니다.
- 3 워크플로 목록에서 워크플로 이름을 하나 이상 선택하고 **제거** 아이콘()을 클릭합니다.
- 4 **확인**을 클릭하여 **기술 관리** 대화상자를 닫고 Model Manager에 변경 내용을 저장합니다.

## 기술 제거

기술을 제거하면 DEM 작업자 및 워크플로에 대한 해당 기술의 모든 연결도 제거됩니다.

### 필수 조건

vRealize Automation Designer 콘솔을 시작합니다.

### 프로시저

- 1 리본에서 **기술 관리**를 클릭합니다.
- 2 **기술 관리** 대화상자의 왼쪽 목록에서 기술 이름을 선택합니다.

- 3 기술 목록의 맨 위쪽에서 **제거** 아이콘()을 클릭합니다.

기술 삭제를 확인하면 해당 기술은 삭제 대상으로 나타내기 위해 이름이 흐리게 표시됩니다.

- 4 **확인**을 클릭하여 **기술 관리** 대화상자를 닫고 Model Manager에 변경 내용을 저장하거나, 기술은 물론 해당 기술과 DEM 및 워크플로 사이의 연결을 삭제하지 않으려면 **취소**를 클릭합니다.

## CloudUtil 명령 참조

이 섹션은 CloudUtil 명령줄 인터페이스의 명령에 대한 참조를 제공합니다.

CloudUtil은 vRealize Automation Designer의 명령줄 인터페이스입니다. Designer를 실행 중인 Windows 시스템에서 명령을 실행합니다. Windows 시스템의 기본 설치 위치는 C:\Program Files (x86)\VMware\WvCAC\Design Center입니다.

**참고** CloudUtil 명령에서 Model Manager는 repository이고, DEM(Distributed Execution Manager)은 agent입니다.

## DEM 명령

DEM 명령을 사용하면 Model Manager에 등록된 Distributed Execution Manager의 목록을 보고 기술과 DEM 간 연결을 추가 또는 제거할 수 있습니다.

### DEM-Add-Skills

등록된 Distributed Execution Manager와 기술을 연결합니다.

#### 개요

```
CloudUtil.exe DEM-Add-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### DEM-Add-Skills 인수

인수	설명
-n   -name	등록된 Distributed Execution Manager의 이름입니다.
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-s   -skills	이 Distributed Execution Manager와 연결할 쉼표로 구분된 기술 목록입니다.
-v   -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

**참고** Model Manager에 기술이 이미 있어야 합니다. [Skill-Install](#)을 참조하십시오.

## DEM-List

Model Manager에 등록된 모든 Distributed Execution Manager 그리고 연결된 해당 기술을 나열합니다.

### 개요

```
CloudUtil.exe DEM-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

### DEM-List 인수

인수	설명
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## DEM-Remove-Skills

기술과 Distributed Execution Manager 간 연결을 제거합니다.

### 개요

```
CloudUtil.exe DEM-Remove-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

### DEM-Remove-Skills 인수

인수	설명
-n   - --name	등록된 Distributed Execution Manager의 이름입니다.
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-s   - --skills	이 Distributed Execution Manager에서 제거할 쉼표로 구분된 기술 목록입니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## File 명령

File 명령을 사용하면 Model Manager에서 파일(보통 스크립트)을 저장하고 관리할 수 있습니다.

## File-Export

Model Manager에서 파일을 내보냅니다.



## 개요

```
CloudUtil.exe File-Export -n|--name <Name> -o|--output <Output File> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Export 인수

인수	설명
-i   -iteration	(선택 사항) Model Manager에서 파일의 버전 문자열입니다. 기본값은 <b>0.0</b> 입니다.
-n   -name	Model Manager에서 파일의 친숙한 이름입니다.
-o   -output	파일 출력 경로입니다.
-repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## File-Import

Model Manager로 파일을 가져옵니다.

### 개요

```
CloudUtil.exe File-Import -n|--name <Name> -f|--filename <File Name> [-d|--description <Description>] [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Import 인수

인수	설명
-d   -description	(선택 사항) 파일에 대한 설명입니다.
-f   -filename	Model Manager로 가져올 파일의 경로입니다.
-i   -iteration	(선택 사항) Model Manager에서 파일의 버전 문자열입니다. 기본값은 <b>0.0</b> 입니다.
-n   -name	Model Manager에서 파일에 할당된 친숙한 이름입니다.
-repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## File-List

Model Manager로 가져온 모든 파일을 나열합니다.

### 개요

```
CloudUtil.exe File-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

## File-List 인수

인수	설명
- repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## File-Remove-All

특정 버전에 대한 모든 개정을 Model Manager에서 제거합니다.

### 개요

```
CloudUtil.exe File-Remove-All -n|--name <Name> [-i|--iteration <Iteration>]
[--repository <Model Manager Root URI>] [-v|--verbose]
```

## File-Remove-All 인수

표 1-20.

인수	설명
-i   - iteration	(선택 사항) Model Manager에서 파일의 버전 문자열입니다. 기본값은 0.0입니다.
-n   - name	Model Manager에서 파일의 친숙한 이름입니다.
- repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## File-Remove-Rev

파일의 특정 개정을 Model Manager에서 제거합니다.

### 개요

```
CloudUtil.exe File-Remove-Rev -n|--name <Name> -r|--revision <Revision> [-i|--iteration <Iteration>]
[--repository <Model Manager Root URI>] [-v|--verbose]
```

## File-Export 인수

인수	설명
-i   - iteration	(선택 사항) Model Manager에서 파일의 버전 문자열입니다. 기본값은 0.0입니다.
-n   - name	Model Manager에서 파일의 친숙한 이름입니다.
-r   - revision	제거할 파일의 개정입니다.

인수	설명
- repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## File-Rollback

Model Manager에서 파일을 지정된 개정으로 되돌립니다.

### 개요

```
CloudUtil.exe File-Rollback -n|--name <Name> -r|--revision <Revision> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Rollback 인수

표 1-21.

인수	설명
-i   - iteration	(선택 사항) Model Manager에서 파일의 버전 문자열입니다. 기본값은 0.0입니다.
-n   - name	Model Manager에서 파일의 친숙한 이름입니다.
-r   - revision	파일을 되돌릴 개정입니다.
- repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## File-Update

이전에 Model Manager로 가져온 파일을 새 개정으로 업데이트합니다.

### 개요

```
CloudUtil.exe File-Update -n|--name <Name> -f|--filename <File Name> [-i|--iteration <Iteration>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### File-Update 인수

인수	설명
-f   - filename	업데이트된 파일의 경로입니다.
-i   - iteration	(선택 사항) Model Manager에서 파일의 버전 문자열입니다. 기본값은 0.0입니다.
-n   - name	Model Manager에서 파일의 친숙한 이름입니다.

인수	설명
-repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Operation 명령

Operation 명령을 사용하면 Model Manager에서 사용자 지정 작업을 관리할 수 있습니다.

### Operation-Create

작업 정의 파일에 기반하여 시스템에서 수행할 수 있는 사용자 지정 작업 또는 작업 집합을 생성합니다.

#### 개요

```
CloudUtil.exe Operation-Create -c|--operationConfig <Operation Definition File> [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Operation-Create 인수

인수	설명
-c   -operationConfig	작업 정의 파일(XML)의 경로입니다.
-repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

### Operation-Delete

Model Manager에서 사용자 지정 작업을 삭제합니다.

#### 개요

```
CloudUtil.exe Operation-Delete -n|--name <Name> [--force] [--repository <Model Manager Root URI>] [-v|--verbose]
```

#### Operation-Delete 인수

인수	설명
-force	(선택 사항) 작업을 강제로 삭제합니다.
-n   -name	Model Manager에서 사용자 지정 작업의 이름입니다.

인수	설명
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Operation-List

Model Manager의 사용자 지정 작업을 모두 나열합니다.

### 개요

```
CloudUtil.exe Operation-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

### Operation-List 인수

인수	설명
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Skill 명령

Skill 명령을 사용하면 Distributed Execution Manager 및 워크플로에 연결된 기술을 관리할 수 있습니다.

## Skill-Install

Model Manager에 기술을 설치합니다.

### 개요

```
CloudUtil.exe Skill-Install -n|--name <Name> [--repository <Model Manager Root URI>] [-v|--verbose]
```

### Skill-Install 인수

인수	설명
-n   - --name	Model Manager에서 기술의 이름입니다.
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Skill-List

Model Manager에 설치된 기술을 모두 나열합니다.

### 개요

```
CloudUtil.exe Skill-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

### Skill-List 인수

인수	설명
--repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Skill-Uninstall

기술을 Model Manager에서 제거합니다.

### 개요

```
CloudUtil.exe Skill-Uninstall -n|--name <Name> [--repository <Model Manager Root URI>] [-v|--verbose]
```

### Skill-Uninstall 인수

인수	설명
-n   - -name	Model Manager에서 제거할 기술의 이름입니다.
--repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

**참고** Distributed Execution Manager 또는 워크플로에 연결되어 있는 기술은 제거할 수 없습니다. [DEM-Remove-Skills](#) 또는 [Workflow-Remove-Skills](#)를 참조하십시오.

## Workflow 명령

Workflow 명령을 사용하면 Model Manager에서 사용자 지정 가능 IaaS 워크플로는 물론 모든 워크플로에 연결된 기술을 관리할 수 있습니다.

## Workflow-Add-Skills

Model Manager에서 워크플로에 기술을 연결합니다.

```
CloudUtil.exe Workflow-Add-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

**표 1-22. Workflow-Add-Skills 인수**

인수	설명
Name	Model Manager에서 워크플로의 이름입니다.
Skills	이 워크플로에 연결할 기술의 쉼표로 구분된 목록입니다.
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

**참고** Model Manager에 기술이 이미 있어야 합니다. [Skill-Install](#)을 참조하십시오.

## Workflow-List

Model Manager에 설치되어 있는 워크플로 및 해당하는 연결된 기술을 모두 나열합니다.

```
CloudUtil.exe Workflow-List [--repository <Model Manager Root URI>] [-v|--verbose]
```

**표 1-23. Workflow-List 인수**

인수	설명
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Workflow-Remove-Skills

Model Manager에서 기술과 워크플로 사이의 연결을 제거합니다.

### 개요

```
CloudUtil.exe Workflow-Remove-Skills -n|--name <Name> -s|--skills <Skills> [--repository <Model Manager Root URI>] [-v|--verbose]
```

## Workflow-Remove-Skills 인수

인수	설명
-n   - -name	Model Manager에서 워크플로의 이름입니다.
- -repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-s   - -skills	이 워크플로에서 제거할 기술의 쉼표로 구분된 목록입니다.
-v   - -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Workflow-Rollback

워크플로를 지정된 개정으로 되돌립니다.

### 개요

```
CloudUtil.exe Workflow-Rollback -n|--name <Name> -r|--revision <Revision> [--repository <Model Manager Root URI>]
[-v|--verbose]
```

## Workflow-Rollback 인수

인수	설명
-n   - -name	Model Manager에서 워크플로의 이름입니다.
- -repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-r   - -revision	워크플로를 되돌릴 개정입니다.
-v   - -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Workflow-Update

사용자 지정 가능한 워크플로를 새 개정으로 업데이트합니다.

```
CloudUtil.exe Workflow-Update -f|--filename <File Name> -n|--name <Name> [-d|--description <Description>] [--repository <Model Manager Root URI>] [-v|--verbose]
```

### 표 1-24. Workflow-Update 인수

인수	설명
File Name	업데이트된 워크플로가 포함된 파일(XAML)의 경로입니다.
Name	업데이트할 워크플로의 이름입니다.
Description	(선택 사항) 워크플로에 대한 설명입니다.



표 1-24. Workflow-Update 인수 (계속)

인수	설명
- repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-v   -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## 가져오기 명령

가져오기 명령을 사용하면 하나 이상의 가상 시스템을 vRealize Automation 배포로 가져올 수 있습니다.

## Machine-BulkRegisterExport

가상 시스템을 vRealize Automation 배포에 가져오는 데 사용되는 CSV 데이터 파일을 생성합니다.

### 개요

```
CloudUtil.exe Machine-BulkRegisterExport [-b|--blueprint] [-m|--managed] [-e|--exportNames] [-p|--properties] -f|--filename <Value> [-g|--group <Value>] [-i|--ignore] [-o|--owner <Value>] [-t|--machinetype <Value>] [-n|--resourceName <Value>] [-r|--resourceType <Value>] [--repository <Value>] [-sn|--sourcename <Value>] [-st|--sourcetype <Value>] -u|--user <value> [-v|--verbose]
```

## Machine-BulkRegisterExport 인수

표 1-25.

인수	설명
-b   -blueprint	(선택 사항) Blueprint 이름을 포함합니다.
-e   -exportNames	(선택 사항) GUID 대신 이름을 내보냅니다.
-f   -filename	시스템 이름 목록이 포함된 CSV 데이터 파일의 이름을 지정합니다(예: filename.csv). 파일은 기본적으로 현재 경로에 저장됩니다. 원하는 디렉토리의 전체 경로를 지정할 수도 있습니다.
-g   -group	(선택 사항) 비즈니스 그룹 이름을 지정합니다(예: 엔지니어링).
-i   -ignore	(선택 사항) 잘못된 인수를 무시합니다.
-m   -managed	(선택 사항) 관리되는 가상 시스템을 내보냅니다. 기본값은 관리되지 않는 가상 시스템을 내보내는 것입니다.
-n   -resourceName	(선택 사항) 리소스 이름별로 필터링하려면 계산 리소스 또는 끝점의 이름을 지정합니다.
-o   -owner	(선택 사항) 가져온 가상 시스템의 소유자를 지정합니다(예: jsmith).
-p   -properties	(선택 사항) 관리되는 가상 시스템의 속성을 내보냅니다.

표 1-25. (계속)

인수	설명
-r   - --resourceType	(선택 사항) 리소스 유형별로 필터링하려면 계산 리소스인 경우 1을 지정하고 끝점인 경우 2를 지정합니다.
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.
-sn   - --sourcename	(선택 사항) 클러스터 또는 끝점의 이름을 지정합니다.
-st   - --sourcetype	(선택 사항) 소스 유형을 클러스터 또는 끝점으로 지정합니다.
-t   - --machinetype	(선택 사항) 내보낼 시스템 유형을 지정합니다(예: 가상, 물리적, 클라우드, AppService, vApp).
-u   - --user	대량 등록을 수행하는 패브릭 관리자를 지정합니다.
-v   - --verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.

## Machine-BulkRegisterImport

하나 이상의 가상 시스템을 대상 vRealize Automation 배포에 가져옵니다.

### 개요

```
CloudUtil.exe Machine-BulkRegisterImport [-b|--batch][--delay <value>] -f|--filename <value> [-i|--ignore] [-h|--humanreadable] -n|--name <value> [--repository <value>] [-s|--skipUser] -t|--time <value> -u|--user <value> [-v|--verbose] [-w|--whatIf]
```

### Machine-BulkRegisterImport 인수

표 1-26.

인수	설명
-b   - --batch	(선택 사항) 배치 크기.
-d   - --delay	(선택 사항) 처리 지연 시간을 hh:mm:ss 형식으로 지정합니다(예: 02:20:10).
-f   - --filename	시스템 이름 목록이 포함되어 있는 CSV 데이터 파일의 이름을 지정합니다. 예를 들면 filename.csv와 같습니다.
-h   --humanreadable	(선택 사항) 입력 파일에는 GUID가 아니라 가상 시스템 이름이 포함됩니다.
-i   - --ignore	(선택 사항) 등록되거나 관리되는 가상 시스템을 무시합니다.
-n   - --name	대상 vRealize Automation으로의 가져오기를 수행하는 작업 대기열의 이름을 지정합니다.
- --repository	(선택 사항) Model Manager의 루트 URI입니다(예: http://hostname/repository). 기본값은 CloudUtil 구성 파일의 <appSettings> 섹션 아래에 있는 repositoryAddress 키에 지정됩니다.

표 1-26. (계속)

인수	설명
-s   - -skipUser	(선택 사항) 사용자가 존재하는지 여부를 확인하지 않고 CSV 데이터 파일의 [소유자] 열에 나열된 값을 시스템의 소유자로 설정합니다. 이 옵션을 선택하면 가져오기에 필요한 시간을 단축할 수 있습니다.
-t   - -time	워크플로 시작 시간을 MM/DD/YYYY hh:mm GMT 형식으로 지정합니다(예: 04/18/2014 10:01 GMT). 지정된 시작 시간은 사용자 Workstation의 로컬 시간이 아니라 서버의 로컬 시간인 것으로 간주됩니다.
-u   - -user	대량 등록을 수행하는 패브릭 관리자를 지정합니다.
-v   - -verbose	(선택 사항) 오류가 발생하면 예외 메시지 대신 스택 추적을 출력합니다.
whatif	(선택 사항) CSV 파일을 검증하지만 가상 시스템은 가져오지 않으려면 설정합니다.

## vRealize Automation 워크플로 작업 참조

VMware에서는 워크플로를 사용자 지정할 때 사용할 수 있는 워크플로 작업 라이브러리를 vRealize Automation Designer에 제공합니다.

**참고** CDK는 vRealize Automation 7.0부터 사용되지 않습니다. vRealize Orchestrator 워크플로를 사용하여 이전에 CDK로 해결했던 사용 사례를 해결할 수 있습니다.

vRealize Automation Designer에는 다섯 가지 Windows Workflow Foundation 작업 범주(제어 흐름, 순서도, 기본 형식, 컬렉션 및 오류 처리)도 포함되어 있습니다.

이 섹션에서는 vRealize Automation Designer의 DynamicOps.Repository.Activities 및 DynamicOps.Cdk.Activities 네임스페이스에 포함된 IaaS 워크플로 작업에 대한 참조를 제공합니다. vRealize Orchestrator 워크플로 호출과 관련된 작업은 [vRealize Orchestrator 워크플로 작업 사용](#)에 설명되어 있습니다.

**참고** IaaS 작업 라이브러리에서 Model Manager는 repository라고 합니다.

## DynamicOps.Repository.Activities

DynamicOps.Repository.Activities 네임스페이스에는 IaaS 워크플로를 위한 기본 워크플로 작업이 포함되어 있습니다.

**참고** CDK는 vRealize Automation 7.0부터 사용되지 않습니다. vRealize Orchestrator 워크플로를 사용하여 이전에 CDK로 해결했던 사용 사례를 해결할 수 있습니다.

## AddLink

지정된 링크를 DataServiceContext에서 추적 중인 개체 집합에 추가합니다.

**표 1-27. AddLink 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	링크를 추가할 DataServiceContext입니다.
Source	Object	새 링크에 대한 소스 개체입니다.
SourceProperty	String	관련 개체를 반환하는 소스 개체에 대한 탐색 속성의 이름입니다.
Target	Object	새 링크에 의해 소스 개체와 관련된 개체입니다.

## AddObject

지정된 개체를 DataServiceContext에서 추적 중인 개체 집합에 추가합니다.

**표 1-28. AddObject 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	개체를 추가할 DataServiceContext입니다.
Instance	Object	DataServiceContext에서 추적할 개체입니다.

## AttachTo

지정된 리소스에 대한 추적을 시작하도록 DataServiceContext에 알립니다.

**표 1-29. AttachTo 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	리소스를 추적해야 하는 DataServiceContext입니다.
Instance	Object	DataServiceContext에서 추적할 리소스입니다. 리소스는 [변경되지 않음] 상태에서 연결되어 있습니다.

## CreateRepositoryServiceContext<T>

Model Manager에 로드된 모델에 대해 컨텍스트를 생성합니다.

vRealize Automation Designer에서 이 작업을 워크플로에 추가하는 경우 RepositoryServiceContext 클래스에서 상속하는 클래스를 선택해야 합니다.

**표 1-30. CreateRepositoryServiceContext<T> 작업 입력 매개 변수**

인수	유형	설명
Uri	URI	(선택 사항) 모델에 연결할 때 사용할 루트 URI입니다.
Username	String	(선택 사항) 컨텍스트에 연결할 때 사용할 사용자 이름입니다.

**표 1-31. CreateRepositoryServiceContext<T> 작업 출력 매개 변수**

인수	유형	설명
Result	RepositoryServiceContext	반환된 특정 유형은 작업이 워크플로에 추가될 때 선택된 클래스의 인스턴스입니다.

## DeleteLink

링크의 상태를 DataServiceContext에서 추적할 링크의 목록에서 삭제됨으로 변경합니다.

**표 1-32. DeleteLink 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	링크를 삭제할 DataServiceContext입니다.
Source	Object	삭제로 표시될 링크의 소스 개체입니다.
SourceProperty	String	대상 개체에 액세스하는 데 사용되는 소스 개체에 대한 탐색 속성의 이름입니다.
Target	Object	소스 개체에 바인딩된 링크와 관련된 대상 개체입니다. 대상 개체는 소스 속성 또는 하위 유형에 의해 식별되는 유형이어야 합니다.

## DeleteObject

지정된 개체의 상태를 DataServiceContext에서 삭제됨으로 변경합니다.

**표 1-33. DeleteObject 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	리소스를 삭제할 DataServiceContext입니다.
Instance	Object	삭제됨 상태로 변경될 추적된 엔티티입니다.

## InvokeRepositoryWorkflow

Model Manager에 설치된 워크플로를 실행합니다.

**표 1-34. InvokeRepositoryWorkflow 작업 입력 매개 변수**

인수	유형	설명
WorkflowType	WorkflowDefinition 엔티티	실행할 워크플로입니다.
WorkflowInputs	Dictionary<string, object>	(선택 사항) 워크플로에 대한 입력입니다.
CallingInstance	WorkflowInstance 엔티티	(선택 사항) 실행된 워크플로를 호출하고 값을 반환할 워크플로입니다.

## LoadProperty

지정된 속성에 대해 지연된 콘텐츠를 데이터 서비스에서 로드합니다.

**표 1-35. LoadProperty 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	로드할 속성이 있는 DataServiceContext입니다.
Instance	Object	로드할 속성이 포함된 엔티티입니다.
InstanceProperty	String	로드하려는 지정된 엔티티의 속성 이름입니다.

## SaveChanges

DataServiceContext에서 추적하는 변경 내용을 스토리지에 저장합니다.

**표 1-36. SaveChanges 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	저장할 변경 내용을 추적하는 DataServiceContext입니다.

## SetLink

지정된 개체 사이에 새 링크가 있으며 SourceProperty 인수에 지정된 속성이 해당 링크를 나타낸다는 정보를 DataServiceContext에 알립니다.

**표 1-37. SetLink 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	링크에 대해 알릴 DataServiceContext입니다.
Source	Object	새 링크에 대한 소스 개체입니다.
SourceProperty	String	새 링크의 대상 개체를 식별하는 소스 개체의 속성입니다.
Target	Object	이 메서드를 호출하여 초기화할 새 링크에 관련된 하위 개체입니다. 대상 개체는 SourceProperty로 식별되는 유형의 하위 유형이어야 합니다. Target을 null로 설정하면 호출이 링크 삭제 작업을 나타냅니다.

## UpdateObject

DataServiceContext에 있는 지정된 개체의 상태를 수정됨으로 변경합니다.

**표 1-38. UpdateObject 작업 입력 매개 변수**

인수	유형	설명
DataServiceContext	RepositoryServiceContext	업데이트할 엔티티를 추적하는 DataServiceContext입니다.
Instance	Object	수정된 상태로 할당할 추적된 엔티티입니다.

## DynamicOps.Cdk.Activities

DynamicOps.Cdk.Activities 네임스페이스에는 IaaS 워크플로를 위한 고급 작업이 포함되어 있습니다.

**참고** CDK는 vRealize Automation 7.0부터 사용되지 않습니다. vRealize Orchestrator 워크플로를 사용하여 이전에 CDK로 해결했던 사용 사례를 해결할 수 있습니다.

## ExecutePowerShellScript

Model Manager에 지정된 이름으로 저장된 PowerShell 스크립트를 실행합니다.

ExecutePowerShellScript 작업을 사용하기 전에 우선 CloudUtil File-Import 명령을 사용하여 Model Manager에서 실행하려는 스크립트를 로드해야 합니다.

**표 1-39. ExecutePowerShellScript 작업 입력 매개 변수**

인수	유형	설명
ScriptName	String	Model Manager에서 실행할 스크립트의 이름입니다.
ScriptVersion	Object	(선택 사항) Model Manager에서 실행할 스크립트의 버전입니다. 기본값은 0.0입니다.
Machineld	Guid	(선택 사항) 지정하는 경우 시스템이 로드되고 모든 해당 속성이 스크립트에 전달됩니다.
Arguments	Dictionary<string,string>	스크립트에 전달할 추가 인수입니다. Machineld가 지정되었을 때 인수와 동일한 이름의 시스템 속성이 있는 경우 (대/소문자를 구분하지 않음) 시스템 속성 값이 인수 값을 재정의합니다.
PSModules	IEnumerable<string>	(선택 사항) 명령 실행 중 PowerShell 런타임으로 로드된 모듈입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.

**표 1-40. ExecutePowerShellScript 작업 출력 매개 변수**

인수	유형	설명
Output	Collection<PSObject>	스크립트의 출력입니다(있는 경우). 오류 시 예외가 발생합니다.

ExecutePowerShellScript의 출력을 처리할 때 vRealize Automation Designer 콘솔에 오류 메시지가 PSObject 유형이 정의되지 않았습니다.가 표시되면 다음 단계를 수행합니다.

- 1 [Designer] 창의 왼쪽 아래에서 **가져오기**를 클릭합니다.
- 2 **System.Management.Automation** 어셈블리를 선택합니다.

## ExecuteSshScript

Model Manager에 지정된 이름으로 저장된 SSH 스크립트를 실행합니다.

ExecuteSshScript 작업을 사용하기 전에 우선 CloudUtil File-Import 명령을 사용하여 Model Manager에서 실행하려는 스크립트를 로드해야 합니다.

**표 1-41. ExecuteSshScript 작업 입력 매개 변수**

인수	유형	설명
ScriptName	String	Model Manager에서 실행할 스크립트의 이름입니다.
Host	String	스크립트를 실행할 서버 이름입니다.
Username	String	호스트에 연결할 때 사용할 사용자 이름입니다.
Password	String	호스트에 연결할 때 사용할 암호입니다.
ScriptVersion	Object	(선택 사항) Model Manager에서 실행할 스크립트의 버전입니다. 기본값은 0.0입니다.
Timeout	TimeSpan	(선택 사항) 스크립트 실행 시간 제한이 발생하는 시간입니다. 기본값은 30분입니다.

**표 1-42. ExecuteSshScript 작업 출력 매개 변수**

인수	유형	설명
EnvironmentVariables	Dictionary<string, string>	스크립트 실행 결과입니다(있는 경우).

## GetMachineName

시스템의 이름을 가져옵니다.

**표 1-43. GetMachineName 작업 입력 매개 변수**

인수	유형	설명
MachineId	Guid	이름을 검색할 시스템입니다.

**표 1-44. GetMachineName 작업 출력 매개 변수**

인수	유형	설명
MachineName	String	MachineId로 식별되는 시스템의 이름입니다.

## GetMachineOwner

시스템 소유자의 사용자 이름을 가져옵니다.

**표 1-45. GetMachineOwner 작업 입력 매개 변수**

인수	유형	설명
MachineId	Guid	소유자를 검색할 시스템입니다.



**표 1-46. GetMachineOwner 작업 출력 매개 변수**

인수	유형	설명
Owner	String	MachinelId로 식별되는 시스템의 소유자로, 소유자가 없는 경우에는 Null입니다.

## GetMachineProperties

시스템과 연결된 사용자 지정 속성의 목록을 가져옵니다.

**표 1-47. GetMachineProperties 작업 입력 매개 변수**

인수	유형	설명
MachinelId	Guid	속성을 검색할 시스템입니다.

**표 1-48. GetMachineProperties 작업 출력 매개 변수**

인수	유형	설명
Properties	Dictionary<string, string>	시스템의 속성 목록입니다. 값이 암호화되어 저장된 경우 암호가 해독된 상태로 반환됩니다.

## GetMachineProperty

시스템에 대한 지정된 속성 값을 가져옵니다.

**표 1-49. GetMachineProperty 작업 입력 매개 변수**

인수	유형	설명
MachinelId	Guid	속성을 검색할 시스템입니다.
PropertyName	String	값을 반환할 속성의 이름입니다.
IsRequired	bool	속성이 필요하지만 찾을 수 없는 경우 작업에서 예외가 발생하고 그렇지 않으면 Null을 반환합니다.

**표 1-50. GetMachineProperty 작업 출력 매개 변수**

인수	유형	설명
PropertyValue	String	PropertyName으로 지정된 속성의 값입니다. 값이 암호화되어 저장된 경우 암호가 해독된 상태로 반환됩니다.

## GetScriptFromName

Model Manager에 지정된 이름으로 저장된 스크립트의 콘텐츠를 가져옵니다.

**표 1-51. GetScriptFromName 작업 입력 매개 변수**

인수	유형	설명
ScriptName	String	Model Manager에서 검색할 스크립트의 이름입니다.
ScriptVersion	Object	(선택 사항) Model Manager에서 검색할 스크립트의 버전입니다. 기본값은 0.0입니다.

**표 1-52. GetScriptFromName 작업 출력 매개 변수**

인수	유형	설명
ScriptContent	String	ScriptName으로 식별되는 스크립트의 콘텐츠입니다.

## InvokePowerShell

PowerShell 명령을 실행합니다.

**표 1-53. InvokePowerShell 작업 입력 매개 변수**

인수	유형	설명
CommandText	String	실행할 명령입니다.
Arguments	IEnumerable<string>	(선택 사항) 명령의 인수입니다.
Input	IEnumerable	(선택 사항) 입력 파이프라인입니다.
IsScript	bool	(선택 사항) CommandText가 스크립트인지 여부를 나타냅니다. 기본값은 False입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
Parameters	Collection	(선택 사항) PowerShell 스크립트에 매개 변수로 전달된 이름-값 쌍의 모음입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
PowerShellVariables	Collection	(선택 사항) PowerShell 런타임에 복사되는 변수입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
PSModules	IEnumerable<string>	(선택 사항) 명령 실행 중 PowerShell 런타임으로 로드된 모듈입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
Runspace	Runspace	(선택 사항) PowerShell runspace를 생성하여 이 인수에 제공하면 동일한 runspace를 여러 PowerShell 호출에 재 사용하여 성능을 개선할 수 있습니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.

**표 1-54. InvokePowerShell 작업 출력 매개 변수**

인수	유형	설명
Output	Collection<PSObject>	명령의 출력입니다(있는 경우). 오류 시 예외가 발생합니다.
Errors	Collection<ErrorRecord>	실행으로 인해 발생하는 오류입니다(있는 경우).

ExecutePowerShellScript의 출력을 처리할 때 vRealize Automation Designer 콘솔에 오류 메시지가 PSObject 유형이 정의되지 않았습니다.가 표시되면 다음 단계를 수행합니다.

- 1 [Designer] 창의 왼쪽 아래에서 **가져오기**를 클릭합니다.
- 2 **System.Management.Automation** 어셈블리를 선택합니다.

## InvokeSshCommand

SSH 명령을 실행합니다.

**표 1-55. InvokeSshCommand 작업 입력 매개 변수**

인수	유형	설명
CommandText	String	실행할 명령입니다.
Host	String	명령을 실행할 대상 서버 이름입니다.
Username	String	호스트에 연결할 때 사용할 사용자 이름입니다.
Password	String	호스트에 연결할 때 사용할 암호입니다.
Timeout	TimeSpan	(선택 사항) 명령 실행 시간이 초과되기 전의 기간입니다. 기본값은 30분입니다.

**표 1-56. InvokeSshCommand 작업 출력 매개 변수**

인수	유형	설명
EnvironmentVariables	Dictionary<string, string>	명령의 출력입니다(있는 경우). 오류 시 예외가 발생합니다.

## LogMachineEvent

시스템 소유자가 볼 수 있는 사용자 로그에 시스템 이벤트를 기록합니다.

**표 1-57. LogMachineEvent 작업 입력 매개 변수**

인수	유형	설명
Machinelid	Guid	기록할 이벤트를 생성하는 시스템입니다.
Message	String	사용자 로그에 쓸 메시지입니다.
Type	String	드롭다운 목록에서 메시지 유형을 선택합니다(정보, 경고, 오류).

## LogMessage

Distributed Execution Manager 로그에 기록합니다.

**표 1-58. LogMessage 작업 입력 매개 변수**

인수	유형	설명
Message	String	DEM 로그에 쓸 메시지입니다.
MessageCategory	String	드롭다운 메뉴에서 범주( <b>디버그</b> , <b>오류</b> , <b>정보</b> , <b>추적</b> )를 선택하거나, 사용자 지정 범주를 입력합니다.
MessageSeverity	String	드롭다운 메뉴에서 심각도를 선택합니다 (System.Diagnostics.TraceEventType에 제공된 심각도 목록에 바인딩됨).

## RunProcess

이 작업을 실행하는 DEM과 동일한 시스템에서 프로세스를 실행합니다.

**참고** vRealize Automation은 RunProcess 작업에 의해 시작된 프로세스에서 UI를 사용자에게 표시할 수 없기 때문에 이러한 프로세스는 비대화형 모드여야 합니다. DEM 시스템에 분리된 프로세스가 남아 있지 않도록 프로세스는 자체 종료 기능을 갖추고 있어야 합니다.

**표 1-59. RunProcess 작업 입력 매개 변수**

인수	유형	설명
Command	String	DEM 시스템에서 실행할 실행 파일의 경로입니다.
WorkingDirectory	String	(선택 사항) 프로세스가 실행되어야 하는 작업 디렉토리입니다.
Arguments	String	(선택 사항) 명령에 전달할 명령줄 인수의 목록입니다.
WaitForExit	bool	(선택 사항) true인 경우 워크플로는 프로세스가 완료될 때까지 기다렸다가 워크플로를 계속합니다. 기본값은 false입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.

## SendEmail

지정된 주소 집합에 이메일을 보냅니다.

**표 1-60. SendEmail 작업 입력 매개 변수**

인수	유형	설명
To	IEnumerable<string>	이메일을 보낼 주소 목록입니다.
From	String	이메일의 "보낸 사람" 필드를 채우는 데 사용할 주소입니다.
Subject	String	이메일의 제목줄입니다.
Body	String	이메일의 본문 텍스트입니다.
Host	String	송신 SMTP 서버의 호스트 이름 또는 IP 주소입니다.
Port	Integer	호스트에 지정된 서버의 SMTP 포트입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
CC	IEnumerable<string>	(선택 사항) 이메일의 참조 필드에 복사할 주소 또는 주소 목록입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
Bcc	IEnumerable<string>	(선택 사항) 이메일의 숨은 참조 필드에 복사할 주소 또는 주소 목록입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.

표 1-60. SendEmail 작업 입력 매개 변수 (계속)

인수	유형	설명
EnableSsl	bool	(선택 사항) SSL을 사용할지 여부를 나타냅니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
UserName	String	Host에 지정된 SMTP 서버에서 인증하는 데 사용할 사용자 이름입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.
Password	String	UserName에 지정된 사용자의 암호입니다. 이 옵션은 [속성] 창에서만 사용할 수 있습니다. [Designer] 창에서는 사용할 수 없습니다.

## SetMachineProperty

시스템에 사용자 지정 속성을 생성하거나 업데이트합니다.

표 1-61. SetMachineProperty 작업 입력 매개 변수

인수	유형	설명
MachineId	Guid	사용자 지정 속성을 생성하거나 업데이트할 시스템입니다.
PropertyName	String	생성하거나 업데이트할 속성의 이름입니다.
PropertyValue	String	속성을 생성하거나 업데이트하는 데 사용할 값입니다.
IsEncrypted	bool	(선택 사항) 속성의 값이 암호화되었는지 여부를 나타냅니다.
IsHidden	bool	(선택 사항) 속성이 숨겨진 속성인지 여부를 나타냅니다.
IsRuntime	bool	(선택 사항) 요청하는 사용자가 요청 시 속성 값을 제공하는지 여부를 나타냅니다(vRealize Automation 콘솔에서 [사용자에게 확인]으로 표시하는 것과 같음).

## SetWorkflowResult

외부 워크플로의 상태를 완료 또는 실패로 설정합니다. 이 값은 ExternalWF.xml 설정에 적용됩니다.

표 1-62. SetWorkflowResult 작업 입력 매개 변수

인수	유형	설명
WorkflowId	Guid	상태를 설정할 워크플로입니다.
Next State	WorkflowState	드롭다운 메뉴에서 <b>완료</b> 또는 <b>실패</b> 를 선택합니다.