

Erweiterbarkeit des Lebenszyklus

vRealize Automation 7.3



vmware®

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

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

Falls Sie Anmerkungen zu dieser Dokumentation haben, senden Sie Ihre Kommentare und Vorschläge an:

docfeedback@vmware.com

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

VMware Global, Inc.
Zweigniederlassung Deutschland
Freisinger Str. 3
85716 Unterschleißheim/Lohhof
Germany
Tel.: +49 (0) 89 3706 17000
Fax: +49 (0) 89 3706 17333
www.vmware.com/de

Inhalt

| | | |
|----------|--|----------|
| 1 | Erweiterbarkeit des Lebenszyklus | 4 |
| | Übersicht über die Erweiterbarkeit von Maschinen | 4 |
| | Erweiterbarkeit des Maschinenlebenszyklus | 5 |
| | Auswählen eines Szenarios für die Lebenszyklus-Erweiterbarkeit | 6 |
| | Erweitern der Maschinenlebenszyklen mithilfe von vRealize Orchestrator | 7 |
| | Erweitern der Maschinenlebenszyklen mithilfe der vRealize Orchestrator -Checkliste | 7 |
| | Konfigurieren des vRealize Automation -Plug-Ins für die Erweiterbarkeit von Maschinen | 8 |
| | Anpassen von IaaS-Workflows mithilfe von vRealize Orchestrator | 13 |
| | Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation | 14 |
| | Mit vRealize Automation bereitgestellte Ereignisthemen | 14 |
| | Terminologie zu Workflow-Abonnements und zum Ereignisbrokerdienst | 16 |
| | Blockierbare und antwortfähige Ereignisthemen | 17 |
| | Best Practices für das Erstellen von vRealize Orchestrator -Workflows für Workflow-Abonnements | 19 |
| | Einstellungen für das Workflow-Abonnement | 19 |
| | Arbeiten mit Bereitstellungs- und Lebenszyklus-Workflow-Abonnements | 24 |
| | Arbeiten mit Genehmigungs-Workflow-Abonnements | 44 |
| | Fehlerbehebung von Workflow-Abonnements | 51 |
| | Erweitern der Maschinenlebenszyklen mithilfe von vRealize Automation Designer | 54 |
| | Erweitern der Maschinenlebenszyklen mithilfe der vRealize Automation Designer -Checkliste | 54 |
| | Installieren und Konfigurieren von vRealize Automation Designer | 55 |
| | Anpassen von IaaS-Workflows mithilfe von vRealize Automation Designer | 59 |
| | Workflows und Distributed Execution Management | 77 |
| | Zuordnen von Workflows und DEM-Workern mithilfe von Qualifikationen | 77 |
| | Entfernen von Zuordnungen zwischen Qualifikationen und DEM-Workern | 78 |
| | Entfernen von Zuordnungen zwischen Qualifikationen und Workflows | 78 |
| | Entfernen einer Qualifikation | 79 |
| | Befehlsreferenz für CloudUtil | 79 |
| | DEM-Befehle | 79 |
| | File-Befehle | 81 |
| | Operation-Befehle | 84 |
| | Skill-Befehle | 86 |
| | Workflow-Befehle | 87 |
| | Importbefehle | 90 |
| | vRealize Automation -Workflowaktivitätsreferenz | 92 |
| | DynamicOps.Repository.Activities | 93 |
| | DynamicOps.Cdk.Activities | 96 |

Erweiterbarkeit des Lebenszyklus

1

Wenn Sie vRealize Orchestrator mit vRealize Automation verwenden, können Sie die Verwaltung des Lebenszyklus von IaaS-Maschinen erweitern.

Für die Erweiterung von vRealize Automation ist es erforderlich, bereitgestellte vRealize Orchestrator-Workflows zu verwenden und benutzerdefinierte Workflows zu erstellen.

Dieses Kapitel behandelt die folgenden Themen:

- [Übersicht über die Erweiterbarkeit von Maschinen](#)
- [Erweitern der Maschinenlebenszyklen mithilfe von vRealize Orchestrator](#)
- [Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation](#)
- [Erweitern der Maschinenlebenszyklen mithilfe von vRealize Automation Designer](#)
- [Workflows und Distributed Execution Management](#)
- [Befehlsreferenz für CloudUtil](#)
- [vRealize Automation-Workflowaktivitätsreferenz](#)

Übersicht über die Erweiterbarkeit von Maschinen

Für die Bereitstellung oder Außerbetriebnahme einer neuen Maschine ist insbesondere bei unternehmenskritischen Systemen die Interaktion mit einer Reihe von Managementsystemen erforderlich. Beispielsweise mit DNS-Servern, Lastausgleichsdiensten, Konfigurationsverwaltungsdatenbanken (Configuration Management Database, CMDB), IP-Adressen-Verwaltungssystemen usw.

Erweiterbarkeit des Maschinenlebenszyklus

In verschiedenen vordefinierten Phasen des IaaS-Lebenszyklus können Sie benutzerdefinierte Logik mithilfe von IaaS-Statusänderungsworkflows, sogenannten Workflow-Stubs, hinzufügen. Mit den Workflow-Stubs können Sie vRealize Orchestrator für die bidirektionale Integration von externen Managementsystemen aufrufen.

Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe [Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation](#).

Durch Erstellen eines Statusänderungsworkflows können Sie die Ausführung eines Workflows auslösen, bevor der IaaS-Masterworkflow in einen bestimmten Status wechselt. Beispielsweise können Sie benutzerdefinierte Workflows für die Integration in eine externe Datenbank und zum Aufzeichnen von Informationen in verschiedenen Phasen des Maschinenlebenszyklus erstellen.

- Erstellen Sie einen benutzerdefinierten Workflow, der ausgeführt wird, bevor der Masterworkflow in den Status „MachineProvisioned“ wechselt, um Informationen wie den Maschinenbesitzer, Genehmiger usw. aufzuzeichnen.
- Erstellen Sie einen benutzerdefinierten Workflow, der ausgeführt wird, bevor eine Maschine in den Status „MachineDisposing“ wechselt, um den Zeitpunkt, zu dem die Maschine gelöscht wurde, sowie Daten wie etwa die Ressourcennutzung bei der letzten Datenerfassung, der letzten Anmeldung usw. aufzuzeichnen.

In den Abbildungen für den Masterworkflow sind die Hauptstatus des Masterworkflows dargestellt, wobei die Status gelb markiert sind, die Sie mithilfe von IaaS-Workflow-Stubs anpassen können. In der Tabelle „Anpassbare Statusänderungsworkflows“ finden Sie die verfügbaren Workflow-Stubs, deren Position im Masterworkflowstatus sowie Beispiele für benutzerdefinierte Logik, die Sie in jedem Status zum Erweitern des Maschinenlebenszyklus verwenden könnten.

Abbildung 1-1. Masterworkflowstatus für die Bereitstellung von Maschinen



Abbildung 1-2. Masterworkflowstatus für den Import von Maschinen

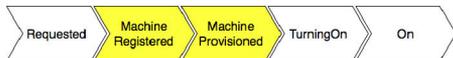


Abbildung 1-3. Masterworkflowstatus für den Ablauf von Maschinen-Leases



Abbildung 1-4. Masterworkflowstatus für das Löschen von Maschinen

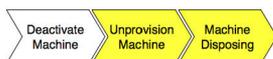


Tabelle 1-1. Anpassbare Statusänderungsworkflows

| Masterworkflowstatus | Anpassbarer Workflowname | Beispiele für die Erweiterbarkeit |
|----------------------|--------------------------|--|
| BuildingMachine | WFStubBuildingMachine | Bereiten Sie das Erstellen der Maschine auf dem Hypervisor vor. Erstellen Sie einen Datensatz für die Konfigurationsverwaltungsdatenbank (Configuration Management Database, CMDB), rufen Sie ein externes System zum Zuweisen einer IP-Adresse zu einer Maschine auf und geben Sie dann beim Löschen der Maschine mithilfe von WFStubMachineDisposing die IP-Adresse an den Pool zurück. |
| RegisterMachine | WFStubMachineRegistered | Fügen Sie eine importierte Maschine einem Anwendungsbereitstellungstool hinzu, um Updates zu erhalten und Übereinstimmungsüberprüfungen durchzuführen. |
| MachineProvisioned | WFStubMachineProvisioned | Die Maschine ist auf dem Hypervisor vorhanden und zusätzliche Anpassungen werden zu diesem Zeitpunkt abgeschlossen, beispielsweise Anpassungen des Gast-Agents. Verwenden Sie diesen Workflow-Stub, um einen CMDB-Datensatz mit der DHCP-IP-Adresse und Speicherinformationen zu aktualisieren. Mithilfe von WFStubMachineProvisioned vorgenommene Anpassungen werden in der Regel mithilfe von WFStubUnprovisionMachine rückgängig gemacht. |
| Expired | WFStubMachineExpired | Verschieben Sie eine abgelaufene Maschine auf kostengünstigen Speicher, um die Archivierungskosten zu senken, und aktualisieren Sie den CMDB-Datensatz und das Fakturierungssystem entsprechend der Änderungen bei Speicher und Kosten. |
| UnprovisionMachine | WFStubUnprovisionMachine | Entfernen Sie Maschinen aus Active Directory-Konten. Mithilfe von WFStubMachineProvisioned vorgenommene Anpassungen werden in der Regel mithilfe von WFStubUnprovisionMachine rückgängig gemacht. |
| Disposing | WFStubMachineDisposing | Geben Sie IP-Adressen an den Pool zurück. |

Auswählen eines Szenarios für die Lebenszyklus-Erweiterbarkeit

Sie können vRealize Orchestrator oder vRealize Automation Designer verwenden, um Maschinenlebenszyklen zu erweitern.

Sie können die Maschinenlebenszyklen erweitern, indem Sie mithilfe von vRealize Automation Designer vRealize Orchestrator aufrufen, oder Sie können vRealize Orchestrator direkt verwenden. Mit beiden Methoden können Sie benutzerdefinierte Logik in vorab festgelegte Phasen des Lebenszyklus der IaaS-Maschine einfügen, indem Sie benutzerdefinierte vRealize Orchestrator-Workflows erstellen und diese dann

in die Stubs der Zustandsänderungsworkflows einfügen. Allerdings können Sie benutzerdefinierte Zustandsänderungslogik nur auf bestimmte Blueprints beschränken, wenn Sie vRealize Orchestrator direkt verwenden, und Sie können die Ausführung von Workflows nur mit vRealize Automation Designer auf bestimmte Distributed Execution Manager beschränken.

Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe [Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation](#).

Tabelle 1-2. Auswählen eines Szenarios für die Lebenszyklus-Erweiterbarkeit

| Szenario | Prozedur |
|--|--|
| Fügen Sie benutzerdefinierte Logik in vorab festgelegte Phasen des Lebenszyklus der IaaS-Maschine ein und wenden Sie diese benutzerdefinierte Logik auf bestimmte Blueprints an. | Erweitern der Maschinenlebenszyklen mithilfe der vRealize Orchestrator-Checkliste |
| Fügen Sie benutzerdefinierte Logik in vorab festgelegte Phasen des Lebenszyklus der IaaS-Maschine ein und wenden Sie diese benutzerdefinierte Logik global auf alle Blueprints an. | Erweitern der Maschinenlebenszyklen mithilfe der vRealize Automation Designer-Checkliste |
| Beschränken Sie die Ausführung des Workflows auf bestimmte Distributed Execution Manager mithilfe von Qualifikationen in vRealize Automation Designer. Qualifikationen sind mit einem Tag vergleichbar, das Sie sowohl auf Workflows als auch auf DEM Worker-Instanzen anwenden können. Beispiel: Sie möchten Cloud-Bereitstellungsworkflows auf einen bestimmten DEM beschränken, der auf einem Host mit dem erforderlichen Netzwerkzugriff auf Amazon-URLs ausgeführt wird. | Zuordnen von Workflows und DEM-Workern mithilfe von Qualifikationen |

Erweitern der Maschinenlebenszyklen mithilfe von vRealize Orchestrator

In vordefinierten Phasen des IaaS-Maschinenlebenszyklus können Sie benutzerdefinierte Logik hinzufügen, indem Sie vRealize Orchestrator-Workflows erstellen und dann mithilfe von vRealize Orchestrator die benutzerdefinierten Workflows dem Lebenszyklus von Maschinen hinzufügen, die mit bestimmten Blueprints erstellt wurden.

Erweitern der Maschinenlebenszyklen mithilfe der vRealize Orchestrator -Checkliste

„Erweitern der Maschinenlebenszyklen mithilfe der vRealize Orchestrator-Checkliste“ bietet eine Übersicht über die erforderlichen Schritte, um vRealize Orchestrator zu installieren und zu konfigurieren und damit die IaaS-Maschinenlebenszyklen anzupassen.

Tabelle 1-3. Erweitern der Maschinenlebenszyklen mithilfe der vRealize Orchestrator - Checkliste

| Aufgabe | Details |
|--|--|
| <input type="checkbox"/> Konfigurieren Sie einen vRealize Automation-Host für Ihre vRealize Orchestrator-Instanz | Hinzufügen eines vRealize Automation-Hosts |
| <input type="checkbox"/> Konfigurieren Sie einen IaaS-Host für Ihre vRealize Orchestrator-Instanz | Hinzufügen eines IaaS-Hosts |
| <input type="checkbox"/> Installieren Sie die vRealize Orchestrator-Anpassungen zum Erweitern der IaaS-Maschinenlebenszyklen. | Installieren der vRealize Orchestrator-Anpassung |
| <input type="checkbox"/> Erstellen Sie einen vRealize Automation-Endpoint für Ihre vRealize Orchestrator-Instanz. | Erstellen eines vRealize Orchestrator-Endpoints |
| <input type="checkbox"/> Verwenden Sie die Workflow-Vorlage aus dem Unterverzeichnis „Erweiterbarkeit“ der vRealize Automation-Plug-In-Bibliothek, um einen benutzerdefinierten vRealize Orchestrator-Workflow zum Ausführen während des Maschinenlebenszyklus zu erstellen. Sie können mehrere Workflows in demselben Zustand für den gleichen Blueprint ausführen, solange Sie diese unter einem Wrapper-Workflow verschachteln. | Informationen zum Entwickeln von Workflows mit vRealize Orchestrator finden Sie in der vRealize Orchestrator-Dokumentation. Informationen zu Schulungen in vRealize Orchestrator-Entwicklung für vRealize Automation-Integrationen finden Sie in den verfügbaren Schulungskursen von VMware Education sowie im Lehrmaterial von VMware Learning. |
| <input type="checkbox"/> Führen Sie den bereitgestellten Workflow aus, der Ihren benutzerdefinierten Workflow in einen IaaS-Workflow-Stub einfügt und einen Blueprint zum Aufrufen des IaaS-Workflow-Stubs konfiguriert. | Zuweisen eines Zustandsänderungsworkflows zu einem Blueprint und seinen virtuellen Maschinen |
| <p>Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation.</p> | |

Konfigurieren des vRealize Automation -Plug-Ins für die Erweiterbarkeit von Maschinen

Sie konfigurieren Ihre vRealize Automation- und IaaS-Hosts, installieren die Anpassungen für die Erweiterbarkeit von Maschinen und erstellen einen vRealize Automation-Endpoint für Ihre vRealize Orchestrator-Instanz.

Hinzufügen eines vRealize Automation -Hosts

Sie können einen Workflow ausführen, um einen vRealize Automation-Host hinzuzufügen und die Verbindungsparameter für den Host zu konfigurieren.

Vorgehensweise

- 1 Wählen Sie im Dropdown-Menü des Orchestrator-Clients **Ausführen** oder **Design** aus.
- 2 Klicken Sie auf die Ansicht **Workflows**.
- 3 Erweitern Sie **Bibliothek > vRealize Automation > Konfiguration**.
- 4 Klicken Sie mit der rechten Maustaste auf den Workflow **vRA-Host hinzufügen** und wählen Sie **Workflow starten** aus.
- 5 Geben Sie im Textfeld **Hostname** einen eindeutigen Namen für den Host ein.
- 6 Geben Sie im Textfeld **Host-URL** die URL-Adresse des Hosts ein.

Beispiel: *https://hostname*.

- 7 Geben Sie im Textfeld **Mandant** den Namen des Mandanten ein.

Um die vollständige Funktionalität des Plug-Ins für einen Mandanten zu verwenden, erstellen Sie einen dedizierten vRealize Automation-Host für jeden Mandanten.

- 8 Wählen Sie aus, ob die SSL-Zertifikate automatisch ohne Bestätigung durch den Benutzer installiert werden sollen.
- 9 (Optional) Um zu konfigurieren, wie lange vRealize Orchestrator auf eine Verbindung oder Antwort von vRealize Automation wartet, geben Sie in den Textfeldern **Zeitüberschreitung bei Verbindung (in Sekunden)** und **Zeitüberschreitung bei Vorgang (in Sekunden)** Zeitüberschreitungsintervalle ein.
- 10 Wählen Sie den Typ der Verbindung zum Host im Dropdown-Menü **Sitzungsmodus** aus.

| Option | Aktionen |
|-----------------------------|---|
| Freigegebene Sitzung | Geben Sie die Anmeldedaten für einen vRealize Automation-Benutzer in die Textfelder Authentifizierungsbenutzernamen und Authentifizierungskennwort ein. |
| Pro Benutzersitzung | Stellen Sie die Verbindung mit den Anmeldedaten des Benutzers her, der gegenwärtig angemeldet ist. Sie müssen beim Orchestrator-Client mit den Anmeldedaten des vRealize Automation-Systemadministrators angemeldet sein. Um diese Option mit einem externen vRealize Orchestrator-Server zu verwenden, müssen Sie den Orchestrator-Server in der vRealize Automation-Komponentenregistrierung registrieren. |

- 11 Klicken Sie auf **Übernehmen**.

Weiter

Fügen Sie einen Host für die vRealize Automation-Infrastrukturadministration hinzu.

Hinzufügen eines IaaS-Hosts

Sie können einen Workflow ausführen, um den IaaS-Host eines vRealize Automation-Hosts hinzuzufügen und die Verbindungsparameter zu konfigurieren.

Vorgehensweise

- 1 Wählen Sie im Dropdown-Menü des Orchestrator-Clients **Ausführen** oder **Design** aus.
- 2 Klicken Sie auf die Ansicht **Workflows**.
- 3 Erweitern Sie **Bibliothek > vRealize Automation > Infrastrukturadministration > Konfiguration**.
- 4 Klicken Sie mit der rechten Maustaste auf **IaaS-Host hinzufügen** und wählen Sie **Workflow starten** aus.
- 5 Wählen Sie den vRealize Automation-Host, für den Sie einen IaaS-Host konfigurieren möchten, aus dem Dropdown-Menü **vCAC-Host** aus.
- 6 Geben Sie im Textfeld **Hostname** einen eindeutigen Namen für den Host ein.
- 7 Geben Sie die URL der Maschine ein, auf der Ihr Model Manager installiert ist.
Beispiel: „https://model_manager_machine.com“.
- 8 Um die SSL-Zertifikate zu installieren, wählen Sie **Ja**.
- 9 Um einen Proxy für den Zugriff auf Ihre Model Manager-Maschine zu verwenden, wählen Sie **Ja**.
Wenn Sie diese Option auswählen, müssen Sie den Proxy-Host und den Proxy-Port auf der folgenden Seite angeben.
- 10 Klicken Sie auf **Weiter**.
- 11 Wenn Sie einen expliziten Proxy konfigurieren, geben Sie die Proxy-Host-URL und den Port ein.
- 12 Klicken Sie auf **Weiter**.
- 13 Klicken Sie auf **Nein**, um Ihre eigenen Zeitüberschreitungsintervalle zu konfigurieren.
- 14 (Optional) Um zu konfigurieren, wie lange vRealize Orchestrator auf eine Verbindung oder Antwort von vRealize Automation wartet, geben Sie in den Textfeldern **Zeitüberschreitung bei Verbindung (in Sekunden)** und **Zeitüberschreitung bei Vorgang (in Sekunden)** Zeitüberschreitungsintervalle ein.
- 15 Klicken Sie auf **Weiter**.

16 Wählen Sie den Authentifizierungstyp des Hosts aus.

| Option | Beschreibung |
|--------|---|
| SSO | Wählen Sie diese Option, um vCenter Single Sign-On zu verwenden. |
| NTLM | Wählen Sie diese Option, um die protokollbasierte Authentifizierung des NT LAN Manager (NTLM) nur dann zu aktivieren, wenn Ihre Active Directory-Infrastruktur auf der NTLM-Authentifizierung basiert. Wenn Sie diese Option auswählen, müssen Sie die zusätzlichen NTLM-Anmelde- daten und Authentifizierungsoptionen verwenden. |

17 Wenn Sie NTLM ausgewählt haben, klicken Sie auf **Weiter** und geben Sie den Namen der Workstation-Maschine und den NetBIOS-Domänennamen ein.

18 Klicken Sie auf **Übernehmen**.

Installieren der vRealize Orchestrator -Anpassung

Sie können einen Workflow ausführen, um die angepassten Statusänderungsworkflow-Stubs und Orchestrator-Menüvorgangworkflows zu installieren.

Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe [Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation](#).

Vorgehensweise

- 1 Wählen Sie im Dropdown-Menü des Orchestrator-Clients **Ausführen** oder **Design** aus.
- 2 Klicken Sie auf die Ansicht **Workflows**.
- 3 Wählen Sie **Bibliothek > vCloud Automation Center > Infrastrukturadministration > Erweiterbarkeit > Installation** aus.
- 4 Klicken Sie mit der rechten Maustaste auf den Workflow **vCO-Anpassung installieren** und wählen Sie **Workflow starten** aus.
- 5 Wählen Sie einen IaaS-Host aus.
- 6 Klicken Sie auf **Weiter**.
- 7 Wählen Sie die Lebenszyklusphasen aus, denen Sie benutzerdefinierte Logik hinzufügen möchten, indem Sie mindestens einen zu installierenden Statusänderungsworkflow-Stub auswählen.
- 8 Klicken Sie auf **Übernehmen**.

Erstellen eines vRealize Orchestrator -Endpoints

Sie können einen vRealize Orchestrator-Endpoint für die Verbindung zu einem vRealize Orchestrator-Server erstellen.

Sie können mehrere Endpoints für die Verbindungsherstellung mit unterschiedlichen vRealize Orchestrator-Servern konfigurieren, aber für jeden Endpoint müssen Sie die Priorität festlegen.

Beim Ausführen von vRealize Orchestrator-Workflows versucht vRealize Automation zuerst den vRealize Orchestrator Endpoint mit der höchsten Priorität. Wenn dieser Endpoint nicht erreichbar ist, folgt der Endpoint mit der nächsthöheren Priorität, bis ein vRealize Orchestrator-Server verfügbar ist, der den Workflow ausführen kann.

Voraussetzungen

- Melden Sie sich an der vRealize Automation-Konsole als **laaS-Administrator** an.

Vorgehensweise

- 1 Wählen Sie **Infrastruktur > Endpoints > Endpoints** aus.
- 2 Wählen Sie **Neu > Orchestrierung > vRealize Orchestrator** aus.
- 3 Geben Sie einen Namen und optional eine Beschreibung ein.
- 4 Geben Sie eine URL mit dem vollqualifizierten Namen oder der IP-Adresse des vRealize Orchestrator-Servers und die vRealize Orchestrator-Portnummer ein.

Das Transportprotokoll muss HTTPS sein. Wenn kein Port angegeben wurde, wird der Standardport 443 verwendet.

Um die in der vRealize Automation-Appliance eingebettete Standard-vRealize Orchestrator-Instanz zu verwenden, geben Sie **https://vrealize-automation-appliance-hostname:443/vco** ein.

- 5 Geben Sie Ihre vRealize Orchestrator-Anmeldedaten in den Feldern **Benutzername** und **Kennwort** ein, um eine Verbindung zum vRealize Orchestrator-Endpoint herzustellen.

Die verwendeten Anmeldedaten sollten für alle vRealize Orchestrator-Workflows von laaS über Berechtigungen zum Ausführen verfügen.

Um die in der vRealize Automation-Appliance eingebettete vRealize Orchestrator-Standardinstanz zu verwenden, lautet der Benutzername **administrator@vsphere.local** und das Kennwort ist das Administratorkennwort, das bei der Konfiguration von SSO angegeben wurde.

- 6 Geben Sie eine ganze Zahl größer oder gleich 1 in das Textfeld **Priorität** ein.
Ein niedrigerer Wert gibt eine höhere Priorität an.
- 7 (Optional) Klicken Sie auf **Eigenschaften** und fügen Sie benutzerdefinierte Eigenschaften, Eigenschaftsgruppen oder Ihre eigenen Eigenschaftsdefinitionen für den Endpoint hinzu.
- 8 Klicken Sie auf **OK**.

Anpassen von IaaS-Workflows mithilfe von vRealize Orchestrator

Sie verwenden einen einzelnen Workflow in vRealize Orchestrator, um Ihre benutzerdefinierte Logik in die IaaS-Workflow-Stubs einzufügen und Ihre benutzerdefinierten Lebenszyklen den Maschinen-Blueprints zuzuweisen.

Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe [Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation](#).

Sie müssen Ihre benutzerdefinierten vRealize Orchestrator-Workflows so auslegen, dass Zeichenfolgeeingaben akzeptiert werden. Wenn Ihr benutzerdefinierter Workflow einen komplexen Datentyp erwartet, erstellen Sie einen Wrapper-Workflow, der diesen komplexen Wert sucht und in eine Zeichenfolge übersetzt. Ein Beispiel für einen Wrapping-Workflow finden Sie in der Beispiel-Workflowvorlage unter **Bibliothek > vRealize Automation > Infrastruktur > Erweiterbarkeit**.

Zuweisen eines Zustandsänderungsworkflows zu einem Blueprint und seinen virtuellen Maschinen

Sie konfigurieren benutzerdefinierte vRealize Orchestrator-Workflows so, dass sie in bestimmten Phasen im Master-Maschinenworkflow ausgeführt werden, indem Sie den benutzerdefinierten Workflow dem Stub eines Zustandsänderungsworkflows zuordnen und die Workflows einem Blueprint zuweisen.

Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe [Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation](#).

Voraussetzungen

Verwenden Sie die Workflowvorlage aus dem Unterverzeichnis „Erweiterbarkeit“ der vRealize Automation-Plug-In-Bibliothek, um einen benutzerdefinierten Workflow zum Ausführen während des Maschinenlebenszyklus zu erstellen.

Vorgehensweise

- 1 Wählen Sie im Dropdown-Menü des Orchestrator-Clients **Ausführen** oder **Design** aus.
- 2 Klicken Sie auf die Ansicht **Workflows**.
- 3 Wählen Sie **Bibliothek > vRealize Automation > Infrastruktur > Erweiterbarkeit** aus.

- 4 Klicken Sie mit der rechten Maustaste auf den Workflow **Zustandsänderungsworkflow zu einem Blueprint und dessen virtuellen Maschinen zuweisen**, und wählen Sie die Option **Workflow starten**.
- 5 Wählen Sie die Lebenszyklusphase zur Ausführung des Workflows, indem Sie einen Stub aus dem Dropdown-Menü **Zu aktivierender vCAC Workflow-Stub** auswählen.
- 6 Wählen Sie einen IaaS-Host aus.
- 7 Klicken Sie auf **Weiter**.
- 8 Wählen Sie den Blueprint aus, dem Sie den Workflow zuweisen möchten.
- 9 Wählen Sie, ob diese Workflows auf vorhandene Maschinen, die über diesen Blueprint bereitgestellt wurden, angewendet werden sollen.
- 10 Wählen Sie den Workflow aus, den Sie während des Maschinenlebenszyklus ausführen möchten.
- 11 Konfigurieren Sie, welche Workfloweingabewerte der Maschine als benutzerdefinierte Eigenschaften hinzugefügt werden.
 - a Fügen Sie vCO-Workfloweingaben als Blueprint-Eigenschaften hinzu.
 - b Fügen Sie die Eingabewerte für die letzte vCO-Workflowausführung als Blueprint-Eigenschaften hinzu.
- 12 Klicken Sie auf **Übernehmen**.

Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation

Sie erstellen Workflow-Abonnements, die den Ereignisbrokerdienst verwenden, um die registrierten Dienste auf Ereignismeldungen in vRealize Automation zu überwachen, und führen dann einen angelegenen vRealize Orchestrator-Workflow aus, wenn die Bedingungen im Abonnement erfüllt sind. Zum Konfigurieren des Abonnements geben Sie das Ereignisthema, die Auslösebedingungen und den Workflow, der beim Auslösen ausgeführt wird, an.

Mandantenadministratoren können die Workflow-Abonnements speziell für ihren Mandanten erstellen und verwalten.

Der Systemadministrator kann System-Workflow-Abonnements erstellen und verwalten. Die erstellten System-Workflow-Abonnements sind für Ereignisse in jedem Mandanten und für Systemereignisse aktiv.

Mit vRealize Automation bereitgestellte Ereignisthemen

Ereignisthemen beschreiben den Typ der Ereignismeldung, die von anderen Diensten an den Ereignisbrokerdienst gesendet werden. Sie wählen ein Ereignisthema aus und konfigurieren das Workflow-Abonnement basierend auf dem Thema.

Tabelle 1-4. Ereignisthemen

| Ereignisthemaname | Beschreibung | Dienst |
|--|---|---------------------|
| Blueprint-Komponente abgeschlossen | Eine Blueprint-Komponente, die Teil eines zusammengesetzten Blueprints ist, beendet die Bereitstellung. Die Komponente ist ein beliebiger Blueprint, der Teil eines zusammengesetzten Blueprints ist. | composition-service |
| Blueprint-Komponente angefordert | Eine Blueprint-Komponente, die Teil eines zusammengesetzten Blueprints ist, wird angefordert. Die Komponente ist ein beliebiger Blueprint, der Teil eines zusammengesetzten Blueprints ist. | composition-service |
| Blueprint-Konfiguration | Ein Blueprint wird erstellt, aktualisiert oder gelöscht. | composition-service |
| Business-Gruppen-Konfiguration | Eine Business-Gruppe wird erstellt, aktualisiert oder gelöscht. | identity |
| Katalogelementanforderung abgeschlossen | Ein zusammengesetzter Blueprint hat die Bereitstellung beendet. Dieses Ereignisthema enthält alle Blueprint-Komponenten. Es enthält keine eigenständigen XaaS-Blueprints. | composition-service |
| Katalogelement angefordert | Ein zusammengesetzter Blueprint wird angefordert. Dieses Ereignisthema enthält keine XaaS-Blueprints. | composition-service |
| Komponentenaktion abgeschlossen | Eine Aktion wurde auf einer bereitgestellten Blueprint-Komponente ausgeführt, als eine Bereitstellungsaktion angefordert wurde. | composition-service |
| Komponentenaktion angefordert | Eine Aktion, die auf einer bereitgestellten Blueprint-Komponente ausgeführt werden soll, wird angefordert, wenn eine Bereitstellungsaktion angefordert wird. | composition-service |
| Bereitstellungsaktion abgeschlossen | Eine Aktion auf einem bereitgestellten Blueprint wurde abgeschlossen, darunter auch das Ausführen aller Komponententaktionen. | composition-service |
| Bereitstellungsaktion angefordert | Eine Aktion auf einem bereitgestellten Blueprint wird angefordert. | composition-service |
| EventLog-Standardereignis | Dem Ereignisprotokoll wird ein Standard-eintrag hinzugefügt. Der Protokolleintrag wird nicht an Abonnenten verteilt. | eventlog-service |
| Abschluss des IPAM-IP-Lebenszyklus-Ereignisses | Eine Anforderung nach einer IP-Zuteilung oder einer Aufhebung einer Zuteilung ist abgeschlossen. | ipam-service |

Tabelle 1-4. Ereignisthemen (Fortsetzung)

| Ereignisthemaname | Beschreibung | Dienst |
|--|--|---------------------------|
| Maschinenlebenszyklus | Eine angegebene IaaS-Aktion wird auf einer bereitgestellten Maschine ausgeführt. | iaas-service |
| Maschinenbereitstellung | Eine IaaS-Maschine wird gerade bereitgestellt. | iaas-service |
| Orchestrierungsserverkonfiguration | Eine vRealize Orchestrator-Serverkonfiguration wird erstellt, aktualisiert, gelöscht oder geändert, um eine andere Standardinstanz zu verwenden. | o11n-gateway-service |
| Orchestrierungsserverkonfiguration (XaaS) – veraltet | Eine vRealize Orchestrator-Serverkonfiguration wird erstellt, aktualisiert, gelöscht oder geändert, um eine andere Standardinstanz zu verwenden. | advanced-designer-service |
| Nach der Genehmigung | Eine Richtlinienebene nach der Genehmigung wird zur Verwendung der Ereignisabonnemientoption konfiguriert. | approval-service |
| Vor der Genehmigung | Eine Richtlinienebene vor der Genehmigung wird zur Verwendung der Ereignisabonnemientoption konfiguriert. | approval-service |
| Fertigstellungsereignis der Ressourcenerückforderung | Eine Ressourcen-Lease ist abgelaufen, und die Ressourcen werden zurückgefordert. | management-service |

Terminologie zu Workflow-Abonnements und zum Ereignisbrokerdienst

Beim Arbeiten mit den Workflow-Abonnements und dem Ereignisbrokerdienst stoßen Sie möglicherweise auf Terminologie speziell für die Abonnements und den Ereignisbrokerdienst.

Tabelle 1-5. Terminologie zu Workflow-Abonnements und zum Ereignisbrokerdienst

| Begriff | Beschreibung |
|----------------------|---|
| Ereignisthema | Beschreibt Ereignisse mit derselben logischen Intention und derselben Struktur. Jedes Ereignis ist eine Instanz eines Ereignisthemas. |
| Ereignis | Bezeichnet eine Statusänderung beim Producer oder den Elementen, die vom Producer verwaltet werden. Das Ereignis ist jene Komponente, die Informationen zum Auftreten des Ereignisses aufzeichnet. |
| Meldung | Überträgt Informationen im Zusammenhang mit dem Ereignis zwischen den verschiedenen Diensten und Komponenten. Beispielsweise vom Producer an den Ereignisbrokerdienst, oder vom Ereignisbrokerdienst an die Abonnenten. |
| Ereignisbrokerdienst | Mit diesem Dienst werden Meldungen versendet, die von einem Producer an die abonnierten Verbraucher veröffentlicht werden. |

Tabelle 1-5. Terminologie zu Workflow-Abonnements und zum Ereignisbrokerdienst (Fortsetzung)

| Begriff | Beschreibung |
|-------------------------------|--|
| Nutzlast | Die Ereignisdaten. |
| Abonnement | Gibt an, dass ein Abonnent über ein Ereignis informiert werden möchte, indem er ein Ereignisthema abonniert und die Kriterien definiert, durch die die Benachrichtigung ausgelöst wird. |
| Abonnent | Bezieht die vom Ereignisbrokerdienst veröffentlichten Ereignisse basierend auf der Abonnementdefinition. Der Abonnent wird möglicherweise auch als Verbraucher bezeichnet. |
| Anbieter | Registriert Ereignisthemen im Ereignisbrokerdienst. |
| Producer | Veröffentlicht Ereignisse im Ereignisbrokerdienst. |
| Systemadministrator | Ein Benutzer mit Berechtigungen zum Erstellen, Lesen, Aktualisieren und Löschen der Mandanten-Workflow-Abonnements und der System-Workflow-Abonnements mit der API oder dem vRealize Automation-Plug-In. vRealize Automation weist keine Benutzeroberfläche für den Systemadministrator auf. |
| Mandantenadministrator | Der Benutzer mit Berechtigungen zum Erstellen, Lesen, Aktualisieren und Löschen von Mandanten-Workflow-Abonnements für seinen Mandanten. |
| Workflow-Abonnement | Legt das Ereignisthema und die Bedingungen, die einen vRealize Orchestrator-Workflow auslösen, fest. |
| System-Workflow-Abonnement | Ein spezielles Workflow-Abonnement, das auf Systemereignisse und auf Ereignisse in allen Mandanten reagiert. |
| Mandanten-Workflow-Abonnement | Ein spezielles Workflow-Abonnement, das angibt, welche Bedingungen einen vRealize Orchestrator-Workflow für Ereignisse im selben Mandanten auslösen. |

Blockierbare und antwortfähige Ereignisthemen

Ereignisthemen können blockierbare und antwortfähige Ereignisse unterstützen. Das Verhalten eines Workflow-Abonnements hängt davon ab, ob das Thema diese Ereignistypen unterstützt, und wie Sie das Workflow-Abonnement konfigurieren.

Nicht blockierbare Ereignisthemen

Mit nicht blockierbaren Ereignisthemen können nur nicht blockierende Abonnements erstellt werden. Nicht blockierende Abonnements werden asynchron ausgelöst, und Sie können sich nicht auf die Reihenfolge verlassen, in der die Abonnements ausgelöst werden, oder darauf, dass die vRealize Orchestrator-Workflows ausgeführt werden. Nicht blockierende Abonnements geben nur eine Antwort zurück, wenn das Thema antwortfähig ist.

Blockierbare Ereignisthemen

Einige Ereignisthemen unterstützen die Blockierung. Wenn ein Workflow-Abonnement als blockierend markiert ist, werden alle Meldungen, die die konfigurierten Bedingungen erfüllen, erst von anderen Workflow-Abonnements mit übereinstimmenden Bedingungen empfangen, wenn der erste Workflow abgeschlossen ist. Wenn Sie mehrere blockierende Workflow-Abonnements für das gleiche Ereignisthema haben, priorisieren Sie die Abonnements.

Blockierende Abonnements werden in der Reihenfolge der Priorität ausgeführt. Der höchste Prioritätswert ist 0 (null). Wenn Sie über mehr als ein blockierendes Abonnement für das gleiche Ereignisthema mit der gleichen Prioritätsstufe verfügen, werden die Abonnements in alphabetischer Reihenfolge basierend auf den Namen ausgeführt. Nachdem alle blockierenden Abonnements verarbeitet wurden, wird die Meldung an alle nicht blockierenden Abonnements gleichzeitig gesendet. Da die blockierenden Workflow-Abonnements synchron ausgeführt werden, umfasst die geänderte Ereignisnutzlast das aktualisierte Ereignis, wenn die nachfolgenden Workflow-Abonnements benachrichtigt werden.

Je nach ausgewähltem Workflow und Ihren Zielen wenden Sie die Blockierung auf ein oder mehrere Workflow-Abonnements an.

Beispiel: Sie haben zwei Bereitstellungsworkflow-Abonnements, bei denen der zweite Workflow von den Ergebnissen des ersten abhängt. Beim ersten wird während der Bereitstellung eine Eigenschaft geändert, und der zweite zeichnet die neue Eigenschaft, z. B. einen Maschinennamen, in einem Dateisystem auf. Das Abonnement „ChangeProperty“ hat die Priorität 0 und „RecordProperty“ hat die Priorität 1, da es die Ergebnisse des Abonnements „ChangeProperty“ verwendet. Wenn eine Maschine bereitgestellt wird, wird die Ausführung des Abonnements „ChangeProperty“ gestartet. Da die Bedingungen des Abonnements „RecordProperty“ auf Bedingungen nach der Bereitstellung basieren, löst eine Meldung das Abonnement „RecordProperty“ aus. Da der Workflow „ChangeProperty“ aber ein blockierender Workflow ist, wird die Meldung erst dann empfangen, wenn der Workflow abgeschlossen ist. Nachdem der Name geändert und der erste Workflow abgeschlossen wurde, wird der zweite Workflow ausgeführt und der Name im Dateisystem aufgezeichnet.

Selbst wenn ein Ereignisthema die Blockierung unterstützt, können Sie ein nicht blockierendes Workflow-Abonnement erstellen, wenn das Workflow-Abonnement keine abhängigen nachfolgenden Workflows aufweist. Das Workflow-Abonnement wird ausgelöst und führt den vRealize Orchestrator-Workflow ohne weitere Interaktion durch vRealize Automation oder das externe System aus.

Antwortfähige Ereignisthemen

Einige Ereignisthemen unterstützen Antworten vom abonnierten Dienst. Der Dienst, der das antwortfähige Ereignisthema registriert hat, kann ein Antwortereignis akzeptieren, das die Workflowsausgabe bereitstellt. In der Regel handelt es sich dabei um das Ergebnis einer Interaktion mit einem System oder Benutzer. Die Antwort-Ausgabeparameter müssen die im Antwortschema definierten Kriterien erfüllen, damit der vRealize Automation-Dienst, der das ursprüngliche antwortfähige Ereignis veröffentlicht hat, sie verarbeiten kann. Beispielsweise sind Workflow-Abonnements vor und nach der Genehmigung antwortfähig. Wenn Sie einen Workflow erstellen, der eine Genehmigungsanforderung an ein externes System sendet, wird die Antwort (genehmigt oder abgelehnt) von vRealize Automation verarbeitet und das Katalogelement bereitgestellt, oder der Benutzer wird benachrichtigt, dass die Anforderung abgelehnt wurde.

Die Antwort kann die Ausgabe des vRealize Orchestrator-Workflows sein, oder sie kann ein Fehler sein, wenn für den Workflow eine Zeitüberschreitung eintritt oder er fehlschlägt. Wenn die Antwort aus den Workflow-Ausgabeparametern stammt, muss die Antwort das richtige Antwortschemaformat aufweisen.

Best Practices für das Erstellen von vRealize Orchestrator - Workflows für Workflow-Abonnements

Ein Workflow-Abonnement basiert auf einem bestimmten Themenschema. Um sicherzustellen, dass die Abonnements vRealize Orchestrator-Workflows starten können, müssen Sie sie mit den richtigen Eingabeparametern konfigurieren, damit sie mit den Ereignisdaten funktionieren.

Workflow-Eingabeparameter

Der benutzerdefinierte Workflow, den Sie erstellen, kann alle Parameter oder einen einzelnen Parameter enthalten, der alle Daten in der Nutzlast verbraucht.

- Um einzelne Parameter einzuschließen, konfigurieren Sie einen oder mehrere Parameter. Stellen Sie sicher, dass der Name und Typ mit dem im Schema angegebenen Namen und Typ übereinstimmen.
- Um einen einzelnen Parameter zu verwenden, konfigurieren Sie einen Parameter des Typs `Properties`. Sie können einen beliebigen sinnvollen Namen angeben. Beispielsweise können Sie `payload` als Parameternamen verwenden.

Workflow-Ausgabeparameter

Der benutzerdefinierte Workflow, den Sie erstellen, kann Ausgabeparameter enthalten, die für erforderliche nachfolgende Ereignisse für einen Antwortereignis-Thementyp relevant sind.

Wenn ein Ereignisthema eine Antwort erwartet, müssen die Workflow-Ausgabeparameter mit dem Antwortschema übereinstimmen.

Einstellungen für das Workflow-Abonnement

Die Abonnementoptionen legen basierend auf Ereignismeldungen in vRealize Automation fest, wann ein Workflow ausgeführt wird. Verwenden Sie die Optionen zum Verwalten Ihrer Abonnements.

Ein Abonnement stellt die Absicht eines Benutzers dar, Ereignisse für ein bestimmtes Ereignisthema zu abonnieren und einen Workflow auszuführen, wenn ein Ereignis für das Thema empfangen wird, das den definierten Bedingungen entspricht.

Sie müssen ein Mandantenadministrator sein, um ein Workflow-Abonnement erstellen zu können. Alle Workflow-Abonnements sind spezifisch für Ihren Mandanten.

Wählen Sie zum Verwalten Ihrer Workflow-Abonnements **Administration > Ereignisse > Abonnements** aus.

Tabelle 1-6. Workflow-Abonnementoptionen

| Option | Beschreibung |
|------------------------------------|--|
| Neu | Erstellen Sie ein neues Abonnement. |
| Bearbeiten | Ändern Sie das ausgewählte Abonnement. Wenn das Abonnement veröffentlicht wird, sind die gespeicherten Änderungen sofort aktiviert. Für ein veröffentlichtes oder nicht veröffentlichtes Abonnement können Sie weder das Ereignisthema bearbeiten noch die Blockierungsoption ändern. |
| Veröffentlichen | Aktivieren Sie das Abonnement. Die Ereignisse des Ereignisbrokerdienstes werden verarbeitet und die Abonnementbedingungen ausgewertet. Wenn eine konfigurierte Bedingung wahr ist, wird der Workflow ausgelöst. |
| Veröffentlichung rückgängig machen | Stellen Sie den Entwurfsstatus eines Abonnements wieder her. Das Abonnement ist in Ihrer Umgebung nicht mehr aktiv und empfängt keine Ereignisse mehr. Wenn Sie ein Abonnement erneut veröffentlichen, beginnt das Abonnement, neue Ereignisse zu empfangen. Vergangene Ereignisse werden nicht empfangen. |
| Löschen | Löschen Sie das ausgewählte Abonnement. |

- [Einstellungen auf der Registerkarte „Ereignisthema“ für Workflow-Abonnements](#)

Ereignisthemen sind Klassen von Ereignissen, die in vRealize Automation bereitgestellt werden. Sie wählen das Ereignisthema aus, für das Sie das Abonnement definieren.

- [Einstellungen auf der Registerkarte „Bedingungen“ für Workflow-Abonnements](#)

Die Bedingungen, die Sie für das Abonnement konfigurieren, bestimmen, ob die Ausführung des Workflows basierend auf den Ereignisdaten ausgelöst wird.

- [Einstellungen auf der Registerkarte „Workflow“ für Workflow-Abonnements](#)

Der von Ihnen ausgewählte vRealize Orchestrator-Workflow wird ausgeführt, wenn die Abonnementbedingungen als wahr ausgewertet werden.

- [Einstellungen auf der Registerkarte „Details“ für Workflow-Abonnements](#)

Die Abonnementdetails bestimmen, wie das Abonnement verarbeitet wird.

Einstellungen auf der Registerkarte „Ereignisthema“ für Workflow-Abonnements

Ereignisthemen sind Klassen von Ereignissen, die in vRealize Automation bereitgestellt werden. Sie wählen das Ereignisthema aus, für das Sie das Abonnement definieren.

Diese Seite dient zu Informationszwecken. Es ist keine Eingabe von Werten erforderlich.

Um ein Ereignisthema für Ihre Workflow-Abonnements auszuwählen, wählen Sie **Administration > Ereignisse > Abonnements** aus. Klicken Sie auf **Neu** und wählen Sie ein Ereignisthema aus.

Tabelle 1-7. Optionen für Ereignisthemen

| Option | Beschreibung |
|--------------|---|
| Ereignis-ID | Bezeichner des Ereignisthemas. |
| Name | Name des Ereignisthemas. |
| Beschreibung | Beschreibung des Ereignisthemas. |
| Herausgeber | Name des Dienstes, für den dieses Ereignisthema registriert ist. |
| Blockierbar | Gibt an, ob Sie ein blockierendes Abonnement für dieses Ereignisthema erstellen können. Blockierende Abonnements werden verwendet, um die Nutzlast eines Ereignisses zu ändern oder um die benutzerdefinierte Logik synchron auszuführen, wenn die Ergebnisse eines zweiten Workflows für das gleiche Ereignis von den Ergebnissen des ersten Workflows abhängen. |
| Antwortfähig | Gibt an, ob ein Ereignisthemaabonnement ein Antwortereignis an den Dienst veröffentlichen kann, der das Ereignis ursprünglich erzeugt hat. Wenn der Wert „Ja“ lautet, wird nach Abschluss des Workflows eine Antwort an den Dienst gesendet, der das ursprüngliche Ereignis veröffentlicht hat. Die Antwort enthält die Ausgabe des vRealize Orchestrator-Workflows und ggf. Fehlerdetails. |
| Schema | Beschreibt die Struktur der Nutzlast des Ereignisses. Sie können das Schema zum Erstellen von Workflows verwenden, die die Nutzlastinformationen verwenden können. |

Einstellungen auf der Registerkarte „Bedingungen“ für Workflow-Abonnements

Die Bedingungen, die Sie für das Abonnement konfigurieren, bestimmen, ob die Ausführung des Workflows basierend auf den Ereignisdaten ausgelöst wird.

Wenn Sie **Auf Bedingungen basierend ausführen** auswählen, können die verfügbaren Optionen Folgendes beinhalten:

- **Daten.** Informationen in der Ereignismeldung speziell zum ausgewählten Ereignisthema. Wenn Sie beispielsweise eine Bedingung für das Ereignisthema Maschinenlebenszyklus erstellen, beziehen sich die Datenfelder auf Blueprints und Maschinen. Wenn Sie ein „Vor Genehmigung“-Ereignisthema auswählen, beziehen sich die Datenfelder auf Genehmigungsrichtlinien.

Darüber hinaus können Sie Bedingungen für Felder hinzufügen, die nicht im Schema enthalten sind, indem Sie im Textfeld über der Baumstruktur den Pfad eingeben. Verwenden Sie das Format **`${PATH}`**. PATH ist der Pfad im Schema. Die Knoten trennen Sie mithilfe von `~`. Beispiel: **`${data~machine~properties~SomeCustomProperty}`**.

- **Kernwerte für die Ereignismeldung.** Allgemeine Informationen zur Ereignismeldung. Beispielsweise der Ereignistyp, der Zeitstempel oder der Benutzername.

Um ein Ereignisthema für Ihre Workflow-Abonnements auszuwählen, wählen Sie **Administration > Ereignisse > Abonnements** aus. Klicken Sie auf **Neu**.

Tabelle 1-8. Optionen auf der Registerkarte „Bedingungen“

| Option | Beschreibung |
|-------------------------------------|---|
| Für alle Ereignisse ausführen | Der ausgewählte Workflow wird ausgeführt, wenn die Meldung für dieses Ereignisthema empfangen wird. |
| Auf Bedingungen basierend ausführen | <p>Der ausgewählte Workflow wird ausgeführt, wenn die Ereignismeldung erkannt wird und das Ereignis die konfigurierten Bedingungen erfüllt.</p> <p>Wenn Sie diese Option auswählen, müssen Sie Bedingungen basierend auf den Ereignisdaten angeben, um den ausgewählten Workflow für dieses Abonnement auszulösen.</p> <ul style="list-style-type: none"> ■ Einzelne Bedingung. Der Workflow wird ausgelöst, wenn die konfigurierte Klausel zutrifft. ■ Alle folgenden Optionen. Der Workflow wird ausgelöst, wenn alle Klauseln zutreffen und Sie mindestens zwei Bedingungen angegeben haben. ■ Eine der folgenden Optionen. Der Workflow wird ausgelöst, wenn mindestens eine Klausel zutrifft und Sie mindestens zwei Bedingungen angegeben haben. ■ Nicht die folgenden. Der Workflow wird ausgelöst, wenn keine der Klauseln zutrifft. <p>Wenn Sie eine Bedingung basierend auf einen Konstantenwert erstellen, wird der Wert unter Nichtbeachtung von Groß- und Kleinschreibung verarbeitet. Wenn die Bedingung beispielsweise lautet: Blueprint-Name enthält „UNIX“, Ihre Blueprints jedoch „Unix“ im Namen verwenden, verarbeitet die Bedingung dennoch korrekt.</p> <p>Um den Namen der Bedingung in Übereinstimmung mit dem Blueprint-Namen zu bringen, müssen Sie zunächst den Wert in etwas ändern, das nicht dieselbe Zeichenfolge enthält. Um beispielsweise die UNIX-Bedingung zu bearbeiten, ändern Sie den Wert in „xxxx“, speichern Sie ihn, ändern Sie dann „xxxx“ in „Unix“ und speichern Sie ihn.</p> |

Einstellungen auf der Registerkarte „Workflow“ für Workflow-Abonnements

Der von Ihnen ausgewählte vRealize Orchestrator-Workflow wird ausgeführt, wenn die Abonnementbedingungen als wahr ausgewertet werden.

Der Workflow, den Sie ausführen möchten, muss bereits in vRealize Orchestrator vorhanden und unter **Administration > vRO-Konfiguration > Serverkonfiguration** angegeben sein.

Um einen Workflow für Ihre Workflow-Abonnements auszuwählen, wählen Sie **Administration > Ereignisse > Abonnements**. Klicken Sie auf **Neu**.

Tabelle 1-9. Optionen auf der Registerkarte „Workflow“

| Option | Beschreibung |
|--------------------------|---|
| Einen Workflow auswählen | Navigiert zum Workflow. |
| Ausgewählter Workflow | Zeigt Informationen zum Workflow, einschließlich Ein- und Ausgabeparametern, an. Auf diese Weise können Sie sicherstellen, dass es sich um den Workflow handelt, den Sie ausführen möchten. |

Einstellungen auf der Registerkarte „Details“ für Workflow-Abonnements

Die Abonnementdetails bestimmen, wie das Abonnement verarbeitet wird.

Wählen Sie zum Verwalten Ihrer Workflow-Abonnements **Administration > Ereignisse > Abonnements** aus. Klicken Sie auf **Neu**.

Die Optionen hängen vom Typ des Ereignisthemas ab.

Tabelle 1-10. Optionen auf der Registerkarte „Details“

| Optionen | Beschreibungen |
|---------------------------|--|
| Name | Standardmäßig ist der angezeigte Name der Name des ausgewählten Workflows. Dieser Name wird in der Abonnementliste angezeigt. Der Name muss im Mandanten eindeutig sein. |
| Priorität | Die Reihenfolge, in der die blockierenden Abonnements ausgeführt werden. Null ist die höchste Priorität. Wenn ein Ereignisthema mehrere blockierende Workflow-Abonnements mit derselben Priorität enthält, werden die Abonnements in alphabetischer Reihenfolge basierend auf den Abonnementnamen verarbeitet. Diese Option steht nur für blockierende Workflow-Abonnements zur Verfügung. |
| Zeitüberschreitung (Min.) | Geben Sie an, nach wie vielen Minuten der Workflow abgeschlossen sein muss, bevor er als fehlgeschlagen gilt. Wenn der Workflow nicht in der angegebenen Zeit abgeschlossen wird, wird der Workflow abgebrochen und die Meldung an das nächste Abonnement in der Prioritätenliste gesendet. Wenn Sie keinen Wert angeben, gilt ein unbegrenzter Zeitüberschreitungswert. Für Dienste, die eine Antwort auf blockierende oder antwortfähige Ereignisse erwarten, gelten möglicherweise eigene Standardwerte für die Zeitüberschreitung. Beispielsweise gilt für IaaS-Bereitstellungs- und Lebenszyklus-Ereignisthemen eine Zeitüberschreitung von 30 Minuten. Dieser Wert wird auf dem IaaS-Server konfiguriert. Genehmigungsthemen weisen einen Standardwert von 24 Stunden auf. Dieser Wert wird im System konfiguriert. |

Tabelle 1-10. Optionen auf der Registerkarte „Details“ (Fortsetzung)

| Optionen | Beschreibungen |
|----------------|---|
| Beschreibung | Standardmäßig ist die angezeigte Beschreibung die Beschreibung des Workflows. |
| Wird blockiert | <p>Bestimmt, ob der Workflow nachfolgende Workflows für das gleiche Ereignisthema blockieren kann, damit diese keine Ereignismeldung erhalten, während auf eine Antwort gewartet wird.</p> <p>Abonnements mit aktivierter Blockierung erhalten Meldungen für das gleiche Ereignisthema vor Abonnements, die nicht für Blockierung konfiguriert sind, in Reihenfolge der Priorität. Wenn der Workflow abgeschlossen ist, wird eine Meldung an das blockierende Abonnement mit der nächsten Priorität gesendet. Nach der Verarbeitung aller blockierenden Abonnements wird die Meldung an alle nicht blockierenden Abonnements gleichzeitig gesendet.</p> <p>Die Blockierungsoption ist nur verfügbar, wenn das Ereignisthema blockierbar ist. Diese Informationen werden auf der Registerkarte „Ereignisthema“ bereitgestellt.</p> <p>Die Blockierbarkeit wird auf der Registerkarte „Ereignisthema“ angegeben.</p> <ul style="list-style-type: none"> ■ Wenn Sie das Kontrollkästchen nicht aktivieren, werden nachfolgende Workflows nicht vom Ereignisbroker blockiert. ■ Wenn Sie das Kontrollkästchen aktivieren, berechnet der Ereignisbroker basierend auf den konfigurierten Bedingungen, welche Workflow-Abonnements für dieses Ereignis berechtigt sind, und führt die Workflows in der Prioritätsreihenfolge aus. Der Ereignisbroker wartet auf eine Antwort vom jeweiligen Workflow, bevor der nächste ausgeführt wird. Alle durch die Ausführung des aktuellen Workflows geänderten Parameter werden an den nächsten Workflow in der Warteschlange übergeben. <p>Während auf eine Antwort gewartet wird, werden so lange keine anderen Workflows über das Ereignis benachrichtigt, bis das belegte System antwortet.</p> <p>Sie können diese Option nicht ändern, nachdem das Workflow-Abonnement veröffentlicht wurde.</p> |

Arbeiten mit Bereitstellungs- und Lebenszyklus-Workflow-Abonnements

Sie erstellen Bereitstellungs- und Lebenszyklus-Workflow-Abonnements, damit Sie mithilfe von vRealize Orchestrator die Verwaltung von IaaS-Maschinen erweitern können. Die Bereitstellungsabonnements erweitern die möglichen Funktionen während des Bereitstellungsvorgangs. Die Lebenszyklusabonnements erweitern die möglichen Funktionen, wenn der Benutzer die bereitgestellten Elemente verwaltet.

Integration von IaaS -Diensten

Sie erstellen ein Workflow-Abonnement basierend auf einem Ereignisthema vom Typ „Bereitstellung“ oder „Lebenszyklus“, das einen benutzerdefinierten vRealize Orchestrator-Workflow basierend auf einer vom IaaS-Dienst generierten Meldung ausführt. vRealize Automation enthält zwei Ereignisthemen, die Sie für die IaaS-Integration verwenden können.

- **Maschinenbereitstellung.** Erstellen Sie Workflow-Abonnements, die Workflows beim Bereitstellen und Löschen von IaaS-Maschinen ausführen.
- **Maschinenlebenszyklus.** Erstellen Sie Workflow-Abonnements, die Workflows im Zusammenhang mit Verwaltungsaktionen ausführen, die ein Besitzer auf der bereitgestellten Maschine ausführt.

Konfigurieren von vRealize Orchestrator -Workflows für Bereitstellungs- und Lebenszyklusworkflows

Sie müssen Ihre vRealize Orchestrator-Workflows zur Unterstützung der IaaS-Servicemeldung konfigurieren.

Schema für Bereitstellungs- und Lebenszyklus-Ereignisthemen

Die Maschinenbgbereitstellungs- und Maschinenlebenszyklus-Ereignisthemen verwenden das gleiche Lebenszyklusschema. Die Unterschiede liegen in den auslösenden Zuständen. Die Maschinenbereitstellung empfängt Meldungen, die auf Bereitstellungszuständen und -ereignissen basieren, und der Maschinenlebenszyklus empfängt Meldungen, die auf aktiven Zuständen und Ereignissen basieren. Bereitstellungszustände sind z. B. „BuildingMachine“ und „Disposing“. Lebenszykluszustände sind z. B. „InstallTools“ und „Off“.

Die Ereignismeldung ist die Ereignisdatennutzlast. Die Ereignisdatennutzlast weist folgende Struktur auf.

```
{
  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 */
}

```

Die vRealize Orchestrator-Parameter werden der Ereignisnutzlast nach Name und Typ zugewiesen.

Bei der Verwendung von `virtualMachineEvent` und `workflowNextState` als Ausgabeparameter müssen die eingegebenen Werte einen Zustand oder ein Ereignis aus dem Workflow repräsentieren, der/das das Ereignis ausgelöst und den aktuellen vRealize Orchestrator-Workflow gestartet hat. Um die möglichen Zustände und Ereignisse des Lebenszyklus zu prüfen, siehe [Lebenszykluszustände des VMPS Master Workflow](#) und [Bereitstellungslebenszykluszustände nach Maschinentyp](#).

Arbeiten mit benutzerdefinierten Erweiterbarkeitseigenschaften

Die benutzerdefinierten Eigenschaften der virtuellen Maschine sind nicht in der Ereignisnutzlast enthalten, es sei denn, sie sind als benutzerdefinierte Erweiterbarkeitseigenschaft für den Lebenszykluszustand angegeben. Sie können diese Eigenschaften zu IaaS-Endpoints, Reservierungen, Blueprints, Anforderungen und anderen Objekten hinzufügen, die benutzerdefinierte Eigenschaften unterstützen.

Die benutzerdefinierte Eigenschaft, die Sie zu einem Objekt hinzufügen, hat folgendes Format: `Extensibility.Lifecycle.Properties.{workflowName}.{stateName}`.

Angenommen, Sie möchten ausgeblendete Eigenschaften und alle Eigenschaften, die mit „Virtual“ beginnen, hinzufügen, wenn der Zustand der virtuellen Maschine „BuildingMachine“ lautet. In diesem Fall fügen Sie die benutzerdefinierten Eigenschaften im Blueprint zur Maschine hinzu. Der Name der benutzerdefinierten Eigenschaft in diesem Beispiel lautet `Extensibility.Lifecycle.Properties.VMPSMasterWorkflow32.BuildingMachine`, und die Werte sind `__*` und `Virtual*` (durch ein Komma getrennt).

Der doppelte Unterstrich (`__*`) umfasst die ausgeblendeten Eigenschaften. Der Wert `Virtual*` umfasst alle Eigenschaften, die mit „Virtual“ beginnen. Das Sternchen (*) ist ein Platzhalter und kann als einziger Wert verwendet werden, aber diese Verwendung des Platzhalterzeichens führt zur Übertragung großer Datenmengen.

Wenn mehrere, nacheinander ausgelöste Workflow-Abonnements mit benutzerdefinierten Eigenschaften vorhanden sind, müssen die entsprechenden Einträge in die Workflows aufgenommen werden, um sicherzustellen, dass die Nutzlastprüfung die benutzerdefinierten Eigenschaften beibehält.

Tabelle 1-11. Aufgabeneinträge zum Beibehalten von benutzerdefinierten Eigenschaften

| Zustand | Aufgabeneinträge |
|--|---|
| Hinzugefügte oder aktualisierte benutzerdefinierte Eigenschaften | <code>virtualMachineAddOrUpdateProperties = payload.virtualMachineAddOrUpdateProperties new Properties();</code> |
| Gelöschte benutzerdefinierte Eigenschaften | <code>virtualMachineDeleteProperties = payload.virtualMachineDeleteProperties new Properties();</code> |

Erstellen eines vRealize Orchestrator -Workflows basierend auf dem Lebenszyklus oder Bereitstellungsschema

Der benutzerdefinierte Workflow, den Sie erstellen, benötigt einen Eingabeparameter `payload` des Typs `Properties`. Die Bereitstellungs- oder Lebenszyklus-Ereignisdatennutzlast wird in diesem Parameter abgelegt, wenn der Workflow in vRealize Orchestrator ausgeführt wird. Sie können auch separate Eingabeparameter einschließen, die mit dem Namen und dem Typ der Felder in der Ereignisnutzlast übereinstimmen.

Lebenszykluszustandsdefinitionen für Workflow-Abonnements

Wenn Sie Workflow-Abonnementbedingungen basierend auf Lebenszykluszuständen konfigurieren, sind möglicherweise die folgenden Definitionen beim Identifizieren der Werte hilfreich.

Jede Meldung enthält ein `lifecycleState`-Element, das auf den Änderungen des IaaS-Maschinenzustands basiert.

Dieses Element weist die folgende Meldungsstruktur auf.

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

Tabelle 1-12. LifecycleState-Elemente

| Eigen-schaft | Beschreibung | Format und Werte | Beispiele |
|---------------------|---|--|--|
| state | Enthält den Workflow-namen und den Zustandsnamen. | {workflowName}.{stateName} | <ul style="list-style-type: none"> ■ VMPSMasterWorkflow32.Requested ■ VMPSMasterWorkflow32.MachineActivated ■ BasicVmWorkflow.BuildComplete |
| phase | Enthält die Phase, die eine Meldung ausgelöst hat. | PRE, POST, EVENT | <ul style="list-style-type: none"> ■ PRE. Ein Ereignis wird veröffentlicht, wenn in diesen Zustand gewechselt wird. ■ POST. Ein Ereignis wird veröffentlicht, wenn dieser Zustand beendet wird. ■ EVENT. Ein Ereignis wird veröffentlicht, wenn ein IaaS-Ereignis in diesem Zustand empfangen wird. |
| event | Enthält das Ereignis. Diese Eigenschaft ist optional und ist nur für die Phase EVENT vorhanden. | {workflowName}.{stateName}.EVENT. {eventName} | <ul style="list-style-type: none"> ■ VMPSMasterWorkflow32.Requested.EVENT.OnProvisionMachine ■ VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnBuildSuccess ■ BasicVmWorkflow.CreatingMachine.EVENT.OnCreatingMachineComplete |

Lebenszykluszustände des VMPS Master Workflow

Die Lebenszykluszustände des VMPS Master Workflow stellen den Lebenszyklus einer virtuellen IaaS-Maschine von der Anforderung bis zur Vernichtung dar. Sie können Zustände und Ereignisse des VMPS Master Workflow verwenden, wenn Sie auslösende Bedingungen basierend auf Lebenszykluszustands-Ereignissen und -Namen erstellen.

Jede virtuelle Maschine durchläuft vier grundlegende Phasen.

- Anfordern. Umfasst Genehmigungen.
- Bereitstellen. Umfasst verschiedene Bereitstellungstypen wie Erstellen, Klonen, Kickstart oder WIM.
- Verwalten. Umfasst Aktionen wie z. B. Einschalten, Ausschalten oder Snapshot.
- Löschen. Umfasst Deaktivieren, Aufheben der Bereitstellung und Löschen der Maschine.

Diese grundlegenden Phasen sind im Masterworkflow enthalten. Sie können die VMPSMasterWorkflow32-Zustände verwenden, wenn Sie Bedingungen für die folgenden Ereignisthemen erstellen:

- Maschinenlebenszyklus
- Maschinenbereitstellung

Die globalen Ereigniszustände sind Meldungen, die vom VMPS Master Workflow an den Ereignisbroker gesendet werden. Globale Ereignisse können jederzeit ausgelöst werden.

Tabelle 1-13. Globale Ereignisse

| Zustand (Thema) | Events [Auslösezeichenfolge] (Thema) |
|-----------------|---|
| Global | <ul style="list-style-type: none"> ■ onBuildFailure [BuildFailure] (Provision) ■ OnBuildSuccess [BuildSuccess] (Provision) ■ OnFinalizeMachine [Destroy] (Provision) ■ OnForceUnregisterEvent [ForceUnregister] (Provision) ■ ReconfigureVM.Pending [ReconfigureVM.Pending] (Active) ■ ReconfigureVM.ExecutionUpdated [ReconfigureVM.ExecutionUpdated] (Active) ■ ReconfigureVM.RetryRequestMade [ReconfigureVM.RetryRequestMade] (Active) ■ ReconfigureVM.Failed [ReconfigureVM.Failed] (Active) ■ ReconfigureVM.Successful [ReconfigureVM.Successful] (Active) ■ ReconfigureVM.Complete [ReconfigureVM.Complete] (Active) ■ ReconfigureVM.Canceled [ReconfigureVM.Canceled] (Active) |

Die aktiven globale Zustände sind Aktionen, die Sie auf bereitgestellten Maschinen ausführen können.

Tabelle 1-14. Aktive Ereignisse

| Zustand | Events [Auslösezeichenfolge] (Thema) |
|---------|---|
| Active | <ul style="list-style-type: none"> ■ OnExpireLease [Expire] (Active) ■ OnForceExpire [ForceExpire] (Active) ■ onReprovision [Reprovision] (Active) ■ onResetBuildSuccess [ResetBuildSuccess] (Active) |

Im Masterworkflow treten Bereitstellungsereignisse während des Maschinenbereitstellungs-Lebenszyklus auf. Aktive Ereignisse sind Aktionen, die Sie auf bereitgestellten Maschinen ausführen können. Eine Darstellung des Masterworkflows finden Sie unter [VMPS Master Workflow – Beispiel](#)

Jeder Maschinentyp weist einen eigenen Bereitstellungsworkflow auf. Weitere Informationen zu den einzelnen Maschinentypen finden Sie unter [Bereitstellungslebenszykluszustände nach Maschinentyp](#)

Tabelle 1-15. VMPSMasterWorkflow32-Zustände und -Ereignisse

| Zustand (Thema) | Events [Auslösezeichenfolge] (Thema) |
|---|---|
| BuildingMachine | |
| <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | |
| DeactivateMachine | |
| <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | |
| Disposing | |
| <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | <ul style="list-style-type: none"> ■ OnDisposeComplete(Provision) ■ OnDisposeTimeout(Provision) ■ OnUnregisterMachine [Unregister] (Provision) |

Tabelle 1-15. VMPSMasterWorkflow32-Zustände und -Ereignisse (Fortsetzung)

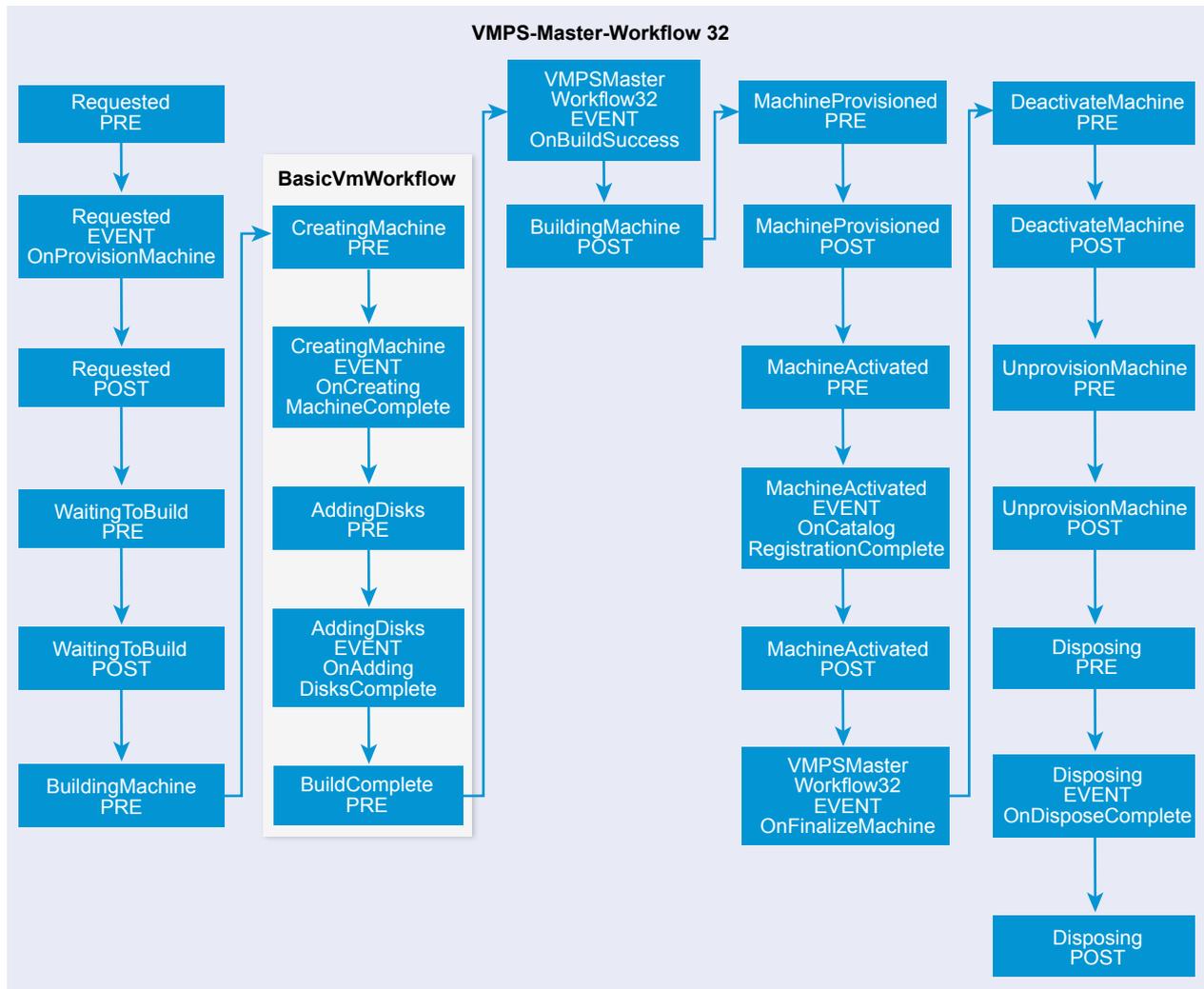
| Zustand (Thema) | Events [Auslösezeichenfolge] (Thema) |
|---|---|
| Expired <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnActiveExpiredMachine [ActivateExpiredMachine] (Active) ■ TurnOffFromExpired [TurnOffExpiredMachine] (Active) |
| InstallTools <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ InstallToolsComplete(Active) ■ TimeoutInstallTools(Active) |
| Leased | <ul style="list-style-type: none"> ■ OnChangeLease [Change Lease] (Active) ■ OnUpdateDescription [UpdateDescription] (Active) ■ OnUpdateOwner [UpdateOwner] (Active) |
| MachineActivated <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | <ul style="list-style-type: none"> ■ OnCatalogRegistrationComplete [CatalogRegistrationComplete] (Provision) |
| MachineProvisioned <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | |
| Off <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnForceOn [ForceOn] (Active) ■ OnResetOff [Turn Off] (Active) ■ OnTurnOn [Turn On] (Active) |
| On <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnForceOff [ForceOff] (Active) ■ onInstallTools [InstallTools] (Active) ■ OnReboot [Reboot] (Active) ■ OnReset [Reset] (Active) ■ OnResetOn [Turn On] (Active) ■ OnShutdown [Shutdown] (Active) ■ OnSuspend [Suspend] (Active) ■ OnTurnOff [Turn Off] (Active) |
| Rebooting <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnRebootComplete(Active) ■ TimeoutFromReboot(Active) |
| RegisterMachine <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | <ul style="list-style-type: none"> ■ onRegisterComplete(Provision) ■ RegisterTimeout(Provision) |
| Requested <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | <ul style="list-style-type: none"> ■ OnProvisionMachine [Provision] (Provision) |
| Resetting <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnResetComplete(Active) ■ TimeoutFromReset(Active) |
| ShuttingDown <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnShutdownComplete(Active) ■ TimeoutFromShutdown(Active) |

Tabelle 1-15. VMPSMasterWorkflow32-Zustände und -Ereignisse (Fortsetzung)

| Zustand (Thema) | Events [Auslösezeichenfolge] (Thema) |
|---|---|
| Suspending <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnSuspendComplete(Active) ■ TimeoutFromSuspend(Active) |
| TurningOff <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnTurningOffComplete(Active) ■ TimeoutFromPowerOff(Active) |
| TurningOn <ul style="list-style-type: none"> ■ Pre(Active) ■ Post(Active) | <ul style="list-style-type: none"> ■ OnTurningOnComplete(Active) ■ TimeoutPowerOn(Active) |
| UnprovisionMachine <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | |
| WaitingToBuild <ul style="list-style-type: none"> ■ Pre(Provision) ■ Post(Provision) | |

VMPS Master Workflow - Beispiel

Der VMPS-Workflow ist der Masterworkflow, in den die anderen Bereitstellungsworkflows eingebettet sind. Dieses Beispiel enthält den Basic VM Workflow, um den Lebenszyklus einer virtuellen Maschine zu veranschaulichen. Dies ist kein spezieller Workflow in Ihrer Umgebung.



Bereitstellungslebenszykluszustände nach Maschinentyp

Die Lebenszykluszustände nach Maschinentyp gelten spezifisch für bestimmte Typen virtueller Maschinen. Neben dem Masterworkflow können Sie die Bereitstellungsworkflowzustände und -ereignisse verwenden, wenn Sie Auslösebedingungen für Workflow-Abonnements erstellen.

Blade Logic Bare Metal

| Status (Thema) | Ereignisse (Thema) |
|--|--------------------|
| BuildFinished | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| CreatingMachine | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |

Opware Bare Metal

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| BuildFinished | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| OpwareRegister | <ul style="list-style-type: none"> ■ OnOpwareRegister(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |

Cloud Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| BuildComplete | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| CloudProvisioning | <ul style="list-style-type: none"> ■ OnCloudProvisioningTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| FailedProvisioning | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |

App Service Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|--|
| AppServiceProvisioning | <ul style="list-style-type: none"> ■ OnAppServiceProvisioningTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| BuildComplete | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| FailedProvisioning | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |

Basic VM Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| AddingDisks | <ul style="list-style-type: none"> ■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| BuildComplete | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| CreatingMachine | <ul style="list-style-type: none"> ■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| FailedProvisioning | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |

Opware Virtual

| Status (Thema) | Ereignisse (Thema) |
|------------------------------------|---|
| AddingDisks ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision) |
| BuildFinished ■ Pre(Provision) | |
| CreatingVM ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCreateVMComplete(Provision) ■ OnCreateVMTimeout(Provision) |
| InitialPowerOn ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision) |
| OpwareRegister ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnOpwareRegister(Provision) |

Cloud Linux Kickstart Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| BuildComplete ■ Pre(Provision) | |
| CreatingMachine ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision) |
| CustomizeOS ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision) |
| FailedProvisioning ■ Pre(Provision) | |
| InitialPowerOn ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision) |
| InstallingOS ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInstallingOSComplete(Provision) ■ OnInstallingOSTimeout(Provision) |

Clone Workflow

| Status (Thema) | Ereignisse (Thema) |
|--------------------------------------|---|
| BuildComplete ■ Pre(Provision) | |
| CloneMachine ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCloneMachineComplete(Provision) ■ OnCloneMachineTimeout(Provision) |
| CustomizeMachine ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCustomizeMachineComplete(Provision) ■ OnCustomizeMachineTimeout(Provision) |
| CustomizeOS | <ul style="list-style-type: none"> ■ OnCustomizeOS(Provision) ■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision) |

| Status (Thema) | Ereignisse (Thema) |
|---|---|
| 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) | |

Cloud WIM Image Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| BuildComplete ■ Pre(Provision) | |
| CreatingMachine ■ Pre(Provision) | ■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision) |
| FailedProvisioning ■ Pre(Provision) | |
| InitialPowerOn ■ Pre(Provision) | ■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision) |
| InstallOS ■ Pre(Provision) | ■ onInstallOSComplete(Provision) ■ OnInstallOSTimeout(Provision) |
| Reboot ■ Pre(Provision) | ■ OnRebootComplete(Provision) ■ OnRebootTimeout(Provision) |
| SetupOS ■ Pre(Provision) | ■ OnSetupOSComplete(Provision) ■ OnSetupOSTimeout(Provision) |

External Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| AddingDisks ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision) |
| BuildComplete ■ Pre(Provision) | |
| CreatingMachine ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision) |
| EpiRegister ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnEpiRegisterComplete(Provision) |
| FailedProvisioning ■ Pre(Provision) | |
| InitialPowerOn ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision) |

Linux Kickstart Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| AddingDisks ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision) |
| BuildComplete ■ Pre(Provision) | |
| CreatingMachine ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision) |
| CustomizeOS ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision) |
| EjectingCD ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnEjectingCDComplete(Provision) ■ OnEjectingCDTimeout(Provision) |
| FailedProvisioning ■ Pre(Provision) | |
| InitialPowerOn ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision) |
| InstallingOS ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInstallingOSComplete(Provision) ■ OnInstallingOSTimeout(Provision) |

Physical Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|---|---|
| FailedProvisioning <ul style="list-style-type: none"> Pre(Provision) | |
| FinalizeProvisioning <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnFinalizeProvisioningTimeout(Provision) |
| InitializeProvisioning <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnInitializeProvisioningTimeout(Provision) |
| InitialPowerOn <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnInitialPowerOnTimeout(Provision) |
| InstallOS <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnInstallOSComplete(Provision) OnInstallOSTimeout(Provision) |
| Reboot <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnRebootComplete(Provision) OnRebootTimeout(Provision) |
| SetupOS <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnSetupOSComplete(Provision) OnSetupOSTimeout(Provision) |

Physical PXE Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|---|---|
| CheckHardwareType <ul style="list-style-type: none"> Pre(Provision) | |
| CleanPxe <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnCleanPxeTimeout(Provision) |
| FailedProvisioning <ul style="list-style-type: none"> Pre(Provision) | |
| FinalizeProvisioning <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnFinalizeProvisioningTimeout(Provision) |
| InitializeProvisioning <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnInitializeProvisioningTimeout(Provision) |
| InitialPowerOn <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnInitialPowerOnTimeout(Provision) |
| InstallOS <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnInstallOSComplete(Provision) OnInstallOSTimeout(Provision) |
| Reboot <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnRebootComplete(Provision) OnRebootTimeout(Provision) |
| SetupOS <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnSetupOSComplete(Provision) OnSetupOSTimeout(Provision) |
| SetupPxe <ul style="list-style-type: none"> Pre(Provision) | <ul style="list-style-type: none"> OnSetupPxeTimeout(Provision) |

Physical SCCM Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| CheckHardwareType ■ Pre(Provision) | |
| Complete ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCompleteProvisioningComplete(Provision) ■ OnCompleteProvisioningTimeout(Provision) |
| FailedProvisioning ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnFailedProvisioningTimeout(Provision) |
| FinalizeProvisioning ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnFinalizeProvisioningTimeout(Provision) |
| InitializeProvisioning ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitializeProvisioningTimeout(Provision) |
| InitialPowerOn ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitialPowerOnTimeout(Provision) |
| SccmRegistration ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnSccmRegistrationTimeout(Provision) |

Physical SCCM PXE Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| CheckHardwareType ■ Pre(Provision) | |
| CleanPxe ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCleanPxeTimeout(Provision) |
| Complete ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnCompleteProvisioningComplete(Provision) ■ OnCompleteProvisioningTimeout(Provision) |
| Disposing ■ Pre(Provision) | |
| FailedProvisioning ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnFailedProvisioningTimeout(Provision) |
| FinalizeProvisioning ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnFinalizeProvisioningTimeout(Provision) |
| InitializeProvisioning ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitializeProvisioningTimeout(Provision) |
| InitialPowerOn ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnInitialPowerOnTimeout(Provision) |
| SccmRegistration ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnSccmRegistrationTimeout(Provision) |
| SetupPxe ■ Pre(Provision) | <ul style="list-style-type: none"> ■ OnSetupPxeTimeout(Provision) |

vApp Clone Workflow

| Status (Thema) | Events [Auslösezeichenfolge] (Thema) |
|--|--|
| Global | <ul style="list-style-type: none"> ■ OnFailProvisioning [FailProvisioning] (Provision) ■ OnMasterProvisioned [MasterProvisioned] (Provision) |
| BuildComplete | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| CloneMachine | <ul style="list-style-type: none"> ■ OnCloneMachineComplete(Provision) ■ OnCloneMachineTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| CustomizeMachine | <ul style="list-style-type: none"> ■ OnCustomizeMachineComplete(Provision) ■ OnCustomizeMachineTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| CustomizeOS | <ul style="list-style-type: none"> ■ OnCustomizeOS(Provision) ■ OnCustomizeOSComplete(Provision) ■ OnCustomizeOSTimeout(Provision) |
| FailedProvisioning | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| FinalizeProvisioning | <ul style="list-style-type: none"> ■ OnFinalizeComplete(Provision) ■ OnFinalizeTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| InitialPowerOn | <ul style="list-style-type: none"> ■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| WaitingForMaster | <ul style="list-style-type: none"> ■ OnWaitingForMasterTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |

Virtual SCCM Provisioning Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| AddingDisks | <ul style="list-style-type: none"> ■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| BuildComplete | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| CreatingMachine | <ul style="list-style-type: none"> ■ CreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| Disposing | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| EjectingCD | <ul style="list-style-type: none"> ■ OnEjectingCDComplete(Provision) ■ OnEjectingCDTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| FailedProvisioning | |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |
| InitialPowerOn | <ul style="list-style-type: none"> ■ OnInitialPowerOnComplete(Provision) ■ OnPowerOnTimeout(Provision) |
| <ul style="list-style-type: none"> ■ Pre(Provision) | |

| Status (Thema) | Ereignisse (Thema) |
|--------------------------------------|---|
| InstallingOS ■ Pre(Provision) | ■ OnInstallingOSComplete(Provision) ■ OnInstallingOSTimeout(Provision) |
| SccmRegistration ■ Pre(Provision) | ■ OnSccmRegistrationTimeout(Provision) |

WIM Image Workflow

| Status (Thema) | Ereignisse (Thema) |
|--|---|
| AddingDisks ■ Pre(Provision) | ■ OnAddingDisksComplete(Provision) ■ OnAddingDisksTimeout(Provision) |
| BuildComplete ■ Pre(Provision) | |
| CreatingMachine ■ Pre(Provision) | ■ OnCreatingMachineComplete(Provision) ■ OnCreatingMachineTimeout(Provision) |
| EjectingCD ■ Pre(Provision) | ■ OnEjectingCDComplete(Provision) ■ OnEjectingCDTimeout(Provision) |
| FailedProvisioning ■ Pre(Provision) | |
| InitialPowerOn ■ Pre(Provision) | ■ OnInitialPowerOnComplete(Provision) ■ OnInitialPowerOnTimeout(Provision) |
| InstallOS ■ Pre(Provision) | ■ onInstallOSComplete(Provision) ■ OnInstallOSTimeout(Provision) |
| Reboot ■ Pre(Provision) | ■ OnRebootComplete(Provision) ■ OnRebootTimeout(Provision) |
| SetupOS ■ Pre(Provision) | ■ OnSetupOSComplete(Provision) ■ OnSetupOSTimeout(Provision) |

Konfigurieren der Zeitüberschreitungswerte für Zustände und Ereignisse

Der Standard-Zeitüberschreitungswert für alle Zustände und Ereignisse beträgt 30 Minuten und wird in den globalen vRealize Automation-Einstellungen konfiguriert. Einige Workflows benötigen möglicherweise mehr Zeit für eine erfolgreiche Ausführung. Um verschiedene Workflows in Ihrer Umgebung zu berücksichtigen, können Sie Überschreibungswerte für die Zeitüberschreitung einzelner Workflows oder Zustände hinzufügen.

Um den Standard-Zeitüberschreitungswert zu ändern, wählen Sie **Infrastruktur > Administration > Globale Einstellungen** aus und bearbeiten Sie den Wert für **Zeitüberschreitung für Erweiterbarkeits-Lebenszyklusmeldungen**. Wenn Sie Änderungen an der globalen Einstellung vornehmen, müssen Sie den Manager Service neu starten.

Um einzelne Zeitüberschreitungswerte zu konfigurieren, fügen Sie die Workflow- oder Ereigniseigenschaft im Abschnitt `appSetting` der Datei `ManagerService.exe.config` auf dem IaaS-Server hinzu. Diese Datei befindet sich in der Regel im Verzeichnis `%System-Drive%\Program Files x86\VMware\vCAC\Server`. Erstellen Sie immer eine Kopie der Datei, bevor Sie sie bearbeiten. Wenn Sie Änderungen an den individuellen Einstellungen vornehmen, müssen Sie den Manager Service neu starten.

Das grundlegende Format für die Schlüssel ist dem in den folgenden Beispielen vergleichbar.

- Für einen Workflow. `Extensibility.{workflow}.Timeout`
- Für Ereignisse. `Extensibility.{workflow}.{state}.EVENT.{event}.Timeout`
- Für Zustände. `Extensibility.{workflow}.{state}.(PRE/POST).Timeout`

Verwenden Sie die folgenden Werte als Beispiele beim Hinzufügen von Schlüsseln im Abschnitt `appSetting`. Das Format für den Zeitüberschreitungswert ist `T.HH:mm:ss.ms`. `T` steht für Tag und `ms` für Millisekunden. Tag und Millisekunden sind optional. Stunden, Minuten und Sekunden sind erforderlich.

- Um die Zeitüberschreitung für den gesamten `BasicVmWorkflow-Workflow` auf 30 Minuten festzulegen, fügen Sie `<add key="Extensibility.BasicVmWorkflow.Timeout" value="00:30:00"/>` hinzu.
- Um die Zeitüberschreitung für das globale Ereignis „`OnFinalizeMachine`“ im „`VMPSMasterWorkflow32`“ auf zwei Stunden festzulegen, fügen Sie `<add key="Extensibility.VMPSMasterWorkflow32.VMPSMasterWorkflow32.EVENT.OnFinalizeMachine.Timeout" value="02:00:00"/>` hinzu.
- Um die Zeitüberschreitung für den Zustand vor der Anforderung von „`VMPSMasterWorkflow32`“ auf 2 Tage festzulegen, fügen Sie `<add key="Extensibility.VMPSMasterWorkflow32.Requested.PRE.Timeout" value="2.00:00:00"/>` hinzu.

Konfigurieren des Fehlerverhaltens für Zustände und Ereignisse

Für den Umgang mit Zeitüberschreitungen und Fehlern bei Workflow-Abonnements gibt es ein Standardverhalten. Sie können das Verhalten für Maschinen in Ihrer Umgebung anpassen.

IaaS steuert die Ereigniszeitüberschreitung und Fehlerverarbeitung über den Ereignisbrokerdienst.

Bei jedem Zustandsübergang sendet `SendEBSMessage` ein Ereignis an den Ereignisbrokerdienst und wartet auf eine Antwort. Standardmäßig wird eine Zeitüberschreitung oder ein Fehler, der vom Ereignisbrokerdienst gemeldet wird, protokolliert, und der Workflow wird fortgesetzt.

Wenn während der folgenden Zustände eine Zeitüberschreitung oder ein Fehler im Masterworkflow auftritt, wird der Workflow zwangsweise in den Fehlerzustand versetzt und nicht fortgesetzt.

Tabelle 1-16. Ausnahmen, bei denen Workflows nicht fortgesetzt werden

| Zustand, in dem der Fehler auftritt | Fehlerzustand |
|-------------------------------------|--------------------|
| PRE MachineProvisioned | UnprovisionMachine |
| PRE BuildingMachine | Disposing |
| PRE RegisterMachine | Finalized |

Um das Zeitüberschreitungs- oder Fehlererhalten anzupassen, können Sie der Maschine benutzerdefinierte Eigenschaften für alle Ereignisse oder Zustände hinzufügen, in denen Sie ein Ereignis auslösen oder eine Zustandsänderung erzwingen möchten. Verwenden Sie die folgenden Beispiele, um die benutzerdefinierten Eigenschaften zu konfigurieren.

- `Extensibility.Lifecycle.Error.Event.{Workflow}.{State}`. Der Wert der Eigenschaft ist der Name des Ereignisses, das im Workflow im Fall einer Zeitüberschreitung oder eines Fehlers ausgelöst wird.
- `Extensibility.Lifecycle.Error.State.{Workflow}.{State}`. Der Wert der Eigenschaft ist der Name des Zustands, in den der Workflow zwangsweise übergeht, wenn eine Zeitüberschreitung oder ein Fehler auftritt.

Szenario: Erstellen eines VM-Snapshots vom Typ „Nach der Bereitstellung“

Als Mandantenadministrator möchten Sie, dass Ihre Servicekatalogbenutzer einen Snapshot vom Typ „Nach der Bereitstellung“ ihrer virtuellen Maschinen erstellen können, damit sie die neue virtuelle Maschine wiederherstellen können, anstatt eine neue virtuelle Maschine anzufordern.

Vorgehensweise

1 Szenario: Erstellen eines vRealize Orchestrator-Workflows für eine Snapshot-Aktion vom Typ „Nach der Bereitstellung“

Sie erstellen einen vRealize Orchestrator-Workflow, der den erforderlichen Eingabeparameter akzeptiert. Sie konfigurieren den Workflow so, dass Ihr Ziel nach der Bereitstellung erreicht wird.

2 Szenario: Erstellen eines Snapshot-Workflow-Abonnements vom Typ „Nach der Bereitstellung“

Als Mandantenadministrator möchten Sie einen Snapshot jeder erstellten virtuellen Maschine anfertigen. Sie konfigurieren ein Workflow-Abonnement basierend auf dem Ereignisthema „Maschinenbereitstellung“ und veröffentlichen es, um es zu aktivieren.

Szenario: Erstellen eines vRealize Orchestrator -Workflows für eine Snapshot-Aktion vom Typ „Nach der Bereitstellung“

Sie erstellen einen vRealize Orchestrator-Workflow, der den erforderlichen Eingabeparameter akzeptiert. Sie konfigurieren den Workflow so, dass Ihr Ziel nach der Bereitstellung erreicht wird.

Informationen zum Erstellen von Ordnern und Workflows für vRealize Orchestrator finden Sie unter *Entwickeln mit VMware vRealize Orchestrator*.

Voraussetzungen

Melden Sie sich bei der vRealize Orchestrator-Instanz, die für vRealize Automation konfiguriert ist, mit Berechtigungen zum Erstellen eines Workflows an.

Vorgehensweise

- 1 Erstellen Sie einen Ordner für Ihre Workflow-Abonnement-Workflows in der Workflowbibliothek.

- Erstellen Sie einen neuen Workflow.

Benennen Sie den Workflow für dieses Szenario
Automatisierungssnapshot „Nach der Bereitstellung“.

- Fügen Sie folgenden Eingabeparameter hinzu.

| Name | Typ |
|----------|------------|
| Nutzlast | Properties |

- Fügen Sie eine skriptfähige Aufgabe hinzu, die den Eingabeparameter akzeptiert und einen Snapshot der virtuellen Maschine erstellt.
- Speichern Sie den Workflow.

Weiter

Sie erstellen ein Workflow-Abonnement, das Ihren Workflow „Automatisierungssnapshot 'Nach der Bereitstellung'“ ausführt. [Szenario: Erstellen eines Snapshot-Workflow-Abonnements vom Typ „Nach der Bereitstellung“](#)

Szenario: Erstellen eines Snapshot-Workflow-Abonnements vom Typ „Nach der Bereitstellung“

Als Mandantenadministrator möchten Sie einen Snapshot jeder erstellten virtuellen Maschine anfertigen. Sie konfigurieren ein Workflow-Abonnement basierend auf dem Ereignisthema „Maschinenbereitstellung“ und veröffentlichen es, um es zu aktivieren.

Sie konfigurieren das Workflow-Abonnement, um einen Workflow zum Erstellen eines Snapshots auszuführen, wenn eine virtuelle Maschine bereitgestellt wird und die erkannte Ereignismeldung den Status „aktiviert“ aufweist.

Voraussetzungen

- Melden Sie sich an der vRealize Automation-Konsole als **Mandantenadministrator** an.
- Konfigurieren Sie ein vCenter Server-Plug-In als vRealize Orchestrator-Endpoint. Siehe *Konfigurieren von vRealize Automation*.
- Stellen Sie sicher, dass ein vSphere-VM-Blueprint vorhanden ist.
- Stellen Sie sicher, dass ein vRealize Orchestrator-Workflow vorhanden ist, der einen Snapshot einer virtuellen Maschine erstellt. Den Workflow „Snapshot erstellen“ des vRealize Automation-Plug-Ins können Sie nicht verwenden. Der verfügbare Snapshot-Workflow ist abhängig von der XaaS-Integration. Siehe [Konfigurieren von vRealize Orchestrator-Workflows für Bereitstellungs- und Lebenszyklusworkflows](#).

Vorgehensweise

- Wählen Sie **Administration > Ereignisse > Abonnements** aus.
- Klicken Sie auf das Symbol **Neu (+)**.
- Wählen Sie **Maschinenbereitstellung** aus.

- 4 Klicken Sie auf **Weiter**.
- 5 Konfigurieren Sie auf der Registerkarte „Bedingungen“ die Auslösebedingungen.
 - a Wählen Sie **Auf Bedingungen basierend ausführen** aus.
 - b Wählen Sie im Dropdown-Menü **Bedingung** die Option **Alle folgenden Optionen** aus
 - c Konfigurieren Sie die folgenden Bedingungen:

| Eigenschaft | Operator | Wert |
|--|------------|---|
| Daten > Maschine > Maschinentyp | Ist gleich | Konstante > Virtuelle Maschine |
| Daten > Lebenszykluszustand > Lebenszykluszustandsname | Ist gleich | Konstante > VMPSMasterWorkflow32.MachineActivated |
| Daten > Lebenszykluszustand > Zustandsphase | Ist gleich | Konstante > POST |

- d Klicken Sie auf **Weiter**.
- 6 Navigieren Sie auf der Registerkarte „Workflow“ in der Orchestrator-Baumstruktur und wählen Sie Ihren Workflow **Automatisierungssnapshot „Nach der Bereitstellung“** aus.
- 7 Klicken Sie auf **Weiter**.
- 8 Nehmen Sie auf der Registerkarte „Details“ in den Feldern **Name** und **Beschreibung** eine Eingabe vor.

In diesem Szenario geben Sie **VM-Snapshot „Nach der Bereitstellung“** als Name und **Snapshot erstellen, wenn eine neue virtuelle Maschine bereitgestellt und aktiviert wird** als Beschreibung ein.
- 9 Klicken Sie auf **Fertig stellen**.
- 10 Wählen Sie die Zeile „VM-Snapshot 'Nach der Bereitstellung'“ aus und klicken Sie auf **Veröffentlichen**.

Das Workflow-Abonnement ist aktiv und löst Ihren Snapshot-Workflow aus, wenn eine Ereignismeldung darauf hinweist, dass eine angeforderte virtuelle Maschine bereitgestellt und aktiviert ist.

Weiter

Zum Testen des Workflow-Abonnements fordern Sie eine virtuelle Maschine im Servicekatalog an. Nachdem die Anforderung auf eine erfolgreiche Bereitstellung hinweist, überprüfen Sie, ob der Snapshot erstellt wurde.

Arbeiten mit Genehmigungs-Workflow-Abonnements

Sie erstellen Workflow-Abonnements vom Typ „Vor Genehmigung“ und „Nach Genehmigung“, damit Sie eine Genehmigungsanforderung zur Verarbeitung an ein externes System senden können. Die Antwort in Form einer Genehmigung oder Ablehnung wird dann von vRealize Automation verarbeitet.

Integration des Genehmigungsdiensts

Sie erstellen ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“, das einen benutzerdefinierten vRealize Orchestrator-Workflow ausführt, mit dem die Genehmigungsanforderung in einem System außerhalb von vRealize Automation verarbeitet wird.

Für die Genehmigungsebene der Genehmigungsrichtlinie können Sie **Ereignisabonnement verwenden** als Genehmiger auswählen. Dies ist eine von mehreren möglichen Ebenen in einer Genehmigungsrichtlinie. Wenn ein Servicekatalogbenutzer ein Objekt anfordert, auf das eine Genehmigungsrichtlinie angewendet wird, die den Genehmiger **Ereignisabonnement verwenden** enthält, sendet der Genehmigungsdienst eine Meldung an den Ereignisbrokerdienst mit folgendem Resultat.

- Bei einem veröffentlichten Workflow-Abonnement mit übereinstimmenden Kriterien führt vRealize Orchestrator Ihren Genehmigungs- oder Ablehnungsworkflow aus.
- Wenn Sie über ein veröffentlichtes Workflow-Abonnement ohne übereinstimmende Kriterien verfügen, wenn Sie die Veröffentlichung des Workflow-Abonnements rückgängig gemacht haben oder wenn Sie nicht über ein veröffentlichtes Abonnement verfügen, lautet die Genehmigungsebene „Genehmigt“ und der Genehmigungsvorgang wechselt zur nächsten Genehmigungsebene.

Das Genehmigungs-Workflow-Abonnement empfängt Meldungen vom Genehmigungsdienst und vergleicht die Meldungen mit den konfigurierten Kriterien für Genehmigungsabonnements. Wird eine Meldung gefunden, die mit den Kriterien übereinstimmt, wird die Ausführung des ausgewählten vRealize Orchestrator-Workflows gestartet. Die veröffentlichten Ereignisdaten werden an den Workflow als Eingabe übergeben und mit der im Workflow angegebenen Methode verarbeitet. Die Ergebnisse des Workflows werden an vRealize Automation zurückgegeben und die Anforderung wird verarbeitet. Wenn „Genehmigt“ zurückgegeben wird, wird die nächste Genehmigungsebene ausgewertet. Wenn „Abgelehnt“ zurückgegeben wird, wird die Anforderung abgelehnt. Empfängt der Genehmigungsdienst innerhalb von 24 Stunden keine Antwort (die standardmäßige Zeitüberschreitung für den Genehmigungsdienst), wird die Anforderung als abgelehnt verarbeitet.

Konfigurieren von vRealize Orchestrator -Workflows für Genehmigungsereignisthemen

Für Ihren benutzerdefinierten vRealize Orchestrator-Workflow müssen Sie die Unterstützung der Genehmigungsmeldung und die Antwort mit ordnungsgemäß formatierten Informationen, die von vRealize Automation verarbeitet werden können, konfigurieren.

Genehmigungsereignisthema-Schema

Das Ereignismeldungsschema vom Typ „Vor Genehmigung“ und „Nach Genehmigung“ enthält die Feldnamen und Werte, die in der Anforderung enthaltenen Informationen sowie Informationen zur Quelle der Anforderung.

Die Ereignisdatennutzlast weist folgende Struktur auf.

```
{
  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
  }
}
```

Eigenschaftsnamen und Eigenschaftswerte können die benutzerdefinierten Eigenschaften oder Systemeigenschaften sein, die Sie auf der Ebene der Genehmigungsrichtlinie konfigurieren. Diese Eigenschaften werden in der Genehmigungsrichtlinie konfiguriert, damit der Genehmiger die Werte während eines Genehmigungsvorgangs ändern kann. Wenn beispielsweise CPU vorhanden ist, kann der Genehmiger die Anzahl der CPUs im Genehmigungsanforderungsformular reduzieren.

Die Antwortereignisdatennutzlast ist jene Information, die durch den Workflow an vRealize Automation zurückgegeben wurde. Der Inhalt der Antwortnutzlast bestimmt, ob die Anforderung genehmigt oder abgelehnt wird.

```
{
  approved : BOOLEAN,

  // Property values
  fieldValues : Properties
}
```

Der `approved`-Parameter in der Antwortereignisnutzlast ist entweder „true“ für genehmigte Anforderungen oder aber „false“ für abgelehnte Anforderungen. Die Eigenschaftswerte sind die benutzerdefinierten Eigenschaften oder Systemeigenschaften, die durch den vRealize Orchestrator-Workflow geändert und an vRealize Automation zurückgegeben in den Genehmigungsvorgang einbezogen wurden.

Es wird empfohlen, für den vRealize Orchestrator-Workflow einen Ausgabeparameter für `businessJustification` zu konfigurieren. Mithilfe dieses Parameters können Sie Kommentare, die der Genehmiger im externen System eingegeben hat, an den Genehmigungsvorgang von vRealize Automation übergeben. Kommentare sind für Genehmigungen oder Ablehnungen möglich.

Erstellen eines vRealize Orchestrator -Workflows basierend auf dem Genehmigungsschema

Der von Ihnen erstellte benutzerdefinierte Genehmigungsworkflow muss einen Eingabeparameter, mit einem aussagekräftigen Namen, für den der Typ `Properties` konfiguriert ist, aufweisen. Die Genehmigungs-Ereignisdatennutzlast wird diesem Parameter hinzugefügt, wenn die Ausführung des Workflow-Abonnements ausgelöst wird.

Die Ausgabeparameter des Workflows, die an vRealize Automation zurückgesendet werden, sind `approved : Boolean` und `fieldValues : Properties`. Der zurückgegebene Parameter `approved : Boolean` bestimmt, ob die Genehmigungsebene „Genehmigt“ oder „Abgelehnt“ lautet. Der Parameter `fieldValues : Properties` enthält die Werte, die im externen System geändert wurden.

Szenario: Senden von Softwareanforderungen an ein externes System zur Genehmigung

Als Mandantenadministrator möchten Sie, dass Benutzer außerhalb von vRealize Automation eine Softwarekomponente genehmigen, wenn ein Servicekatalogbenutzer eine Maschine anfordert, die Software enthält. Sie konfigurieren eine Genehmigungsrichtlinie, die die Genehmigung für alle Softwarebereitstellungen erfordert, sowie ein Workflow-Abonnement, das ausgeführt wird, wenn Meldungen vor der Genehmigung empfangen werden, die mit Ihren definierten Bedingungen übereinstimmen.

Vorgehensweise

1 [Szenario: Erstellen eines vRealize Orchestrator-Workflows für Genehmigungs-Workflow-Abonnements](#)

Sie erstellen einen vRealize Orchestrator-Workflow, der die erforderlichen Genehmigungseingabeparameter aus vRealize Automation akzeptiert und die erforderlichen Ausgabeparameter zurückgibt, um den Genehmigungsvorgang abzuschließen.

2 [Szenario: Erstellen einer Genehmigungsrichtlinie für die externe Genehmigung](#)

Als Mandantenadministrator erstellen Sie eine Genehmigungsrichtlinie, von der eine Ereignismeldung generiert wird, die vom Genehmigungsdienst veröffentlicht wird. Wenn Sie ein Workflow-Abonnement mit Kriterien definiert haben, die mit der Ereignismeldung übereinstimmen, führt vRealize Orchestrator den ausgewählten Workflow aus.

3 [Szenario: Erstellen eines Workflow-Abonnements vor der Genehmigung](#)

Als Mandantenadministrator erstellen Sie ein Workflow-Abonnement vor der Genehmigung, das einen vRealize Orchestrator-Workflow ausführt, wenn eine Servicekataloganforderung eine Genehmigungsanforderung erstellt, die den konfigurierten Bedingungen entspricht.

Szenario: Erstellen eines vRealize Orchestrator -Workflows für Genehmigungs-Workflow-Abonnements

Sie erstellen einen vRealize Orchestrator-Workflow, der die erforderlichen Genehmigungseingabeparameter aus vRealize Automation akzeptiert und die erforderlichen Ausgabeparameter zurückgibt, um den Genehmigungsvorgang abzuschließen.

Sie müssen den Workflow so konfigurieren, dass Ihr Genehmigungsziel erreicht wird. Informationen zum Erstellen von Ordnern und Workflows für vRealize Orchestrator finden Sie unter *Entwickeln mit VMware vRealize Orchestrator*.

Voraussetzungen

Melden Sie sich bei der vRealize Orchestrator-Instanz, die für vRealize Automation konfiguriert ist, mit Berechtigungen zum Erstellen eines Workflows an.

Vorgehensweise

- 1 Erstellen Sie einen Ordner für Ihre Workflow-Abonnement-Workflows in der Workflowbibliothek.
- 2 Erstellen Sie einen neuen Workflow.

Benennen Sie den Workflow für dieses Szenario **Genehmigungsanforderung automatisieren**.

- a Fügen Sie folgenden Eingabeparameter hinzu.

| Name | Typ |
|-------|------------|
| input | Properties |

- b Fügen Sie folgende Ausgabeparameter hinzu.

| Name | Typ |
|-------------|------------|
| approved | boolean |
| fieldValues | Properties |

- 3 Erstellen Sie eine skriptfähige Aufgabe, mit der die Ein- und Ausgabeparameter verarbeitet werden.
- 4 Speichern Sie den Workflow.

Weiter

Sie erstellen eine Genehmigungsrichtlinie, die das Workflow-Abonnement als Genehmiger verwendet.

[Szenario: Erstellen einer Genehmigungsrichtlinie für die externe Genehmigung](#)

Szenario: Erstellen einer Genehmigungsrichtlinie für die externe Genehmigung

Als Mandantenadministrator erstellen Sie eine Genehmigungsrichtlinie, von der eine Ereignismeldung generiert wird, die vom Genehmigungsdienst veröffentlicht wird. Wenn Sie ein Workflow-Abonnement mit Kriterien definiert haben, die mit der Ereignismeldung übereinstimmen, führt vRealize Orchestrator den ausgewählten Workflow aus.

Voraussetzungen

- Melden Sie sich bei der vRealize Automation-Konsole als **Mandantenadministrator** oder **Genehmigungsadministrator** an.

Vorgehensweise

- 1 Wählen Sie **Administration > Genehmigungsrichtlinien** aus.
- 2 Erstellen Sie eine Genehmigungsrichtlinie für Ihre Softwarekomponenten.
 - a Klicken Sie auf das Symbol **Neu (+)**.
 - b Wählen Sie **Richtlinientyp auswählen** aus.
 - c Wählen Sie in der Liste **Servicekatalog – Katalogelementanforderung – Softwarekomponente** aus.
 - d Klicken Sie auf **OK**.
 - e Konfigurieren Sie die folgenden Optionen:

| Option | Konfiguration |
|---------------------|--|
| Name | Geben Sie Externe Genehmigung von Software ein. |
| Beschreibung | Geben Sie An externes Genehmigungssystem gesendete Genehmigungsanforderung ein. |
| Status | Wählen Sie Active aus. |

- 3 Klicken Sie auf der Registerkarte **Vor der Genehmigung** auf das Symbol **Hinzufügen (+)**.
- 4 Konfigurieren Sie die Registerkarte **Ebeneninformationen** mit den Auslöserkriterien und den Genehmigungsaktionen.
 - a Geben Sie im Textfeld **Name** den Namen **Externe Ebene für Software** ein.
 - b Geben Sie im Textfeld **Beschreibung** die Beschreibung **An externes Genehmigungssystem gesendete Softwaregenehmigungsanforderung** ein.
 - c Wählen Sie **Immer erforderlich** aus.
 - d Wählen Sie **Ereignisabonnement verwenden** aus.
- 5 Klicken Sie auf **OK**.

Weiter

- Erstellen Sie ein Workflow-Abonnement vom Typ „Vor Genehmigung“, das Ereignismeldungen basierend auf der konfigurierten Genehmigungsebene empfängt. Siehe [Szenario: Erstellen eines Workflow-Abonnements vor der Genehmigung](#).
- Wenden Sie die Genehmigungsrichtlinie auf eine Softwarekomponente in einer Berechtigung an. Siehe *Konfigurieren von vRealize Automation*.

Szenario: Erstellen eines Workflow-Abonnements vor der Genehmigung

Als Mandantenadministrator erstellen Sie ein Workflow-Abonnement vor der Genehmigung, das einen vRealize Orchestrator-Workflow ausführt, wenn eine Servicekataloganforderung eine Genehmigungsanforderung erstellt, die den konfigurierten Bedingungen entspricht.

Voraussetzungen

- Melden Sie sich an der vRealize Automation-Konsole als **Mandantenadministrator** an.
- Konfigurieren Sie eine Genehmigungsrichtlinienebene mit dem Namen „Externe Ebene für Software“. Siehe [Szenario: Erstellen einer Genehmigungsrichtlinie für die externe Genehmigung](#).
- Erstellen Sie einen benutzerdefinierten vRealize Orchestrator-Workflow, der die Anforderung an Ihr externes System sendet. Verwenden Sie in diesem Szenario den Workflow „Genehmigungsanforderung automatisieren“.

Vorgehensweise

- 1 Wählen Sie **Administration > Ereignisse > Abonnements** aus.
- 2 Klicken Sie auf das Symbol **Neu (+)**.
- 3 Klicken Sie auf **Vor der Genehmigung**.
- 4 Klicken Sie auf **Weiter**.
- 5 Konfigurieren Sie auf der Registerkarte **Bedingungen** die Auslösebedingungen.
 - a Wählen Sie **Auf Bedingungen basierend ausführen** aus.
 - b Konfigurieren Sie im Dropdown-Menü **Bedingung** die folgende Bedingung:

| Eigenschaft | Operator | Wert |
|---|------------|--------------------------------|
| Daten > Informationen zur Anforderung > Genehmigungsebenename | Ist gleich | Externe Ebene für die Software |

- c Klicken Sie auf **Weiter**.
- 6 Durchsuchen Sie auf der Registerkarte „Workflow“ die Orchestrator-Baumstruktur und wählen Sie den Workflow **Genehmigungsanforderung automatisieren** aus.
- 7 Klicken Sie auf **Weiter**.
- 8 Geben Sie auf der Registerkarte „Details“ den Namen und die Beschreibung ein.
Geben Sie in diesem Szenario **Software extern** als Name und **Sendet Genehmigungsanforderung an externes System** als Beschreibung ein.
- 9 Geben Sie im Textfeld **Zeitüberschreitung (Min.)** den Wert „120“ ein.
Der Zeitraum, den Sie für die Zeitüberschreitung des Abonnement-Workflows eingeben, hängt vom Zielsystem ab. Wenn vRealize Automation in der angegebenen Anzahl Minuten keine Antwort des Zielsystems verarbeitet, wird die Anforderung automatisch abgelehnt.

Wenn Sie keinen Wert bereitstellen, beträgt die Zeitüberschreitung standardmäßig 24 Stunden.

10 Klicken Sie auf **Fertig stellen**.

11 Wählen Sie die Zeile „Software extern“ aus und klicken Sie auf **Veröffentlichen**.

Das Ereignisabonnement vor der Genehmigung „Software extern“ empfängt jetzt Ereignismeldungen vor der Genehmigung.

Weiter

- Wenn Sie die Genehmigungsrichtlinie auf eine Softwarekomponente in einer aktiven Berechtigung angewendet haben, fordern Sie das Element im Servicekatalog an und stellen Sie sicher, dass Ihre Genehmigungsrichtlinie und das Workflow-Abonnement wie gewünscht funktionieren.

Fehlerbehebung von Workflow-Abonnements

Zur Fehlerbehebung von Workflow-Abonnements gehören häufig auftretende Probleme. Möglicherweise müssen Sie auch verschiedene Protokolle überprüfen.

- [Fehlerbehebung von vRealize Orchestrator-Workflows, die nicht gestartet werden](#)
Sie konfigurieren ein Workflow-Abonnement, um einen benutzerdefinierten Workflow auszuführen, wenn die Ereignismeldung empfangen wird, aber der Workflow wird nicht ausgeführt.
- [Fehlerbehebung für Bereitstellungsanforderungen, die zu lange dauern](#)
Die Bereitstellung einer IaaS-Maschine kann zehn oder mehr Stunden dauern.
- [Fehlerbehebung eines vRealize Orchestrator-Workflows, der für eine Genehmigungsanforderung nicht ausgeführt wird](#)
Sie haben ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“ konfiguriert, um einen vRealize Orchestrator-Workflow auszuführen. Der Workflow wird nicht ausgeführt, wenn eine Maschine, die den definierten Kriterien entspricht, im Servicekatalog angefordert wird.
- [Fehlerbehebung einer abgelehnten Genehmigungsanforderung, die genehmigt sein sollte](#)
Sie konfigurieren ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“, das den angegebenen vRealize Orchestrator-Workflow ausführt, aber die Anforderung wird abgelehnt, obwohl Sie wissen, dass sie genehmigt wurde.
- [Fehlerbehebung einer abgelehnten Genehmigungsanforderung](#)
Sie konfigurieren ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“, das den angegebenen vRealize Orchestrator-Workflow ausführt, aber die Anforderung wird unerwartet abgelehnt.

Fehlerbehebung von vRealize Orchestrator -Workflows, die nicht gestartet werden

Sie konfigurieren ein Workflow-Abonnement, um einen benutzerdefinierten Workflow auszuführen, wenn die Ereignismeldung empfangen wird, aber der Workflow wird nicht ausgeführt.

Lösung

1 Stellen Sie sicher, dass Sie das Workflow-Abonnement veröffentlicht haben.

- 2 Stellen Sie sicher, dass die Bedingungen des Workflow-Abonnements ordnungsgemäß konfiguriert sind.
- 3 Stellen Sie sicher, dass der vRealize Orchestrator-Server den angegebenen Workflow aufweist.

Fehlerbehebung für Bereitstellungsanforderungen, die zu lange dauern

Die Bereitstellung einer IaaS-Maschine kann zehn oder mehr Stunden dauern.

Ursache

Wenn Sie ein Workflow-Abonnement zur Auslösung in einem Bereitstellungszustand konfiguriert haben, kann es vorkommen, dass zwei Instanzen des Manager-Dienstes in Ihrer Umgebung ausgeführt werden.

Lösung

- ◆ Stellen Sie sicher, dass nur eine Instanz der Manager-Dienste aktiv ist. Wenn beide aktiv sind, werden auch Fehler wie diese in den Protokollen angezeigt.

Wenn mehr als eine Instanz aktiv ist, wird eine Fehlermeldung wie die folgende in die Protokolle der Manager-Dienste geschrieben.

```
[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'.
```

Fehlerbehebung eines vRealize Orchestrator -Workflows, der für eine Genehmigungsanforderung nicht ausgeführt wird

Sie haben ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“ konfiguriert, um einen vRealize Orchestrator-Workflow auszuführen. Der Workflow wird nicht ausgeführt, wenn eine Maschine, die den definierten Kriterien entspricht, im Servicekatalog angefordert wird.

Ursache

Um ein Workflow-Abonnement erfolgreich für eine Genehmigung auszuführen, müssen Sie sicherstellen, dass alle Komponenten ordnungsgemäß konfiguriert sind.

Lösung

- 1 Stellen Sie sicher, dass die Genehmigungsrichtlinie aktiv ist und dass Sie **Ereignisabonnement verwenden** für eine Genehmigungsebene in der Richtlinie ausgewählt haben.
- 2 Stellen Sie sicher, dass die Genehmigungsrichtlinie in Ihrer Berechtigung ordnungsgemäß angewendet wird.

- 3 Stellen Sie sicher, dass Ihr Workflow-Abonnement ordnungsgemäß konfiguriert und veröffentlicht ist.
- 4 Überprüfen Sie die Ereignisprotokolle auf Meldungen im Zusammenhang mit Genehmigungen.

Fehlerbehebung einer abgelehnten Genehmigungsanforderung, die genehmigt sein sollte

Sie konfigurieren ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“, das den angegebenen vRealize Orchestrator-Workflow ausführt, aber die Anforderung wird abgelehnt, obwohl Sie wissen, dass sie genehmigt wurde.

Lösung

- 1 Überprüfen Sie den Workflow in vRealize Orchestrator.
 - a Melden Sie sich bei vRealize Orchestrator mit Administratorrechten an.
 - b Stellen Sie sicher, dass der Workflow fehlerfrei ausgeführt wurde.
 - c Stellen Sie sicher, dass für die Parameter `approval` und `fieldValues` die erwarteten Werte zurückgegeben wurden.
- 2 Überprüfen Sie die Anforderung in vRealize Automation.
 - a Melden Sie sich bei vRealize Automation als der Benutzer an, der das abgelehnte Element angefordert hat.
 - b Klicken Sie auf die Registerkarte **Anforderungen**.
 - c Öffnen Sie die abgelehnte Anforderung.
 - d Klicken Sie auf **Genehmigungsstatus** und überprüfen Sie die weiteren Informationen in der Spalte „Begründung“.

Falls ein Fehler aufgetreten ist, werden Informationen zu dem Fehler als Begründungsdaten angezeigt.

Fehlerbehebung einer abgelehnten Genehmigungsanforderung

Sie konfigurieren ein Workflow-Abonnement vom Typ „Vor Genehmigung“ oder „Nach Genehmigung“, das den angegebenen vRealize Orchestrator-Workflow ausführt, aber die Anforderung wird unerwartet abgelehnt.

Problem

Alle Genehmigungsebenen vor dieser externen Genehmigungsebene wurden genehmigt und diese Genehmigungsebene hätte genehmigt werden sollen, wurde aber als abgelehnt verarbeitet.

Ursache

Eine mögliche Ursache ist ein interner Fehler, als vRealize Orchestrator versucht hat, den Workflow auszuführen. Beispielsweise fehlt der Workflow oder der vRealize Orchestrator-Server wird nicht ausgeführt.

Lösung

- 1 Wählen Sie **Administration > Ereignisse > Ereignisprotokolle** aus.

- 2 Überprüfen Sie die Protokolle auf Meldungen im Zusammenhang mit Genehmigungen.

Erweitern der Maschinenlebenszyklen mithilfe von vRealize Automation Designer

In vordefinierten Phasen des IaaS-Maschinenlebenszyklus können Sie benutzerdefinierte Logik mithilfe von vRealize Automation Designer hinzufügen, um die Statusänderungsworkflow-Stubs direkt zu bearbeiten und optional benutzerdefinierte vRealize Orchestrator-Workflows aufzurufen.

Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe [Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation](#).

Erweitern der Maschinenlebenszyklen mithilfe der vRealize Automation Designer -Checkliste

„Erweitern der Maschinenlebenszyklen mithilfe der vRealize Automation Designer-Checkliste“ bietet eine Übersicht über die erforderlichen Schritte, um vRealize Automation Designer zu installieren und zu konfigurieren und damit die IaaS-Maschinenlebenszyklen anzupassen.

Tabelle 1-17. Erweitern der Maschinenlebenszyklen mithilfe der vRealize Automation Designer -Checkliste

| Aufgabe | Details |
|---|---|
| <input type="checkbox"/> Laden Sie vRealize Automation Designer herunter und installieren Sie das Programm. | Installieren von vRealize Automation Designer |
| <input type="checkbox"/> Erstellen Sie einen vRealize Automation-Endpoint für Ihre vRealize Orchestrator-Instanz. | Erstellen eines vRealize Orchestrator-Endpoints |
| <input type="checkbox"/> Ordnen Sie Ihren vRealize Orchestrator-Endpoint einem Maschinen-Blueprint zu. | Zuordnen eines vRealize Orchestrator-Endpoints zu einem Blueprint |

Tabelle 1-17. Erweitern der Maschinenlebenszyklen mithilfe der vRealize Automation Designer -Checkliste (Fortsetzung)

| Aufgabe | Details |
|--|--|
| <input type="checkbox"/> Verwenden Sie vRealize Automation Designer-Aktivitäten, um einen IaaS-Workflow-Stub anzupassen. | Anpassen eines IaaS-Workflows |
| <p>Hinweis Die Workflow-Stubs werden durch die Workflow-Abonnements des Ereignisbrokers ersetzt. Sie sind noch verfügbar, werden unterstützt und können verwendet werden, werden aber voraussichtlich in einer zukünftigen Version von vRealize Automation entfernt. Um eine zukünftige Produktkompatibilität sicherzustellen, sollten Sie zum Ausführen benutzerdefinierter Workflows auf Basis von Zustandsänderungen die Workflow-Abonnements verwenden. Siehe Konfigurieren von Workflow-Abonnements zum Erweitern von vRealize Automation.</p> | |
| <p>Optional können Sie mithilfe von vRealize Orchestrator-Workflowaktivitäten benutzerdefinierte vRealize Orchestrator-Workflows aufrufen.</p> | |
| <input type="checkbox"/> Nachdem Sie einen benutzerdefinierten Zustandsänderungsworkflow erstellt haben, muss ein Mandantenadministrator oder Business-Gruppenmanager ihn für bestimmte Blueprints aktivieren, indem eine benutzerdefinierte Eigenschaft hinzugefügt wird. | Konfigurieren eines Blueprints zum Aufrufen eines Zustandsänderungsworkflows |

Installieren und Konfigurieren von vRealize Automation Designer

Sie können vRealize Automation Designer auf einem Windows-Computer installieren und für die Kommunikation mit einer Model Manager-Remote-Instanz konfigurieren. Wenn Sie IaaS-Workflows zum Aufrufen von vRealize Orchestrator-Workflows verwenden, müssen Sie auch die vRealize Orchestrator-Instanz in IaaS konfigurieren.

Installieren von vRealize Automation Designer

Sie können vRealize Automation Designer auf einem Windows-Computer installieren und für die Kommunikation mit einer Model Manager-Remote-Instanz konfigurieren.

Voraussetzungen für vRealize Automation Designer

vRealize Automation Designer wird in der Regel nicht auf einem Server, sondern auf einem Entwicklercomputer installiert.

Unterstützte Betriebssysteme

Die für vRealize Automation Designer unterstützten Betriebssysteme sind in der *Übersicht über die Unterstützung von vRealize Automation* auf der VMware vRealize Automation-Dokumentationsseite aufgeführt.

Systemkonfigurationsanforderungen

In der *Übersicht über die Unterstützung von vRealize Automation* für Ihre vRealize Automation-Version finden Sie mögliche aktualisierte Informationen.

- .NET Framework 4.5 muss installiert sein.
- Der vRealize Automation Designer-Host benötigt Netzwerkzugriff auf die IaaS-Websitekomponenten (insbesondere die Model Manager Web-Komponente).
- Wenn der Model Manager remote installiert ist, muss das für die Model Manager Web-Komponente verwendete Zertifikat auf dem vRealize Automation Designer-Host vertrauenswürdig sein.

Herunterladen des Installationsprogramms für vRealize Automation Designer

Das Installationsprogramm für vRealize Automation Designer können Sie von der vRealize Automation-Appliance herunterladen.

Voraussetzungen

- Melden Sie sich als lokaler Administrator am Windows-Computer an.
- Achten Sie bei Verwendung von Internet Explorer darauf, dass „Verstärkte Sicherheitskonfiguration“ deaktiviert ist. Siehe `res://iesetup.dll/SoftAdmin.htm`.

Vorgehensweise

- 1 Öffnen Sie einen Browser.
- 2 Navigieren Sie zur Downloadseite für das Windows-Installationsprogramm, indem Sie den Hostnamen verwenden (`https://vra-va-hostname.domain.name:5480/installer/`).
- 3 Klicken Sie auf **vRealize Automation Designer**.
- 4 Speichern Sie das Installationsprogramm, wenn Sie dazu aufgefordert werden.

Weiter

[Installieren von vRealize Automation Designer.](#)

Installieren von vRealize Automation Designer

Das vRealize Automation Designer-Installationsprogramm ist ein Windows-Installations-Assistent.

Voraussetzungen

[Herunterladen des Installationsprogramms für vRealize Automation Designer.](#)

Vorgehensweise

- 1 Navigieren Sie zu dem Verzeichnis, in das Sie das Installationsprogramm heruntergeladen haben.
- 2 Klicken Sie mit der rechten Maustaste auf `DesignCenter-Setup.exe` und wählen Sie **Als Administrator ausführen** aus.
- 3 Klicken Sie auf der Begrüßungsseite auf **Weiter**.

4 Lesen Sie die Lizenzvereinbarung, wählen Sie **Ich akzeptiere die Bedingungen der Lizenzvereinbarung** und klicken Sie anschließend auf **Weiter**.

5 Klicken Sie auf der Seite für benutzerdefiniertes Setup auf **Weiter**.

6 Geben Sie den vollqualifizierten Domännennamen und den Port der Model Manager Web-Instanz im Format *hostname:port* ein.

Der Standardport lautet 443.

7 Geben Sie die Anmeldedaten des Model Manager-Dienstes ein.

8 Klicken Sie auf **Weiter**.

Das Installationsprogramm validiert die Kombination aus Model Manager-Host und Anmeldedaten, indem versucht wird, auf Model Manager zuzugreifen. Wenn ein Fehler zurückgegeben wird, müssen Sie die richtige Kombination aus Model Manager-Host und Anmeldedaten angeben, um den Vorgang fortsetzen zu können.

9 Klicken Sie auf **Installieren**.

10 Klicken Sie auf **Fertig stellen**.

Weiter

Starten Sie vRealize Automation Designer im Windows-Startmenü, indem Sie zum Installationsverzeichnis navigieren.

Konfigurieren der vRealize Orchestrator -Endpoints

Wenn Sie vRealize Automation-Workflows zum Aufrufen von vRealize Orchestrator-Workflows verwenden, müssen Sie die vRealize Orchestrator-Instanz oder den Server als Endpoint konfigurieren.

Sie können einen vRealize Orchestrator-Endpoint einem Maschinen-Blueprint zuordnen, um sicherzustellen, dass alle vRealize Orchestrator-Workflows für die mit diesem Blueprint bereitgestellten Maschinen unter Verwendung dieses Endpoints ausgeführt werden.

vRealize Automation enthält standardmäßig eine eingebettete vRealize Orchestrator-Instanz. Es wird empfohlen, dass Sie diese als vRealize Orchestrator-Endpoint für das Ausführen von vRealize Automation-Workflows in einer Testumgebung oder für das Erstellen eines Proof-of-Concepts verwenden.

Sie können auch ein Plug-In auf einem externen vRealize Orchestrator-Server installieren.

vCenter Orchestrator-Integration - Voraussetzungen

Wenn Sie vRealize Automation-Workflows verwenden, um vRealize Orchestrator-Workflows mit Eingabe- oder Ausgabeparametern des Typs `VC:VirtualMachine` auszuführen, stellen Sie sicher, dass Sie über die vRealize Orchestrator-Workflows zum Konvertieren der virtuellen Maschinentypen zwischen vRealize Orchestrator und IaaS verfügen.

Die erforderlichen Workflows sind als Teil des vCenter-Plug-Ins standardmäßig in vRealize Orchestrator 5.5 und höher enthalten.

Wenn Sie vRealize Orchestrator 5.1 verwenden, installieren Sie das vRealize Automation-Integrationspaket für vRealize Orchestrator. Laden Sie das Paket `com.vmware.library.vcenter.vcac-integration.package` von der vRealize Orchestrator-Community-Website unter <http://communities.vmware.com/community/vmtn/server/vcenter/orchestrator> herunter. Importieren Sie das Paket auf jedem vRealize Orchestrator-Server, den Sie als Endpoint in IaaS eingerichtet haben.

Informationen zum Importieren von Paketen in vRealize Orchestrator finden Sie in der vRealize Orchestrator-Dokumentation.

Erstellen eines vRealize Orchestrator -Endpoints

Sie können einen vRealize Orchestrator-Endpoint für die Verbindung zu einem vRealize Orchestrator-Server erstellen.

Sie können mehrere Endpoints für die Verbindungsherstellung mit unterschiedlichen vRealize Orchestrator-Servern konfigurieren, aber für jeden Endpoint müssen Sie die Priorität festlegen.

Beim Ausführen von vRealize Orchestrator-Workflows versucht vRealize Automation zuerst den vRealize Orchestrator Endpoint mit der höchsten Priorität. Wenn dieser Endpoint nicht erreichbar ist, folgt der Endpoint mit der nächsthöheren Priorität, bis ein vRealize Orchestrator-Server verfügbar ist, der den Workflow ausführen kann.

Voraussetzungen

- Melden Sie sich an der vRealize Automation-Konsole als **IaaS-Administrator** an.

Vorgehensweise

- 1 Wählen Sie **Infrastruktur > Endpoints > Endpoints** aus.
- 2 Wählen Sie **Neu > Orchestrierung > vRealize Orchestrator** aus.
- 3 Geben Sie einen Namen und optional eine Beschreibung ein.
- 4 Geben Sie eine URL mit dem vollqualifizierten Namen oder der IP-Adresse des vRealize Orchestrator-Servers und die vRealize Orchestrator-Portnummer ein.

Das Transportprotokoll muss HTTPS sein. Wenn kein Port angegeben wurde, wird der Standardport 443 verwendet.

Um die in der vRealize Automation-Appliance eingebettete Standard-vRealize Orchestrator-Instanz zu verwenden, geben Sie **`https://vrealize-automation-appliance-hostname:443/vco`** ein.

- 5 Geben Sie Ihre vRealize Orchestrator-Anmeldedaten in den Feldern **Benutzername** und **Kennwort** ein, um eine Verbindung zum vRealize Orchestrator-Endpoint herzustellen.

Die verwendeten Anmeldedaten sollten für alle vRealize Orchestrator-Workflows von IaaS über Berechtigungen zum Ausführen verfügen.

Um die in der vRealize Automation-Appliance eingebettete vRealize Orchestrator-Standardinstanz zu verwenden, lautet der Benutzername **`administrator@vsphere.local`** und das Kennwort ist das Administratorkennwort, das bei der Konfiguration von SSO angegeben wurde.

- 6 Geben Sie eine ganze Zahl größer oder gleich 1 in das Textfeld **Priorität** ein.
Ein niedrigerer Wert gibt eine höhere Priorität an.
- 7 (Optional) Klicken Sie auf **Eigenschaften** und fügen Sie benutzerdefinierte Eigenschaften, Eigenschaftsgruppen oder Ihre eigenen Eigenschaftsdefinitionen für den Endpoint hinzu.
- 8 Klicken Sie auf **OK**.

Zuordnen eines vRealize Orchestrator -Endpoints zu einem Blueprint

Sie können einen bestimmten vRealize Orchestrator-Endpoint zur Verwendung mit einem Blueprint angeben.

Wenn IaaS einen vRealize Orchestrator-Workflow für beliebige, über diesen Blueprint bereitgestellte Maschinen ausführt, wird immer der zugeordnete Endpoint verwendet. Wenn der Endpoint nicht erreichbar ist, schlägt der Workflow fehl.

Voraussetzungen

Melden Sie sich an der vRealize Automation-Konsole als **Infrastrukturarchitekt** an.

Vorgehensweise

- 1 Wählen Sie **Design > Blueprints** aus.
- 2 Erstellen Sie einen neuen Blueprint oder bearbeiten Sie einen vorhandenen Blueprint.
Wenn Sie einen vorhandenen Blueprint bearbeiten, gilt der von den Ihnen angegebene vRealize Orchestrator-Endpoint nur für neue Maschinen, die über den aktualisierten Blueprint bereitgestellt wurden. Über den Blueprint bereitgestellte, vorhandene Maschinen verwenden weiterhin den Endpoint mit der höchsten Priorität, es sei denn, Sie fügen diese Eigenschaft manuell zur Maschine hinzu.
- 3 Klicken Sie auf die Registerkarte **Eigenschaften**.
 - a Klicken Sie auf **Neue Eigenschaft**.
 - b Geben Sie **VMware.VCenterOrchestrator.EndpointName** im Textfeld **Name** ein.
Bei Eigenschaftsnamen wird die Groß- und Kleinschreibung berücksichtigt.
 - c Geben Sie den Namen eines vRealize Orchestrator-Endpoints im Textfeld **Wert** ein.
 - d Klicken Sie auf das Symbol **Speichern** (✔).
- 4 Klicken Sie auf **OK**.

Anpassen von IaaS-Workflows mithilfe von vRealize Automation Designer

VMware bietet eine Reihe von Workflows, die Sie mit vRealize Automation Designer anpassen können. Dazu gehören Zustandsänderungsworkflows und Menüvorgangworkflows.

IaaS-Workflows werden mithilfe von Microsoft Windows Workflow Foundation 4 (im Lieferumfang von .NET Framework 4 enthalten) erstellt. Informationen zu Windows Workflow Foundation und der Erstellung von Workflows finden Sie in der Microsoft-Dokumentation. vRealize Automation bietet außerdem mehrere vRealize Automation Designer-Aktivitäten zur Ausführung und Überwachung von vRealize Orchestrator-Workflows.

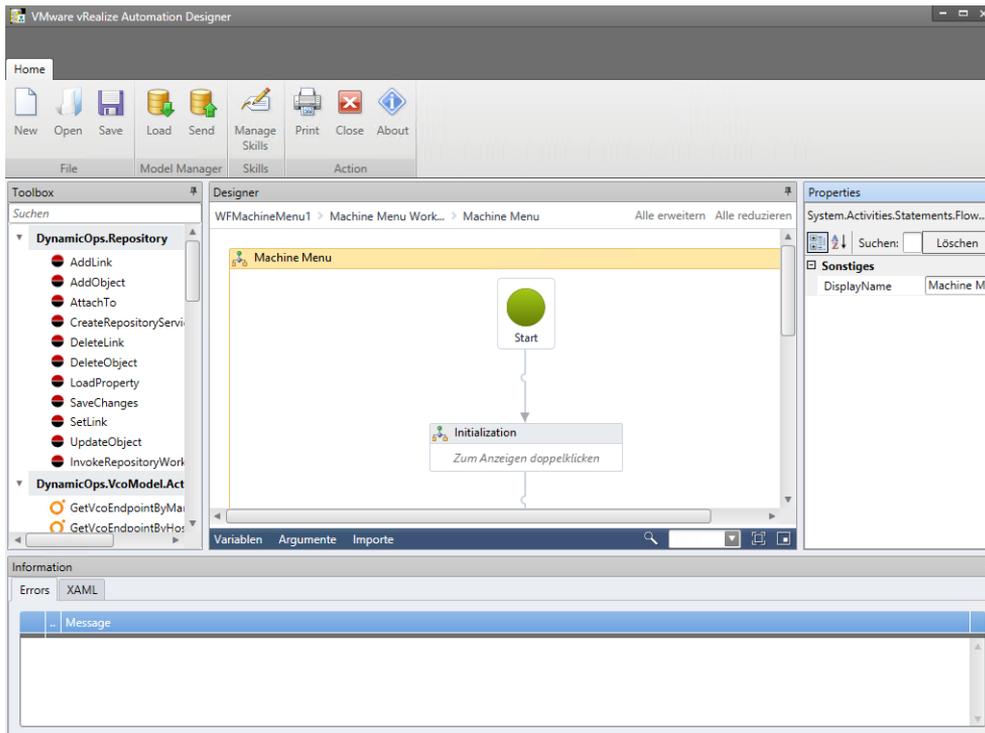
Die von VMware bereitgestellten anpassbaren Workflowvorlagen weisen Best Practices für die Strukturierung von Workflows mit separaten Sequenzen für die Initialisierung, die benutzerdefinierte Logik und den Abschluss auf. Der gesamte Workflow wird von einem TryCatch-Block für die Fehlerbehandlung umschlossen. Alle nicht abgefangenen oder zurückgesandten Ausnahmen werden von dem Distributed Execution Manager protokolliert, der den Workflow ausführt.

Nachdem Sie einen benutzerdefinierten IaaS-Workflow erstellt haben, muss ein Blueprint-Autor den Workflow in bestimmten Blueprints aktivieren.

vRealize Automation Designer -Konsole

Die vRealize Automation Designer-Konsole enthält einen grafischen Workflow-Editor zum Anpassen von IaaS-Workflows.

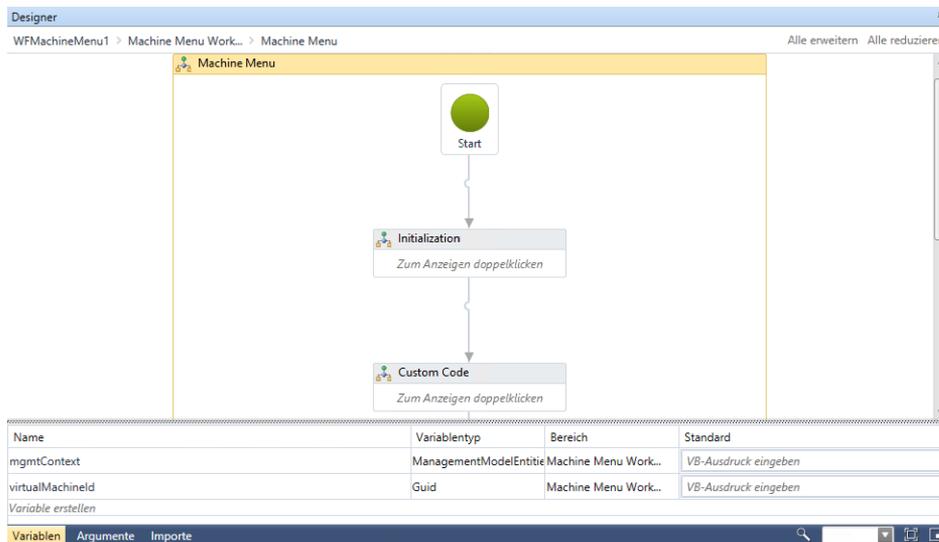
Sie benötigen lokale Administratorrechte auf dem vRealize Automation Designer-Host (in der Regel ein Entwicklercomputer), um die vRealize Automation Designer-Konsole zu starten.



Der Toolbox-Bereich auf der linken Seite ermöglicht den Zugriff auf die Bibliothek mit vRealize Automation-Workflowaktivitäten. Sie können Aktivitäten aus der Toolbox in den Designer-Bereich ziehen, um sie einem Workflow hinzuzufügen. Im Eigenschaftenbereich werden die konfigurierbaren Eigenschaften der aktuell im Designer-Bereich ausgewählten Aktivität angezeigt. Diese Benutzeroberfläche ist dem Workflow-Designer in Visual Studio sehr ähnlich.

Mithilfe der Detailregisterkarten am unteren Rand des Designer-Bereichs können Sie Variablen innerhalb des Bereichs der ausgewählten Aktivität oder Argumente für die ausgewählte Aktivität anzeigen und bearbeiten.

Hinweis Variablen und Argumente werden als Visual Basic-Ausdrücke angegeben. Bei Variablennamen wird jedoch die Groß-/Kleinschreibung nicht beachtet, während bei Argumentnamen die Groß-/Kleinschreibung beachtet wird. Informationen zu gültigen Argumenten für die IaaS-Workflowaktivitäten finden Sie in [vRealize Automation-Workflowaktivitätsreferenz](#).



Auf der Registerkarte „Importe“ werden importierte Namespaces angezeigt, unter denen Sie Entitätstypen auswählen können, die dem Workflow hinzugefügt werden sollen.

Im reduzierbaren Informationsbereich am unteren Rand der Konsole werden Fehler bei Konfigurationsaktivitäten angezeigt. Er ermöglicht außerdem den Zugriff auf die XAML-Darstellung des Workflows.

IaaS-Workflowtypen

Sie können zwei Typen von Workflows mithilfe von vRealize Automation Designer anpassen: Zustandsänderungsworkflows und Menüvorgangworkflows.

- Ein Zustandsänderungsworkflow wird ausgeführt, wenn der Masterworkflow zwischen Zuständen wechselt, z. B. in einer bestimmten Phase während des Bereitstellungsprozesses für eine neue Maschine.
- Wenn ein Benutzer eine Option aus dem Menü „Aktion“ im Servicekatalog oder dem Maschinenmenü auf der Registerkarte „Infrastruktur“ auswählt, wird ein Menüvorgangworkflow ausgeführt.

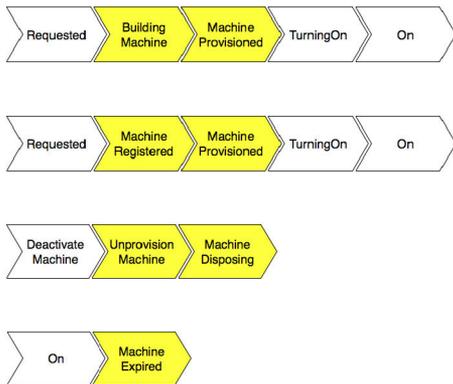
Zustandsänderungsworkflows

Durch Erstellen eines Statusänderungsworkflows können Sie die Ausführung eines Workflows auslösen, bevor der IaaS-Masterworkflow in einen bestimmten Status wechselt.

Beispielsweise können Sie benutzerdefinierte Workflows für die Integration in eine externe Datenbank erstellen und Informationen in verschiedenen Phasen des Maschinenlebenszyklus aufzeichnen:

- Erstellen Sie einen benutzerdefinierten Workflow, der ausgeführt wird, bevor der Masterworkflow in den Zustand „MachineProvisioned“ wechselt, um Informationen wie Maschinenbesitzer, Genehmiger usw. aufzuzeichnen.
- Erstellen Sie einen benutzerdefinierten Workflow, der ausgeführt wird, bevor eine Maschine in den Status „MachineDisposing“ wechselt, um den Zeitpunkt, zu dem die Maschine gelöscht wurde, sowie Daten wie etwa die Ressourcennutzung bei der letzten Datenerfassung, der letzten Anmeldung usw. aufzuzeichnen.

Die folgenden Abbildungen zeigen die wichtigsten Zustände des Masterworkflows.



vRealize Automation Designer bietet einen anpassbaren Workflow für jeden dieser Zustände.

Tabelle 1-18. Anpassbare Statusänderungsworkflows

| Masterworkflowstatus | Anpassbarer Workflowname |
|----------------------|--------------------------|
| BuildingMachine | WFStubBuildingMachine |
| Disposing | WFStubMachineDisposing |
| Expired | WFStubMachineExpired |
| MachineProvisioned | WFStubMachineProvisioned |
| RegisterMachine | WFStubMachineRegistered |
| UnprovisionMachine | WFStubUnprovisionMachine |

Konfigurieren eines Zustandsänderungsworkflows - Übersicht

Sie können einen Zustandsänderungsworkflow mithilfe von vRealize Automation Designer anpassen. Ein Blueprint-Autor kann ihn dann für bestimmte Blueprints aktivieren.

Es folgt eine Übersicht über die erforderlichen Schritte, mit denen Zustandsänderungsworkflows aktiviert werden:

- 1 Ein Workflowentwickler passt eine der Vorlagen für Zustandsänderungsworkflows mithilfe von vRealize Automation Designer an. Siehe [Anpassen eines IaaS-Workflows](#).

Jeder IaaS-Workflow kann einen vRealize Orchestrator-Workflow aufrufen. Weitere Informationen finden Sie unter [Verwenden von vRealize Orchestrator-Workflowaktivitäten](#).

- 2 Ein Mandantenadministrator oder Business-Gruppenmanager konfiguriert einen Blueprint, um den benutzerdefinierten Workflow für die von diesem Blueprint bereitgestellten Maschinen aufzurufen. Siehe [Konfigurieren eines Blueprints zum Aufrufen eines Zustandsänderungsworkflows](#).

Menüvorgangsworkflows

Wenn ein Benutzer eine Option aus dem Menü „Aktionen“ im Servicekatalog oder dem Maschinenmenü auf der Registerkarte „Infrastruktur“ auswählt, wird ein Menüvorgangsworkflow ausgeführt.

Beispielsweise können Sie einen benutzerdefinierten Workflow erstellen, mit dem Benutzer zum Erstellen eines Support-Tickets im Zusammenhang mit einer Maschine berechtigt werden. Wählen Sie hierzu „Supportproblem melden“ im Maschinenmenü aus.

vRealize Automation Designer stellt Vorlagen für das Anpassen von Menüvorgangsworkflows bereit.

Zusätzlich zur Workflowdefinition hängt ein Menüvorgangsworkflow von einer Vorgangskonfigurationsdatei ab, die die Aspekte der benutzerdefinierten Menüoption definiert, wie z. B. den Anzeigetext, welche Rollen Zugriff darauf haben, und die Maschinenzustände, für die der Vorgang zur Verfügung steht.

Hinweis Ein XaaS-Architekt kann mithilfe von XaaS benutzerdefinierte Aktionen für beliebige Katalogelemente definieren. Zum Erstellen von anderen benutzerdefinierten Aktionen als den über vSphere oder vCloud Director bereitgestellten ist vRealize Automation 6.1 oder höher erforderlich.

Konfigurieren eines Workflows – Übersicht

Sie können ein Menüvorgangsworkflow mithilfe von vRealize Automation Designer und dem Befehlszeilenprogramm CloudUtil anpassen. Ein Blueprint-Autor kann ihn dann für bestimmte Blueprints aktivieren.

Es folgt eine Übersicht über die erforderlichen Schritte, mit denen Menüvorgangsworkflows aktiviert werden:

- 1 Ein Workflowentwickler passt eine der Vorlagen für Menüvorgangsworkflows mithilfe von vRealize Automation Designer an. Siehe [Anpassen eines IaaS-Workflows](#).

Jeder IaaS-Workflow kann einen vRealize Orchestrator-Workflow aufrufen. Weitere Informationen finden Sie unter [Verwenden von vRealize Orchestrator-Workflowaktivitäten](#).

- 2 Ein Workflowentwickler konfiguriert den Menüvorgang in Model Manager. Siehe [Konfigurieren eines Menüvorgangs](#).
- 3 Ein Workflowentwickler registriert den neuen Menüvorgang im Servicekatalog. Siehe [Registrieren neuer Menüvorgänge im Servicekatalog](#).

- Ein Mandantenadministrator oder Business-Gruppenmanager konfiguriert einen Blueprint, um den Menüvorgang für die von diesem Blueprint bereitgestellten Maschinen zu aktivieren. Siehe [Konfigurieren eines Blueprints zum Aktivieren eines Menüvorgangworkflows](#).

Wenn der Menüvorgang im Servicekatalog verwendet werden soll, müssen dafür auch Berechtigungen für Benutzer erteilt werden. Weitere Informationen finden Sie unter *Mandantenverwaltung*.

Anpassen eines IaaS-Workflows

vRealize Automation Designer ermöglicht das Bearbeiten der anpassbaren Workflows und das Aktualisieren von Workflows in Model Manager.

Voraussetzungen

Starten Sie vRealize Automation Designer.

Vorgehensweise

- Klicken Sie auf **Laden**.
- Wählen Sie den anzupassenden Workflow aus.

| Option | Beschreibung |
|---------------------------------|--|
| WFMachineMenuN | Anpassbarer Menüvorgangworkflow |
| WFStubBuildingMachine | Anpassbarer Zustandsänderungswflow, der ausgeführt wird, bevor eine Maschine in den BuildingMachine-Zustand wechselt. |
| WFStubMachineDisposing | Anpassbarer Zustandsänderungswflow, der ausgeführt wird, bevor eine Maschine in den Disposing-Zustand wechselt. |
| WFStubMachineExpired | Anpassbarer Zustandsänderungswflow, der ausgeführt wird, bevor eine Maschine in den Expired-Zustand wechselt. |
| WFStubMachineProvisioned | Anpassbarer Zustandsänderungswflow, der ausgeführt wird, bevor eine Maschine in den MachineProvisioned-Zustand wechselt. |
| WFStubMachineRegistered | Anpassbarer Zustandsänderungswflow, der ausgeführt wird, bevor eine Maschine in den RegisterMachine-Zustand wechselt. |
| WFStubUnprovisionMachine | Anpassbarer Zustandsänderungswflow, der ausgeführt wird, bevor eine Maschine in den UnprovisionMachine-Zustand wechselt. |

- Klicken Sie auf **OK**.
Der Workflow wird im Bereich „Designer“ angezeigt.
- Passen Sie den Workflow an, indem Sie Aktivitäten aus der Toolbox in den Bereich „Designer“ ziehen und deren Argumente konfigurieren.
- Wenn Sie die Bearbeitung des Workflows abgeschlossen haben, aktualisieren Sie den Workflow in Model Manager, indem Sie auf **Senden** klicken.

Der Workflow wird gespeichert und beim nächsten Laden eines Workflows als eine neue Version in der Liste angezeigt. Sie können jederzeit auf eine frühere Version eines Workflows zugreifen. Siehe [Zurücksetzen auf eine frühere Version eines Workflows](#).

Verwenden von vRealize Orchestrator -Workflowaktivitäten

Mit den vRealize Automation Designer-Aktivitäten können Sie vRealize Orchestrator-Workflows synchron oder asynchron aufrufen.

Ein vRealize Orchestrator-Endpoint wird mithilfe einer der folgenden Methoden angegeben:

- `VirtualMachineId` ist der Name der Variablen, die die virtuelle Maschinen-ID darstellt. Eine virtuelle Maschine mit dieser ID wird ausgewählt, und der Wert, der aus der benutzerdefinierten Eigenschaft `VMware.VCenterOrchestrator.EndpointName` für eine virtuelle Maschine abgerufen wird, wird als vRealize Orchestrator-Endpoint-Name verwendet.
- `GetVcoEndpointByManagementEndpoint` gibt den Wert einer benutzerdefinierten Eigenschaft in einem angegebenen `ManagementEndpoint`-Objekt zurück. Wenn `CustomPropertyName` nicht angegeben ist, wird der Wert der `VMware.VCenterOrchestrator.EndpointName`-Eigenschaft verwendet.
- `GetVcoEndpointByHost` gibt den Wert einer benutzerdefinierten Eigenschaft auf einem angegebenen Host zurück. Wenn `CustomPropertyName` nicht angegeben ist, wird der Wert der `VMware.VCenterOrchestrator.EndpointName`-Eigenschaft verwendet.

Synchrone Ausführung

Die `InvokeVcoWorkflow`-Aktivität ruft einen vRealize Orchestrator-Workflow auf und blockiert die weitere Ausführung des übergeordneten IaaS-Workflows, bis der vRealize Orchestrator-Workflow abgeschlossen wurde. Diese Aktivität gibt die Ausgabeparameter für den vRealize Orchestrator-Workflow zurück.

Darüber hinaus unterstützt die synchrone Ausführung die folgende Eigenschaft:

- `WorkflowTimeout` ist ein Zeitüberschreitungswert in Sekunden. Wenn der vRealize Orchestrator-Workflow nicht in der angegebenen Zeit abgeschlossen wird, wird eine Ausnahme generiert, anstatt den Workflow so lange zu blockieren, bis eine Antwort zurückgegeben wird. Wenn kein Wert definiert ist, oder der Wert null bereitgestellt wird, wird die Zeitüberschreitung nicht aktiviert. Der Workflowstatus wird während dieses Zeitraums alle 10 Sekunden überprüft, es sei denn, die Abfragezeit für den Endpoint wird durch die Angabe eines Werts in der benutzerdefinierten Eigenschaft `VMware.VCenterOrchestrator.PollingInterval` geändert.

Asynchrone Ausführung

Die `InvokeVcoWorkflowAsync`-Aktivität ist eine „Auslösen und Vergessen“-Aktivität, die einen vRealize Orchestrator-Workflow aufruft und die Ausführung von Aktivitäten im IaaS-Workflow fortsetzt, ohne abzuwarten, bis der vRealize Orchestrator-Workflow abgeschlossen wurde.

Diese Aktivität gibt entweder ein eindeutiges Workflowausführungstoken zurück, mit dem der Workflow überwacht werden kann, oder aber einen Fehler, falls der REST API-Aufruf des vRealize Orchestrator-Servers fehlgeschlagen ist (z. B., wenn der Server nicht erreichbar war).

Zwei zusätzliche Aktivitäten sind für die Verwendung mit dieser Aktivität verfügbar:

- `GetVcoWorkflowExecutionStatus` ermöglicht das Abrufen des Status für den vRealize Orchestrator-Workflow.

- `WaitForVcoWorkflowCompletion` ermöglicht das Blockieren der weiteren Ausführung des IaaS-Workflows, bis der vRealize Orchestrator-Workflow abgeschlossen wurde oder eine Zeitüberschreitung eingetreten ist. Mithilfe dieser Aktivität können Sie die Ergebnisse eines vRealize Orchestrator-Workflows abrufen, den Sie asynchron ausführen.

Aufrufen eines vRealize Orchestrator -Workflows

Sie können die Aktivität `InvokeVcoWorkflow` oder `InvokeVcoWorkflowAsync` verwenden, um einen vRealize Orchestrator-Workflow aus einem IaaS-Workflow aufzurufen.

Einige vRealize Orchestrator-Workflows erfordern die Benutzerinteraktion während der Ausführung. Für diese Workflows wird die Benutzereingabeaufforderung im vRealize Orchestrator-Client anstatt in der vRealize Automation-Konsole angezeigt. Der Endbenutzer in vRealize Automation kann also nicht erkennen, dass ein Workflow auf eine Eingabe wartet.

Um Workflows zu vermeiden, die bei Benutzereingabe blockieren, rufen Sie keine vRealize Orchestrator-Workflows, die eine Benutzerinteraktion erfordern, über IaaS-Workflows auf.

Vorgehensweise

- 1 Öffnen Sie in vRealize Automation Designer einen Workflow und navigieren Sie zu dem Kontext, in dem Sie einen vRealize Orchestrator-Workflow aufrufen möchten.
- 2 Ziehen Sie die Aktivität `InvokeVcoWorkflow` oder `InvokeVcoWorkflowAsync` in den Bereich „Designer“.
- 3 Wählen Sie den auszuführenden vCenter Orchestrator-Workflow aus.
 - a Klicken Sie unter „Allgemein“ auf das Auslassungszeichen neben „Workflow“.
 - b Wählen Sie im Dialogfeld „Nach vCO-Workflow suchen“ einen Workflow aus.
 - c Klicken Sie auf **OK**.

In den Abschnitten „Eingaben“ und „Ausgaben“ werden die Eingabe- und Ausgabeparameter des ausgewählten Workflows angezeigt.

4 Geben Sie im Bereich „Eigenschaften“ einen der folgenden Zielparameter ein.

- `VirtualMachineId` ist der Name der Variablen, die die virtuelle Maschinen-ID darstellt. Eine virtuelle Maschine mit dieser ID wird ausgewählt, und der Wert, der aus der benutzerdefinierten Eigenschaft `VMware.VCenterOrchestrator.EndpointName` für eine virtuelle Maschine abgerufen wird, wird als vRealize Orchestrator-Endpoint-Name verwendet.
- `VcoEndpointName` ist der Endpoint-Name, der zum Ausführen des Workflows verwendet wird. Wenn dieser Parameter angegeben ist, überschreibt dieser Wert den `VirtualMachineId`-Wert, wenn Sie den vRealize Orchestrator-Endpoint auswählen.
- `WorkflowTimeout` ist ein Zeitüberschreitungswert in Sekunden. Wenn der vRealize Orchestrator-Workflow nicht in der angegebenen Zeit abgeschlossen wird, wird eine Ausnahme generiert, anstatt den Workflow so lange zu blockieren, bis eine Antwort zurückgegeben wird. Wenn kein Wert definiert ist, oder der Wert null bereitgestellt wird, wird die Zeitüberschreitung nicht aktiviert. Der Workflowstatus wird während dieses Zeitraums alle 10 Sekunden überprüft, es sei denn, die Abfragezeit für den Endpoint wird durch die Angabe eines Werts in der benutzerdefinierten Eigenschaft `VMware.VCenterOrchestrator.PollingInterval` geändert.

5 Geben Sie die Parameter für den vRealize Orchestrator-Workflow an.

- Geben Sie die Werte in der Aktivität im Bereich „Designer“ ein.
- Klicken Sie im Bereich „Eigenschaften“ auf das Auslassungszeichen neben **InputParameters** oder **OutputParameters**, um das Dialogfeld „Parameter“ zu öffnen. In diesem Dialogfeld wird der IaaS-Typ der einzelnen Parameter angezeigt. Wenn der Parametertyp fett formatiert ist, ist der Parameter erforderlich.

Zeigen Sie auf das Textfeld für einen beliebigen Parameter, um eine QuickInfo einzublenden, die den vRealize Orchestrator-Typ angibt.

Wenn Sie die Aktivität `InvokeVcoWorkflowAsync` verwenden, werden die Ausgabeparameter des vRealize Orchestrator-Workflows mit den entsprechenden Typen für Informationszwecke angezeigt, aber Sie können keinen Ausdruck für den Parameter in dieser Aktivität angeben.

Weiter

Um die Ergebnisse eines Workflows abzurufen, den Sie asynchron ausführen, verwenden Sie die Aktivität `WaitForVcoWorkflowCompletion`.

Abrufen des Status eines vRealize Orchestrator -Workflows

Sie können den Status eines vRealize Orchestrator-Workflows überprüfen, der mit der `InvokeVcoWorkflowAsync`-Aktivität mithilfe der `GetVcoWorkflowExecutionStatus`-Aktivität aufgerufen wurde.

Voraussetzungen

[Aufrufen eines vRealize Orchestrator-Workflows](#) mithilfe der `InvokeVcoWorkflowAsync`-Aktivität.

Vorgehensweise

- 1 Öffnen Sie in vRealize Automation Designer einen Workflow, in dem Sie die `InvokeVcoWorkflowAsync`-Aktivität verwendet haben.

- 2 Navigieren Sie zu dem Kontext, in dem Sie den Status des vRealize Orchestrator-Workflows überprüfen möchten.
- 3 Ziehen Sie die `GetVcoWorkflowExecutionStatus`-Aktivität in den Bereich „Designer“.
- 4 Geben Sie im Bereich „Eigenschaften“ den Namen der Variablen ein, die die virtuelle Maschinen-ID in `VirtualMachineId` darstellt.

Anpassbare Workflows enthalten eine Variable, die standardmäßig mit dem Namen `virtualMachineId` bezeichnet wird, der während der Initialisierung festgelegt wird.
- 5 Erstellen Sie eine Variable des Typs `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken`.
- 6 Geben Sie den Namen der Token-Variablen als `executionToken`-Ausgabeparameter für die `InvokeVcoWorkflowAsync`-Aktivität an.
- 7 Geben Sie denselben Variablennamen als `WorkflowExecutionToken`-Eigenschaft der `GetVcoWorkflowExecutionStatus`-Aktivität an.
- 8 Erstellen Sie eine Variable des Typs „String“.
- 9 Geben Sie den Namen der String-Variablen als `VcoWorkflowExecutionStatus`-Eigenschaft der `GetVcoWorkflowExecutionStatus`-Aktivität an.

Wenn der Workflow ausgeführt wird, wird der Wert der `VcoWorkflowExecutionStatus`-Variablen auf den Status des vRealize Orchestrator-Workflows festgelegt.

Abrufen der Ergebnisse eines vRealize Orchestrator -Workflows

Wenn Sie einen vRealize Orchestrator-Workflow asynchron aufrufen und dann die Ergebnisse des abgeschlossenen Workflows zu einem späteren Zeitpunkt abrufen möchten, können Sie die `WaitForVcoWorkflowCompletion`-Aktivität verwenden.

Die `WaitForVcoWorkflowCompletion`-Aktivität blockiert die Ausführung des IaaS-Workflows, bis der vRealize Orchestrator-Workflow abgeschlossen ist oder eine Zeitüberschreitung eintritt. Die Aktivität gibt die Ergebnisse des vRealize Orchestrator-Workflows zurück, wenn er erfolgreich abgeschlossen wurde, bzw. einen Fehler, wenn der Workflow fehlschlägt, oder null, wenn eine Zeitüberschreitung für den Workflow eintritt.

Voraussetzungen

[Aufrufen eines vRealize Orchestrator-Workflows](#) mithilfe der `InvokeVcoWorkflowAsync`-Aktivität.

Vorgehensweise

- 1 Öffnen Sie in vRealize Automation Designer einen Workflow, in dem Sie die `InvokeVcoWorkflowAsync`-Aktivität verwendet haben.
- 2 Navigieren Sie zu dem Kontext, in dem Sie die Ergebnisse des vRealize Orchestrator-Workflows abrufen möchten.
- 3 Ziehen Sie die `WaitForVcoWorkflowCompletion`-Aktivität in den Bereich „Designer“.

- 4 Geben Sie im Bereich „Eigenschaften“ den Namen der Variablen ein, die die virtuelle Maschinen-ID in `VirtualMachineId` darstellt.

Anpassbare Workflows enthalten eine Variable, die standardmäßig mit dem Namen `virtualMachineId` bezeichnet wird, der während der Initialisierung festgelegt wird.

- 5 Erstellen Sie eine Variable des Typs „DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken“.
- 6 Erstellen Sie eine Variable des Typs `DynamicOps.VcoModel.Common.VcoWorkflowExecutionToken`.
- 7 Geben Sie den Namen der Token-Variablen als `executionToken`-Ausgabeparameter für die `InvokeVcoWorkflowAsync`-Aktivität an.
- 8 Geben Sie denselben Variablennamen als `WorkflowExecutionToken`-Eigenschaft der `WaitForVcoWorkflowCompletion`-Aktivität an.
- 9 Rufen Sie die Ausgabe des vRealize Orchestrator-Workflows ab.
- Erstellen Sie eine Variable des Typs `DynamicOps.VcoModel.Common.VcoWorkflowExecutionResult`.
 - Geben Sie den Namen der Ergebnisvariablen als `WorkflowOutput`-Eigenschaft der `WaitForVcoWorkflowCompletion`-Aktivität an.

Wenn der Workflow ausgeführt wird, wird der Wert der Variablen auf die Ergebnisse des vRealize Orchestrator-Workflows, sofern vorhanden, festgelegt.

vRealize Orchestrator - und IaaS-Objekttypen

Wenn Sie entweder die `InvokeVcoWorkflow`-Aktivität oder die `InvokeVcoWorkflowAsync`-Aktivität in vRealize Automation Designer verwenden, werden die Eingabe- und Ausgabeeigenschaften für die Aktivität automatisch basierend auf den Parametern für den ausgewählten vRealize Orchestrator-Workflow aufgefüllt.

Einfache vRealize Orchestrator-Typen werden in die folgenden IaaS-Typen konvertiert:

Tabelle 1-19. vRealize Orchestrator - und IaaS-Objekttypen

| vRealize Orchestrator-Typ | IaaS-Typ |
|---------------------------|---------------------------|
| string | string |
| boolean | bool |
| number | decimal |
| SecureString | string |
| Text | string |
| Array/T | Array<T> |
| Properties | Dictionary<string,object> |

Tabelle 1-19. vRealize Orchestrator - und IaaS-Objekttypen (Fortsetzung)

| vRealize Orchestrator-Typ | IaaS-Typ |
|---------------------------|----------------|
| Date | DateTime |
| VC:VirtualMachine | VirtualMachine |

Hinweis Wenn Sie vRealize Orchestrator 5.1 verwenden, müssen Sie das vRealize Automation-Integrationspaket installiert haben, um die Konvertierung von VC:VirtualMachine-Objekttypen in VirtualMachine zu ermöglichen.

Alle anderen vRealize Orchestrator-Typen werden in den IaaS-Typ VcoSdkObject konvertiert.

Konfigurieren eines Blueprints zum Aufrufen eines Zustandsänderungsworkflows

Nachdem Sie einen benutzerdefinierten Zustandsänderungsworkflow erstellt haben, muss ein Mandantadministrator oder Business-Gruppenmanager diesen für bestimmte Blueprints aktivieren, indem eine benutzerdefinierte Eigenschaft hinzugefügt wird.

Jedem Zustandsänderungsworkflow ist eine bestimmte benutzerdefinierte Eigenschaft zugeordnet. Wenn eine Maschine in einen Zustand mit einem entsprechenden Zustandsänderungsworkflow wechselt, prüft IaaS, ob die Maschine über die entsprechende benutzerdefinierte Eigenschaft verfügt. Falls dies der Fall ist, wird der zugeordnete Workflow ausgeführt. Wenn z. B. eine Maschine über die benutzerdefinierte Eigenschaft ExternalWFStubs.MachineProvisioned verfügt, wird der WFStubMachineProvisioned-Workflow ausgeführt, bevor der Masterworkflow in den Zustand „MachineProvisioned“ wechselt.

Benutzerdefinierte Eigenschaften können aus einer Reihe von Quellen auf eine Maschine angewendet werden. In der Regel wird die Eigenschaft für einen Zustandsänderungsworkflow aber in einem Blueprint angegeben, der den Workflow für alle Maschinen aktiviert, die über diesen Blueprint bereitgestellt werden.

Voraussetzungen

Melden Sie sich an der vRealize Automation-Konsole als **Mandantenadministrator** oder **Business-Gruppenmanager** an.

Vorgehensweise

- 1 Wählen Sie **Design > Blueprints** aus.
- 2 Zeigen Sie auf den Namen eines Blueprints und klicken Sie auf **Bearbeiten**.
- 3 Klicken Sie auf die Registerkarte **Eigenschaften**.
- 4 Klicken Sie auf **Neue Eigenschaft**.

- 5 Geben Sie den Namen der benutzerdefinierten Eigenschaft, die dem zu aktivierenden Workflow zugeordnet ist, im Textfeld **Name** ein.

| Anpassbarer Workflowname | Zugeordneter Eigenschaftsname |
|--------------------------|------------------------------------|
| WFStubMachineProvisioned | ExternalWFStubs.MachineProvisioned |
| WFStubBuildingMachine | ExternalWFStubs.BuildingMachine |
| WFStubMachineDisposing | ExternalWFStubs.MachineDisposing |
| WFStubUnprovisionMachine | ExternalWFStubs.UnprovisionMachine |
| WFStubMachineRegistered | ExternalWFStubs.MachineRegistered |
| WFStubMachineExpired | ExternalWFStubs.MachineExpired |

- 6 Lassen Sie das Textfeld **Wert** leer.

Der Workflow hängt vom Vorhandensein der Eigenschaft ab, nicht von einem bestimmten Wert.

- 7 Klicken Sie auf das Symbol **Speichern** (✔).

- 8 Klicken Sie auf **OK**.

Der Workflow ist jetzt für neue Maschinen aktiviert, die über diesen Blueprint bereitgestellt werden.

Konfigurieren eines Menüvorgangworkflows

Nachdem Sie einen Menüvorgangworkflow angepasst haben, sind zusätzliche Konfigurationsschritte erforderlich, bevor der Menüvorgang Benutzern in der vRealize Automation-Konsole zur Verfügung steht.

Konfigurieren eines Menüvorgangs

Um einen Menüvorgang zu konfigurieren, erstellen Sie eine Vorgangskonfigurationsdatei und installieren sie im Model Manager.

Vorgehensweise

1 Erstellen einer Vorgangskonfigurationsdatei

Die Vorgangskonfigurationsdatei ist für Menüvorgangworkflows erforderlich. Hiermit werden die Aspekte der benutzerdefinierten Menüoption in der vRealize Automation-Konsole angegeben, wie beispielsweise der Anzeigetext, welche Rollen Zugriff auf die Option haben sowie die Maschinenstatus, für die die Option verfügbar ist.

2 Installieren eines Vorgangs im Model Manager

Mithilfe des Befehlszeilenprogramms CloudUtil können Sie einen Vorgang im Model Manager installieren.

Weiter

Wenn der Menüvorgang im Servicekatalog verwendet werden soll, muss er beim Servicekatalog registriert werden, damit Benutzern die Berechtigung für diesen Menüvorgang erteilt werden kann. [Registrieren neuer Menüvorgänge im Servicekatalog](#).

Erstellen einer Vorgangskonfigurationsdatei

Die Vorgangskonfigurationsdatei ist für Menüvorgangsworkflows erforderlich. Hiermit werden die Aspekte der benutzerdefinierten Menüoption in der vRealize Automation-Konsole angegeben, wie beispielsweise der Anzeigetext, welche Rollen Zugriff auf die Option haben sowie die Maschinenstatus, für die die Option verfügbar ist.

Vorgehensweise

- 1 Erstellen Sie eine neue XML-Datei.

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

- 2 Erstellen Sie das Root-Element customOperations.

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

Für das Element muss der XML-Namespace `http://www.dynamicops.com/schemas/2009/OperationConfig/` angegeben werden.

- 3 Fügen Sie für jeden Vorgang, den Sie definieren möchten, ein operation-Element in customOperations hinzu.

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

Für das operation-Element sind die folgenden Attribute verfügbar:

| Attribut | Beschreibung |
|--------------------|---|
| name | Der Name des Workflows, der von diesem Vorgang ausgeführt wird. |
| displayName | Eine Beschreibung für die Option im Maschinenmenü. |

4 Geben Sie die Rollen an, denen der Zugriff auf den Menüvorgang gewährt werden soll.

a Fügen Sie das `authorizedTasks`-Element hinzu.

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

b Fügen Sie für jede Rolle, der Sie Zugriff auf den Vorgang gewähren möchten, ein `task`-Element hinzu. Beispiel:

```
<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>
```

Dies sind die gültigen Inhalte des `task`-Elements:

| Elementinhalt | Beschreibung |
|--|---|
| VRM User Custom Event | Gewährt allen Benutzern den Zugriff auf den Vorgang. |
| VRM Support Custom Event | Gewährt Supportbenutzern den Zugriff auf den Vorgang. |
| Group Administrator Custom Event | Gewährt Business-Gruppenmanagern den Zugriff auf den Vorgang. |
| Enterprise Administrator Custom Event | Gewährt Fabric-Administratoren den Zugriff auf den Vorgang. |
| VRM Administrator Custom Event | Gewährt nur IaaS-Administratoren den Zugriff auf den Vorgang. |

5 (Optional) Geben Sie die Maschinenstatus an, für die der Vorgang verfügbar ist.

a Fügen Sie das `machineStates`-Element hinzu.

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

b Fügen Sie für jeden Status, in dem der Vorgang verfügbar sein soll, ein `state`-Element hinzu.

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

Als Wert kommt einer der möglichen Maschinenstatus in Frage. Eine vollständige Aufstellung der Maschinenstatus finden Sie unter *IaaS-Konfiguration für virtuelle Plattformen*, *IaaS-Konfiguration für physische Maschinen* oder *IaaS-Konfiguration für Cloud-Plattformen*.

Wenn dieses Element ausgelassen wird, ist der Vorgang für alle Maschinenstatus verfügbar.

Es folgt ein Beispiel für eine vollständige Vorgangskonfigurationsdatei:

```
<?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>
```

Installieren eines Vorgangs im Model Manager

Mithilfe des Befehlszeilenprogramms `CloudUtil` können Sie einen Vorgang im Model Manager installieren.

Voraussetzungen

[Erstellen einer Vorgangskonfigurationsdatei.](#)

Vorgehensweise

1 Öffnen Sie eine Eingabeaufforderung mit erweiterten Berechtigungen.

2 Führen Sie den Befehl CloudUtil.exe mit folgenden Argumenten aus.

- ```
CloudUtil.exe Operation-Create -c <Pfad zu Vorgangsdefinitionsdatei>
```
- Optional können Sie einen Model Manager-Host angeben und bei einem Fehler eine Stapelüberwachung anfordern.

```
CloudUtil.exe Operation-Create -c <Pfad zu Vorgangsdefinitionsdatei>
--repository <Model Manager-Root-URI> -v
```

### Weiter

Wenn der Menüvorgang im Servicekatalog verwendet werden soll, muss er beim Servicekatalog registriert werden, damit Benutzern die Berechtigung für diesen Menüvorgang erteilt werden kann. [Registrieren neuer Menüvorgänge im Servicekatalog](#).

### Registrieren neuer Menüvorgänge im Servicekatalog

Nach der Installation von neuen Menüvorgängen muss der Workflowentwickler sie im Servicekatalog registrieren, damit Benutzern Berechtigungen für sie erteilt werden können.

### Voraussetzungen

- [Konfigurieren eines Menüvorgangs](#).
- Melden Sie sich am der IaaS Model Manager-Host als lokaler Benutzer mit **Administratorrechten** an.

### Vorgehensweise

- 1 Öffnen Sie eine Eingabeaufforderung mit erweiterten Berechtigungen.
- 2 Navigieren Sie zum IaaS-Root-Installationsverzeichnis.  
In einer typischen-Installation ist dies C:\Programme (x86)\VMware\vCAC.
- 3 Navigieren Sie zu Server\Model Manager Data\Cafe.
- 4 Führen Sie den folgenden Befehl aus:

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

### Weiter

Ein Mandantenadministrator oder Business-Gruppenmanager muss eine Berechtigung für die neue Aktion erteilen, bevor sie Benutzern im Servicekatalog zur Verfügung steht. Weitere Informationen finden Sie unter *Mandantenverwaltung*.

### Konfigurieren eines Blueprints zum Aktivieren eines Menüvorgangworkflows

Ein Menüvorgangworkflow wird für über einen bestimmten Blueprint bereitgestellte Maschinen aktiviert, indem die Sicherheitskonfiguration für den Blueprint aktualisiert wird.

### Voraussetzungen

Melden Sie sich an der vRealize Automation-Konsole als **Mandantenadministrator** oder **Business-Gruppenmanager** an.

### Vorgehensweise

- 1 Wählen Sie **Design > Blueprints** aus.
- 2 Zeigen Sie auf den Namen eines Blueprints und klicken Sie auf **Bearbeiten**.
- 3 Klicken Sie auf die Registerkarte **Aktionen**.
- 4 Aktivieren Sie das Kontrollkästchen, das dem Vorgang entspricht, den Sie aktivieren möchten.
- 5 Klicken Sie auf **OK**.

Der Menüvorgang ist jetzt für alle über diesen Blueprint bereitgestellten Maschinen aktiviert und steht allen Benutzerrollen zur Verfügung, die in der Vorgangskonfigurationsdatei angegeben sind.

### Weiter

Wenn der Menüvorgang im Servicekatalog verwendet werden soll, müssen dafür auch Berechtigungen für Benutzer erteilt werden. Weitere Informationen finden Sie unter *Mandantenverwaltung*.

## Zurücksetzen auf eine frühere Version eines Workflows

Das Dialogfeld „**Workflow laden**“ zeigt alle Versionen eines Workflows in Model Manager an, damit Sie Zugriff auf den vollständigen Versionsverlauf der Workflows haben.

Jedes Mal, wenn Sie einen Workflow an Model Manager senden, werden die Version und der Zeitstempel aktualisiert.

### Voraussetzungen

Starten Sie die vRealize Automation Designer-Konsole.

### Vorgehensweise

- 1 Klicken Sie auf **Laden**.
- 2 Wählen Sie die Version des Workflows, auf die Sie diesen zurücksetzen möchten.  
Die ursprünglichen Workflows, die von VMware bereitgestellt werden, weisen die Version 0 (null) auf.
- 3 Klicken Sie auf **OK**.
- 4 Aktualisieren Sie den Workflow in Model Manager, indem Sie auf **Senden** klicken.

Die frühere Version wird zur neuesten Version in Model Manager. Wenn Sie beispielsweise die Versionen 1 und 2 eines Workflows erstellt haben und dann die Version 0 laden und speichern, sind die Versionen 0 und 3 identisch, und Sie haben den Workflow auf die von VMware bereitgestellte Version zurückgesetzt.

## Workflows und Distributed Execution Management

Sie können Qualifikationen verwenden, um die Ausführung von Workflows auf bestimmte Distributed Execution Manager zu beschränken.

Eine Qualifikation ist mit einem Tag vergleichbar, das Sie sowohl auf Workflows als auch auf DEM Worker-Instanzen anwenden können. Wenn einem Workflow keine Qualifikationen zugeordnet sind, kann er von jedem DEM-Worker ausgeführt werden. Wenn einem Workflow eine oder mehrere Qualifikationen zugeordnet sind, kann er nur von DEM-Workern ausgeführt werden, die jeder der Qualifikationen zugeordnet sind.

Qualifikationen sind nützlich, wenn für einen bestimmten Workflow ein DEM auf einem Host mit bestimmten Voraussetzungen installiert werden muss. Beispiel: Sie möchten Cloud-Bereitstellungsworkflows auf einen bestimmten DEM beschränken, der auf einem Host mit dem erforderlichen Netzwerkzugriff auf Amazon-URLs ausgeführt wird.

Qualifikationen können auch verwendet werden, um Workflows einem bestimmten Datacenter-Standort zuzuordnen. Sie können beispielsweise einen DEM in Ihrem Datacenter in Boston und einen anderen in Ihrem Datacenter in London installieren und Vorgänge anhand von Qualifikationen an ein bestimmtes Datacenter weiterleiten.

## Zuordnen von Workflows und DEM-Workern mithilfe von Qualifikationen

Sie ordnen Workflows einem bestimmten DEM-Worker oder einem Satz von Worker-Instanzen zu, indem Sie eine Qualifikation in Model Manager hinzufügen und diese dann einem oder mehreren Workflows und DEM-Workern zuordnen.

### Voraussetzungen

Starten Sie die vRealize Automation Designer-Konsole.

### Vorgehensweise

- 1 Klicken Sie im Menüband auf **Qualifikationen verwalten**.
- 2 Geben Sie im Textfeld oben links im Dialogfeld **Qualifikationen verwalten** den Namen einer neuen Qualifikation ein und klicken Sie auf die Schaltfläche „Hinzufügen“.  
  
Der Qualifikationsname muss eindeutig sein. Wenn der Name der neuen Qualifikation mit dem Namen einer vorhandenen Qualifikation übereinstimmt, ist die Schaltfläche „Hinzufügen“ nicht verfügbar.
- 3 Wählen Sie den Namen der Qualifikation in der Liste auf der linken Seite aus.
- 4 Ordnen Sie die Qualifikation einem oder mehreren DEM-Workern zu.
  - a Klicken Sie neben „Distributed Execution Manager“ auf das Symbol **Hinzufügen** ().
  - b Wählen Sie im Dialogfeld **DEMs auswählen** eine oder mehrere DEM-Worker-Instanzen aus.
  - c Klicken Sie auf **OK**.

5 Ordnen Sie die Qualifikation einem oder mehreren Workflows zu.

- a Klicken Sie neben „Workflows“ auf das Symbol **Hinzufügen** (.
- b Wählen Sie im Dialogfeld **Workflows auswählen** einen oder mehrere Workflows aus.
- c Klicken Sie auf **OK**.

Die Workflows, die dieser Qualifikation zugeordnet sind, können nur von den DEM-Workern ausgeführt werden, die dieser Qualifikation zugeordnet sind.

6 Wenn Sie das Hinzufügen von Qualifikationen und Zuordnen zu DEM-Workern und Workflows abgeschlossen haben, klicken Sie auf **OK**, um das Dialogfeld **Qualifikationen verwalten** zu schließen und Ihre Änderungen in Model Manager zu speichern.

## Entfernen von Zuordnungen zwischen Qualifikationen und DEM-Workern

Wenn Sie die Zuordnung zwischen einer Qualifikation und einem DEM-Worker entfernen, kann diese Worker-Instanz die mit der Qualifikation verbundenen Workflows nicht mehr ausführen.

### Voraussetzungen

Starten Sie die vRealize Automation Designer-Konsole.

### Vorgehensweise

- 1 Klicken Sie im Menüband auf **Qualifikationen verwalten**.
- 2 Wählen Sie im Dialogfeld **Qualifikationen verwalten** den Namen der Qualifikation in der Liste auf der linken Seite aus.
- 3 Wählen Sie in der Distributed Execution Manager-Liste den Namen einer oder mehrerer DEM-Worker-Instanzen aus und klicken Sie auf das Symbol **Entfernen** (.
- 4 Klicken Sie auf **OK**, um das Dialogfeld **Qualifikationen verwalten** zu schließen und Ihre Änderungen am Model Manager zu speichern.

## Entfernen von Zuordnungen zwischen Qualifikationen und Workflows

Wenn Sie die Zuordnung zwischen einer Qualifikation und einem Workflow entfernen, ist dieser Workflow nicht mehr auf die DEM-Worker beschränkt, die dieser Qualifikation zugeordnet sind.

### Voraussetzungen

Starten Sie die vRealize Automation Designer-Konsole.

### Vorgehensweise

- 1 Klicken Sie im Menüband auf **Qualifikationen verwalten**.
- 2 Wählen Sie im Dialogfeld **Qualifikationen verwalten** den Namen der Qualifikation in der Liste auf der linken Seite aus.

- 3 Wählen Sie den Namen eines oder mehrerer Workflows in der Workflowliste aus und klicken Sie auf das Symbol **Entfernen** ().
- 4 Klicken Sie auf **OK**, um das Dialogfeld **Qualifikationen verwalten** zu schließen und Ihre Änderungen am Model Manager zu speichern.

## Entfernen einer Qualifikation

Durch das Entfernen einer Qualifikation werden auch deren Zuordnungen zu DEM-Workern und Workflows entfernt.

### Voraussetzungen

Starten Sie die vRealize Automation Designer-Konsole.

### Vorgehensweise

- 1 Klicken Sie im Menüband auf **Qualifikationen verwalten**.
- 2 Wählen Sie im Dialogfeld **Qualifikationen verwalten** den Namen der Qualifikation in der Liste auf der linken Seite aus.
- 3 Klicken Sie auf das Symbol **Entfernen** () oben in der Liste der Qualifikationen.  
Nachdem Sie bestätigt haben, dass Sie die Qualifikation löschen möchten, wird deren Name abgeblendet dargestellt, um anzugeben, dass sie zum Löschen markiert ist.
- 4 Klicken Sie auf **OK**, um das Dialogfeld **Qualifikationen verwalten** zu schließen und die Änderungen in Model Manager zu speichern. Klicken Sie auf **Abbrechen**, wenn Sie die Qualifikation und deren Zuordnungen zu DEMs und Workflows nicht löschen möchten.

## Befehlsreferenz für CloudUtil

Dieser Abschnitt enthält eine Beschreibung der Befehle in der CloudUtil-Befehlszeilenschnittstelle.

CloudUtil ist die Befehlszeilenschnittstelle für vRealize Automation Designer. Sie führen die Befehle auf dem Windows-Computer aus, auf dem Sie vRealize Automation Designer ausführen. Der standardmäßige Installationspeicherort auf dem Windows-Computer ist C:\Programme (x86)\VMware\vCAC\Design Center.

---

**Hinweis** In den CloudUtil-Befehlen wird der Model Manager als `repository` bezeichnet, und ein Distributed Execution Manager (DEM) wird als `agent` bezeichnet.

---

## DEM-Befehle

Mithilfe der DEM-Befehle können Sie eine Liste der Distributed Execution Manager anzeigen, die beim Model Manager registriert sind, und Zuordnungen zwischen Qualifikationen und DEMs hinzufügen bzw. entfernen.

## DEM-Add-Skills

Ordnet einem registrierten Distributed Execution Manager Qualifikationen zu.

### Übersicht

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

### Argumente für DEM-Add-Skills

| Argument      | Beschreibung                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n   -name    | Der Name einer registrierten Distributed Execution Manager-Instanz.                                                                                                                                                  |
| -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -s   -skills  | Kommagetrennte Liste der Qualifikationen, die diesem Distributed Execution Manager zugeordnet werden sollen.                                                                                                         |
| -v   -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

**Hinweis** Die Qualifikationen müssen in Model Manager bereits vorhanden sein. Siehe [Skill-Install](#).

## DEM-List

Zeigt eine Liste aller Distributed Execution Manager, die beim Model Manager registriert sind, und die zugehörigen Qualifikationen an.

### Übersicht

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

### Argumente für DEM-List

| Argument      | Beschreibung                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## DEM-Remove-Skills

Entfernt die Zuordnung zwischen Qualifikationen und einem Distributed Execution Manager.

## Übersicht

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

### Argumente für DEM-Remove-Skills

| Argument      | Beschreibung                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n   -name    | Der Name einer registrierten Distributed Execution Manager-Instanz.                                                                                                                                                  |
| -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -s   -skills  | Kommagetrennte Liste der Qualifikationen, die aus diesem Distributed Execution Manager entfernt werden sollen.                                                                                                       |
| -v   -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## File-Befehle

Mithilfe der File-Befehle können Sie Dateien (in der Regel Skripts) im Model Manager speichern und verwalten.

### File-Export

Exportiert eine Datei aus dem Model Manager.

#### Übersicht

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

### Argumente für File-Export

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -i   -iteration | (Optional) Versionszeichenfolge der Datei im Model Manager. Der Standardwert ist 0,0.                                                                                                                                |
| -n   -name      | Der Anzeigename der Datei im Model Manager.                                                                                                                                                                          |
| -o   -output    | Der Pfad für die Dateiausgabe.                                                                                                                                                                                       |
| -repository     | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose   | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

### File-Import

Importiert eine Datei in den Model Manager.

## Übersicht

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

## Argumente für File-Import

| Argument          | Beschreibung                                                                                                                                                                                                         |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -d   -description | (Optional) Die Beschreibung der Datei.                                                                                                                                                                               |
| -f   -filename    | Der Pfad zu einer Datei, die in den Model Manager importiert werden soll.                                                                                                                                            |
| -i   -iteration   | (Optional) Versionszeichenfolge der Datei im Model Manager. Der Standardwert ist <b>0,0</b> .                                                                                                                        |
| -n   -name        | Der Anzeigename, der der Datei im Model Manager zugewiesen werden soll.                                                                                                                                              |
| -repository       | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose     | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## File-List

Listet alle Dateien auf, die in den Model Manager importiert werden.

## Übersicht

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

## Argumente für File-List

| Argument      | Beschreibung                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## File-Remove-All

Entfernt alle Revisionen für eine bestimmte Dateiversion aus dem Model Manager.

## Übersicht

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

## Argumente für File-Remove-All

Tabelle 1-20.

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -i   -iteration | (Optional) Versionszeichenfolge der Datei im Model Manager. Der Standardwert ist <b>0, 0</b> .                                                                                                                       |
| -n   -name      | Der Anzeigename der Datei im Model Manager.                                                                                                                                                                          |
| -repository     | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose   | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## File-Remove-Rev

Entfernt eine bestimmte Dateirevision aus dem Model Manager.

### Übersicht

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

## Argumente für File-Export

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -i   -iteration | (Optional) Versionszeichenfolge der Datei im Model Manager. Der Standardwert ist <b>0, 0</b> .                                                                                                                       |
| -n   -name      | Der Anzeigename der Datei im Model Manager.                                                                                                                                                                          |
| -r   -revision  | Die zu entfernende Dateirevision.                                                                                                                                                                                    |
| -repository     | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose   | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## File-Rollback

Stellt die angegebene Revision einer Datei im Model Manager wieder her.

### Übersicht

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

## Argumente für File-Rollback

Tabelle 1-21.

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -i   -iteration | (Optional) Versionszeichenfolge der Datei im Model Manager. Der Standardwert ist 0, 0.                                                                                                                               |
| -n   -name      | Der Anzeigename der Datei im Model Manager.                                                                                                                                                                          |
| -r   -revision  | Die wiederherzustellende Dateirevision.                                                                                                                                                                              |
| -repository     | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose   | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## File-Update

Aktualisiert eine zuvor importierte Datei im Model Manager mit einer neuen Revision.

### Übersicht

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

### Argumente für File-Update

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -f   -filename  | Pfad zur aktualisierten Datei.                                                                                                                                                                                       |
| -i   -iteration | (Optional) Versionszeichenfolge der Datei im Model Manager. Der Standardwert ist 0, 0.                                                                                                                               |
| -n   -name      | Der Anzeigename der Datei im Model Manager.                                                                                                                                                                          |
| -repository     | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose   | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Operation-Befehle

Mit den Operation-Befehlen können Sie benutzerdefinierte Vorgänge im Model Manager verwalten.

### Operation-Create

Erstellt einen oder mehrere benutzerdefinierte Vorgänge, die auf einer Maschine basierend auf einer Vorgangsdefinitionsdatei ausgeführt werden können.

## Übersicht

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

### Argumente für Operation-Create

| Argument              | Beschreibung                                                                                                                                                                                                         |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -c   -operationConfig | Der Pfad zu einer Vorgangsdefinitionsdatei (XML).                                                                                                                                                                    |
| -repository           | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose         | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Operation-Delete

Löscht einen benutzerdefinierten Vorgang aus dem Model Manager.

### Übersicht

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

### Argumente für Operation-Delete

| Argument      | Beschreibung                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -force        | (Optional) Erzwingt das Löschen des Vorgangs.                                                                                                                                                                        |
| -n   -name    | Der Name des benutzerdefinierten Vorgangs im Model Manager.                                                                                                                                                          |
| -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Operation-List

Listet alle benutzerdefinierten Vorgänge im Model Manager auf.

### Übersicht

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

## Argumente für Operation-List

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   - -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Skill-Befehle

Mit den Skill-Befehlen können Sie die Qualifikationen für Distributed Execution Manager und Workflows verwalten.

### Skill-Install

Installiert eine Qualifikation im Model Manager.

#### Übersicht

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

### Argumente für Skill-Install

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n   - -name    | Der Name der Qualifikation im Model Manager.                                                                                                                                                                         |
| - -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   - -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

### Skill-List

Listet alle im Model Manager installierten Qualifikationen auf.

#### Übersicht

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

## Argumente für Skill-List

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   - -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Skill-Uninstall

Deinstalliert eine Qualifikation im Model Manager.

### Übersicht

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

### Argumente für Skill-Uninstall

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n   - -name    | Der Name der Qualifikation, die im Model Manager deinstalliert werden soll.                                                                                                                                          |
| - -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   - -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

**Hinweis** Eine Qualifikation kann nicht deinstalliert werden, wenn sie einem Distributed Execution Manager oder einem Workflow zugeordnet ist. Siehe [DEM-Remove-Skills](#) oder [Workflow-Remove-Skills](#).

## Workflow-Befehle

Mit den Workflow-Befehlen können Sie anpassbare IaaS-Workflows im Model Manager sowie die den Workflows zugeordneten Qualifikationen verwalten.

### Workflow-Add-Skills

Ordnet Qualifikationen einem Workflow im Model Manager zu.

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

**Tabelle 1-22. Argumente für Workflow-Add-Skills**

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name            | Der Name eines Workflows im Model Manager.                                                                                                                                                                           |
| Skills          | Kommagetrennte Liste der Qualifikationen, die diesem Workflow zugeordnet werden sollen.                                                                                                                              |
| - -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   - -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

**Hinweis** Die Qualifikationen müssen in Model Manager bereits vorhanden sein. Siehe [Skill-Install](#).

## Workflow-List

Zeigt eine Liste aller Workflows, die im Model Manager installiert sind, und die zugehörigen Qualifikationen an.

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

**Tabelle 1-23. Argumente für Workflow-List**

| Argument        | Beschreibung                                                                                                                                                                                                         |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| - -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   - -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Workflow-Remove-Skills

Entfernt die Zuordnung zwischen Qualifikationen und einem Workflow im Model Manager.

### Übersicht

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

### Argumente für Workflow-Remove-Skills

| Argument      | Beschreibung                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n   - -name  | Der Name eines Workflows im Model Manager.                                                                                                                                                                           |
| - -repository | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |

| Argument      | Beschreibung                                                                                                 |
|---------------|--------------------------------------------------------------------------------------------------------------|
| -s   -skills  | Kommagetrennte Liste der Qualifikationen, die aus diesem Workflow entfernt werden sollen.                    |
| -v   -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben. |

## Workflow-Rollback

Stellt eine bestimmte Revision eines Workflows wieder her.

### Übersicht

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

### Argumente für Workflow-Rollback

| Argument       | Beschreibung                                                                                                                                                                                                         |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -n   -name     | Der Name des Workflows im Model Manager.                                                                                                                                                                             |
| -repository    | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -r   -revision | Die wiederherzustellende Workflowrevision.                                                                                                                                                                           |
| -v   -verbose  | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Workflow-Update

Aktualisiert einen anpassbaren Workflow mit einer neuen Revision.

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

**Tabelle 1-24. Argumente für Workflow-Update**

| Argument      | Beschreibung                                                                                                                                                                                                         |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| File Name     | Der Pfad zu einer Datei (XAML), die den aktualisierten Workflow enthält.                                                                                                                                             |
| Name          | Der Name des zu aktualisierenden Workflows.                                                                                                                                                                          |
| Description   | (Optional) Die Beschreibung des Workflows.                                                                                                                                                                           |
| -repository   | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -v   -verbose | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Importbefehle

Mithilfe der Importbefehle können Sie eine oder mehrere virtuelle Maschinen in eine vRealize Automation-Bereitstellung importieren.

### Machine-BulkRegisterExport

Erstellt eine CSV-Datendatei, die verwendet wird, um virtuelle Maschinen in eine vRealize Automation-Bereitstellung zu importieren.

#### Übersicht

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

#### Argumente für „Machine-BulkRegisterExport“

Tabelle 1-25.

| Argument             | Beschreibung                                                                                                                                                                                                                                   |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -b   - -blueprint    | (Optional) Schließen Sie den Blueprint-Namen ein.                                                                                                                                                                                              |
| -e   - -exportNames  | (Optional) Exportiert Namen anstelle von GUIDs.                                                                                                                                                                                                |
| -f   - -filename     | Geben Sie den Namen der CSV-Datendatei mit einer Liste von Maschinennamen an, z. B. filename.csv. Die Datei wird standardmäßig im aktuellen Pfad gespeichert. Sie können auch den vollständigen Pfad zu einem gewünschten Verzeichnis angeben. |
| -g   - -group        | (Optional) Geben Sie den Business-Gruppennamen an, z. B. „Engineering“.                                                                                                                                                                        |
| -i   - -ignore       | (Optional) Ungültige Argumente werden ignoriert.                                                                                                                                                                                               |
| -m   - -managed      | (Optional) Verwaltete virtuelle Maschinen werden exportiert. Standardmäßig werden nicht verwaltete virtuelle Maschinen exportiert.                                                                                                             |
| -n   - -resourceName | (Optional) Geben Sie den Namen der Computing-Ressource oder des Endpoints an, um nach Ressourcenname zu filtern.                                                                                                                               |
| -o   - -owner        | (Optional) Geben Sie den Besitzer der importierten virtuellen Maschine an, z. B. „jschmidt“.                                                                                                                                                   |
| -p   - -properties   | (Optional) Eigenschaften für verwaltete virtuelle Maschinen werden exportiert.                                                                                                                                                                 |
| -r   - -resourceType | (Optional) Geben Sie „1“ für die Computing-Ressource oder „2“ für den Endpoint an, um nach Ressourcentyp zu filtern.                                                                                                                           |

Tabelle 1-25. (Fortsetzung)

| Argument          | Beschreibung                                                                                                                                                                                                         |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -repository       | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben. |
| -sn   -sourcename | (Optional) Geben Sie den Namen des Clusters oder des Endpoints an.                                                                                                                                                   |
| -st   -sourcetype | (Optional) Geben Sie den Quelltyp als Cluster oder als Endpoint an.                                                                                                                                                  |
| -t   -machinetype | (Optional) Geben Sie den zu exportierenden Maschinentyp an, z. B. virtuell, physisch, Cloud, AppService oder vApp.                                                                                                   |
| -u   -user        | Geben Sie den Fabric-Administrator an, der die Massenregistrierung durchführt.                                                                                                                                       |
| -v   -verbose     | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                         |

## Machine-BulkRegisterImport

Importiert eine oder mehrere virtuelle Maschinen in eine vRealize Automation-Zielbereitstellung.

### Übersicht

```
CloudUtil.exe Machine-BulkRegisterImport [-b|--batch] [-d|--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]
```

### Argumente für „Machine-BulkRegisterImport“

Tabelle 1-26.

| Argument             | Beschreibung                                                                                                         |
|----------------------|----------------------------------------------------------------------------------------------------------------------|
| -b   -batch          | (Optional) Batchgröße.                                                                                               |
| -d   -delay          | (Optional) Geben Sie die Verzögerung bei der Verarbeitung im Format hh:mm:ss ein, z. B. 02:20:10.                    |
| -f   -filename       | Geben Sie den Namen der CSV-Datendatei an, die die Liste der Maschinennamen enthält. Beispiel: filename.csv.         |
| -h   --humanreadable | (Optional) Die Eingabedatei enthält die Namen der virtuellen Maschinen, nicht die GUIDs.                             |
| -i   -ignore         | (Optional) Registrierte oder verwaltete virtuelle Maschinen werden ignoriert.                                        |
| -n   -name           | Geben Sie den Namen der Arbeitswarteschlange an, um den Import an die vRealize Automation-Zielinstanz durchzuführen. |

Tabelle 1-26. (Fortsetzung)

| Argument       | Beschreibung                                                                                                                                                                                                                                                       |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -repository    | (Optional) Der Stamm-URI von Model Manager, z. B. „http://hostname/repository“. Der Standardwert ist in der Konfigurationsdatei für CloudUtil im Schlüssel „repositoryAddress“ im Abschnitt <appSettings> angegeben.                                               |
| -s   -skipUser | (Optional) Legt den Besitzer der Maschine auf den Wert fest, der in der Spalte „Besitzer“ der CSV-Datendatei aufgeführt ist, ohne dass überprüft wird, ob der Benutzer vorhanden ist. Bei Auswahl dieser Option verringert sich die für den Import benötigte Zeit. |
| -t   -time     | Geben Sie die Workflowstartzeit im Format MM/TT/JJJJ hh:mm GMT an, z. B. 04/18/2014 10:01 GMT. Für die angegebene Startzeit wird von der Ortszeit des Servers ausgegangen und nicht von der Ortszeit der Workstation des Benutzers.                                |
| -u   -user     | Geben Sie den Fabric-Administrator an, der die Massenregistrierung durchführt.                                                                                                                                                                                     |
| -v   -verbose  | (Optional) Wenn ein Fehler auftritt, wird eine Stapelüberwachung anstatt nur der Ausnahmemeldung ausgegeben.                                                                                                                                                       |
| whatif         | (Optional) Legen Sie fest, dass die CSV-Datei validiert wird, aber keine virtuellen Maschinen importiert werden.                                                                                                                                                   |

## vRealize Automation -Workflowaktivitätsreferenz

VMware stellt mit vRealize Automation Designer eine Bibliothek mit Workflowaktivitäten zum Anpassen von Workflows bereit.

**Hinweis** Das CDK läuft in vRealize Automation 7.0 aus. Die vRealize Orchestrator-Workflows können für Anwendungsfälle genutzt werden, für die zuvor das CDK verwendet wurde.

Darüber hinaus gibt es in vRealize Automation Designer die folgenden fünf Kategorien von Windows Workflow Foundation-Aktivitäten: Ablaufsteuerung, Flussdiagramm, Primitive, Auflistung und Fehlerbehandlung.

Dieser Abschnitt enthält eine Referenz mit den IaaS-Workflowaktivitäten von vRealize Automation Designer in den Namespaces `DynamicOps.Repository.Activities` und `DynamicOps.Cdk.Activities`. Aktivitäten im Zusammenhang mit Aufrufen von vRealize Orchestrator-Workflows werden unter [Verwenden von vRealize Orchestrator-Workflowaktivitäten](#) beschrieben.

**Hinweis** In der Bibliothek mit IaaS-Aktivitäten wird der Model Manager als `repository` bezeichnet.

## DynamicOps.Repository.Activities

Der `DynamicOps.Repository.Activities`-Namespace enthält grundlegende Workflowaktivitäten für IaaS-Workflows.

**Hinweis** Das CDK läuft in vRealize Automation 7.0 aus. Die vRealize Orchestrator-Workflows können für Anwendungsfälle genutzt werden, für die zuvor das CDK verwendet wurde.

### AddLink

Fügt den angegebenen Link zu den Objekten hinzu, die von `DataServiceContext` nachverfolgt werden.

**Tabelle 1-27. Eingabeparameter für die AddLink-Aktivität**

| Argument                        | Typ                                   | Beschreibung                                                                                          |
|---------------------------------|---------------------------------------|-------------------------------------------------------------------------------------------------------|
| <code>DataServiceContext</code> | <code>RepositoryServiceContext</code> | Der <code>DataServiceContext</code> , dem der Link hinzugefügt werden soll.                           |
| <code>Source</code>             | <code>Object</code>                   | Das Quellobjekt für den neuen Link.                                                                   |
| <code>SourceProperty</code>     | <code>String</code>                   | Der Name der Navigationseigenschaft im Quellobjekt, von der das zugehörige Objekt zurückgegeben wird. |
| <code>Target</code>             | <code>Object</code>                   | Das Objekt, das über den neuen Link mit dem Quellobjekt verknüpft ist.                                |

### AddObject

Fügt das angegebene Objekt zu den Objekten hinzu, die von `DataServiceContext` nachverfolgt werden.

**Tabelle 1-28. Eingabeparameter für die AddObject-Aktivität**

| Argument                        | Typ                                   | Beschreibung                                                                    |
|---------------------------------|---------------------------------------|---------------------------------------------------------------------------------|
| <code>DataServiceContext</code> | <code>RepositoryServiceContext</code> | Der <code>DataServiceContext</code> , dem das Objekt hinzugefügt werden soll.   |
| <code>Instance</code>           | <code>Object</code>                   | Das Objekt, das durch <code>DataServiceContext</code> nachverfolgt werden soll. |

### AttachTo

Weist `DataServiceContext` an, mit der Nachverfolgung der angegebenen Ressource zu beginnen.

**Tabelle 1-29. Eingabeparameter für die AttachTo-Aktivität**

| Argument                        | Typ                                   | Beschreibung                                                                                                                               |
|---------------------------------|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| <code>DataServiceContext</code> | <code>RepositoryServiceContext</code> | Der <code>DataServiceContext</code> , von dem die Ressource nachverfolgt werden soll.                                                      |
| <code>Instance</code>           | <code>Object</code>                   | Die Ressource, die durch <code>DataServiceContext</code> nachverfolgt werden soll. Die Ressource wird im Status „Unverändert“ hinzugefügt. |

## CreateRepositoryServiceContext<T>

Erstellt einen Kontext für ein Modell, das in Model Manager geladen wird.

Wenn Sie diese Aktivität einem Workflow in vRealize Automation Designer hinzufügen, müssen Sie eine Klasse auswählen, die Werte aus der Klasse RepositoryServiceContext übernimmt.

**Tabelle 1-30. Eingabeparameter für die Aktivität „CreateRepositoryServiceContext<T>“**

| Argument | Typ    | Beschreibung                                                                                |
|----------|--------|---------------------------------------------------------------------------------------------|
| Uri      | URI    | (Optional) Root-URI, der bei der Verbindungsherstellung mit dem Modell verwendet wird.      |
| Username | String | (Optional) Benutzername, der bei der Verbindungsherstellung mit dem Kontext verwendet wird. |

**Tabelle 1-31. Ausgabeparameter für die Aktivität „CreateRepositoryServiceContext<T>“**

| Argument | Typ                      | Beschreibung                                                                                                                            |
|----------|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| Result   | RepositoryServiceContext | Der spezifische zurückgegebene Typ ist eine Instanz der Klasse, die ausgewählt wurde, als die Aktivität zum Workflow hinzugefügt wurde. |

## DeleteLink

Ändert den Status des Links in der Liste mit den Links, die von DataServiceContext nachverfolgt werden, in „Gelöscht“.

**Tabelle 1-32. Eingabeparameter für die DeleteLink-Aktivität**

| Argument           | Typ                      | Beschreibung                                                                                                                                                                  |
|--------------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DataServiceContext | RepositoryServiceContext | Der DataServiceContext, aus dem der Link gelöscht werden soll.                                                                                                                |
| Source             | Object                   | Das Quellobjekt im Link, der zum Löschen markiert werden soll.                                                                                                                |
| SourceProperty     | String                   | Der Name der Navigationseigenschaft im Quellobjekt, die für den Zugriff auf das Zielobjekt verwendet wird.                                                                    |
| Target             | Object                   | Das für den Link verwendete Zielobjekt, das an das Quellobjekt gebunden ist. Das Zielobjekt muss den von der Quelleigenschaft festgelegten Typ oder einen Untertyp aufweisen. |

## DeleteObject

Ändert den Status des zu löschenden Objekts in DataServiceContext.

**Tabelle 1-33. Eingabeparameter für die DeleteObject-Aktivität**

| Argument           | Typ                      | Beschreibung                                                                |
|--------------------|--------------------------|-----------------------------------------------------------------------------|
| DataServiceContext | RepositoryServiceContext | Der DataServiceContext, aus dem die Ressource gelöscht werden soll.         |
| Instance           | Object                   | Die nachverfolgte Entität, deren Status in „Gelöscht“ geändert werden soll. |

## InvokeRepositoryWorkflow

Führt einen im Model Manager installierten Workflow aus.

**Tabelle 1-34. Eingabeparameter für die InvokeRepositoryWorkflow-Aktivität**

| Argument        | Typ                        | Beschreibung                                                                                  |
|-----------------|----------------------------|-----------------------------------------------------------------------------------------------|
| WorkflowType    | WorkflowDefinition-Entität | Der auszuführende Workflow.                                                                   |
| WorkflowInputs  | Dictionary<String, Object> | (Optional) Eingaben für den Workflow.                                                         |
| CallingInstance | WorkflowInstance-Entität   | (Optional) Der Workflow, der den ausgeführten Workflow aufruft und an den zurückgegeben wird. |

## LoadProperty

Lädt zurückgestellten Inhalt für eine angegebene Eigenschaft über den Datendienst.

**Tabelle 1-35. Eingabeparameter für die LoadProperty-Aktivität**

| Argument           | Typ                      | Beschreibung                                                           |
|--------------------|--------------------------|------------------------------------------------------------------------|
| DataServiceContext | RepositoryServiceContext | Der DataServiceContext, über den die Eigenschaft geladen werden soll.  |
| Instance           | Object                   | Die Entität, die die zu ladende Eigenschaft enthält.                   |
| InstanceProperty   | String                   | Der Eigenschaftsname der angegebenen Entität, die geladen werden soll. |

## SaveChanges

Speichert die vom DataServiceContext nachverfolgten Änderungen im Speicher.

**Tabelle 1-36. Eingabeparameter für die SaveChanges-Aktivität**

| Argument           | Typ                      | Beschreibung                                                             |
|--------------------|--------------------------|--------------------------------------------------------------------------|
| DataServiceContext | RepositoryServiceContext | Der DataServiceContext, der die zu speichernden Änderungen nachverfolgt. |

## SetLink

Benachrichtigt den DataServiceContext, dass ein neuer Link zwischen den angegebenen Objekten vorhanden ist und dass der Link durch die im Argument SourceProperty angegebene Eigenschaft dargestellt wird.

**Tabelle 1-37. Eingabeparameter für die SetLink-Aktivität**

| Argument           | Typ                      | Beschreibung                                                                                                                                                                                                                                                                                          |
|--------------------|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DataServiceContext | RepositoryServiceContext | Der DataServiceContext, der über den Link benachrichtigt werden soll.                                                                                                                                                                                                                                 |
| Source             | Object                   | Das Quellobjekt für den neuen Link.                                                                                                                                                                                                                                                                   |
| SourceProperty     | String                   | Die Eigenschaft im Quellobjekt, die das Zielobjekt des neuen Links identifiziert.                                                                                                                                                                                                                     |
| Target             | Object                   | Das untergeordnete Objekt des neuen Links, das durch Aufrufen dieser Methode initialisiert werden soll. Das Zielobjekt muss ein Untertyp des durch „SourceProperty“ identifizierten Untertyps sein. Wenn „Target“ auf null festgelegt ist, stellt der Aufruf einen Vorgang zum Löschen des Links dar. |

## UpdateObject

Ändert den Status des angegebenen Objekts im DataServiceContext in „Geändert“.

**Tabelle 1-38. Eingabeparameter für die UpdateObject-Aktivität**

| Argument           | Typ                      | Beschreibung                                                                 |
|--------------------|--------------------------|------------------------------------------------------------------------------|
| DataServiceContext | RepositoryServiceContext | Der DataServiceContext, der die zu aktualisierende Entität nachverfolgt.     |
| Instance           | Object                   | Die nachverfolgte Entität, der der Status „Geändert“ zugewiesen werden soll. |

## DynamicOps.Cdk.Activities

Der DynamicOps.Cdk.Activities-Namespace enthält erweiterte Aktivitäten für IaaS-Workflows.

**Hinweis** Das CDK läuft in vRealize Automation 7.0 aus. Die vRealize Orchestrator-Workflows können für Anwendungsfälle genutzt werden, für die zuvor das CDK verwendet wurde.

## ExecutePowerShellScript

Führt ein PowerShell-Skript aus, das in Model Manager unter dem angegebenen Namen gespeichert ist.

Bevor Sie die ExecutePowerShellScript-Aktivität verwenden, müssen Sie zunächst das auszuführende Skript mit dem Befehl `CloudUtil File-Import` in Model Manager laden.

**Tabelle 1-39. Eingabeparameter für die ExecutePowerShellScript-Aktivität**

| Argument      | Typ    | Beschreibung                                                                                                                           |
|---------------|--------|----------------------------------------------------------------------------------------------------------------------------------------|
| ScriptName    | String | Name des auszuführenden Skripts in Model Manager.                                                                                      |
| ScriptVersion | Object | (Optional) Version des auszuführenden Skripts in Model Manager. Der Standardwert ist „0,0“.                                            |
| Machineld     | Guid   | (Optional) Wenn dieser Parameter angegeben ist, wird die Maschine geladen, und alle ihre Eigenschaften werden an das Skript übergeben. |

**Tabelle 1-39. Eingabeparameter für die ExecutePowerShellScript-Aktivität (Fortsetzung)**

| Argument  | Typ                       | Beschreibung                                                                                                                                                                                                                                                                                 |
|-----------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Arguments | Dictionary<string,string> | Zusätzliche Argumente zum Übergeben an das Skript. Wenn „Machineld“ angegeben ist und eine Maschineneigenschaft mit dem gleichen Namen wie ein Argument vorhanden ist (ohne Beachtung der Groß- und Kleinschreibung), überschreibt der Wert der Maschineneigenschaft den Wert des Arguments. |
| PSModules | IEnumerable<string>       | (Optional) Module, die während der Befehlsausführung in die PowerShell-Laufzeit geladen werden.<br><br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                                                                            |

**Tabelle 1-40. Ausgabeparameter für die ExecutePowerShellScript-Aktivität**

| Argument | Typ                  | Beschreibung                                                                        |
|----------|----------------------|-------------------------------------------------------------------------------------|
| Output   | Collection<PSObject> | Die Ausgabe des Skripts, sofern vorhanden. Bei Fehlern wird eine Ausnahme gemeldet. |

Wenn Sie die Fehlermeldung Typ `PSObject` ist nicht definiert in der vRealize Automation Designer-Konsole erhalten, während Sie mit der Ausgabe von `ExecutePowerShellScript` arbeiten, führen Sie folgende Schritte aus:

- 1 Klicken Sie in der unteren linken Ecke des Bereichs „Designer“ auf **Importe**.
- 2 Wählen Sie die **System.Management.Automation**-Assembly aus.

## ExecuteSshScript

Führt ein SSH-Skript aus, das in Model Manager unter dem angegebenen Namen gespeichert ist.

Bevor Sie die `ExecuteSshScript`-Aktivität verwenden, müssen Sie zunächst das auszuführende Skript mit dem Befehl `CloudUtil File-Import` in Model Manager laden.

**Tabelle 1-41. Eingabeparameter für die ExecuteSshScript-Aktivität**

| Argument      | Typ      | Beschreibung                                                                                                                |
|---------------|----------|-----------------------------------------------------------------------------------------------------------------------------|
| ScriptName    | String   | Name des auszuführenden Skripts in Model Manager.                                                                           |
| Host          | String   | Name des Servers, für den das Skript ausgeführt wird.                                                                       |
| Username      | String   | Benutzername, der bei der Verbindungsherstellung mit dem Host verwendet wird.                                               |
| Password      | String   | Kennwort, das bei der Verbindungsherstellung mit dem Host verwendet wird.                                                   |
| ScriptVersion | Object   | (Optional) Version des auszuführenden Skripts in Model Manager. Der Standardwert ist „0,0“.                                 |
| Timeout       | TimeSpan | (Optional) Zeitspanne, nach der eine Zeitüberschreitung bei der Skriptausführung eintritt. Der Standardwert ist 30 Minuten. |

**Tabelle 1-42. Ausgabeparameter für die ExecuteSshScript-Aktivität**

| Argument             | Typ                        | Beschreibung                                    |
|----------------------|----------------------------|-------------------------------------------------|
| EnvironmentVariables | Dictionary<string, string> | Ergebnis der Skriptausführung, falls vorhanden. |

## GetMachineName

Ruft den Namen einer Maschine ab.

**Tabelle 1-43. Eingabeparameter für die GetMachineName-Aktivität**

| Argument  | Typ  | Beschreibung                                    |
|-----------|------|-------------------------------------------------|
| MachineId | Guid | Die Maschine, deren Name abgerufen werden soll. |

**Tabelle 1-44. Ausgabeparameter für die GetMachineName-Aktivität**

| Argument    | Typ    | Beschreibung                                           |
|-------------|--------|--------------------------------------------------------|
| MachineName | String | Der Name der durch MachineId identifizierten Maschine. |

## GetMachineOwner

Ruft den Benutzernamen des Besitzers einer Maschine ab.

**Tabelle 1-45. Eingabeparameter für die GetMachineOwner-Aktivität**

| Argument  | Typ  | Beschreibung                                        |
|-----------|------|-----------------------------------------------------|
| MachineId | Guid | Die Maschine, deren Besitzer abgerufen werden soll. |

**Tabelle 1-46. Ausgabeparameter für die GetMachineOwner-Aktivität**

| Argument | Typ    | Beschreibung                                                                                            |
|----------|--------|---------------------------------------------------------------------------------------------------------|
| Owner    | String | Der durch MachineId identifizierte Besitzer der Maschine, oder null, falls kein Besitzer vorhanden ist. |

## GetMachineProperties

Ruft die Liste der benutzerdefinierten Eigenschaften ab, die einer Maschine zugeordnet sind.

**Tabelle 1-47. Eingabeparameter für die GetMachineProperties-Aktivität**

| Argument  | Typ  | Beschreibung                                               |
|-----------|------|------------------------------------------------------------|
| MachineId | Guid | Die Maschine, deren Eigenschaften abgerufen werden sollen. |

**Tabelle 1-48. Ausgabeparameter für die GetMachineProperties-Aktivität**

| Argument   | Typ                        | Beschreibung                                                                                                               |
|------------|----------------------------|----------------------------------------------------------------------------------------------------------------------------|
| Properties | Dictionary<string, string> | Die Liste der Maschineneigenschaften. Werte werden entschlüsselt zurückgegeben, wenn sie verschlüsselt gespeichert wurden. |

## GetMachineProperty

Ruft den Wert der angegebenen Eigenschaft für eine Maschine ab.

**Tabelle 1-49. Eingabeparameter für die GetMachineProperty-Aktivität**

| Argument     | Typ    | Beschreibung                                                                                                                              |
|--------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------|
| Machineld    | Guid   | Die Maschine, von der eine Eigenschaft abgerufen wird.                                                                                    |
| PropertyName | String | Name der Eigenschaft, deren Wert zurückgegeben wird.                                                                                      |
| IsRequired   | bool   | Wenn die Eigenschaft erforderlich ist und nicht gefunden wird, löst die Aktivität eine Ausnahme aus, andernfalls wird null zurückgegeben. |

**Tabelle 1-50. Ausgabeparameter für die GetMachineProperty-Aktivität**

| Argument      | Typ    | Beschreibung                                                                                                                           |
|---------------|--------|----------------------------------------------------------------------------------------------------------------------------------------|
| PropertyValue | String | Von PropertyName angegebener Wert der Eigenschaft. Der Wert wird entschlüsselt zurückgegeben, wenn er verschlüsselt gespeichert wurde. |

## GetScriptFromName

Ruft die Inhalte des Skripts ab, das im Model Manager unter dem angegebenen Namen gespeichert ist.

**Tabelle 1-51. Eingabeparameter für die GetScriptFromName-Aktivität**

| Argument      | Typ    | Beschreibung                                                                                  |
|---------------|--------|-----------------------------------------------------------------------------------------------|
| ScriptName    | String | Der Name des abzurufenden Skripts im Model Manager.                                           |
| ScriptVersion | Object | (Optional) Die Version des abzurufenden Skripts im Model Manager. Der Standardwert ist „0,0“. |

**Tabelle 1-52. Ausgabeparameter für die GetScriptFromName-Aktivität**

| Argument      | Typ    | Beschreibung                                              |
|---------------|--------|-----------------------------------------------------------|
| ScriptContent | String | Die Inhalte des durch ScriptName identifizierten Skripts. |

## InvokePowerShell

Führt einen PowerShell-Befehl aus.

**Tabelle 1-53. Eingabeparameter für die InvokePowerShell-Aktivität**

| Argument    | Typ                 | Beschreibung                         |
|-------------|---------------------|--------------------------------------|
| CommandText | String              | Auszuführender Befehl.               |
| Arguments   | IEnumerable<string> | (Optional) Argumente für den Befehl. |
| Input       | IEnumerable         | (Optional) Die Eingabe-Pipeline.     |

**Tabelle 1-53. Eingabeparameter für die InvokePowerShell-Aktivität (Fortsetzung)**

| Argument            | Typ                 | Beschreibung                                                                                                                                                                                                                                                                                                                      |
|---------------------|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IsScript            | bool                | (Optional) Gibt an, ob CommandText ein Skript ist. Der Standardwert lautet „Falsch“.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                                                                                                                                |
| Parameters          | Collection          | (Optional) Auflistung der Name/Wert-Paare, die als Parameter an das PowerShell-Skript übergeben werden.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                                                                                                             |
| PowerShellVariables | Collection          | (Optional) Variablen, die in die PowerShell-Laufzeit kopiert werden.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                                                                                                                                                |
| PSModules           | IEnumerable<string> | (Optional) Module, die während der Befehlsausführung in die PowerShell-Laufzeit geladen werden.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                                                                                                                     |
| Runspace            | Runspace            | (Optional) Wenn ein PowerShell-Runspace erstellt und für dieses Argument bereitgestellt wird, können Sie den gleichen Runspace in mehreren PowerShell-Aufrufen wiederverwenden, was zu einer Verbesserung der Leistung führen kann.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar. |

**Tabelle 1-54. Ausgabeparameter für die InvokePowerShell-Aktivität**

| Argument | Typ                     | Beschreibung                                                                        |
|----------|-------------------------|-------------------------------------------------------------------------------------|
| Output   | Collection<PSObject>    | Die Ausgabe des Befehls, sofern vorhanden. Bei Fehlern wird eine Ausnahme gemeldet. |
| Errors   | Collection<ErrorRecord> | Fehler als Folge der Ausführung, sofern vorhanden.                                  |

Wenn Sie die Fehlermeldung Typ `PSObject` ist nicht definiert in der vRealize Automation Designer-Konsole erhalten, während Sie mit der Ausgabe von `ExecutePowerShellScript` arbeiten, führen Sie folgende Schritte aus:

- 1 Klicken Sie in der unteren linken Ecke des Bereichs „Designer“ auf **Importe**.
- 2 Wählen Sie die **System.Management.Automation**-Assembly aus.

## InvokeSshCommand

Führt einen SSH-Befehl aus.

**Tabelle 1-55. Eingabeparameter für die InvokeSshCommand-Aktivität**

| Argument    | Typ    | Beschreibung                                        |
|-------------|--------|-----------------------------------------------------|
| CommandText | String | Auszuführender Befehl.                              |
| Host        | String | Der Servername, für den der Befehl ausgeführt wird. |

**Tabelle 1-55. Eingabeparameter für die InvokeSshCommand-Aktivität (Fortsetzung)**

| Argument | Typ      | Beschreibung                                                                                                                                  |
|----------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| Username | String   | Benutzername, der bei der Verbindungsherstellung mit dem Host verwendet wird.                                                                 |
| Password | String   | Kennwort, das bei der Verbindungsherstellung mit dem Host verwendet wird.                                                                     |
| Timeout  | TimeSpan | (Optional) Der Zeitraum, nach dessen Ablauf bei der Ausführung des Befehls eine Zeitüberschreitung auftritt. Der Standardwert ist 30 Minuten. |

**Tabelle 1-56. Ausgabeparameter für die InvokeSshCommand-Aktivität**

| Argument             | Typ                        | Beschreibung                                                                        |
|----------------------|----------------------------|-------------------------------------------------------------------------------------|
| EnvironmentVariables | Dictionary<string, string> | Die Ausgabe des Befehls, sofern vorhanden. Bei Fehlern wird eine Ausnahme gemeldet. |

## LogMachineEvent

Protokolliert ein Maschineneignis im Benutzerprotokoll, das für den Maschinenbesitzer sichtbar ist.

**Tabelle 1-57. Eingabeparameter für die LogMachineEvent-Aktivität**

| Argument  | Typ    | Beschreibung                                                                         |
|-----------|--------|--------------------------------------------------------------------------------------|
| Machineld | Guid   | Die Maschine, die das zu protokollierende Ereignis generiert.                        |
| Message   | String | Die Nachricht, die in das Benutzerprotokoll geschrieben werden soll.                 |
| Type      | String | Wählen Sie in der Dropdown-Liste einen Nachrichtentyp aus („Info“, „Warn“, „Error“). |

## LogMessage

Protokolliert in das Distributed Execution Manager-Protokoll.

**Tabelle 1-58. Eingabeparameter für die LogMessage-Aktivität**

| Argument        | Typ    | Beschreibung                                                                                                                                                      |
|-----------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Message         | String | Die Nachricht, die in das DEM-Protokoll geschrieben werden soll.                                                                                                  |
| MessageCategory | String | Wählen Sie im Dropdown-Menü eine Kategorie aus ( <b>Debug</b> , <b>Error</b> , <b>Info</b> , <b>Trace</b> ) oder geben Sie eine benutzerdefinierte Kategorie ein. |
| MessageSeverity | String | Wählen Sie im Dropdown-Menü einen Schweregrad aus. Dieser Wert ist an die Liste der Schweregrade in <code>System.Diagnostics.TraceEventType</code> gebunden.      |

## RunProcess

Führt einen Prozess auf derselben Maschine wie der DEM aus, der diese Aktivität ausführt.

**Hinweis** vRealize Automation kann dem Benutzer für Prozesse, die von der RunProcess-Aktivität gestartet wurden, keine Benutzeroberfläche anzeigen. Daher können diese Prozesse nicht interaktiv sein. Um zu vermeiden, dass verwaiste Prozesse auf der DEM-Maschine zurückbleiben, müssen die Prozesse zudem automatisch beendet werden.

**Tabelle 1-59. Eingabeparameter für die RunProcess-Aktivität**

| Argument         | Typ    | Beschreibung                                                                                                                                                                                                                                                            |
|------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Command          | String | Pfad zur ausführbaren Datei, die auf der DEM-Maschine ausgeführt werden soll.                                                                                                                                                                                           |
| WorkingDirectory | String | (Optional) Das Arbeitsverzeichnis, unter dem der Prozess ausgeführt werden soll.                                                                                                                                                                                        |
| Arguments        | String | (Optional) Die Liste der Befehlszeilenargumente, die an den Befehl übergeben werden sollen.                                                                                                                                                                             |
| WaitForExit      | bool   | (Optional) Wenn dieser Parameter wahr ist, wartet der Workflow, bis der Prozess abgeschlossen ist, bevor der Workflow fortgesetzt wird. Der Standardwert lautet „Falsch“.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar. |

## SendEmail

Sendet eine E-Mail an die angegebenen Adressen.

**Tabelle 1-60. Eingabeparameter für die SendEmail-Aktivität**

| Argument | Typ                 | Beschreibung                                                                                                                                                                              |
|----------|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| To       | IEnumerable<string> | Die Liste der Adressen, an die die E-Mail gesendet werden soll.                                                                                                                           |
| From     | String              | Die Adresse, die in das Feld „Von“ der E-Mail eingegeben werden muss.                                                                                                                     |
| Subject  | String              | Die Betreffzeile für die E-Mail.                                                                                                                                                          |
| Body     | String              | Der Textkörper der E-Mail.                                                                                                                                                                |
| Host     | String              | Der Hostname oder die IP-Adresse des SMTP-Ausgangsservers.                                                                                                                                |
| Port     | Integer             | Der SMTP-Port des im Host angegebenen Servers.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                              |
| CC       | IEnumerable<string> | (Optional) Die Adresse oder Adressliste, an die eine Kopie der E-Mail gesendet werden soll.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar. |

**Tabelle 1-60. Eingabeparameter für die SendEmail-Aktivität (Fortsetzung)**

| Argument  | Typ                 | Beschreibung                                                                                                                                                                                             |
|-----------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bcc       | IEnumerable<string> | (Optional) Die verborgene Adresse oder Adressliste, an die eine Kopie der E-Mail gesendet werden soll.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.     |
| EnableSsl | bool                | (Optional) Gibt an, ob SSL verwendet werden soll.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                                          |
| UserName  | String              | Der Benutzername, der zur Authentifizierung bei dem in Host angegebenen SMTP-Server verwendet werden soll.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar. |
| Password  | String              | Das in UserName angegebene Kennwort des Benutzers.<br>Diese Option ist nur im Bereich „Eigenschaften“ und nicht im Bereich „Designer“ verfügbar.                                                         |

## SetMachineProperty

Erstellt oder aktualisiert eine benutzerdefinierte Eigenschaft auf einer Maschine.

**Tabelle 1-61. Eingabeparameter für die SetMachineProperty-Aktivität**

| Argument      | Typ    | Beschreibung                                                                                                                                                                          |
|---------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Machineld     | Guid   | Maschine, auf der die benutzerdefinierte Eigenschaft erstellt oder aktualisiert wird.                                                                                                 |
| PropertyName  | String | Name der zu erstellenden oder aktualisierenden Eigenschaft.                                                                                                                           |
| PropertyValue | String | Wert, mit dem die Eigenschaft erstellt oder aktualisiert wird.                                                                                                                        |
| IsEncrypted   | bool   | (Optional) Gibt an, ob der Wert der Eigenschaft verschlüsselt ist.                                                                                                                    |
| IsHidden      | bool   | (Optional) Gibt an, ob die Eigenschaft eine ausgeblendete Eigenschaft ist.                                                                                                            |
| IsRuntime     | bool   | (Optional) Gibt an, ob der anfordernde Benutzer den Eigenschaftswert zur Anforderungszeit angibt (entspricht der Markierung „Eingabeaufforderung“ in der Konsole).vRealize Automation |

## SetWorkflowResult

Legt den Status eines externen Workflows auf „Vollständig“ oder „Fehlgeschlagen“ fest, um von den Einstellungen in ExternalWF.xml berücksichtigt zu werden.

**Tabelle 1-62. Eingabeparameter für die SetWorkflowResult-Aktivität**

| <b>Argument</b> | <b>Typ</b>    | <b>Beschreibung</b>                                                                       |
|-----------------|---------------|-------------------------------------------------------------------------------------------|
| WorkflowId      | Guid          | Der Workflow, für den der Status festgelegt werden soll.                                  |
| Next State      | WorkflowState | Wählen Sie im Dropdown-Menü die Option <b>Vollständig</b> oder <b>Fehlgeschlagen</b> aus. |