2 Architektur und Produktübersicht

Nachdem wir Ihnen im ersten Kapitel einen grundlegenden Überblick über die Themen Cloud und Automatisierung gegeben haben, legen wir jetzt im zweiten Kapitel den Fokus auf die Architektur des *vRealize Automation Center*. Die Hauptkomponenten sind:

- vRealize Automation Center Appliance:
  Dies ist eine vorkonfigurierte virtuelle Appliance im *Open Virtualization Format* (OVF). Sie umfasst die Webkonsole zur Selfservice-Provisionierung und dient als Administrationsoberfläche.
- vRealize Automation Infrastructure as a Service (IaaS):
  Diese Komponente ist für das Zusammenspiel mit Hypervisoren, Cloud-Umgebungen und physischen Rechnern zum Deployen von neuen Rechnern verantwortlich.
- Identity Appliance:
  Diese vorkonfigurierte Appliance stellt Single-Sign-on-Dienste für vRealize Automation bereit. Falls im Unternehmen zumindest auf *vSphere 5.5 Update 1* aktualisiert worden ist, kann VMware vCenter Server diese Funktion übernehmen.

Daneben existiert noch eine Reihe von weiteren Bausteinen, die wir im Folgenden alle detailliert beschreiben werden. Anschließend zeigen wir auf, welche Versionen von vRealize Automation existieren und wie diese sich lizenztechnisch unterscheiden.

2.1 Architekturüberblick

Neben den bereits erwähnten Hauptkomponenten existieren noch weitere Komponenten, die je nach lizenzierter Edition vorhanden sind:

- VMware IT Business Management Suite Standard Edition:
  Mithilfe dieses Tools ist es möglich, die vorhandene Infrastruktur kostentechnisch zu überprüfen. Dabei kann geklärt werden, wie sich Kosten zusammen-

setzen, wer welche Kosten verursacht, wie Systeme ausgelastet sind oder ob sich die Anschaffung neuer Hardware lohnt.

- Application Services (vorher als VMware Cloud Application Director bekannt): Dabei handelt es sich um eine Appliance, mit der komplette Applikationen und Services – d.h. das Zusammenspiel von Betriebssystemen und Applikationen – automatisch deployt und verwaltet werden können.

Das Zusammenspiel dieser Komponenten ist in Abbildung 2–1 genauer dargestellt.

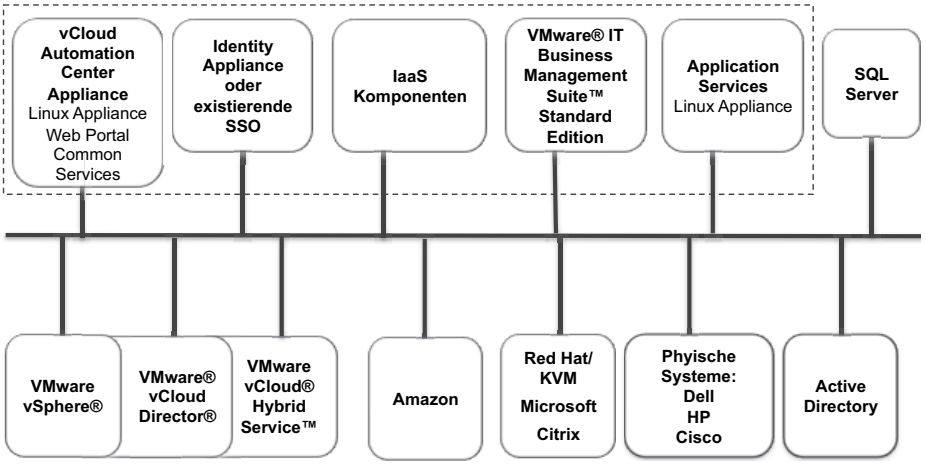

Abb. 2–1 *Logische Architektur von vCloud Automation Center*

Für die erfolgreiche Konfiguration der IaaS-Komponenten ist darüber hinaus noch ein Microsoft SQL Server notwendig. Erwähnenswert ist auch das Zusammenspiel zwischen Active Directory und der Identity Appliance. In den meisten Firmen existiert Active Directory als zentrales Tool zur Verwaltung von Benutzern. Die Identity Appliance ermöglicht eine Anbindung von Active Directory-Umgebungen an vRealize Automation, so dass Benutzer und Gruppen nicht in vRealize Automation verwaltet werden müssen.

Wie wir bereits im ersten Kapitel erwähnt haben, dient vRealize Automation als Cloud-Management-Plattform für Private, Public und Hybrid Clouds. Für die private Cloud werden VMware vSphere, VMware Cloud Director, aber auch Red Hat/KVM, Microsoft Hyper-V oder auch Citrix Xen Server unterstützt. Im Public-Cloud-Bereich kann Amazon angebunden werden. Selbstverständlich darf auch VMware nicht fehlen: vRealize Automation kann VMwares vCloud Air verwalten.

Eine Unterstützung für physische Systeme ist auch gegeben. Zum jetzigen Zeitpunkt werden Systeme mit HP ILO, Dell iDRAC und Cisco UCS unterstützt.

2.2 vRealize-Automation-Komponenten im Detail

2.2.1 vCloud Automation Center Appliance

Die Linux-basierte Appliance stellt sowohl das User-Interface zur Administration als auch das Selfservice-Portal samt Service Catalog für Endbenutzer parat.

Intern besteht die bereits vorkonfigurierte Appliance aus folgenden Komponenten:

- SuSE Linux Enterprise Server (SLES) für VMware
- vPostgresSQL als mitgelieferte Datenbank
- tcServer als Java-Webserver
- vCenter Orchestrator (vCO)
- RabbitMQ Message Broker

2.2.2 vCloud Automation Center Infrastructure as a Service (IaaS)

Die IaaS-Komponente besteht aus verschiedenen Elementen (siehe Abb. 2–2).

- IaaS-Webseite
- Distributed Execution Managers (DEMs)
- Agents
- Model Manager
- Manager Service
- Microsoft-SQL-Server-Datenbank

Die IaaS-Komponenten sind alle in Microsoft .NET implementiert und bildeten in früheren Versionen des Cloud Automation Centers (bis Version 5.2) das alleinige Produkt ab. Folglich müssen sie auch auf Microsoft Windows installiert werden. In den nächsten Abschnitten sehen wir uns diese Komponenten detailliert an.

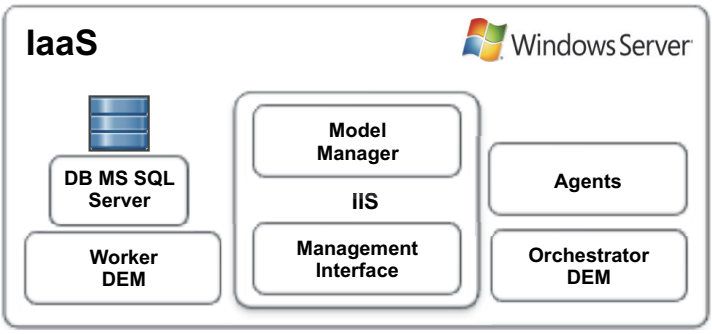

Abb. 2–2 *IaaS-Komponenten*

2.2.2.1 IaaS-Webseite

Die IaaS-Webseite basiert auf dem *Internet Information Server* (IIS) und stellt sowohl den Model Manager als auch das Management Interface zur Kommunikation mit der vCloud Automation Center Appliance dar. Dazu kommuniziert sie mit dem Model Manager, um aktuelle Informationen von den DEMs, Agenten und der Datenbank zu erhalten.

2.2.2.2 Model Manager

Der Model Manager stellt das zentrale Element der IaaS-Komponenten dar. Seine eigentliche Aufgabe ist es, mit externen Systemen sowie Datenbanken zu kommunizieren, Businesslogik zu speichern und diese mittels DEMs auszuführen. Als Beispiel für externe Systeme können *VMware vSphere*, *Microsoft Virtual Machine Manager* oder *Cisco UCS Manager* genannt werden. Intern besteht das Model aus folgenden Bestandteilen:

Data Model und REST-Schnittstelle

Die IaaS-Komponenten speichern alle ihre Daten in einer Microsoft-SQL-Server-Datenbank. Der Zugriff auf diese Daten erfolgt jedoch über eine REST-Schnittstelle. Die Daten werden nach außen als Entitäten veröffentlicht. Beispiele für Entitäten sind alle durch vRealize Automation verwalteten virtuellen Maschinen, alle Hosts oder der verfügbare Speicher.

REST-Webservices

Representational State Transfer (REST) ist ein Architekturstil, mit dem Webservices realisiert werden. Aufgrund seiner Einfachheit, Kompatibilität und Skalierbarkeit stellt dieser Ansatz heute die beliebteste Technik dar, um Webservices zu implementieren.

Security-Informationen zum Zugriff auf Daten und Workflows

Zusätzlich speichert der Model Manager, wer welche Daten sehen kann und welche Aktionen ausgeführt werden können.

Workflows

Wie bereits gesagt, ermöglicht vRealize Automation das Provisionieren von virtuellen und physischen Maschinen auf unterschiedlichen Plattformen. Jedoch schaut der Provisionierungsprozess je nach zu bereitstellendem System unterschiedlich aus. Die dazugehörigen verschiedenen Workflows und deren auszuführende Schritte sind im Model gespeichert. Zusätzlich zu den bereits vorgegebenen Workflows ist es in vRealize Automation möglich, eigene Workflows zu implementieren, die auch im Model Manager gespeichert werden. Beispielsweise könnte

nach Abschluss einer Provisionierung ein Skript ausgeführt werden oder ein Ereignis mitgeloggt werden. Zusammengefasst kann man sagen, dass diese Workflows beschreiben, welche einzelnen Aktivitäten durch vRealize Automation ausgeführt werden.

Events und Trigger

Zur Laufzeit muss vRealize Automation zu jeder Zeit wissen, ob eine neue Workflow-Instanz erzeugt und ausgeführt werden soll. In der Regel geschieht dies auf Wunsch des Benutzers, der zum Beispiel eine neue virtuelle Maschine anfordert. Technisch ist es aber möglich, zusätzliche Events und Trigger zu hinterlegen. Ein Event kann eine Änderung der Daten im Model sein, das Starten einer Aktion durch einen Auftragsplaner (Scheduler), eine Benutzeraktion oder der Zugriff einer externen Applikation. Nachdem ein Ereignis aufgetreten ist, muss der Model Manager entscheiden, ob mittels eines Triggers ein Workflow gestartet werden soll. Events definieren somit, wann eine Aktivität bzw. ein Workflow ausgeführt werden soll.

Informationen zur verteilten Ausführung von Workflows

Die letzte offene Frage ist, welche Workflows an welchem Ort ausgeführt werden. Dies stellt eine durchaus relevante Frage dar, weil vRealize Automation verschiedene Hypervisoren bzw. Cluster an unterschiedlichen Standorten verwalten kann. Um Workflows auszuführen, kommuniziert der Model Manager mit Distributed Execution Managers (DEMs). DEMs werden im nächsten Abschnitt genauer erläutert – kurz gesagt führen diese die Workflows aus.

In der Praxis kann es unterschiedliche Strategien zum Ausführen von Workflows bzw. zum Provisionieren von Maschinen geben. Beispielsweise kann eine Maschine an einem Ort erzeugt werden, der möglichst nah an vRealize Automation angebunden ist. Wenn es weltweit verschiedene Standorte gibt, ist es jedoch auch manchmal notwendig, einen Rechner physisch möglichst nah beim Endbenutzer zu provisionieren. Eine weitere Möglichkeit wäre wiederum, die verfügbaren Datacenter möglichst gleichmäßig auszulasten.

Um diesen Anforderungen nachzukommen, können im Model Manager *Skills* hinterlegt werden. Dabei handelt es sich um eine einfache Zuordnung zwischen Workflows und DEMs. Jeder Workflow kann auf diese Art und Weise die Skills definieren, die er zum Ausführen benötigt. Ist einem DEM dieser Skill auch zugeordnet, kann der Model Manager diesen DEM mit der Ausführung eines Workflows beauftragen.

Eine weitere Besonderheit beim Zusammenspiel zwischen dem Model Manager und dem DEM ist, dass der DEM den benötigten Code zusammen mit allen benötigten Ressourcen (z.B. aufzurufenden Skripten) vom Model Manager herunterladen kann.

Dabei fragen DEMs in regelmäßigen Abständen beim Model Manager an, ob gerade Workflows zur Ausführung anstehen. Falls dies zutrifft, wird der Workflow samt allen notwendigen Artefakten heruntergeladen (siehe Abb. 2–3).

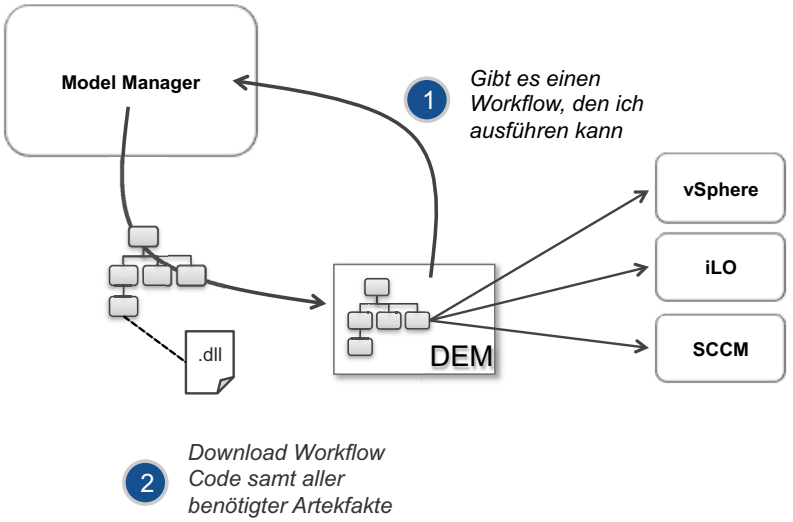

Abb. 2–3 *Zusammenspiel zwischen Model Manager und DEM*

2.2.2.3 Manager Service

Der Manager Service (auch bekannt als *vCloud Automation Center Service*) ist für die Koordination und Kommunikation zwischen den vRealize-Automation-Agenten, der vRealize-Automation-Datenbank, Active Directory und SMTP zuständig. Um Daten von der vRealize-Automation-Datenbank abzurufen, kommuniziert dieser Dienst mit dem Model Manager.

2.2.2.4 Distributed Execution Managers (DEMs)

Unter DEMs versteht man bei vRealize Automation diejenigen Komponenten, die mit dem Model Manager interagieren und Workflows auf externen Systemen ausführen. Es existieren zwei verschiedene DEM-Typen:

- Ein *DEM Orchestrator* kommuniziert mit dem Model Manager und plant die Ausführung von Workflows. Diese führt der DEM Orchestrator jedoch nicht selber aus, sondern übergibt sie einem DEM Worker. Der Orchestrator überwacht vielmehr die konkrete Ausführung und gibt in regelmäßigen Abständen Rückmeldung über den aktuellen Zustand an den Model Manager. Es kann zur gleichen Zeit nur einen aktiven DEM Orchestrator geben. Es empfiehlt sich aber aus Redundanzgründen, einen zweiten passiven DEM Orchestrator bereitzustellen, der beim Ausfall des primären DEM Orchestrators aktiviert werden kann.

- *DEM Worker* übernehmen die tatsächliche Ausführung der Workflows. Im Gegensatz zum DEM Orchestrator können verschiedene DEM Worker zur gleichen Zeit aktiv sein. Während es sich empfiehlt, den DEM Orchestrator möglichst nahe am vRealize Automation Server zu installieren, sollten DEM Worker in der Nähe der externen Systeme konfiguriert werden.

2.2.2.5 Agenten

Jedoch haben DEMs nicht immer genug Logik implementiert, um auf allen externen Plattformen alle Aufrufe selbst durchführen zu können. Aus diesem Grund ist diese Logik in Agenten ausgelagert worden. (Genauer gesagt gilt dies vor allem für Systeme, für die historisch als Erstes eine Anbindung implementiert wurde – neuerdings wurde die DEM-Logik für externe Systeme direkt erweitert, wie zum Beispiel für KVM). vRealize Automation Center besitzt dabei folgende verschiedene Agentenarten:

- *Virtualisierungs-Proxy-Agenten* werden benutzt, um Daten von virtualisierten Umgebungen abzurufen (zum Beispiel bereits vorhandene Templates oder Cluster-Informationen) oder um virtuelle Maschinen zu provisionieren. Virtualisierungs-Proxy-Agenten werden als Windows-Dienst installiert und existieren für vSphere-, Hyper-V- oder Xen-Server. Beachten Sie aber, dass nicht für jeden Hypervisor ein Agent installiert werden muss. So gelingt die Integration mit KVM-Hypervisoren beispielsweise ohne den Einsatz von Agenten.
- Die zweite Gruppe von Proxies dient zur *Virtual Desktop Integration* (VDI). Als Beispiel können hier *External Provisioning Integration* (EDI) oder *VDI Windows PowerShell*-Agenten genannt werden.
- *Windows Management Instrumentation*-(WMI-)Agenten ermöglichen das Sammeln von Daten (zum Beispiel ermitteln sie den Active Directory-Status eines Rechners) für Rechner, die von vRealize Automation verwaltet werden. Dies ist vor allem für das Provisionieren mittels WIM (Windows Imaging File Format) relevant.

Aber auch auf den zu provisionierenden Gastbetriebssystemen können Agenten installiert werden. Dabei ist aber zu beachten, dass diese bereits im Template vorinstalliert sein müssen. Guest-Agenten ermöglichen folgende Funktionalitäten:

- Nach der Provisionierung können Disk-Operationen ausgeführt werden. Dies umfasst beispielsweise das Partitionieren, das Formatieren oder das Mounten von Festplatten.
- Sobald eine Maschine provisioniert ist, ist es oft notwendig, ein Skript auszuführen. Auf diese Art und Weise kann der Rechner weiter konfiguriert werden bzw. kann weitere Software nachinstalliert werden.
- Oft ist es auch notwendig, weitere Netzwerkkarten des Betriebssystems zu konfigurieren. Dazu bietet vRealize Automation spezielle Network Operations.