

## 6 Blueprints

Nachdem wir Ihnen in den vorigen Kapiteln gezeigt haben, wie Sie vRealize Automation installieren und konfigurieren, zeigen wir nun, wie Maschinen mit vRealize Automation provisioniert werden können. Um dies zu ermöglichen, müssen Blueprints erstellt und im Service Catalog veröffentlicht werden. In vRealize Automation gibt es vier verschiedene Arten von Blueprints: *Physical Blueprints*, *Virtual Blueprints*, *Cloud Blueprints* und *Multimachine Blueprints*. Nach einer Einführung beschreiben wir die ersten drei Blueprints detailliert in diesem Kapitel, und eine ausführliche Erläuterung der Multimachine Blueprints folgt in Kapitel 7.

### 6.1 Einführung in Blueprints

Ein Blueprint kann – kurz gefasst – als eine Spezifikation einer virtuellen, physischen oder Cloud-Maschine samt ihres Lebenszyklus beschrieben werden und bestimmt als solcher die Eigenschaften der Maschine und wie sie bereitgestellt wird. Dies umfasst diejenigen Informationen, die in den folgenden Abschnitten beschrieben werden.

#### Notwendige Ressourcen

Wenn ein Benutzer eine Maschine aus dem Service Catalog anfordert, wird im Hintergrund der dazugehörige Blueprint geöffnet. Im Blueprint sind alle Informationen enthalten, die vRealize Automation benötigt, um den Provisionierungsvorgang zu initiieren. Dies umfasst als Erstes die Art des Blueprints. Auch die im Blueprint definierten Ressourcen werden ausgelesen und auf ihr Vorhandensein geprüft.

#### Bereitstellungsart

Natürlich ist auch die Art der Provisionierung interessant. Bei virtuellen Maschinen ist das Erzeugen mittels Cloning sehr beliebt, während physische Maschinen beispielsweise mittels eines PXE-Boot-Servers installiert werden können.

Für jede Art der Provisionierung hat vRealize Automation im Model Manager den passenden Workflow gespeichert.

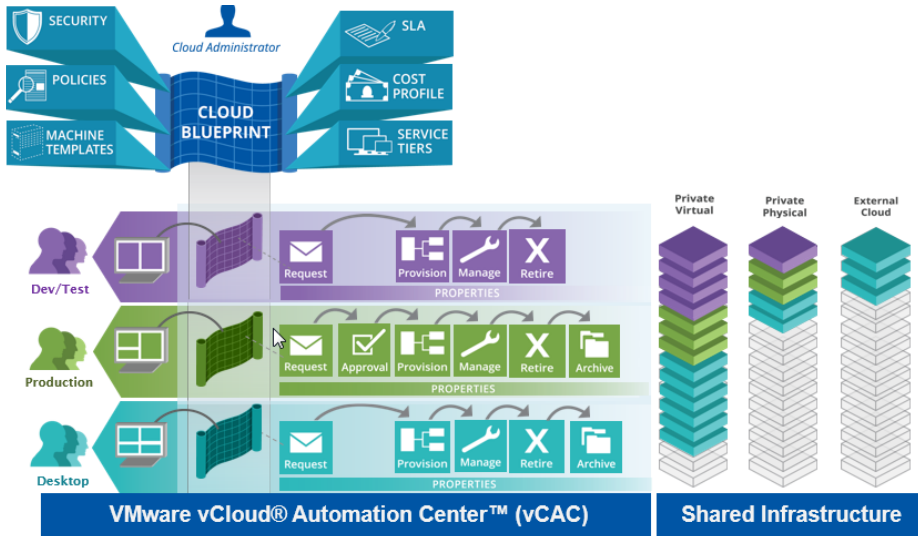


Abb. 6-1 vRealize Automation-Blueprints

### Lebenszyklus

Blueprints regeln darüber hinaus den ganzen Lebenszyklus einer Maschine – von der Anforderung, der Genehmigung über den Betrieb bis zum Archivieren bzw. dem Löschen der Maschine. Abbildung 6-1 zeigt exemplarisch den Lebenszyklus verschiedener Maschinen (Desktop, Produktion, Dev/Test) und wie die Provisionierung mittels eines Cloud-Blueprint definiert wird.

### Erweiterbarkeit

Eine besondere Stärke von vRealize Automation ist darüber hinaus die Möglichkeit, während des Lebenszyklus eigene Workflows auszuführen. Beispielsweise kann definiert werden, dass vor dem Beginn der Provisionierung eine IP-Adresse von einem IP-Adress-Managementsystem angefordert wird, die dann im weiteren Provisionierungsprozess benutzt werden soll.

### Globale und lokale Blueprints

Blueprints können entweder global in einem Tenant gültig sein oder nur in einer spezifischen Business Group definiert sein. Für beide Arten müssen bei der Bereitstellung die entsprechenden Berechtigungen vorhanden sein.

Dabei können Tenant-Administratoren globale Blueprints anlegen und diese bei Bedarf auch als Master Blueprint ausweisen, damit sie als Template benutzt und kopiert werden können.

Business Group Manager können innerhalb ihrer Business Group eigene Blueprints erzeugen, diese können jedoch nicht mit anderen Business Groups geteilt werden.

### Machine Leases

Da vRealize Automation es mittels Blueprints Endbenutzern erlaubt, relativ einfach neue Maschinen zu erzeugen, und dieses Feature natürlich oft gebraucht wird, kann es zu dem Punkt kommen, an dem nicht mehr genug Ressourcen für neue virtuelle Maschinen zur Verfügung stehen. Aus diesem Grund existieren Machine Leases, mit denen man die Bereitstellungsdauer zeitlich begrenzen kann. Ist diese Zeit abgelaufen, kann die Dauer entweder weiter verlängert werden oder die Maschine wird archiviert oder endgültig zerstört. Auf diese Art und Weise wird sichergestellt, dass vergebene Betriebsmittel wieder zurückgegeben werden.

### Reclamations

Nichtsdestotrotz kann es passieren, dass nicht mehr genügend Ressourcen vorhanden sind, um neue Maschinen bereitzustellen. In diesem Fall ist es möglich, wenig genutzte Maschinen zu identifizieren. Dabei kann nach geringer CPU-, Speicher-, Festplatten- oder auch Netzwerkauslastung gesucht werden. Sind mögliche Kandidaten gefunden worden, kann mithilfe eines Reclamation Workflow der Besitzer der Maschine gefragt werden, ob der Rechner noch weiterhin benötigt wird. Je nach Antwort können dann die Betriebsmittel wieder freigegeben werden.

### Konfigurationsänderungen zur Laufzeit

Blueprints erlauben es auch, dass Benutzer in gewissen Maßen Änderungen an einer virtuellen oder Cloud-Maschine vornehmen. So ist es beispielsweise möglich, einem Rechner 4 GB Arbeitsspeicher zu geben. Falls der Benutzer jedoch mehr benötigt, kann er diesen Wert vergrößern. Voraussetzung ist natürlich, dass dies während der Blueprint-Erstellung berücksichtigt und erlaubt wurde.

## 6.2 Allgemeine Blueprint-Einstellungen

Wie besprochen, existieren verschiedene Blueprint-Arten. Da die Konfiguration sich nicht in allen Punkten unterscheidet, sollen zuerst die Punkte erläutert werden, die allen Blueprints gemeinsam sind.

Blueprints finden sich unter **Infrastructure > Blueprints > Blueprints**. Unter dem Menüpunkt **New Blueprint** kann die Art des Blueprints ausgewählt werden, also zum Beispiel **Virtual > vSphere(vCenter)**.

Die sich dann öffnende Seite hat vier verschiedene Registerkarten: **Blueprint Information**, **Build Information**, **Properties** und **Actions**.

**New Blueprint - vSphere (vCenter)**

Create a blueprint to define a catalog item for infrastructure provisioning. You also can copy an existing blueprint to use as a starting point.

Blueprint Information    Build Information    Properties    Actions

\* Name: Win2008 R2 64 bit Plain

Description:

Blueprint options:  Master (can be copied)  
 Display location on request  
 Shared blueprint (can be shared across groups)

Reservation policy:

\* Machine prefix: Use group default

Maximum per user: Unlimited

\* Archive (days): 0

Cost (daily): \$1.000

OK    Cancel

**Abb. 6–2**    *Blueprint Information*

### Blueprint Information

Abbildung 6–2 zeigt die Registerkarte **Blueprint Information**. Hier können Sie die folgenden Einstellungen vornehmen:

1. Geben Sie im Feld **Name** eine Bezeichnung für den Blueprint.
2. Im Feld **Description** können Sie optional eine Beschreibung hinterlegen. Diese wird dann in den Service Catalog übernommen und hilft Benutzern bei der Auswahl des richtigen Blueprints.
3. Das Feld **Master (can be copied)** gibt an, ob der Blueprint als Vorlage fungieren kann und somit mit all seinen Einstellungen kopiert werden kann.
4. In der Checkbox **Display location on request** kann angegeben werden, ob Benutzer den Standort sehen können, an dem die Maschine provisioniert werden soll. Dies kann unter Umständen bei der Auswahl des richtigen Blueprints helfen.
5. Falls die Option **Shared blueprint** gesetzt ist, ist der Blueprint in allen Business Groups sichtbar.
6. Im Feld **Reservation policy** kann der Blueprint mit einer Policy (wie in Kap. 5 beschrieben) verknüpft werden.
7. Das **Machine prefix** wird für die Namensgebung der erzeugten Maschine benötigt.
8. Im Feld **Maximum per user** kann angegeben werden, wie viele Instanzen ein Benutzer von einem Blueprint in höchstem Fall anfordern kann.
9. **Archive (days)** gibt an, was mit der Maschine passieren soll, sobald der Lease ausläuft. Wenn dort kein Wert gesetzt ist, wird die Maschine gelöscht, sobald der Machine Lease ausgelaufen ist. Falls eine Archivierungsdauer

festgelegt wird, wird die Maschine ausgeschaltet und kann innerhalb des Zeitraums wieder reaktiviert werden.

- Das unterste Feld, **Cost (daily)**, bestimmt die Kosten für die Maschine. In Abschnitt 5.2.9 »*Cost Profiles*« wurden bereits die anfallenden Kosten für die Betriebsmittel Speicher, Prozessor und Festplatte definiert. Im Betrieb können aber noch weitere Kosten anfallen, beispielsweise weil die Maschine gewartet werden muss. Dies können Sie an dieser Stelle definieren. Zusammen mit den Kosten aus den Cost Profiles wird dann eine Summe gebildet, um die Gesamtkosten anzugeben.

#### **Tipp: »Display location on request«**

Damit später im Service Catalog auch der Ort, an dem eine Maschine provisioniert werden soll, richtig angezeigt wird, müssen Sie noch Vorarbeiten erledigen.

Dazu müssen Sie sich auf dem Rechner anmelden, auf dem die IaaS-Website läuft. Im vRealize-Automation-Server-Verzeichnis (normalerweise %SystemDrive%\Program Files x86\VMware\vCAC\Server\WebSite\XmlData findet sich die Datei DataCenter-Locations.xml. Für jeden Standort, der hinzugefügt werden soll, muss anschließend ein XML-Fragment mit dem Ort eingefügt werden, also beispielsweise:

```
<CustomDataType>
  <Data Name="Nuernberg" Description="Nuernberg Datacenter" />
  <Data Name="Muenchen" Description="Muenchen Datacenter" />
</CustomDataType
```

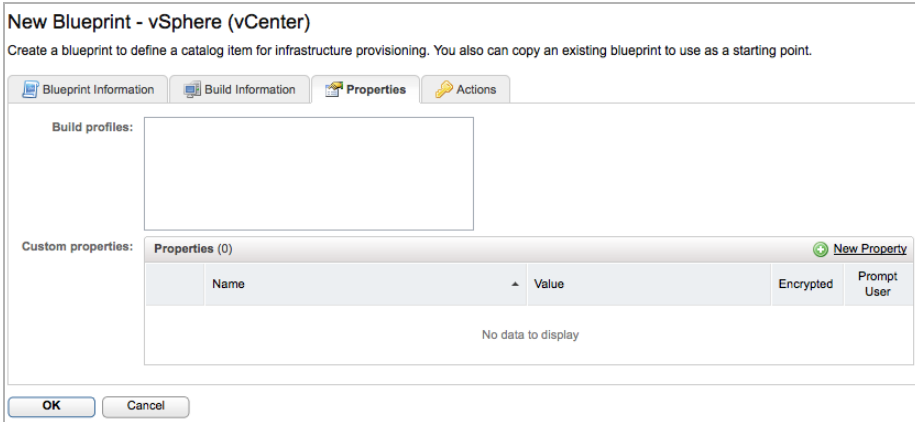
Nach dem Speichern muss der Manager Service neu gestartet werden. Anschließend können Sie unter **Infrastructure > Compute Resources > Compute Resources** die Ortsinformationen hinzufügen. Dazu gehen Sie auf die entsprechende Ressource, klicken auf **Edit** und wählen anschließend den Ort aus.

## **Build Information**

Auf der Registerkarte **Build Information** definieren Sie die Art der Provisionierung und die zu verwendenden Ressourcen. Da diese Registerkarte sich je nach Blueprint-Art unterscheidet, soll sie an dieser Stelle noch nicht weiter thematisiert werden.

## **Properties**

Die Registerkarte **Properties** (siehe Abb. 6–3) nimmt eine wichtige Stellung bei der Anpassung und Erweiterbarkeit von Blueprints ein. Um sie zu verstehen, muss man sich die Funktionsweise eines Bereitstellungsworkflows vergegenwärtigen. Das Ausführen eines Workflows ist dabei ein komplexer Vorgang, und es existieren an vielen Stellen mögliche Erweiterungspunkte, wie zum Beispiel:



**Abb. 6-3** Definition von Custom Properties

- Nach dem Provisionieren soll eine Maschine einer Active Directory-Organisationseinheit hinzugefügt werden.
- Die Anzahl der Cores pro Socket soll eingeschränkt werden, um Lizenzkosten niedrig zu halten.
- Nach dem Provisionieren soll ein Skript ausgeführt werden, um weitere Software zu installieren.
- Für einen Imaging-Vorgang soll eine Sysprep-Datei spezifiziert werden.

Die Liste an Beispielen könnte an dieser Stelle beliebig fortgesetzt werden. Es ist aber klar, dass nicht alle möglichen Szenarien in der grafischen Oberfläche angezeigt werden können. Aus diesem Grund existieren Custom Properties. In Custom Properties können Werte geschrieben werden, die von einem Workflow zur Laufzeit ausgelesen werden und das Verhalten von Workflows ändern können. Wenn beispielsweise eine Custom Property für einen Virtual-Center-Folder in vSphere gesetzt ist, wird ein Clone-Workflow die Maschine dort auch platzieren. Da die Liste der möglichen Properties sehr umfangreich ist, werden wir sie später in Kapitel 10 genauer thematisieren.

Beim Hinzufügen einer Custom Property werden folgende Eigenschaften abgefragt:

- der Name der Property
- ein Wert für die Property
- Optional kann definiert werden, ob der Wert verschlüsselt werden soll (zum Beispiel zum Speichern von Passwörtern).
- Falls ein Wert zur Definitionszeit noch nicht bekannt ist, kann er auch vom Benutzer abgefragt werden (beispielsweise eine Kostenstelle).

Ein konkretes Beispiel für eine Custom Property ist der **Hostname**. Falls diese Property gesetzt ist, wird der Name nicht wie üblich aus dem Machine Prefix generiert, sondern kann vom Benutzer zur Bereitstellungszeit angegeben werden (siehe Abb. 6-4).

Die Liste an Properties kann in vRealize Automation auch erweitert werden. Dafür existiert das *Property Dictionary*. Dies ist vor allem dann sinnvoll, wenn man eigene Workflows implementiert, die Input aus der Blueprint-Definition benötigen.

Custom properties:		Properties (0)		New Property	
	* Name	Value	Encrypted	Prompt User	
	Hostname		<input type="checkbox"/>	<input checked="" type="checkbox"/>	

**Abb. 6-4** Die Custom Property »Hostname«

Bei der Konfiguration von Blueprints wird jedoch zumeist nicht nur eine einzelne Custom Property benötigt, sondern mehrere. Da sich aber das manuelle Hinzufügen von Custom Properties langwierig gestalten kann und zudem die Gefahr besteht, dies zu vergessen, fasst vRealize Automation die Custom Properties zu *Build Profiles* zusammen. Build Profiles können selbst definiert werden, oder Sie können bereits vorhandene Build Profiles nutzen. Beispielsweise bietet vRealize Automation beim Provisionieren für Active Directory ein spezielles Build Profile an, das alle Eigenschaften umfasst, die für das Entfernen von Active Directory benötigt werden (siehe Abb. 6-5).

**New Build Profile**  
Create a build profile to facilitate adding a set of custom properties to a blueprint. You also can copy an existing build profile to use as a starting point.

\* Name: Remove from Active Directory  
Description:

Add from property set: ActiveDirectoryCleanupPlugin [Load]

Custom properties:

Properties (5)		New Property	
Name	Value	Encrypted	Prompt User
Plugin.AdMachineCleanup.Delete	true	No	No
Plugin.AdMachineCleanup.Execute	true	No	No
Plugin.AdMachineCleanup.MoveToOu		No	No
Plugin.AdMachineCleanup.Password		No	No
Plugin.AdMachineCleanup.UserName	Username of AD account with sufficient rights to delete, disable, rename or move AD accounts	No	No

OK Cancel

**Abb. 6-5** Build Profiles

## Actions

Sobald ein Benutzer eine Maschine fertig provisioniert hat, ist es möglich, diese Maschine von vRealize Automation aus zu verwalten. Dabei können verschiedene Tätigkeiten an der Maschine ausgeführt werden, wie zum Beispiel das Verbinden mittels Remote Desktop. Die verschiedenen Aktionen, die auf einer Maschine möglich und erlaubt sein sollen, können Sie auf der Registerkarte **Actions** definieren. Beachten Sie dabei, dass sich hier noch nicht um Benutzerberechtigungen handelt, sondern nur generell darum, welche Aktionen definiert sind (siehe Abb. 6-6).

### New Blueprint - vSphere (vCenter)

Create a blueprint to define a catalog item for infrastructure provisioning. You also can copy an existing blueprint to use as a starting point.

Blueprint Information
Build Information
Properties
Actions

Select the actions that machine owners can perform on this blueprint:

Machine operations:

- Destroy
- Reprovision
- Expire
- Power on
- Power off, Shut down, Suspend
- Reboot, Power cycle
- Connect by using VMRC
- Connect by using RDP or SSH
- Connect by using VDI

Snapshots

Allow snapshots:  Yes  No

Reconfigure

Allow reconfigure:  Yes  No

- Enable execution selector
- Enable power action selector

OK
Cancel

**Abb. 6-6** Definition von Actions

## Veröffentlichen von Blueprints

Nachdem ein Blueprint fertig erstellt worden ist, fehlt noch ein Schritt, damit dieser im Service Catalog hinzugefügt werden kann: Er muss veröffentlicht werden. Führen Sie dazu die folgenden Schritte aus:

1. Wechseln Sie auf die Seite **Infrastructure > Blueprints > Blueprints**.
2. Klicken Sie mit der rechten Maustaste auf den Blueprint, den Sie veröffentlichen wollen, und wählen Sie anschließend aus dem Kontextmenü **Publish** aus.

## 6.3 Virtuelle Blueprints

Die am häufigsten anzutreffenden Blueprints in vRealize Automation sind die virtuellen Blueprints. Virtuelle Blueprints unterstützen dabei verschiedene Plattformen:

- vSphere (vCenter)
- Hyper-V
- Hyper-V (SCVMM)
- KVM (RHEV)
- Xen Server
- Generic