Using VMware vCloud Application Director

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.
You can find the most up-to-date technical documentation on the VMware Web site at:
http://www.vmware.com/support/
The VMware Web site also provides the latest product updates.
If you have comments about this documentation, submit your feedback to:
docfeedback@vmware.com
Contents

1 Introducing vCloud Application Director 9
   vCloud Application Director Overview 9
   Core Architectural Principles 10
   Key Concepts 12

2 Install and Configure vCloud Application Director 17

3 Installing vCloud Application Director 19
   Preparing to Install vCloud Application Director 19
   Start the vCloud Application Director Appliance 25
   Troubleshooting Problems Connecting to the vCloud Application Director Web Interface 27
   Unlock Your darwin_user Account 27
   Restarting vCloud Application Director 28
   Configure vCloud Application Director to Use a Proxy for External URLs 28
   Register vCloud Application Director to vCloud Automation Center 30
   Upgrading vCloud Application Director 31

4 Setting Up Users and Groups 33
   Overview of Roles, Users, and Groups 33
   Predefined Users, Groups, and Roles 36
   Start the CLI Remotely 36
   Create Users and Groups with vCloud Application Director CLI 37
   Create and Activate an LDAP Configuration 38
   Import a SSL Certificate for Secure LDAP Connection 39

5 Using the vCloud Application Director Web Interface 41
   Log In to vCloud Application Director 41
   Using the vCloud Application Director Web Interface 42

6 Setting Up Application Provisioning for the vCloud Automation Center Environment 47
   Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates 48
   Creating Virtual Machine Templates in vCloud Automation Center 49
   Updating Existing Virtual Machine Templates in vCloud Automation Center 57
   Register the vCloud Automation Center Cloud Provider and Template 57
   Create a vCloud Automation Center Deployment Environment 59

7 Setting Up Application Provisioning for the vCloud Director Environment 69
   Virtual Machine Requirements for Creating vCloud Director Custom Templates 70
Creating Windows Virtual Machine Templates in vCloud Director 71
Create Linux Virtual Machine Templates in vCloud Director 75
Verify Cloud Template Configuration from the vCloud Director Catalog 77
Updating Existing Virtual Machine Templates in vCloud Director 77
Exporting Virtual Machine Templates with OVF Format 78
Register the vCloud Director Cloud Provider and Template 78
Create a vCloud Director Deployment Environment 79

8 Setting Up Application Provisioning for the Amazon EC2 Environment 81
Configure Amazon EC2 Environment for vCloud Application Director 82
Virtual Machine Requirements for Creating Amazon EC2 Custom Templates 86
Create Amazon EC2 Virtual Machine Templates or AMIs 87
Register the Amazon EC2 Cloud Provider and Template 88
Create an Amazon EC2 Deployment Environment 89

9 Developing vCloud Application Director Components 91
Defining Component Actions 91
Configuring Component Properties 93
vCloud Application Director Predefined Properties 100
Secured Component Properties 101
Required Component Properties 101
Best Practices for Developing Components 101

10 Managing the vCloud Application Director Catalog 103
Add Operating Systems to the Catalog 104
Add Tags to the Catalog 104
Add a Service to the Catalog 105
Import a Puppet Service to the Catalog 112
Add an External Service to the Catalog 113
Add a Logical Template to the Catalog 121
Add a Policy to the Catalog 124
Add a Custom Task to the Catalog 129

11 Creating Applications 133
Create an Application 134
Creating an Advanced Blueprint 138
Copy an Application Version 142
Copy an Application 143
Delete an Application Version 143
Working with Marketplace 144

12 Deploying Applications 147
Set Up and Configure a Deployment Profile 147
Deploy an Application 155
Publish a Deployment Profile to the vCloud Automation Center Service Catalog 157
Request a vCloud Automation Center Service Catalog Item 158
Understanding the Deployment and Update Process 159
Using the Deployment Summary Page 160
13 Updating Application Deployments 163
   Initiate an Update Process to Scale Out Deployments 164
   Initiate an Update Process to Scale In Deployments 166
   Initiate an Update Process to Modify Configurations 169
   Use an Existing Update Profile 171
   Promote an Update Profile 172
   Rollback an Update Process 174
   Troubleshoot Failed Update Process to Scale Deployments 175
   Troubleshoot Failed Update Process to Modify Configuration 175

14 Deploying Predefined Catalog Components 177
   Using the Sample Templates 177
   Available Sample Applications for Deployment 178
   Update Deployed Sample Applications 183
   Sample vCloud Application Director Catalog Tasks 186
   Sample vCloud Application Director Catalog Services 191

15 Managing Deployments 193
   View Deployment Task and Blueprint Details for an Application 193
   View Deployed VM Details and Execution Plan of an Application 195
   Start a Policy Scan 197
   Tear Down an Application from the Cloud 197
   Delete an Application Deployment from vCloud Application Director 199
   Cancel a Deployment or an Update Process 199
   View Policy Compliance Summary 200

16 Using the vCloud Application Director CLI 203
   General CLI Options 203
   Managing Users and Groups 204
   Managing LDAP Configurations 206
   Managing Cloud Tunnels 207
   Deploying and Updating an Application Using CLI 208
   Using CLI to Tear Down a Deployment 212

17 Using the CLI Import and Export Functions 213
   Using the CLI Export Command 213
   Using the CLI Import Command 216

Index 221
Using VMware vCloud Application Director

vCloud Application Director automates application provisioning in the cloud including deploying, configuring, and updating the application’s components and dependent middleware platform services on infrastructure clouds. vCloud Application Director 6.0 simplifies complex deployments of custom and packaged applications on infrastructure clouds that are based on vCloud Director, vSphere, and Amazon Elastic Compute Cloud (Amazon EC2).

This documentation describes how to install and use vCloud Application Director to create, deploy, manage, and update applications across virtual and cloud-based infrastructures.

Intended Audience

This information is intended for anyone who wants to install and use vCloud Application Director for application deployments. This audience includes application architects and application deployers who work in collaboration with application infrastructure administrators and cloud administrators.
VMware® vCloud™ Application Director is a model-based application provisioning solution that simplifies creating and standardizing application deployment topologies on multiple infrastructure clouds. Application architects can use a graphic-based canvas with a drag-and-drop interface to model application deployment topologies called application blueprints.

These application blueprints define the structure of the application, enable the use of standardized application infrastructure components, and include installation dependencies and default configurations for custom and packaged enterprise applications. Application blueprints are logical deployment topologies that are portable across VMware-based IaaS clouds such as vCloud Director and vCloud Automation Center and across public clouds such as Amazon EC2.

This chapter includes the following topics:

- “vCloud Application Director Overview,” on page 9
- “Core Architectural Principles,” on page 10
- “Key Concepts,” on page 12

vCloud Application Director Overview

vCloud Application Director has a model-driven, open, and extensible architecture. With its catalog of standard components, or services, vCloud Application Director automates and manages the update life cycle of deployments for multitier enterprise applications in hybrid cloud environments.

Enterprise users can standardize, deploy, configure, update, and scale complex applications in dynamic cloud environments. These applications can range from simple Web applications to complex custom applications and packaged applications.

To automate application deployments to a supported cloud environment, users must configure the components in the cloud abstraction layer (CAL). The cloud template contains a predefined reusable machine image that includes an operating system and data that is applied to a virtual machine when it is created. A cloud template is mapped to a logical template in the vCloud Application Director catalog. The cloud provider offers a cloud instance for deployment. The deployment environment provides an environment for the cloud provider. Both the cloud provider and deployment environment map components from the cloud environment to vCloud Application Director.

Application architects can use the drag-and-drop interface to create visual application blueprints. Application architects can use the prepopulated and extensible catalog of standard logical templates, application infrastructure service, components, and scripts to model an application blueprint. These blueprints standardize the structure of the application, including software components, dependencies, and configurations, for repeated deployments.
Application blueprints are portable across deployment environments. After a blueprint is available, application development, QA, and release teams can work in the standards set by IT. Teams can repeatedly deploy a standard blueprint, customize configurations as allowed, and deploy within IT-approved deployment environments.

From an application blueprint, you can create different deployments using deployment profiles to test prototypes or deploy mission-critical multitier applications in production environments. From these saved blueprints, the application deployer can generate execution plans for deploying the application to a private or public cloud. You can also initiate an update process to scale clustered nodes of deployed applications and change the configuration or code of deployed applications when a new version is available.

Figure 1-1. vCloud Application Director Provisioning and Updating Workflow

---

**Core Architectural Principles**

vCloud Application Director is designed to automate deployments of complex applications across any IaaS cloud. vCloud Application Director 6.0 automates deployments in IaaS cloud environments with vCloud Director, vCloud Automation Center to deploy to vSphere, and Amazon EC2.

- Deploying Any Application and Middleware Service on page 11
  With vCloud Application Director, application architects can use virtual machine templates and scripts to model an application deployment with middleware services and applications.

- Multicloud Support on page 11
  vCloud Application Director is designed to deploy the same application to multiple types of clouds.

- vCloud Application Director Extensibility and Open Architecture on page 11
  vCloud Application Director is optimized for vCloud components and is extensible to other components.

- Standardization in vCloud Application Director on page 12
  With vCloud Application Director, you can create reusable services using standardized configuration properties to meet strict requirements for IT compliance.
vCloud Application Director grants local users, LDAP users, and LDAP groups specific roles for their functions.

**Deploying Any Application and Middleware Service**

With vCloud Application Director, application architects can use virtual machine templates and scripts to model an application deployment with middleware services and applications.

With vCloud Application Director 6.0, you can deploy applications on Windows and Linux operating systems in the vCloud Director and vCloud Automation Center cloud environments. For the Amazon EC2 cloud environment, you can only deploy applications on Linux operating systems.

- Application architects can use virtual machine templates from a standardized cloud provider library defined for their enterprise.
- Application architects can also add application components to the middleware services defined in the application blueprint.

**Multicloud Support**

vCloud Application Director is designed to deploy the same application to multiple types of clouds.

- vCloud Application Director encapsulates deployment settings in deployment profiles, separate from application blueprints. Deployment profiles enable application blueprint portability across private clouds based on vCloud Director, the public clouds based on Amazon EC2, and use the vCloud Automation Center provisioning infrastructure to access the vSphere private and public clouds.
- vCloud Application Director uses a CAL to plug in cloud providers. vCloud Application Director automates deployments to the vCloud Director, the vSphere through vCloud Automation Center, and the Amazon EC2 cloud environment. vCloud Director uses open standards like the vCloud API and the Open Virtualization Format (OVF). vCloud Application Director integrates with vCloud Director through the publicly available IaaS API that is based on vCloud. This integration allows you to deploy applications in private and public deployments of vCloud Director clouds.

vCloud Automation Center uses REST APIs to deploy applications to the vSphere private and public clouds.

vCloud Application Director uses the Amazon AWS APIs and Amazon Machine Images (AMIs) to deploy applications to the Amazon EC2 cloud.

**vCloud Application Director Extensibility and Open Architecture**

vCloud Application Director is optimized for vCloud components and is extensible to other components.

- The vCloud Application Director catalog has predefined (out-of-the-box) services or applications for middleware services. In addition, you can add pointers to virtual machine templates residing in cloud catalogs. The vCloud Application Director catalog also lets you add definitions of dynamically installable custom services on virtual machine templates using install, configure, start, update, rollback, and teardown scripts, and appropriate configurations.

- Deployment execution plans are generated by the system based on the blueprint. These plans help users to track the status and progress of tasks during deployment.

You can add custom task scripts to these plans to run in the deployed virtual machine. You can use these scripts to perform additional tasks such as security patches, audit integrations, quality and compliance reviews using third-party internal IT systems, and running smoke tests.

- To streamline the build to deployment process, organizations can further automate deployment by using the command-line interface to allow their continuous build systems or cloud provisioning portals to generate and deploy an application.
Standardization in vCloud Application Director

With vCloud Application Director, you can create reusable services using standardized configuration properties to meet strict requirements for IT compliance.

- vCloud Application Director provides a model-driven architecture that enables adding IT certified virtual machine templates and middleware services within the application blueprint.
- vCloud Application Director includes a delegation model for overriding configuration name value pairs between catalog administrator, application architect, and deployer to standardize configuration values for application and middleware service.

User Group-Based Security

vCloud Application Director grants local users, LDAP users, and LDAP groups specific roles for their functions.

You can group together local users, LDAP users, and LDAP groups to isolate applications, deployments, and deployment environments per group.

vCloud Application Director also supports an authentication mechanism through vCloud Automation Center. All of the users in the identity stores (IDP) configured in vCloud Automation Center for the tenant registered with vCloud Application Director, can log in with SSO credentials.

Key Concepts

To deploy an application, you must configure CAL. When you model a blueprint you can use virtual machine templates that you obtain directly from a cloud provider library, as well as an extensible catalog of ready-to-use application infrastructure components and scripts. After you model your application deployment topology, you can create dependencies and edit configurations to finalize your execution plan. After you deploy the application, you can also initiate an update process to scale clustered nodes, or change the configuration or code of deployed applications when a new version is available.

The following definitions help you understand the provisioning and updating workflow.

Main components of CAL.

Cloud Template

Cloud template that contains a predefined reusable machine image that includes an operating system and data. When you create virtual machines for your cloud environment, the specifications and software defined in the cloud template are applied to that virtual machine.

<table>
<thead>
<tr>
<th>vCloud Application Director Environment</th>
<th>Cloud Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud template</td>
<td>vCloud Director vApp template</td>
</tr>
<tr>
<td></td>
<td>vCloud Automation Center blueprint</td>
</tr>
<tr>
<td></td>
<td>Amazon Machine Image (AMI)</td>
</tr>
</tbody>
</table>

Cloud provider

A cloud instance for deployment. You can define several cloud providers for a cloud provider type.
Table 1-2. Cloud Provider Mapping to Supported Cloud Environments

<table>
<thead>
<tr>
<th>vCloud Application Director Environment</th>
<th>Cloud Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud provider</td>
<td>vCloud Director organization</td>
</tr>
<tr>
<td></td>
<td>vCloud Automation Center 5.2 provisioning group</td>
</tr>
<tr>
<td></td>
<td>vCloud Automation Center 6.0 business group</td>
</tr>
<tr>
<td></td>
<td>Amazon Region</td>
</tr>
</tbody>
</table>

Cloud provider type

Type of cloud infrastructure on which deployments can be made. vCloud Application Director supports only vCloud Director, vCloud Automation Center, and Amazon EC2.

Deployment environment

An environment in a cloud provider, for example, development, test, staging, and production. A cloud provider can have multiple deployment environments.

Table 1-3. Deployment Environment Mapping to Supported Cloud Environments

<table>
<thead>
<tr>
<th>vCloud Application Director Environment</th>
<th>Cloud Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment environment</td>
<td>vCloud Director organization vDC</td>
</tr>
<tr>
<td></td>
<td>vCloud Automation Center reservation policy</td>
</tr>
<tr>
<td></td>
<td>Amazon VPC</td>
</tr>
</tbody>
</table>

For vCloud Director, a deployment environment maps to an organization virtual datacenter (vDC) for a defined cloud provider and uses resources from that vDC. An organization vDC provides resources to an organization and is partitioned from a provider vDC. Organization vDCs provide an environment where virtual systems can be stored, deployed, and operated. They also provide storage for virtual media, such as floppy disks and CD ROMs. A single organization can have multiple organization vDCs.

For vCloud Automation Center, a deployment environment maps to a reservation policy. If a deployment environment is not selected, vCloud Automation Center assigns a reservation policy depending on the resource requirements of the virtual machine in the vCloud Automation Center blueprint.

For Amazon EC2, a deployment environment maps to a combination of Amazon Virtual Private Cloud (VPC) and one of the Availability Zones in a region.

Main catalog components.

Catalog

Library that contains logical templates, which are pointers to cloud templates. Reusable services that can be used in multiple applications and installed on a virtual machine. Tasks that can perform additional customized tasks in an application deployment.

Service

Scripted software that can be installed on a virtual machine and reused in multiple applications.
### External Services
An application service such as a hardware load-balancer or a preinstalled database that is installed external to the deployment of the application. The application and the external service must be configured to work with each other.

### Logical template
A predefined virtual machine definition in vCloud Application Director. A logical template can be mapped to an actual cloud template in the cloud catalog and supported services. Logical templates allow an application blueprint to remain cloud agnostic.

### Policy
A user-defined set of definitions to govern application life cycle operations. For example, a policy can blacklist the use of certain software based on corporate guidelines. Policies are enforced when they are enabled in specific deployment environments through the use of policy instances.

### Custom Tasks
From the execution plan, you can add custom tasks to perform additional customized tasks such as run security patches in an application deployment. You can create a custom task in the catalog and add it to an application deployment. vCloud Application Director also provides predefined tasks in the catalog that you can use to configure an APT repository, a YUM repository, register a machine with a Red Hat Network, or Join Domain.

### Operating system
Specifies an operating system that the IT organization for logical templates and services supports. A list of operating systems appears in the **Operating systems** menu, and you can add to the list.

### Tag
Organizes the lists of logical templates and services to enhance readability in the blueprint editor. A list of tags appears in the **Tags** menu, and you can add new tags to the list.

### Main components of application provisioning,

#### Application
Logical deployment unit, which defines the relationship between operating system templates, application components and their dependent services that can be distributed across multiple virtual machines.

#### Application components
Custom code used as a template for components such as EAR files, WAR files, and so on. They are custom script packages for the install, configure, start, update, rollback, and teardown actions on a node or service.

#### Application Blueprint
Logical topology of an application for deployment. A blueprint captures the structure of an application with logical nodes, their corresponding services and operating systems, dependencies, default configurations, and network and storage topology requirements.

#### Node
Virtual machine defined in the blueprint.

#### Clustered Node
Cluster of virtual machines defined in the blueprint.

#### Disks
Additional disks to be added to the corresponding virtual machine or node.

#### Properties
Configuration name-value pairs for services and application components. These are variables used by the scripts to set parameters on a script and run various configurations. For example, you can set the installation_path property value and configure installation scripts to use this property to specify the path to use to install a service during the application deployment process.

#### Actions
Life cycle stages for the install, configure, start, update, rollback, and teardown scripts for services and application components to be installed.
<table>
<thead>
<tr>
<th><strong>Storage</strong></th>
<th>Space provided by cloud provider to place the disk into different storages to meet the performance requirements.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment profile</strong></td>
<td>Collection of deployment settings for a blueprint, including cluster size, CPU, memory, cloud templates, and networks.</td>
</tr>
<tr>
<td><strong>Logical network</strong></td>
<td>An entity created as a logical abstraction for a network. A logical network is a cloud agnostic abstraction used to indicate network locality between nodes in an application. A logical network is dynamically mapped to a cloud network during the configuration of a deployment profile.</td>
</tr>
<tr>
<td><strong>Cloud network</strong></td>
<td>A network defined within the cloud deployment environment.</td>
</tr>
<tr>
<td><strong>Execution plan</strong></td>
<td>Task plan for viewing the order in which virtual machines are created and action scripts for catalog and application components are installed, configured, started, and updated. The order in which an agent performs a task is defined in the deployment execution plan.</td>
</tr>
<tr>
<td><strong>Teardown</strong></td>
<td>Remove a vCloud Director vApp and associated virtual machines, vCloud Automation Center virtual machine and the associated virtual machine in vCenter Server, or Amazon EC2 instances of a deployed application from the cloud environment. After the teardown process is complete, all of the other resources such as storage and IP addresses that are used by these virtual machines are returned to their respective pools.</td>
</tr>
</tbody>
</table>
vCloud Application Director includes predefined sample applications, services, and virtual machine templates to help you understand the basic concepts and start using the product. Complete the tasks to install vCloud Application Director, configure, and deploy a predefined sample application to the vCloud Director, vSphere through vCloud Automation Center, or Amazon EC2 environment.

Prerequisites
Familiarize yourself with the vCloud Application Director provisioning workflow. See “vCloud Application Director Overview,” on page 9.

Procedure
1. Install and set up the vCloud Application Director virtual appliance.
   “Preparing to Install vCloud Application Director,” on page 19 and “Start the vCloud Application Director Appliance,” on page 25.
2. Configure vCloud Application Director to use a proxy.
   Perform this task when an application needs to download files from outside the corporate firewall.
   See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
3. (Optional) Enable the predefined user accounts and set account passwords using the CLI, to provide privileges to specific product areas.
   The admin user account is enabled by default.
   See “Predefined Users, Groups, and Roles,” on page 36.
4. Log in to the vCloud Application Director Web interface and familiarize yourself with the product features.
   See “Log In to vCloud Application Director,” on page 41 and “Using the vCloud Application Director Web Interface,” on page 42.
5 Register a cloud provider.

When you register a cloud provider, you map vCloud Application Director to one of the following components in your cloud environment.

<table>
<thead>
<tr>
<th>Cloud Provider Types</th>
<th>Reference Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCloud Director instance and organization</td>
<td>See “Register the vCloud Director Cloud Provider and Template,” on page 78</td>
</tr>
<tr>
<td>vCloud Automation Center provisioning or business group</td>
<td>See “Register the vCloud Automation Center Cloud Provider and Template,” on page 57</td>
</tr>
<tr>
<td>Amazon EC2 Region</td>
<td>See “Register the Amazon EC2 Cloud Provider and Template,” on page 88</td>
</tr>
</tbody>
</table>

6 Create a deployment environment in the registered cloud provider.

Map the vCloud Application Director deployment environment to one of the following components in your cloud environment before you can deploy an application.

<table>
<thead>
<tr>
<th>Deployment Environment</th>
<th>Reference Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization vDC within vCloud Director</td>
<td>See “Create a vCloud Director Deployment Environment,” on page 79</td>
</tr>
<tr>
<td>Reservation policy within vCloud Automation Center</td>
<td>See “Create a vCloud Automation Center Deployment Environment,” on page 59</td>
</tr>
<tr>
<td>Amazon VPC and associated Availability Zone</td>
<td>See “Create an Amazon EC2 Deployment Environment,” on page 89</td>
</tr>
</tbody>
</table>

7 Map the cloud template to a logical template.

When you map a cloud template to a logical template, you are linking the vCloud Application Director logical template to the actual template in the cloud.

See “Add a Logical Template to the Catalog,” on page 121.

8 Deploy a predefined sample application from the Deployment Profile wizard.

See Chapter 14, “Deploying Predefined Catalog Components,” on page 177 and “Create a Deployment Profile,” on page 148.

9 Check the status of the deployment.

During deployment, components are installed and configured based on the dependencies of an application. See “Understanding the Deployment and Update Process,” on page 159.

You can use the user interface to check the status of an application deployment in real time. See “Using the Deployment Summary Page,” on page 160.

10 Troubleshoot deployment failures.

If you experience deployment failures, you can examine the virtual machine-specific logs and deployment logs and troubleshoot the problem.

To access the virtual machine-specific logs, see “View Deployed VM Details and Execution Plan of an Application,” on page 195. To resolve the problem, see the VMware vCloud Application Director Troubleshooting guide.
Installing vCloud Application Director

To install vCloud Application Director 6.0, you can create and deploy the virtual appliance in either vCloud Director or in vSphere vCenter.

This chapter includes the following topics:

- “Preparing to Install vCloud Application Director,” on page 19
- “Start the vCloud Application Director Appliance,” on page 25
- “Troubleshooting Problems Connecting to the vCloud Application Director Web Interface,” on page 27
- “Unlock Your darwin_user Account,” on page 27
- “Restarting vCloud Application Director,” on page 28
- “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28
- “Register vCloud Application Director to vCloud Automation Center,” on page 30
- “Upgrading vCloud Application Director,” on page 31

Preparing to Install vCloud Application Director

Before you begin installing the vCloud Application Director appliance, verify that your computing environment meets the hardware and software system requirements.

vCloud Application Director requires that vCloud Director use specific configuration settings. For previously installed configurations of vCenter and vCloud Director servers, verify that these servers use the settings that work with vCloud Application Director.

- **vCloud Application Director System Requirements** on page 20
  
  The virtual appliance on which you run vCloud Application Director must meet certain hardware and software requirements. In addition, you must open certain ports for vCloud Application Director.

- **Set Up VMware vCenter Cluster Configurations** on page 21
  
  You must set up the storage and network of your vCenter cluster configurations to meet certain requirements so that you can install vCloud Application Director in vCloud Director or vSphere.

- **Set Up vCloud Director for vCloud Application Director** on page 23
  
  You must configure a virtual appliance in vCloud Director to successfully install vCloud Application Director. If you have a previously installed version of vCloud Director, you must check the configuration settings to optimize for vCloud Application Director.
vCloud Application Director System Requirements

The virtual appliance on which you run vCloud Application Director must meet certain hardware and software requirements. In addition, you must open certain ports for vCloud Application Director.

Disk Space and Memory Requirements

vCloud Application Director requires a minimum of 20GB of disk space, 2GB memory, and 1 vCPU with a speed of 2GHz.

**CAUTION** For some organization vDCs, by default vCloud Director sometimes sets the virtual CPU to 0.24GHz, based on the vCloud Director setup. If this setting is the default in your environment, you must set the vCPU speed to 2GHz for the organization vDC in which the vCloud Application Director vApp is deployed. Otherwise, the performance of the vCloud Application Director virtual appliance is affected.

Port Requirements

You must open certain ports for the vCloud Application Director virtual machine.

<table>
<thead>
<tr>
<th>Port</th>
<th>Connection Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP Port 8443</td>
<td>External port for the vCloud Application Director user interface connection through HTTPS.</td>
</tr>
<tr>
<td>TCP Port 8080</td>
<td>Optional port for the vCloud Application Director user interface connection through HTTP.</td>
</tr>
<tr>
<td>TCP Port 80</td>
<td>External port for vami-lighttpd.</td>
</tr>
<tr>
<td>TCP Ports 5671</td>
<td>External port for vFabric RabbitMQ.</td>
</tr>
<tr>
<td>TCP Port 443</td>
<td>External port for the vCloud Application Director user interface to connect to a cloud environment.</td>
</tr>
<tr>
<td>TCP Port 22</td>
<td>Optional port for the external SSH connection.</td>
</tr>
</tbody>
</table>

Licensing Requirements

You must type the serial number of the VMware vCloud Application Director for Release Automation license edition to start a new vCloud Application Director appliance.

Web Interface Support

vCloud Application Director supports the following Web browsers:

- Mozilla Firefox 25.0 and 24.0.
  - The browser is supported on Mac 10.7 and Mac 10.6, Windows 7 64-bit, and Linux.
- Chrome 30.0 and 29.0.
  - The browser is supported on Mac 10.6 and Windows 7 64-bit.
- Internet Explorer 10.0.

Virtualization Software Requirements

To use vCloud Application Director, you must install and set up the following VMware products:

- vSphere ESXi 5.0, 5.1, or 5.5. See vSphere documentation
- VMware vCenter Server 5.0, 5.1, or 5.5. See vSphere documentation
**Supported Operating Systems for Virtual Machine Templates in the vCloud Application Director Catalog**

To create custom virtual machine templates to use in the vCloud Application Director catalog, verify that the supported operating systems are available in the following products:

- vCloud Director. See “Virtual Machine Requirements for Creating vCloud Director Custom Templates,” on page 70.
- Amazon EC2. See “Virtual Machine Requirements for Creating Amazon EC2 Custom Templates,” on page 86.

**NOTE** Linux and Windows virtual machine templates with Federal Information Processing Standard (FIPS) enabled are not supported.

**VMRC Plug-In Support for vCloud Director**

The VMRC plug-in lets you connect to a deployed virtual machine directly from vCloud Application Director. This plug-in works only on supported versions of Mozilla Firefox running on Windows operating systems. The VMRC plug-in is not supported on the Mac operating system.

**Set Up VMware vCenter Cluster Configurations**

You must set up the storage and network of your vCenter cluster configurations to meet certain requirements so that you can install vCloud Application Director in vCloud Director or vSphere.

**Prerequisites**

- Familiarize yourself with the procedures for creating vSphere 5.0, 5.1 or 5.5 resource pools and vCenter clusters that have DRS enabled. See the vSphere documentation center.
- Verify that you have one or more hosts running vSphere ESXi 5.0, 5.1, or 5.5.
- Verify that you have a vCenter cluster with DRS enabled.
- Verify that you have the NTP client running on each of the vSphere ESXi hosts in your vCenter installation.

Synchronize the time on the host on which vCloud Application Director is deployed and on the hosts on which virtual machines are to be deployed. If the time is not synchronized, the vCloud Application Director server might experience problems when communicating with the provisioned virtual machines.

To synchronize the time on the hosts, set an NTP server to the **Configuring Time** option for each ESXi host underlying the vCloud Director system.

**Procedure**

1. Create a cluster with DRS enabled.
Check the configuration settings for datastore and network requirements.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiple hosts in a cluster</td>
<td>■ Verify that all of the hosts have at least one shared datastore, for</td>
</tr>
<tr>
<td></td>
<td>example, shared LUN, NFS, and so on.</td>
</tr>
<tr>
<td></td>
<td>■ Verify that all of the hosts in the cluster have connectivity to at</td>
</tr>
<tr>
<td></td>
<td>least one common network.</td>
</tr>
<tr>
<td></td>
<td>■ To support live virtual machine migration, verify that all of the</td>
</tr>
<tr>
<td></td>
<td>hosts in the cluster are identical.</td>
</tr>
<tr>
<td>Network or SAN storage</td>
<td>■ Verify that your host has shared storage.</td>
</tr>
<tr>
<td></td>
<td>Although vCloud Director and vCloud Application Director can use</td>
</tr>
<tr>
<td></td>
<td>local storage, shared storage ensures future scalability.</td>
</tr>
<tr>
<td></td>
<td>■ Verify that the host has at least one network.</td>
</tr>
</tbody>
</table>

What to do next

For vSphere, deploy the vCloud Application Director appliance. See “Deploy the vCloud Application Director Appliance in vSphere,” on page 22.

For vCloud Director, configure the vApp settings to install vCloud Application Director. See “Set Up vCloud Director for vCloud Application Director,” on page 23.

Deploy the vCloud Application Director Appliance in vSphere

To install vCloud Application Director in vSphere, you must deploy an OVF template.

You can deploy an OVF template from any local file system accessible from the vSphere Client machine, or from a remote Web server. The local file systems can include local disks, removable media, and shared network drives.

Prerequisites

■ Verify that you completed all of the prerequisites listed in “Preparing to Install vCloud Application Director,” on page 19 for vSphere.

■ A vCloud Application Director OVF template must be available for deployment. Familiarize yourself with deploying OVF templates in a vSphere Client. See the vSphere documentation.

Procedure

1. Log in to the vSphere client.
2. Select File > Deploy OVF template.
   The Deploy OVF template wizard opens.
4. Map the template to a network in a pool of IP addresses.
5. Select the Power on after deployment check box.
6. Review the deployment settings and click Finish.

The vCloud Application Director virtual machine deploys and powers on.

What to do next

Initialize the vCloud Application Director appliance. See “Start the vCloud Application Director Appliance,” on page 25.
Set Up vCloud Director for vCloud Application Director

You must configure a virtual appliance in vCloud Director to successfully install vCloud Application Director. If you have a previously installed version of vCloud Director, you must check the configuration settings to optimize for vCloud Application Director.

**Prerequisites**

- Configure your vCenter settings for vCloud Application Director. See “Set Up VMware vCenter Cluster Configurations,” on page 21.

- Familiarize yourself with the procedures for creating organizations and catalogs. See the latest vCloud Director documentation.

- Verify that the vCloud Director administrator uploaded templates to a catalog.

  **Note** If the templates are uploaded to a catalog contained in a different vCloud Organization from the organization in which vCloud Application Director performs application deployments, you must publish the catalog and the templates. Check whether the catalog has the correct attributes so that users can access it.

- Verify that you have a direct-connect network with an external pool of IP addresses that vCloud Application Director can use. Contact your vCloud Director administrator to determine the number of IP addresses in the pool.

**Procedure**

- In vCloud Director, allocate a separate organization for vCloud Application Director.

**What to do next**

Verify your vCloud Director setup. See “Verify Your vCloud Director Environment,” on page 23.

To create or update custom virtual machine templates, see “Creating Windows Virtual Machine Templates in vCloud Director,” on page 71 and “Create Linux Virtual Machine Templates in vCloud Director,” on page 75.

**Verify Your vCloud Director Environment**

If you have a previously installed version of vCloud Director, verify that the vApps can communicate with the virtual machine and external network IP addresses.

vCloud Application Director supports deploying virtual machines to directly connected and NAT-routed networks. Use IPPOOL addressing for deployed virtual machines to communicate with the vCloud Application Director server over the network.

**Note** DHCP addressing for directly connected networks is not supported.

**Prerequisites**

- Configure your vCenter settings for vCloud Application Director. See “Set Up VMware vCenter Cluster Configurations,” on page 21.

- Familiarize yourself with the procedures for customizing vCloud Director. See the latest vCloud Director documentation and VMware knowledge base articles kb.vmware.com/kb/2005829 and kb.vmware.com/kb/2034092.

- Verify that you have the VMRC plug-in for vCloud Director installed to work with your browser. For information about compatible Web browsers, see “vCloud Application Director System Requirements,” on page 20.
Verify that the required TCP ports are open for the vCloud Application Director virtual machine. See “vCloud Application Director System Requirements,” on page 20.

**Procedure**

1. From the vCloud Director user interface, create a vApp with one virtual machine.
2. In the vCloud organization that you map to the vCloud Application Director cloud provider, verify that the virtual machines in the vCloud organization have a vCloud network configuration that allows them to connect to the vCloud Application Director appliance.
3. Deploy the vApp.
4. Use the VMRC plug-in to connect to the virtual machines and ensure that they can ping an external IP.
5. Open TCP ports so that the deployed virtual machines can reach the vCloud Application Director appliance.

**What to do next**

Download and deploy the vCloud Application Director appliance. See “Deploy the vCloud Application Director Appliance in vCloud Director,” on page 24.

### Deploy the vCloud Application Director Appliance in vCloud Director

To install vCloud Application Director in vCloud Director, you must download the appliance.

For information about adding vApps, see the vCloud Director documentation.

**Prerequisites**

Verify that you completed all of the prerequisites listed in “Preparing to Install vCloud Application Director,” on page 19 for vCloud Director.

**Procedure**

1. Download the vCloud Application Director appliance and save both files in the same folder without changing their file names.
   
   The appliance consists of the following files:
   
   ApplicationDirector-`VersionNumber`-`*`-_OVF10.ovf
   
   ApplicationDirector-`VersionNumber`-`*`-system.vmdk

2. Replace `*` with the build number of the appliance.
3. Log in to vCloud Director and select the organization vDC in which to deploy vCloud Application Director.
4. Select the Catalogs view and click the **vApp Templates** tab.
5. Click the **Upload** button ( ![Upload](image)

6. In the Upload OVF as a Template window, complete the requested information.
7. Click **OK** in any certificate warning pop-up windows to continue uploading the appliance.

   Because of the large file size and depending on network connection speed, expect the upload process to take a significant amount of time.

8. Right-click the uploaded template and select **Add to My Cloud**.
9. Follow the prompts to add a vApp.
10. Navigate to **My Cloud**.
Right-click the newly added vApp and click **Start**.

**What to do next**

Initialize the vCloud Application Director appliance. See “Start the vCloud Application Director Appliance,” on page 25.

**Start the vCloud Application Director Appliance**

You must start the newly added vCloud Director vApp or vSphere virtual machine to confirm that your vCloud Application Director installation was successful and to complete the remaining set up procedures.

**NOTE** You cannot upgrade an older version of the vCloud Application Director appliance to the latest version.

You can also use the vCloud Director Web Console to access a powered on virtual machine. Verify that your Web browser has a copy of the vmware-vmrc plug-in installed. See *vCloud API Programming* guide.

**Prerequisites**

- Verify that you have the VMRC plug-in for vCloud Director installed to work with your browser. For information about compatible Web browsers, see “vCloud Application Director System Requirements,” on page 20.
- Verify that the vSphere client meets the Web browser requirements. See VMware vSphere documentation.

**Procedure**

1. Open the vCloud Application Director appliance.
   - From the vCloud Director My Cloud view, click the image under the consoles column that corresponds to your vCloud Application Director appliance and click inside the VMRC console.
     It might take a few seconds until the connection to the appliance is established and the vApp console view opens.
   - From the vSphere client, locate the powered-on virtual machine and click the **Console** tab.
2. At the prompt, type the vCloud Application Director serial number and press Enter.
3. Type a password for the root user account and press Enter.
4. At the prompt, retype the password to confirm it and press Enter.
5. Type a password for the darwin_user account and press Enter.
   - Select a password for the darwin_user account that you can remember easily. When you are prompted for this password again, you have only three attempts to type the correct password.
6. At the prompt, retype the password to confirm it and press Enter.
   - **NOTE** It takes less than a minute for the boot scripts to install and start other required software processes in the background, before you are prompted to set the admin user account password. During this time, the system might appear to be unresponsive.
7. Type a password for the vCloud Application Director admin user account and press Enter.
8. At the prompt, retype the password to confirm it and press Enter.

The boot up script should finish starting the necessary services and display the Web URL and DHCP or Static IP address for accessing the vCloud Application Director server
9 Navigate to the Web URL using a supported browser.
   The Web URL format is https://Application_Director_IP:8443/darwin.

10 Log in as the admin user.
   Use the password you set for the admin user account. The admin user can use the CLI to enable the
   preconfigured user accounts and create additional users and groups with specific roles.

The browser opens the vCloud Application Director user interface. See “Using the vCloud Application
Director Web Interface,” on page 42.

**NOTE** Shut down the vCloud Application Director appliance you deployed in vCloud Director, from the
vCloud Director user interface. Do not use vCenter Server to shut down the appliance.

**What to do next**

If the darwin_user is locked due to multiple failed login attempts, you need to unlock the account to
proceed. See “Unlock Your darwin_user Account,” on page 27.

Log in to vCloud Application Director and familiarize yourself with the product features. See “Log In to
vCloud Application Director,” on page 41 and “Using the vCloud Application Director Web Interface,” on
page 42.

Register a cloud provider for your cloud environment. See “Register the vCloud Director Cloud Provider
and Template,” on page 78, “Register the vCloud Automation Center Cloud Provider and Template,” on
page 57, or “Register the Amazon EC2 Cloud Provider and Template,” on page 88.
Troubleshooting Problems Connecting to the vCloud Application Director Web Interface

A few situations can cause connection problems when you attempt to access the vCloud Application Director Web interface.

Table 3-1. Common Connection Errors

<table>
<thead>
<tr>
<th>Error</th>
<th>Possible Cause</th>
<th>Possible Solution</th>
</tr>
</thead>
</table>
| The vCloud Application Director virtual appliance does not have a working IP address or network. | The following problems might cause a networking error:  
  - Organizational vDC network is not configured properly.  
  - Network settings not specified in the Configure Virtual Machines wizard.  
  - IP address is not specified for IP Assignment setting in the Configure Virtual Machines wizard.  
  - Static IP address is not available.  
  - DHCP server is not available. | You must resolve the networking issue.                                                                |
| The vCloud Application Director virtual appliance temporarily fails to retrieve a working IP address during start up, or the IP address changes after start up. | The following problems might cause a networking error:  
  - A problem was encountered with the network.  
  - Static IP address is not available.  
  - DHCP server is not available.  
  - The IP address was explicitly modified. | Run the following command in the appliance virtual machine with root privileges:  
/home/darwin/tools/darwin_util.sh -a AUTO                                                                 |
| The vFabric tc Server service encounters an error in the vCloud Application Director virtual appliance. | The virtual appliance was not shut down properly and restarted.  
The tc Server service was incorrectly started or restarted. | Navigate to the log file at/home/darwin/tcserver/darwin_user/logs/catalina.out for error details and contact VMware technical support if needed. |

Unlock Your darwin_user Account

If you do not provide the correct password after three attempts when you use the SSH client to log in with your darwin_user account, you are locked out of the darwin_user account.

Prerequisites

- Verify that you are logged in to the VMRC console of the vCloud Application Director appliance.
  - SSH is disabled for the root account.
- Verify that you have root privileges.

Procedure

1. In the VMRC console, log in as the root user.
2. Check the number of failed login attempts from the shell prompt.
   ```bash
   faillog -u darwin_user
   ```
3. Unlock the failed account.
   ```bash
   faillog -u darwin_user -r
   ```
Restarting vCloud Application Director

In some cases, you might have to restart vCloud Application Director.

Procedure

1. Log in to your vCloud Application Director virtual machine using the SSH client or vCloud Director console.
2. Restart vCloud Application Director.
   - If you are logged in with the darwin_user account, type `sudo service vmware-darwin-tcserver restart`.
   - If you are logged in with the root account, type `service vmware-darwin-tcserver restart`.

Configure vCloud Application Director to Use a Proxy for External URLs

Even if you use vCloud Application Director only to deploy applications in a private cloud, some deployments might require access to URLs from outside the corporate firewall. For example, an action script might involve downloading application bits from an open-source Web site. You can configure vCloud Application Director to use a proxy for these cases.

You must complete this task before you create services and applications or before you deploy existing predefined sample services and applications.

vCloud Application Director also contains a proxy file called `darwin_global_noproxy.conf`, which does not define a proxy. You can specify this file as the global_conf property value if a deployment environment does not require a proxy. The Amazon EC2 deployment environment does not require a proxy to deploy an application.

The proxy support is now expanded from global level to deployment environment level. For more information, see “Create a vCloud Automation Center Deployment Environment,” on page 59.

Prerequisites

- Verify that you have access to the virtual machine where vCloud Application Director is installed and have the password for logging in to the operating system with the darwin_user account. This password was set during installation. See “Start the vCloud Application Director Appliance,” on page 25.
- Verify that your vCloud Application Director user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Verify that your vCloud Application Director user account has the ROLE_APP_ARCHITECT application architect role assigned to it.
- Familiarize yourself with the procedure for creating new services. See “Add a Service to the Catalog,” on page 105.

Procedure

1. For vCloud Director or vCloud Automation Center, configure the vCloud Application Director virtual appliance to use a proxy.
   a. Log in to the vCloud Application Director virtual appliance as `darwin_user`.
   b. Type `su` to switch to root user.
c Open the file /home/darwin/tcserver/darwin/webapps/darwin/conf/darwin_global.conf with a text editor.

d Update the proxy IP and port information and save the file.

The proxy URL format is http://proxy:PortNumber.

Existing predefined applications or catalog services that must access a repository by using a `yum` update or `install` command use the proxy specified in this file. The predefined services and applications have the script required to access this proxy.

2 Log in to vCloud Application Director as a user with the catalog administrator role and define proxy-specific properties and scripts for a new service or an existing service.

a For a new service, add a service to the catalog.

b In the new or existing service, add a property with the name global_conf to the service, of type Content, and define the value as `https://DarwinServerIP:8443/darwin/conf/darwin_global.conf`.

**CAUTION** If you add a value to the http_proxy, https_proxy, or ftp_proxy property with service scripts that use the `darwin_global.conf` as a file source, when the service script runs, these configured properties override any existing proxy information in the application.

c Add the following lines to the beginning of each action script for the service that requires a proxy:

```
# Import global conf
. $global_conf
```

d Save the service.

3 Log in to vCloud Application Director as a user with the application architect role and define proxy-specific properties and scripts in the application blueprint to configure an application to use a proxy while creating the application blueprint.

a For the application component, add the global_conf property to the application component, of type Content, and define the value as `https://DarwinServerIP:8443/darwin/conf/darwin_global.conf`.

b Add the following lines to the beginning of each action script for the application component that requires a proxy:

```
# Import global conf
. $global_conf
```

**What to do next**

Log in to vCloud Application Director and familiarize yourself with the product features. See “Log In to vCloud Application Director,” on page 41 and “Using the vCloud Application Director Web Interface,” on page 42.

Register a cloud provider for your cloud environment. See “Register the vCloud Director Cloud Provider and Template,” on page 78, “Register the vCloud Automation Center Cloud Provider and Template,” on page 57, or “Register the Amazon EC2 Cloud Provider and Template,” on page 88.
Register vCloud Application Director to vCloud Automation Center

When you register vCloud Application Director, you use the CLI to establish a connection between the vCloud Application Director environment and the vCloud Automation Center server so that you can access the vCloud Automation Center service catalog.

When the connection between vCloud Application Director and vCloud Automation Center is established, you can publish the deployment profile to the vCloud Automation Center catalog so that other users can request the deployment profile and use it.

**NOTE** You can have only one active registration. If you retry to register, you receive an error message.

**Prerequisites**

- Verify that your user account has the **ROLE_SYSTEM_ADMIN** system administrator role assigned to it.
- Familiarize yourself with the roles available for users. See “Predefined Users, Groups, and Roles,” on page 36.
- Verify that you know the password for the darwin_user. This password was set during installation. See “Start the vCloud Application Director Appliance,” on page 25.
- Start the vCloud Application Director CLI. See “Start the CLI Remotely,” on page 36.
- The vCloud Automation Center 6.0 IaaS environment and endpoints must be properly set up and configured. See *vCloud Automation Center Installation and Configuration Guide* and *vCloud Automation Center Virtual Provisioning Guide*.
- You must have SSO admin credentials available.

**Procedure**

1. In the root shell prompt, register vCloud Application Director to the vCloud Automation Center server.

   ```
   register-vcac-server --componentRegistryUrl vCACServerURL --ssoAdministratorUsername UserName --ssoAdministratorPassword Password --tenant TenantName
   ```

   The **componentRegistryUrl** is the URL for the vCloud Automation Center Self Service Catalog. A sample URL for the componentRegistryUrl is, https://vcac-oci-071.eng.test.com.

   The registration might take approximately 30 seconds to display the vCAC server registered successfully status.

2. Synchronize groups from vCloud Automation Center to vCloud Application Director.

   ```
   sync-business-groups
   ```

3. (Optional) View the vCloud Automation Center server information.

   ```
   list-vcac-info
   ```

4. (Optional) If there are any changes in the vCloud Automation Center server information, register vCloud Application Director to the vCloud Automation Center server.

   ```
   register-vcac-server --componentRegistryUrl vCACServerURL --ssoAdministratorUsername UserName --ssoAdministratorPassword Password --tenant TenantName --update
   ```

5. (Optional) Unregister the vCloud Automation Center server from vCloud Application Director to register the appliance to another vCloud Automation Center server.

   ```
   unregister-vcac-server
   ```
What to do next

When the connection is established you can start publishing vCloud Application Director deployment profiles to the vCloud Automation Center catalog. See “Publish a Deployment Profile to the vCloud Automation Center Service Catalog,” on page 157.

Upgrading vCloud Application Director

You cannot upgrade an older version of the vCloud Application Director appliance to the latest version. You must create and deploy a new virtual appliance in either vCloud Director 5.1.2 or 5.5 or vSphere vCenter 5.0, 5.1, or 5.5, and install the latest version of vCloud Application Director. For more information about licensing requirements, see “vCloud Application Director System Requirements,” on page 20.

You can import and export packages created in vCloud Application Director 5.0 and 5.2 to 6.0. You can also import and export packages between different vCloud Application Director 6.0 appliances. See Chapter 17, “Using the CLI Import and Export Functions,” on page 213.
After you install vCloud Application Director, you can create accounts for users and assign users to groups. Several preconfigured user accounts are available, but are disabled by default.

**Note** For vCloud Application Director 6.0, you must use the command-line interface (CLI) to create groups, user accounts, and LDAP configurations. You must also use the CLI to manage users, assign roles, and import existing users and groups from an LDAP directory.

This chapter includes the following topics:

- “Overview of Roles, Users, and Groups,” on page 33
- “Predefined Users, Groups, and Roles,” on page 36
- “Start the CLI Remotely,” on page 36
- “Create Users and Groups with vCloud Application Director CLI,” on page 37
- “Create and Activate an LDAP Configuration,” on page 38
- “Import a SSL Certificate for Secure LDAP Connection,” on page 39

### Overview of Roles, Users, and Groups

You must determine who can use vCloud Application Director and what tasks those users are authorized to perform. You can selectively assign administrative permissions by assigning roles to specific users. You can limit access to specific deployment environments and cloud templates by associating each user with a specific group in vCloud Application Director.

### Roles

You can specify which functions the user can perform in vCloud Application Director by associating a local user, an LDAP user, or an LDAP group with one or more roles. These functions include managing user accounts, managing the catalog, managing the cloud providers and deployment environments, creating applications, and deploying applications.

The built-in admin user has the `ROLE_SYSTEM_ADMIN`, `ROLE_CLOUD_ADMIN`, `ROLE_CATALOG_ADMIN`, `ROLE_APP_ARCHITECT`, `ROLE_DEPLOYER`, and `ROLE_SYSTEM_INTEGRATOR` roles assigned to it. See “Predefined Users, Groups, and Roles,” on page 36.
Local Users
A local user has information, including password, stored in the vCloud Application Director database. A user with the **ROLE_SYSTEM_ADMIN** role can use the vCloud Application Director command-line interface to perform the following tasks:

- Create vCloud Application Director users.
- Specify any combination of roles a user has, depending on the tasks that the user is required to perform.
- Specify the group to which a user belongs.
- Change passwords.
- Enable or disable user accounts.
- Create LDAP configurations.
- Import LDAP users and groups.
- Manage LDAP users and groups.


All of the users of a group can view applications, deployments, cloud templates, and cloud providers that a local user created. Local users belonging to a vCloud Application Director group cannot view applications, deployments, cloud templates, and cloud providers of another group.

**Note** Each local user can be associated with only one vCloud Application Director group. If a local user who belongs to a different vCloud Application Director group needs access to applications, deployments, templates, or cloud providers that belong to other groups, a system administrator must provide such users with multiple accounts to use and make sure that the accounts belong to the specific group that the user intends to access.

LDAP Users
LDAP users are user accounts that are authenticated by a remote LDAP server during login. Minimal information about an LDAP user is stored in the vCloud Application Director database. The following user information is stored:

- Username
- SID information of the user
- vCloud Application Director group with which the user is associated
- vCloud Application Director roles that are assigned to the user

**Note** vCloud Application Director does not save LDAP user password information.

LDAP users can belong to only one vCloud Application Director group. LDAP users that belong to one group cannot view applications, deployments, cloud templates, or cloud providers of any other group.

Only the user account with the **ROLE_SYSTEM_ADMIN** role can perform these actions:

- Import existing LDAP users and groups to vCloud Application Director.
- Assign the imported users and groups with any combination of roles, depending on the tasks that they are required to perform.
During the import process, the vCloud Application Director server communicates with the LDAP server to determine whether the user account exists in the LDAP directory. After the confirmation is received, the user’s SID information is copied from the LDAP directory and an entry is created in the vCloud Application Director database for that user.

No other synchronization takes place because vCloud Application Director does not store user data. Required user data is retrieved when the user logs in to the system.

**vCloud Application Director Groups**

With vCloud Application Director, you can create vCloud Application Director groups and assign local users, LDAP users, and LDAP groups to one vCloud Application Director group.

vCloud Application Director groups consolidate various vCloud Application Director components that belong together. Assigning a user to a group gives the user access to the following consolidated set of vCloud Application Director components:

- Applications, including specific application versions and deployments
- Logical templates, services, policies, and external services
- Cloud providers, including deployment environments

Each local user account, LDAP user account, and LDAP group account is associated with only one vCloud Application Director group. You can associate an LDAP user or LDAP group with a vCloud Application Director group when you import the user or group from the LDAP directory. If multiple LDAP groups are imported that are associated with different vCloud Application Director groups, and a user belongs to multiple LDAP groups, then the user that belongs to multiple LDAP groups is assigned to only the first group association.

Logical templates, applications, cloud providers, and deployment environments are also associated with the group of the user that created them, to allow limited access to components of vCloud Application Director. You can associate a local user with a group when you create the user account.

For example, when a user in abcGroup creates an abcApp application, all of the local users, LDAP users, and LDAP groups in that group can access the application. A local user in xyzGroup can create an application xyzApp for all of the users in xyzGroup to access. A local user in abcGroup cannot access the xyzApp application and the same is true for a local user in xyzGroup. If a local user’s group is changed, any applications created as a member of that group remain in that group. The limited group access also applies to cloud providers, deployment environments, logical templates, and deployments.

The built-in Default group includes all of the predefined sample applications and logical templates. The built-in user accounts, including the admin user, belong to this Default group.

**LDAP Groups**

LDAP groups are imported from the LDAP server and associated with vCloud Application Director roles. An LDAP group can be assigned a set of roles, in the same way that a local user or LDAP user can be assigned roles. Importing an LDAP group allows all the LDAP users in the group to log in to the vCloud Application Director appliance without being individually imported.

When LDAP users log in, their LDAP group credentials are evaluated. The actual roles assigned to the user are an aggregate of all the roles assigned to the imported LDAP user account and the roles for all of the groups to which the user belongs. This accretion of roles depends on whether the LDAP groups are imported to vCloud Application Director and assigned roles.

Each LDAP group is associated with one vCloud Application Director group.
Predefined Users, Groups, and Roles

vCloud Application Director provides five predefined user accounts with roles that map to specific privileges within product areas.

<table>
<thead>
<tr>
<th>User</th>
<th>Roles Assigned</th>
<th>Product Area Managed</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin</td>
<td>ROLE_SYSTEM_ADMIN, ROLE_CLOUD_ADMIN, ROLE_CATALOG_ADMIN, ROLE_APP_ARCHITECT, ROLE_DEPLOYER, ROLE_SYSTEM_INTEGRATOR</td>
<td>Users, groups, and roles</td>
<td>System administrator manages users and groups.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>All other areas</td>
<td>Because this account includes all roles, you can also perform the actions listed for the other users.</td>
</tr>
<tr>
<td>cloudAdmin</td>
<td>ROLE_CLOUD_ADMIN</td>
<td>Cloud providers and deployment environments</td>
<td>Manage cloud providers and deployment environments.</td>
</tr>
<tr>
<td>catalogAdmin</td>
<td>ROLE_CATALOG_ADMIN</td>
<td>Catalog</td>
<td>Maintain and add services, templates, and tasks to the catalog.</td>
</tr>
<tr>
<td>appArchitect</td>
<td>ROLE_APP_ARCHITECT</td>
<td>Applications</td>
<td>Design and create applications.</td>
</tr>
<tr>
<td>deployer</td>
<td>ROLE_DEPLOYER</td>
<td>Deployments</td>
<td>Initiate deployments, update an existing deployment, and tear down a deployed application from the cloud.</td>
</tr>
</tbody>
</table>

All of the predefined user accounts except the admin user are disabled by default. You must enable these built-in user accounts and set the passwords. See “Create Users and Groups with vCloud Application Director CLI,” on page 37.

The password for the admin user is the admin password that was set the first time the appliance was started.

Start the CLI Remotely

You can start the vCloud Application Director CLI from a remote machine.

As a best practice, run the CLI remotely to reduce server load and avoid shared CLI on the vCloud Application Director appliance. The connection from the remote machine to the CLI is secured.

Prerequisites

- Verify that you know the password for the vCloud Application Director appliance.
- Verify that you installed Java (JDK 1.7) on your remote machine.
- Make sure that the remote machine can connect to the vCloud Application Director appliance using HTTPS.

Procedure

1. Download the darwin-cli.jar file from the vCloud Application Director server
   http://DarwinServerIP/tools/darwin-cli.jar to a folder with write permissions on the remote machine.
   The CLI creates a log file during the session.
2. Open a command prompt and start the client.
   java -jar /PathToJarFolder/darwin-cli.jar
   The Darwin CLI banner appears and the appd> CLI prompt appears.
3 Log in to the vCloud Application Director server.
   
   ```bash
   login --serverUrl https://DarwinServerIP:8443/darwin --username UserName
   ```

   Replace DarwinServerIP with the vCloud Application Director server IP address.

   If you run the `--password` parameter with the login command or a command that lets you add a password, your password is saved as plain text in the darwin-cli-history.log file located in the current directory. By default, the CLI deletes the log file. For added security, delete this log file.

4 Type the password when you are prompted.

   The system returns a message similar to this:

   ```
   You are logged in to https://DarwinServerIP:8443/darwin as UserName.
   ```

**What to do next**

Determine the type of users and groups to create in vCloud Application Director and use the CLI options to enable and manage them. See “Create Users and Groups with vCloud Application Director CLI,” on page 37 and “Managing Users and Groups,” on page 204.

---

**Create Users and Groups with vCloud Application Director CLI**

You must use the vCloud Application Director CLI to create users and groups.

**Prerequisites**

- Verify that your user account has the `ROLE_SYSTEM_ADMIN` system administrator role assigned to it.
- Familiarize yourself with the roles available for users. See “Predefined Users, Groups, and Roles,” on page 36.
- Verify that you know the password for the darwin_user. This password was set during installation. See “Start the vCloud Application Director Appliance,” on page 25.
- Start the vCloud Application Director CLI. See “Start the CLI Remotely,” on page 36.

**Procedure**

1 In the root shell prompt, press the Tab key to display the list of available commands.
2 Type the command to perform the appropriate action.

<table>
<thead>
<tr>
<th>Option</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>List usage information</td>
<td><code>help</code></td>
</tr>
<tr>
<td>Create a group</td>
<td><code>create-group --name GroupName --description Description</code></td>
</tr>
<tr>
<td>Create a user</td>
<td><code>create-user --username UserName --firstName FirstName --lastName LastName --enabled true --roles ROLE_SYSTEM_ADMIN, ROLE_CATALOG_ADMIN, ROLE_CLOUD_ADMIN, ROLE_DEPLOYER, ROLE_APP_ARCHITECT, ROLE_SYSTEM_INTEGRATOR --group GroupName --email EmailAddress</code></td>
</tr>
</tbody>
</table>

If you run the `--password` parameter with the login command or a command that lets you add a password, your password is saved as plain text in the darwin-cli-history.log file located in the current directory. By default, the CLI deletes the log file. For added security, delete this log file.

If you specify all the roles shown in this example, the user is the equivalent of the admin user. Specify only the roles the specific user must have.

The valid values for the `enabled` option are true, yes, 1, false, no, and 0.

For a vCloud Application Director object or entity, if you use multiple words with spaces between the words, enclose the words in quotation marks.
For example, to create a group called Test Group, use the command `create-group --name "Test Group"`.

Often after you enter a command, the system displays many details relating to the user, in addition to indicating whether the command was successful.

**What to do next**

Use the CLI to perform tasks such as changing passwords or enabling and disabling accounts for users and groups. See “Managing Users and Groups,” on page 204.

### Create and Activate an LDAP Configuration

LDAP is a central authentication mechanism that lets you use a login credential to access multiple servers and groups with which you are associated.

If an organization already uses LDAP to handle its authentication and directory services, vCloud Application Director server can integrate with the existing LDAP authentication mechanism. You can use the CLI to create and manage LDAP configurations in vCloud Application Director.

**NOTE** You can have multiple LDAP servers defined in a vCloud Application Director server, but you can have only one LDAP configuration active at any given time for authentication.

#### Prerequisites

- Verify that your user account has the `ROLE_SYSTEM_ADMIN` system administrator role assigned to it.
- Verify that you know the password for the darwin_user. This password was set during installation. See “Start the vCloud Application Director Appliance,” on page 25.
- Start the vCloud Application Director CLI. See “Start the CLI Remotely,” on page 36.

#### Procedure

1. In the root shell, follow the prompts to create an LDAP configuration.
   ```
   create-ldap-config
   ```
   vCloud Application Director creates the LDAP configuration and saves the configuration in its database.

2. (Optional) Check the existence of a user name in the LDAP directory to confirm server connectivity.
   ```
   test-named-ldap-config --configname LDAPConfigName --name LDAPUserName
   ```

3. (Optional) Activate an LDAP configuration if it was not activated during the initial creation or to activate another LDAP configuration.
   ```
   activate-ldap-config --configname LDAPConfigName
   ```

Activating an LDAP configuration in the vCloud Application Director server allows authentication against the named LDAP configuration.

Even after the LDAP configuration is activated, vCloud Application Director always searches the local database first for valid users before performing an SSO or LDAP authentication. This verification action ensures that local users are not locked out if the same user name exists in the local database, SSO group, and the LDAP server. If users have the same credentials in the SSO group as the LDAP server, the SSO authentication takes precedence. If multiple user names exist on the local database, the first entry that was added is used for authentication.
What to do next

To connect to your LDAP server over a secure channel when your LDAP certificate is signed by a local authority, or self-signed, import the LDAP certificate to the vCloud Application Director server openssl trusted list. See “Import a SSL Certificate for Secure LDAP Connection,” on page 39.

Use the CLI to perform tasks such as importing users and groups and updating existing LDAP configurations. See “Managing LDAP Configurations,” on page 206.

Import a SSL Certificate for Secure LDAP Connection

vCloud Application Director server supports most public issuers of SSL certification by using the JDK cacerts file to securely connect to the LDAP server. If the LDAP server certificate is signed by a local authority, or self-signed, you must import the LDAP certificate to the vCloud Application Director server openssl trusted list.

Prerequisites

- Verify that your user account has the ROLE_SYSTEM_ADMIN system administrator role assigned to it.
- Verify that you know the password for the darwin_user. This password was set during installation. See “Start the vCloud Application Director Appliance,” on page 25.
- Copy the LDAP server certificate file as the certificate.pem file to a directory on the vCloud Application Director server.

Procedure

1. Use the SSH client to log in to the vCloud Application Director appliance as the user darwin_user. The password for this account was set during installation.
2. Open a command prompt.
3. Switch user from darwin_user to darwin.
   
   su darwin
4. Change to the keystore directory.
   
   cd /home/darwin/keystore
5. Copy the LDAP certificate.pem file to the directory.
   
   cp Directory_certificate_copied/certificate.pem /home/darwin/keystore/
6. Add the certificate file to the vCloud Application Director server trusted list.
   
   keytool -import -trustcacerts -alias UniqueAlias -file certificate.pem -keystore ./appd.truststore -storepass ""
7. Restart the vCloud Application Director server.
   
   sudo /sbin/service vmware-darwin-tcserver restart

What to do next

Use the CLI to perform tasks such as importing users and groups and updating existing LDAP configurations. See “Managing LDAP Configurations,” on page 206.
You can use the vCloud Application Director Web interface to register cloud providers, maintain the catalog of virtual machine templates and services, create applications, deploy applications, publish a deployment profile to the vCloud Automation Center service catalog, update deployed applications, and tear down deployed applications from the cloud.

**NOTE** You must use the CLI to register vCloud Automation Center server, create groups and user accounts, manage users, assign roles, manage LDAP configurations, and import and export packages between different instances.

You can also use the CLI to deploy applications, update deployed applications, and tear down deployed applications. See “General CLI Options,” on page 203.

This chapter includes the following topics:

- “Log In to vCloud Application Director,” on page 41
- “Using the vCloud Application Director Web Interface,” on page 42

**Log In to vCloud Application Director**

To see the sample applications included with vCloud Application Director, you must log in using an account that belongs to the Default group.

Whether you can perform a particular task after you log in depends on the roles assigned to the user account. The objects you can see depend on the group associated with the user. See Chapter 4, “Setting Up Users and Groups,” on page 33.

**Prerequisites**

- Verify that you have the URL for the vCloud Application Director Web interface. This Web URL appears in the console of the virtual machine that hosts vCloud Application Director when installation is finished. See “Start the vCloud Application Director Appliance,” on page 25.
- Verify that you have credentials for a user account that was set up in vCloud Application Director. You can also use the admin user account and the password that was set during installation. The admin user has all roles associated with it and can perform all of the functions in vCloud Application Director.
- Verify that the supported version of a Web browser is installed on your computer. See “vCloud Application Director System Requirements,” on page 20.

**Procedure**

1. Open a Web browser and type the vCloud Application Director URL.
   
   An example of a URL for accessing the Web interface is https://192.0.2.255:8443/darwin.
2. Type the credentials of a user account with the necessary roles assigned to it.

You can access the product areas that are assigned to your role. For example, if your user account has the deployer role assigned to it, when you log in to the vCloud Application Director Web interface, the Deployments page opens. If a product area is not assigned to your role then the user interface is visible but disabled.

What to do next

Familiarize yourself with the vCloud Application Director features. See “Using the vCloud Application Director Web Interface,” on page 42.

Using the vCloud Application Director Web Interface

With the vCloud Application Director Web interface, you can access almost all of the available functions.

The Web user interface includes the standard features of a Web application. For example, with the search box, you can select filtering criteria that are related to the objects you are searching for. If your search string includes an underscore, vCloud Application Director ignores the underscore as a wildcard. For example, if you have an application called test_qe and another application called testapp, when you type the search string test_ in the search box, both of the applications appear in the search result.

You can access applications, deployments, and the cloud environment depending on the group associated with your user account when you log in to the vCloud Application Director appliance. All of the users can see the sample applications, services, tasks, logical templates, external services, operating systems, and tags included in the vCloud Application Director catalog. If you belong to multiple groups, you can click the down arrow next to the group name and view all of the groups in the drop-down list. To change your group, click the group name in the drop-down list.

Figure 5-1. Available Groups in the vCloud Application Director Appliance

The vCloud Application Director title bar includes a drop-down menu that includes the main categories Applications, Deployments, Clouds, and Catalog. In addition, the drop-down menu displays the subcategories under Clouds and Catalog. From the vCloud Application Director title bar, click the Applications drop-down menu and select another category to navigate from one page to another.
Figure 5-2. Title Bar Menu Options

The Applications page organizes all of the applications. From this page, you can configure the sample or custom applications to deploy to a cloud environment. The Deployments page lists both successful and failed deployments. From this page, you can initiate an update process to scale clustered nodes, modify the configuration, and tear down or delete a deployed application. The Clouds category lets you create cloud providers and deployment environments. The Catalog category lets you create services, external services, policies, logical templates, tasks, operating systems, and tags for your application. You can also use the sample services and other components in the catalog to model an application blueprint.

Click Logout in the vCloud Application Director title bar to log out from the current session.

Using the vCloud Application Director Buttons

The vCloud Application Director interface includes buttons that allow you to perform specific or multiple tasks.

Table 5-1. vCloud Application Director Buttons

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="icon" alt="Promote Update Profile" /></td>
<td>Promote Update Profile</td>
<td>Available on an updated deployment. Indicates that the update profile is promoted to another deployment environment.</td>
</tr>
<tr>
<td><img src="icon" alt="Add Relation" /></td>
<td>Add Relation</td>
<td>Available in the blueprint. Creates relations between components in a blueprint.</td>
</tr>
<tr>
<td><img src="icon" alt="Convert to Node Array" /></td>
<td>Convert to Node Array</td>
<td>Available in the blueprint. Creates a clustered node. You can also use the button to convert a clustered node to a single node.</td>
</tr>
<tr>
<td><img src="icon" alt="Add" /></td>
<td>Add</td>
<td>Available in the blueprint. Adds a NIC to a node from the NICs tab or to add application component properties from the Properties tab.</td>
</tr>
<tr>
<td><img src="icon" alt="Reset" /></td>
<td>Reset</td>
<td>Available on the Deployment Profile wizard. Reverts to the original value. Also available on the Update Deployment Profile wizard, service version, logical template version, and task version pages.</td>
</tr>
<tr>
<td><img src="icon" alt="View Task Information" /></td>
<td>View Task Information</td>
<td>Available in execution plans. View action script text, properties, and logs.</td>
</tr>
<tr>
<td><img src="icon" alt="Expand" /></td>
<td>Expand</td>
<td>Available in execution plans. Displays all virtual machines in a cluster node.</td>
</tr>
</tbody>
</table>
**Table 5-1. vCloud Application Director Buttons (Continued)**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>Collapse</td>
<td>Available in execution plans. Minimizes all virtual machines in a cluster node.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Add Script Task</td>
<td>Available in execution plans. Adds custom tasks to an application deployment.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Notification</td>
<td>Appears when a task is incomplete or a user performs a task that is not allowed.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Expand</td>
<td>Maximizes a status window in the deployment summary page.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Minimize</td>
<td>Collapses a status window in the deployment summary page.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Expand</td>
<td>Maximizes the provisioning tasks in the execution plan status window of the deployment summary page.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>View Logs</td>
<td>Available on a deployment details page or an execution plan after deployment.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Copy Application Version</td>
<td>Creates a new version of an existing application.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Quick Deploy</td>
<td>Available on the Applications page and application version card. Deploys an application without configuring any of the elements in the Deployment Profile wizard.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Update Deployment</td>
<td>Initiates an update process to scale a clustered node or modify configuration of a deployed application.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Teardown Deployment</td>
<td>Removes a deployed application from a cloud environment.</td>
</tr>
</tbody>
</table>

**Understanding the vCloud Application Director Icons**

The vCloud Application Director Web interface includes icons that represent the availability of a function.

**Table 5-2. vCloud Application Director Icons**

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Icon]</td>
<td>Group name</td>
<td>Appears on the Applications, Services, External Services, Logical Templates, and Deployments pages next to the name of the group that owns the component.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Shared Publicly</td>
<td>Appears on the Applications, Services, External Services, Logical Templates, and Deployments pages for components that are shared publicly.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Shared Privately</td>
<td>Appears on the Applications, Services, External Services, Logical Templates, and Deployments pages for components that are private to the group.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Policy Scan Success</td>
<td>Appears on the Deployments page for a deployment or deployed application with a successful policy scan.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Policy Scan Fail</td>
<td>Appears on the Deployments page for a deployment or deployed application with a failed policy scan.</td>
</tr>
<tr>
<td>![Icon]</td>
<td>Auto-Bind Consume</td>
<td>Appears in the blueprint canvas, next to properties that can Auto-Bind with another property, which is set to Expose.</td>
</tr>
</tbody>
</table>
Table 5-2. vCloud Application Director Icons (Continued)

<table>
<thead>
<tr>
<th>Icon</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Auto-Bind Expose</td>
<td>Appears in the blueprint canvas, next to properties that can be used to customize another property, which is set to Consume.</td>
</tr>
<tr>
<td><img src="image" alt="Icon" /></td>
<td>Extra Configuration</td>
<td>Appears in the Deployment Profile wizard and the VM details status window of the deployment summary page for a vCloud Automation Center deployment. In the Deployment Profile wizard, you can use Extra Configuration to override the vCloud Automation Center blueprint custom properties or add to the existing properties.</td>
</tr>
</tbody>
</table>
Setting Up Application Provisioning for the vCloud Automation Center Environment

vCloud Application Director uses the virtual cloud infrastructure features in vCloud Automation Center to deploy applications to the VMware vSphere environment.

As part of the setup process, you must create custom vCloud Automation Center blueprints. A vCloud Automation Center blueprint includes specifications for provisioning virtual, cloud, or physical machines, that determine the machine attributes, how they are provisioned, and their policy and management settings. When you request a machine, you must select the blueprint from which the machine is created. vCloud Automation Center applies a set of custom properties defined in the build profile or in the blueprint to that requested machine. From the vCloud Automation Center interface, you can access the clone or linked clone blueprints and specify properties to override several properties in the vCenter Server template.

**IMPORTANT** vCloud Application Director supports only the vCloud Automation Center virtual blueprint provisioning to vSphere.

In vCloud Application Director, you must register a cloud provider and template. When you register a cloud provider, you map a specific vCloud Automation Center blueprint in a provisioning or business group to a cloud template in vCloud Application Director. This registration process makes this vCloud Automation Center blueprint available in the vCloud Application Director catalog. To deploy an application using this cloud template, you must map a cloud template to a logical template. You must then create a deployment environment and map this deployment environment to an applicable vCloud Automation Center reservation policy.

Familiarize yourself with the key concepts that relate to setting up and configuring a cloud environment for application provisioning. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates,” on page 48
- “Creating Virtual Machine Templates in vCloud Automation Center,” on page 49
- “Updating Existing Virtual Machine Templates in vCloud Automation Center,” on page 57
- “Register the vCloud Automation Center Cloud Provider and Template,” on page 57
- “Create a vCloud Automation Center Deployment Environment,” on page 59
Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates

To create custom virtual machine templates to use in vCloud Application Director, verify that certain software and virtualization requirements are met and that certain Linux and Windows commands are available.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>Use one of the following operating systems:</td>
</tr>
<tr>
<td></td>
<td>- Supported operating systems with 32-bit Red Hat Enterprise Linux 6.4 and CentOS 6.3</td>
</tr>
<tr>
<td></td>
<td>- Supported operating systems with 64-bit Red Hat Enterprise Linux 6.4, CentOS 6.3, and Windows Server 2008 R2 Enterprise with Service Pack 1</td>
</tr>
<tr>
<td>JRE</td>
<td>JRE 1.7.0 must be installed. The preferred and supported JRE can be installed from one of these packages, which are available in the vCloud Application Director virtual appliance:</td>
</tr>
<tr>
<td></td>
<td>- JRE for Windows</td>
</tr>
<tr>
<td></td>
<td>- JRE for Linux</td>
</tr>
<tr>
<td></td>
<td>http://Application_Director_IP/agent/jre-1.7.0_45-lin32.zip</td>
</tr>
<tr>
<td></td>
<td>http://Application_Director_IP/agent/jre-1.7.0_45-lin64.zip. To install JRE on an RPM-based virtual machine template, see “Prepare vCenter Server Linux Virtual Machine Templates,” on page 52.</td>
</tr>
<tr>
<td>Linux agent bootstrap service</td>
<td>Download the Linux agent bootstrap package vmware-appdirector-agent-service-vcac from the VMware product download site at <a href="http://vmware.com/web/vmware/downloads">http://vmware.com/web/vmware/downloads</a>. Install the Linux agent bootstrap script from one of the following packages:</td>
</tr>
<tr>
<td></td>
<td>- http://Application_Director_IP/agent/vmware-appdirector-agent-service-vcac_6.0.0.0-0_x86_64.rpm</td>
</tr>
<tr>
<td></td>
<td>- http://Application_Director_IP/agent/vmware-appdirector-agent-service-vcac_6.0.0.0-0_i386.rpm</td>
</tr>
<tr>
<td></td>
<td>To install the agent bootstrap service on an RPM-based virtual machine template, see “Prepare vCenter Server Linux Virtual Machine Templates,” on page 52.</td>
</tr>
<tr>
<td>Windows agent bootstrap service</td>
<td>Download the http://Application_Director_IP/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip file from the vCloud Application Director virtual appliance.</td>
</tr>
<tr>
<td></td>
<td>To install the agent bootstrap service on a Windows-based virtual machine template, see “Prepare vCenter Server Windows Virtual Machine Templates,” on page 50.</td>
</tr>
<tr>
<td></td>
<td>To install the Windows and Linux guest agent in a template, see “Prepare vCenter Server Windows Virtual Machine Templates,” on page 50 and “Prepare vCenter Server Linux Virtual Machine Templates,” on page 52.</td>
</tr>
<tr>
<td>Supported Windows scripting</td>
<td>vCloud Application Director supports scripting with Windows CMD, PowerShell 2.0, and BeanShell 1.3.0.</td>
</tr>
<tr>
<td>Supported Linux scripting</td>
<td>vCloud Application Director supports scripting with Bash and BeanShell 1.3.0.</td>
</tr>
</tbody>
</table>
Table 6-1. Virtual Machine Requirements for Custom Templates (Continued)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux commands</td>
<td>The following Linux commands must be available on the virtual machine:</td>
</tr>
<tr>
<td></td>
<td>■ wget</td>
</tr>
<tr>
<td></td>
<td>■ md5sum</td>
</tr>
<tr>
<td></td>
<td>■ grep</td>
</tr>
<tr>
<td></td>
<td>■ sed</td>
</tr>
<tr>
<td></td>
<td>■ setsid</td>
</tr>
<tr>
<td></td>
<td>■ awk</td>
</tr>
<tr>
<td></td>
<td>■ ifconfig</td>
</tr>
<tr>
<td></td>
<td>■ apt-get</td>
</tr>
<tr>
<td></td>
<td>■ yum</td>
</tr>
</tbody>
</table>

| Optional services | If you plan to remotely access the virtual machine using Linux `ssh` logging or Windows remote desktop for troubleshooting or for other reasons, the OpenSSH server and client for Linux or Remote Desktop Services (RDS) for Windows must be installed and running properly. |

**IMPORTANT** Because the boot process must not be interrupted, configure the virtual machine so that nothing causes the virtual machine’s boot process to pause before reaching the final operating system login prompt. For example, verify that no processes or scripts prompt for user interaction when the virtual machine starts. This requirement applies only to virtual machine templates created for the vCloud Application Director catalog.

**Creating Virtual Machine Templates in vCloud Automation Center**

To start a clone deployment in vCenter Server with a vCloud Automation Center blueprint, you must create a vCenter Server virtual machine template.

A clone deployment creates a complete and independent virtual machine based on the vCenter Server virtual machine template. A linked clone deployment references a snapshot of a virtual machine. The deployment creates a complete and independent virtual machine based on the vCenter Server snapshot of a virtual machine. You can set up a linked clone deployment from a clone deployment. See [vCloud Automation Center What’s New Guide](#).

- **Prepare vCenter Server Windows Virtual Machine Templates** on page 50
  You must prepare vCenter Server Windows virtual machine templates for vCloud Automation Center blueprints to create clone deployments. vCloud Application Director uses the vCloud Automation Center blueprints to deploy Windows-based applications and services to the VMware vSphere environment.

- **Prepare vCenter Server Linux Virtual Machine Templates** on page 52
  You must prepare vCenter Server Linux virtual machine templates for vCloud Automation Center blueprints to create clone deployments. vCloud Application Director uses the vCloud Automation Center blueprints to deploy Linux-based applications and services to the VMware vSphere environment.

- **Create vCloud Automation Center Blueprints for Clone Deployment** on page 54
  When you create a custom vCloud Automation Center blueprint, you can allow a clone deployment to provision vCloud Application Director Windows-based or Linux-based applications and services to the VMware vSphere environment.
Create and Configure vCloud Automation Center Blueprint for Linked Clone Deployment on page 55

When you create a custom vCloud Automation Center blueprint you can allow a linked clone deployment to provision vCloud Application Director Windows-based or Linux-based applications and services to the VMware vSphere environment. A linked clone is a copy of a virtual machine based on a snapshot. It occupies a small disk space, quick to provision, ideal when performance is not a high priority.

Prepare vCenter Server Windows Virtual Machine Templates

You must prepare vCenter Server Windows virtual machine templates for vCloud Automation Center blueprints to create clone deployments. vCloud Application Director uses the vCloud Automation Center blueprints to deploy Windows-based applications and services to the VMware vSphere environment.

Prerequisites

- Verify that the vCenter Server 5 or 5.1 is installed and configured.
- Verify that a vSphere Windows virtual machine is available. See “Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates,” on page 48.
  The supported operating system is Windows Server 2008 R2 Enterprise with Service Pack 1.
- Verify that the vSphere Windows virtual machine does not have an older version of either a vCloud Automation Center 5.2 or 6.0 guest agent or vCloud Application Director agent bootstrap installed. See vSphere documentation.
- Familiarize yourself with how to convert to a template and create a customization specification for an operating system. See vSphere documentation.
- Verify that all of the network configuration artifacts are removed from the network configuration files.

Procedure

1. Log in to the vSphere client and open a Windows virtual machine.
2. Type the Windows Administrator credentials and open a command prompt.
3. Download and install the supported Java SE 7 Runtime Environment from http://Application_Director_IP/agent/jre-1.7.0_45-win64.zip.

   a. Create a \opt\vmware-jre folder and unzip the JRE file to the folder.
   b. Open a PowerShell command window and type type \opt\vmware-jre\bin\java --version to verify the installation.

   The installed version of Java appears.
4. To install the vCloud Automation Center 6.0 guest agent.

   a. Change directory to C:\.
   b. Download the vCAC-vCAC_Version-Installation ZIP file from the VMware product download site.
   c. Unzip the installation file to C:\ and open the VRMGuestAgent folder.
   a Unzip the vmware-appdirector-agent-bootstrap-windows_VersionNumber.zip file to the \temp folder.
   b Right-click to view the \tmp\vmware-appdirector-agent-bootstrap-windows_VersionNumber\agent_bootstrap.ps1 file properties and click Unblock to disable the security on the file.
   
   **IMPORTANT** If you do not disable this Windows security feature, you cannot use the vCloud Application Director agent bootstrap file.
   
   c Run the rktools.exe utility and add the NTRights.exe utility to the \temp folder.
   
   The NTRights.exe utility is used in the install.bat script to configure the agent bootstrap service to run in the darwin user account. The utility is included in the Windows Server 2003 Resource Kit Tools (rktools.exe), which you can download from the Microsoft download Web site.
   
   d (Optional) If the NTRights.exe utility is not available, manually configure the agent bootstrap service to run in a specific user account after you run the install.bat script.

6 To install the agent bootstrap service for vCloud Automation Center 6.0, complete the following steps.
   a Open a Windows CMD console and navigate to the \temp folder.
   b Type the command to install the agent bootstrap.

   ```bash
   install.bat password=Password cloudProvider=vcac vcacServer=IaaS_Server_FQDN httpsMode=true/false
   ```
   
   The install.bat script creates a user account called darwin for the agent bootstrap service and uses the password you set. The Password must meet the Windows password requirements.

   7 Check the \opt\vmware-appdirector\agent-bootstrapp\agent_reset.bat file output log to verify that the user darwin_user is created.

   If the user is not available, verify that the Windows server password is accurate.

   8 Shut down the Windows virtual machine.

   9 Create a customization specification for the Windows virtual machine.

   A customization specification lets you change Windows operating system properties such as the host name, network settings, and license settings. Customizing guest operating systems can help prevent problems that can result if virtual machines with identical settings are deployed with duplicate host names.
   a Select **Home > Customization Specifications Manager**.
   b Select **New** to create a customization specification.
   c In the vSphere Client Windows Guest Customization wizard, specify the customization specification name.
   d In the NetBIOS section select **Use the virtual machine name** and in the Administrator Password section, type the administrator password of the Windows operating system.
   e Accept the default settings and select **Finish** to save your changes.

   A Windows-based vCloud Automation Center blueprint uses this customization specification information to create a clone or linked deployment.

   10 In the inventory, right-click the Windows virtual machine and select **Template > Convert to Template**. vCenter Server marks the virtual machine as a template and displays the task in the Recent Tasks pane.
What to do next
Create a vCloud Automation Center blueprint for clone or linked clone deployment. See “Create vCloud Automation Center Blueprints for Clone Deployment,” on page 54 or “Create and Configure vCloud Automation Center Blueprint for Linked Clone Deployment,” on page 55.

Prepare vCenter Server Linux Virtual Machine Templates
You must prepare vCenter Server Linux virtual machine templates for vCloud Automation Center blueprints to create clone deployments. vCloud Application Director uses the vCloud Automation Center blueprints to deploy Linux-based applications and services to the VMware vSphere environment.

Prerequisites
- Verify that VMware vCenter Server 5.0, 5.1, or 5.5 is installed and configured.
- Verify that a supported vSphere Linux virtual machine is available. See “Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates,” on page 48.
- Familiarize yourself with how to convert to a template and create a customization specification for an operating system. See vSphere documentation.
- Verify that the vCenter Server virtual machine meets the vCloud Application Director virtual machine requirements. See “Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates,” on page 48.
- Verify that all of the network configuration artifacts are removed from the network configuration files.

Procedure
1. Log in to the vSphere client and open a Linux virtual machine.
2. Log in the virtual machine with root privileges and open a terminal.
3. If you have an existing vCloud Application Director agent bootstrap service, uninstall the service.
   a. Check whether the agent_bootstrap.sh service is running.
      ```bash
      ps -ef | grep agent_bootstrap
      ```
   b. Stop the process
   c. Run the Shell file to remove the agent bootstrap service.
      ```bash
      /opt/vmware-appdirector/agent-bootstrap/agent_reset.sh
      ```
   d. Uninstall the agent bootstrap service.
      ```bash
      rpm --e vmware-appdirector-agent-service-vcac
      ```
4. Download and install the supported JRE packages with the YUM package management tool or APT-GET packaging tool.
   If you have the supported JRE packages installed, skip this step.
   a. Download the JRE package from http://Application_Director_IP/agent/jre-1.7.0_45-lin-ArchitectureName.zip, where the ArchitectureName is 32 or 64.
   b. Create an /opt/vmware-jre directory.
   c. Unzip the JRE package.
      ```bash
      unzip -d /opt/vmware-jre
      ```
   d. Type /opt/vmware-jre/bin/java -version to verify the installation.
      The installed version of Java appears.
5 Download and install the vCloud Automation Center guest agent.
   a From the VMware product download site http://vmware.com/web/vmware/downloads, search for vCloud Automation Center.
   b In the vCloud Automation Center download page, select the supported product version and click View Download.
   c Download and unzip the vCAC-vcAC_Version-Installation.zip file.
   d In the terminal, type `rpm -i gugent-6.0.0-2026.ArchitectureName.rpm` to install the guest agent, where the ArchitectureName is i386 for 32-bit and x86_64 for 64-bit.

6 Download and install the vCloud Application Director agent bootstrap service.
   a Download the agent bootstrap from http://Application_Director_IP/agent/vmware-appdirector-agent-service-vcac_6.0.0.0-0_ArchitectureName.rpm to the /tmp/ folder, where the ArchitectureName is i386 for 32-bit and x86_64 for 64-bit.
   b In the terminal, type `rpm -i vmware-appdirector-agent-service-vcac_6.0.0.0-0_ArchitectureName.rpm` to install the service.

7 Register the vCloud Automation Center guest agent to the vCloud Automation Center server.
   a For a vCloud Automation Center server that requires an SSL connection, type this command.
      ```
      /opt/vmware-appdirector/agent-bootstrap/vcac-register.sh -r vCAC_Port -s IaaS_Server_FQDN
      ```
      The default vCloud Automation Center port number is 443.
   b For a vCloud Automation Center server that does not require an SSL connection, type this command.
      ```
      /opt/vmware-appdirector/agent-bootstrap/vcac-register.sh -r vCAC_Port -n -s vCloud_Automation_Center_Server_Name
      ```
      The default vCloud Automation Center port number is 80.

8 For Red Hat Enterprise Linux 6.4-based virtual machines, verify that the vrm-agent service is available.
   ```
   chkconfig --list | grep vrm-agent
   ```
   The service vrm-agent 0:off 1:off 2:off 3:on 4:off 5:on 6:off appears.

9 For CentOS 6.3-based virtual machines, verify that the demidecode package is installed.
   ```
   rpm -qa | grep dmidecode
   ```
10 (Optional) Install the demidecode package if it is not available.
    ```
    rpm -i dmidecode-2.11-2.e16.ArchitectureName.rpm
    ```
    The ArchitectureName is i686 for 32-bit and x86_64 for 64-bit.

11 Verify that the vmware-appdirector-agent-service-vcac service is available in the Linux virtual machine.
   ```
   rpm -qa | grep vmware-appdirector-agent-service-vcac
   ```
12 Open the cd /usr/share/gugent folder and run the verification command.
   ```
   ./rungugent
   ```
   The verification passes without any error messages.

13 (Optional) Run the agent_reset.sh Shell file to remove any runtime log files generated if you manually started the vmware_appdirector_agent or vrm-agent services for verification.

14 Shut down the Linux virtual machine.
In the inventory, right-click the Linux virtual machine and select **Template > Convert to Template**.

vCenter Server marks that virtual machine as a template and displays the task in the Recent Tasks pane.

**What to do next**

Create a vCloud Automation Center blueprint for clone or linked clone deployment. See “Create vCloud Automation Center Blueprints for Clone Deployment,” on page 54 or “Create and Configure vCloud Automation Center Blueprint for Linked Clone Deployment,” on page 55.

**Create vCloud Automation Center Blueprints for Clone Deployment**

When you create a custom vCloud Automation Center blueprint, you can allow a clone deployment to provision vCloud Application Director Windows-based or Linux-based applications and services to the VMware vSphere environment.

**Prerequisites**

- Verify that vCloud Automation Center 5.2 or 6.0 is installed and configured.
- Verify that the vCloud Automation Center provisioning group that you plan to use with vCloud Application Director is created and configured with Enterprise administrator privileges.
- Verify that a vCenter Server virtual machine is available for the vCloud Automation Center blueprint to create a clone deployments.

You must have the customization specification name available for a Windows-based vCloud Automation Center blueprint. See “Prepare vCenter Server Windows Virtual Machine Templates,” on page 50.

- Familiarize yourself with how to create a blueprint in vCloud Automation Center. See *vCloud Automation Center Operating guide*.

**Procedure**

1. Log in to the vCloud Automation Center server.
2. Select **New Blueprint > Virtual > vSphere (vCenter)**.
3. Complete the information on the **Blueprint Information** tab.
4. On the **Build Information** tab, select the **Clone** option from the **Action** drop-down menu.
5. Click **Clone from** to browse and select the vCenter Server template you created for use in vCloud Automation Center.
6. For a Windows-based vCloud Automation Center blueprint, type the customization specification name in the **Customization spec** text box.
   
   You created the customization specification as part of the vCenter Server virtual machine template preparation.
7. Add information to the **Property** and **Security** tabs as required and click **OK** to save your changes.

The vCloud Automation Center blueprint appears in the **Blueprint** menu.

**What to do next**

Register the vCloud Automation Center blueprint configured for clone deployment for use in vCloud Application Director. See “Register the vCloud Automation Center Cloud Provider and Template,” on page 57.
Create and Configure vCloud Automation Center Blueprint for Linked Clone Deployment

When you create a custom vCloud Automation Center blueprint you can allow a linked clone deployment to provision vCloud Application Director Windows-based or Linux-based applications and services to the VMware vSphere environment. A linked clone is a copy of a virtual machine based on a snapshot. It occupies a small disk space, quick to provision, ideal when performance is not a high priority.

Prerequisites

- Verify that vCloud Automation Center 5.2 or 6.0 is installed and configured.
- Verify that the vCloud Automation Center provisioning group that you plan to use with vCloud Application Director is created and configured with Enterprise administrator privileges.
- Verify that a vCenter Server virtual machine is available for the vCloud Automation Center blueprint to create a linked clone deployment.
  
  You must have the customization specification name available for a Windows-based vCloud Automation Center blueprint. See “Prepare vCenter Server Windows Virtual Machine Templates,” on page 50.
- Familiarize yourself with how to create a blueprint in vCloud Automation Center. See the vCloud Automation Center Operating guide.

Procedure

1. Log in to the vCloud Automation Center server.
2. Select New Blueprint > Virtual > vSphere (vCenter) and name the blueprint vCAC_Clone.
3. Complete the information on the Blueprint Information tab.
4. On the Build Information tab, select the Clone option from the Action drop-down menu.
5. In the Clone from drop-down menu, select the vCenter Server template you created for use in vCloud Automation Center.
6. For a Windows-based vCloud Automation Center blueprint, type the customization specification name in the Customization spec text box.
   
   You created the customization specification as part of the vCenter Server virtual machine template preparation.
7. Click OK to save your changes.
8. Select vCAC_Clone, request a machine from the drop-down menu and accept the default settings.
   
   The request information is derived from the resources defined in the vCenter Server template.
   
   When your machine request is processed, it appears under the Provisioning Group > Group Machine menu. The machine name vCAC_Clone is appended with a unique series of letters and numbers such as vCAC_Clone-VFF17Y0.
9. Log in to the vCAC_Clone-VFF17Y0 machine and stop the vCloud Application Director and vCloud Automation Center services depending on your Windows or Linux operating system.

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vmware-appdirector-agent-service-vcac</td>
<td>vCloud Application Director bootstrap service for Linux</td>
</tr>
<tr>
<td>vrm-agent</td>
<td>vCloud Automation Center agent service for Linux</td>
</tr>
<tr>
<td>Service</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AppDAgentBootstrap</td>
<td>vCloud Application Director bootstrap service for Windows</td>
</tr>
<tr>
<td>VRMAgent or winservice</td>
<td>vCloud Automation Center 5.2 and 6.0 agent service for Windows</td>
</tr>
</tbody>
</table>

10. Run the Windows script `agent_reset.bat` or Linux script `agent_reset.sh` to remove runtime log files from vCAC_Clone-VFF17Y0.

11. Power off the vCAC_Clone-VFF17Y0 machine.

**What to do next**
Create a snapshot from the vCAC_Clone blueprint for a linked clone deployment. See “Create a Snapshot from a vCloud Automation Center Blueprint,” on page 56.

### Create a Snapshot from a vCloud Automation Center Blueprint

When you create a linked clone, you must create a snapshot of a vCloud Automation Center virtual machine. This snapshot requires a small disk space, so a linked clone deployment is faster than a clone deployment.

For more information about linked clones and creating snapshots, see *vCloud Automation Center What’s New Guide*.

**Prerequisites**
- Verify that vCloud Automation Center 5.2 or 6.0 is installed and configured.
- Verify that the vCloud Automation Center provisioning group that you plan to use with vCloud Application Director is created and configured with Enterprise administrator privileges.
- Verify that a vCloud Automation Center blueprint is available to create a snapshot from. See “Create and Configure vCloud Automation Center Blueprint for Linked Clone Deployment,” on page 55.

**Procedure**

1. Select **New Blueprint > Virtual** and name the blueprint `vCAC_LinkedClone`.
2. Complete the information on the **Blueprint Information** tab.
3. On the **Build Information** tab, click the **Linked Clone** option for the **Action** drop-down menu to appear.
4. Click the **Clone from** drop-down menu to create a snapshot of the vCAC_Clone-VFF17Y0 machine.
   a. In the Select Snapshot dialog box, select `vCAC_Clone-VFF17Y0` and click **Next**.
   b. Click **New Snapshot**.
   c. In the New Snapshot dialog box, type `vCAC_Clone-VFF17Y0_snapshot` in the **Description** text box and click **OK**.
      vCloud Automation Center creates a snapshot of vCAC_Clone-VFF17Y0. After the snapshot is created, it appears in the Snapshot Manager.
   d. Click **Finish** and click **OK** to save your changes.
      vCAC_Clone-VFF17Y0_snapshot appears in the **Clone from** text box.
5. For a Windows-based vCloud Automation Center blueprint, type the customization specification name in the **Customization spec** text box.
   You created the customization specification as part of the vCenter Server virtual machine template preparation.
Configure the Property and Security tabs as required and click OK to save your changes. The vCAC_LinkedClone blueprint appears in the Blueprint menu.

What to do next

Register the vCloud Automation Center blueprint configured for linked clone deployment for use in vCloud Application Director. See “Register the vCloud Automation Center Cloud Provider and Template,” on page 57.

Updating Existing Virtual Machine Templates in vCloud Automation Center

To update the content of an existing Linux or Windows template, or to use it for creating a new template, you must run applicable commands to remove the agent bootstrap service.

For a Linux template, the agent_reset.sh command resets the vCloud Application Director agent bootstrap status and deletes existing runtime log files. You can log in to the virtual machine as root and run this command:

/opt/vmware-appdirector/agent-bootstrap/agent_reset.sh

For a Windows template, the agent_reset.bat command deletes existing runtime log files. In a PowerShell command window, type the following command.

\opt\vmware-appdirector\agent-bootstrap\agent_reset.bat

For the Windows template, you can also remove existing vCloud Application Director agent bootstrap and vCloud Automation Center 5.1 guest agent. In a PowerShell command window, type the following command.

\opt\vmware-appdirector\agent-bootstrap\agent_bootstrap_removal.bat

Register the vCloud Automation Center Cloud Provider and Template

vCloud Application Director registers a cloud provider by connecting to a vCloud Automation Center provisioning or business group.

A vCloud Automation Center provisioning or business group is a collection of machines, corresponding to a business, department, or other organizational unit. Each provisioning or business group has access to one or more vCloud Automation Center blueprints used to request machines. These blueprints can belong to one or more provisioning or business group. To request machines, a user must belong to at least one provisioning or business group. The machine resource information in the vCloud Automation Center blueprint is obtained from the cloned or linked cloned VMware vSphere template.

Prerequisites

- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
- Verify that vCloud Automation Center 5.2 or 6.0 is installed and configured.
- Verify that the vCloud Automation Center provisioning or business group that you plan to use with vCloud Application Director is created and configured.
- Verify that you have Enterprise administrator privileges to access the provisioning or business groups.

Users with Enterprise administrator privileges create and maintain provisioning or business groups. For more information about provisioning or business groups, see the vCloud Automation Center Operating guide.

- Make sure that you register vCloud Application Director to vCloud Automation Center using CLI. See “Register vCloud Application Director to vCloud Automation Center,” on page 30.
Verify that at least one vCloud Automation Center blueprint is available in the provisioning or business group for the vCloud Application Director cloud provider to connect to. This blueprint must point to a virtual machine template that meets the vCloud Application Director requirements. See “Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates,” on page 48.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select **Clouds > Cloud Providers**.
2. Click **New** in the toolbar.
3. Complete the cloud provider information.
   
   For vCloud Automation Center 6.0, when you register vCloud Application Director to vCloud Automation Center, the host name and business group name gets populated for the logged in user.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name and Description</td>
<td>Include the vCloud Automation Center provisioning group name in either of these fields. The text from these fields appears in the Deployment Profile wizard under the Cloud Provider column.</td>
</tr>
<tr>
<td>Cloud Provider Type</td>
<td>vCloud Automation Center 5.2 or 6.0 is the supported type.</td>
</tr>
</tbody>
</table>
   | vcAC IP/Host Name/URL| Specify the vCloud Automation Center 5.2 server information. Use the secure vCloud Automation Center IP address or the secure vCloud Automation Center host URL.  
   |                      | An example of a secure vcAC IP, https://192.0.2.115 or https://192.10.1.113:443.  
   |                      | An example of a secure Host Name URL, https://vcac.it.vmware.com.           |
   | User Name and Password| Add Enterprise administrator credentials for the provisioning group in vCloud Automation Center.  
   |                      | The domain name is required when you type the User Name. An example of a User Name is vCAC\Admin. |
   | Provisioning Group   | Click **Select** to select from a list of provisioning group URLs that are available for your cloud provider. |

4. To test whether the values you entered are correct, click **Validate Connection**.
5. Click **Save** to register the cloud provider.
   
   If required entries for the cloud provider were invalid or left blank, you are prompted to correct them when you click **Save**.
6. To register vCloud Automation Center blueprints, click **Edit** in the toolbar.
7. In the Templates section, click **New** to register templates.
8. Select the check boxes next to one or more cloud templates or vCloud Automation Center blueprints in the list, and click **OK**.
   
   The templates you registered are added to the list of cloud templates that you can choose from when you map a logical template to a cloud template.
To change existing information for a vCloud Automation Center cloud provider, click **Edit** in the toolbar, make your changes, and click **Save**.

When vCloud Application Director registers the cloud template or vCloud Automation Center blueprint, the reservation policy information associated with the cloud template is saved. If the reservation policy changes in the vCloud Automation Center environment, you can click **Refresh** to get the updates in the reservation policy. As a best practice, refresh the cloud template information, vCAC Infrastructure IP, and the host name when you edit a vCloud Automation Center cloud provider.

**NOTE:** If the vCloud Automation Center blueprint name changes in the vCloud Automation Center environment, the new blueprint name does not appear in the vCloud Application Director user interface. The new blueprint name is updated only in the cloud template meta data.

The cloud provider is registered and its templates are available for use in vCloud Application Director.

**What to do next**

Map the vCloud Automation Center cloud template to the vCloud Application Director logical templates. See “Add a Logical Template to the Catalog,” on page 121.

Create a deployment environment to deploy an application to the vCloud Automation Center environment. See “Create a vCloud Automation Center Deployment Environment,” on page 59.

**Create a vCloud Automation Center Deployment Environment**

You must map a deployment environment to a vCloud Automation Center reservation policy before you can deploy applications to that cloud environment.

A vCloud Automation Center provisioning group can have several reservations mapped to reservation policies. A reservation has a pool of resources such as memory, storage, and networking assigned to it by a particular provisioning group to build machines. Based on the reservation settings, you can save a virtual machine in a designated storage space and determine the network it can connect to. You use reservation policies to group similar resources to create defined service levels or to make a specific type of resource available for a particular purpose.

You can add a reservation policy with a reservation to a vCloud Automation Center blueprint. When you request a machine from this blueprint, the virtual machine is saved in a designated storage space and assigned to a network already defined in the reservation. If you do not map a reservation policy to a blueprint, vCloud Automation Center assigns a reservation depending on the resource requirements of the virtual machine in the vCloud Automation Center blueprint.

**Prerequisites**

- Verify that your user account has the **ROLE_CLOUD_ADMIN** cloud administrator role assigned to it.
- Verify that vCloud Automation Center 5.2 or 6.0 is installed and configured.
- Verify that the vCloud Automation Center provisioning or business group that you plan to use with vCloud Application Director is created and configured.
- Verify that you have Enterprise administrator privileges to access the provisioning or business groups. Users with Enterprise administrator privileges create and maintain provisioning or business groups. For more information about provisioning or business groups, see the **vCloud Automation Center Operating guide**.
- Verify that a reservation policy is created and configured in the vCloud Automation Center. The vCloud Application Director deployment environment is equivalent to the reservation policy in vCloud Automation Center.
Verify that at least one vCloud Automation Center cloud provider is registered in vCloud Application Director. See “Register the vCloud Automation Center Cloud Provider and Template,” on page 57.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Clouds > Deployment Environments.
2. Click New in the toolbar.
3. Complete the deployment environment information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment Environment name and</td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>Include the reservation policy name in either of these text boxes. The text from these text boxes appears in the Deployment Profile wizard under the Deployment Environment column.</td>
</tr>
<tr>
<td>Cloud Provider</td>
<td>If the cloud provider does not appear in the list, cancel the dialog box and select Clouds &gt; Cloud Providers to add the vCloud Automation Center cloud provider.</td>
</tr>
<tr>
<td>Reservation Policy</td>
<td>Click Select to select from a list of reservation policies that the cloud provider you selected provides. Reservation policies in vCloud Automation Center that include auto in the name do not appear in the reservation policy list. Rename the reservation policy so that it appears in the reservation policy list. If you select No reservation policy, vCloud Automation Center assigns a reservation based on availability of resources.</td>
</tr>
</tbody>
</table>

4. Select a reservation policy and click OK.

A vCloud Automation Center reservation policy is mapped to the vCloud Application Director deployment environment name.

5. Click Save.

6. (Optional) To change the existing information for a vCloud Automation Center deployment environment, click Edit in the toolbar. Make your changes. For certain deployments, you can also edit the properties of the proxy settings. Changing these proxy settings lets you access external data. Click Show to edit the following properties:

   - http.proxyHost
   - http.proxyPort
   - http.proxyUser
   - http.proxyPassword

   If you do not edit the proxy setting properties, then the global proxy settings are considered.

When you create a deployment profile, you can select this deployment environment to deploy applications to vSphere.

What to do next

You can map an external service instance, create a policy instance, or register a solution instance with the deployment environment. See “Map an External Service Instance,” on page 61, “Create a Policy Instance,” on page 61 or “Create a Solution Instance,” on page 64.
Map an External Service Instance

You can map an external service instance to a deployment environment so that the instance is always used when an application is deployed to that deployment environment.

Prerequisites

- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
- Depending on your cloud provider, you must have at least one vCloud Director, vCloud Automation Center, or Amazon EC2 deployment environment mapped in vCloud Application Director. See “Create a vCloud Director Deployment Environment,” on page 79, “Create a vCloud Automation Center Deployment Environment,” on page 59, or “Create an Amazon EC2 Deployment Environment,” on page 89.
- If you are using a custom external service, verify that a preconfigured or custom external service is available in the catalog. See “Add an External Service to the Catalog,” on page 113.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Clouds > Deployment Environments.
2. Select an existing deployment environment.
3. Select the External Service Instance tab and click Register to create an external service instance.
4. Complete the external service instance information for the deployment environment.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name and Description</td>
<td>Include the external service name in either of these fields. The text from the name field appears in the Deployment Profile wizard under the Deployment Environment column.</td>
</tr>
<tr>
<td>External Service Version</td>
<td>If the external service does not appear in the drop-down list, cancel the dialog box and select Catalog &gt; External Services to add a custom external service to the catalog.</td>
</tr>
</tbody>
</table>
| Service Provider        | Click the down arrow to select from a list of service providers that are associated with the external service.  
                          | Note: This option appears when you map a custom external service to the deployment environment. |

The properties defined in the external service version or provider specification version are automatically populated in the Properties section.

5. Click Save when you finish.
6. Click the arrow next to the external service instance name to return to the previous page.

Create a Policy Instance

A policy instance allows an existing policy definition to take effect for all of the application provisioning activities in a specific deployment environment.

Prerequisites

- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
Depending on your cloud provider, you must have at least one vCloud Director, vCloud Automation Center, or Amazon EC2 deployment environment mapped in vCloud Application Director. See “Create a vCloud Director Deployment Environment,” on page 79, “Create a vCloud Automation Center Deployment Environment,” on page 59, or “Create an Amazon EC2 Deployment Environment,” on page 89.

If you are using a custom policy, verify that it is available in the catalog. See “Add a Policy to the Catalog,” on page 124.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Clouds > Deployment Environments**.
2. Select an existing deployment environment.
3. Select the **Policy Instances** tab and click **Create Policy Instance** to create a policy instance.
4. Complete the policy instance information for the deployment environment.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name and Description</strong></td>
<td>Include the policy name in either of these fields. The text from these fields appears in the policy instances section for the deployment environment.</td>
</tr>
<tr>
<td><strong>Policy</strong></td>
<td>If the policy does not appear in the drop-down list, cancel the dialog box and select <strong>Catalog &gt; Policies</strong> to add a custom policy to the catalog.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>Click the down arrow to select from a list of versions that are associated with the policy.</td>
</tr>
<tr>
<td><strong>Critical</strong></td>
<td>Select this check box to stop operations that can result in a non-compliant deployment, irrespective of current compliance state of deployment. Before you use this option, it is recommended that the policy compliance is fully established in a deployment environment. A non-critical policy instance marks the compliance status of the deployment but does not stop the operation.</td>
</tr>
</tbody>
</table>

**Note** If you specify a new property value in the policy instance, the new property value overrides the existing catalog value defined in the policy.

The properties and scripts defined in the policy version are automatically populated in the Properties and Actions sections.

When you create a policy action script, the policy instance makes a copy. Changes to the policy script in the catalog do not affect the existing policy instances associated to that policy. You need to create a policy instance for the script changes to take effect.

5. Click **Save** when you finish.
6. Click the arrow next to the policy instance name to return to the previous page.

**Register a Solution Instance**

To register a solution instance such as Puppet master, you must configure the Puppet master to work with the vCloud Application Director server and create a Puppet master solution instance within a deployment environment in vCloud Application Director.

For the registration process, you can implement either the manual or the automated option. For the automated option instructions, see [http://kb.vmware.com/kb/2068342](http://kb.vmware.com/kb/2068342).
Procedure
1. **Prepare the Puppet Master Environment for Registration** on page 63
   To register a Puppet master within a deployment environment in vCloud Application Director you must prepare the Puppet master to work with the vCloud Application Director server.

2. **Create a Solution Instance** on page 64
   A solution instance such as Puppet master, must be created within a deployment environment in vCloud Application Director so that the Puppet master is always used when an application using Puppet based services is deployed in that deployment environment.

### Prepare the Puppet Master Environment for Registration

To register a Puppet master within a deployment environment in vCloud Application Director you must prepare the Puppet master to work with the vCloud Application Director server.

You can also automate the registration process, see [http://kb.vmware.com/kb/2068342](http://kb.vmware.com/kb/2068342).

**Prerequisites**
- Verify that the Puppet enterprise version 3.0.1 or later or Puppet OpenSource version 3.2.4 or later is installed.
- Make sure that there is a client machine available with Puppet installed besides the Puppet master.

**Procedure**

1. Navigate to the directory on your Puppet master and create a directory called nodes.
   - On Puppet enterprise, navigate to `/etc/puppetlabs/puppet/manifests/`.
   - On Puppet OpenSource, navigate to `/etc/puppet/manifests/`.

2. Open the `/etc/puppetlabs/puppet/manifests/site.pp` or `/etc/puppet/manifests/site.pp` file and add the command `import 'nodes/*.pp'` to the first line of the file.

3. In the command line, create the `empty_manifest_file.pp` file.
   - On Puppet enterprise, type `touch /etc/puppetlabs/puppet/manifests/nodes/empty_manifest_file.pp`
   - On Puppet OpenSource, type `touch /etc/puppet/manifests/nodes/empty_manifest_file.pp`

4. Download the `appdintegn.rb` and `appdintegn.ddl` agent files.

5. Copy the `appdintegn.rb` and `appdintegn.ddl` agent files to the `LIBDIR/mcollective/agent` directory on the Puppet master.
   
   `LIBDIR` is the value of the MCollective libdir setting that is typically set to `/opt/puppet/libexec/mcollective/`.

6. Restart MCollective to view the newly copied agent files.
   - On Puppet enterprise, type `/etc/init.d/pe-mcollective restart`
   - On Puppet OpenSource, type `/etc/init.d/mcollective restart`
7 On a client machine that has Puppet installed, generate a certificate

```
puppet certificate generate Name --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster
```

_Name_ is the name of a string for the vCloud Application Director certificate corresponding to the Puppet master. _TempCredsDir_ is a locally created directory in the /tmp folder such as /tmp/appd_mc/credentials and _CAPuppetMaster_ is the host name or IP address of the Puppet master.

8 (Optional) On the Puppet master, if the auto-sign for the certificate is not turned on type the command

```
sudo puppet cert sign Name.
```

9 Copy the signed public certificate file to the MCollective authorized client list.

```
cp CertDirectory/Name.pem AuthorizedClientsDirectory
```

_CertDirectory_ is the Puppet master certdir config setting and _AuthorizedClientsDirectory_ is the config setting for the MCollective server plugin.ssl_client_cert_dir.

A sample Puppet enterprise copy command, _cp /etc/puppetlabs/puppet/ssl/ca/signed/vmware-appd.pem /etc/puppetlabs/mcollective/ssl/clients/_

10 On the client machine that has Puppet installed, get the signed certificates from the Puppet master.

```
puppet certificate find Name --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster
```

```
puppet certificate find ca --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster
```

11 On the client machine that has Puppet installed, get the public certificate from the Puppet master.

- On Puppet enterprise, type

```
puppet certificate find pe-internal-mcollective-servers --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster.
```

- On Puppet OpenSource, type

```
puppet certificate find mcollective-servers --ssldir TempCredsDir --ca-location remote --ca_server CAPuppetMaster.
```

vCloud Application Director creates the node definition file *.pp for each virtual machine it provisions in the /etc/puppetlabs/puppet/manifests/nodes or /etc/puppet/manifests/nodes directory.

What to do next

Register the Puppet master within a deployment environment in vCloud Application Director. See “Create a Solution Instance,” on page 64.

Create a Solution Instance

A solution instance such as Puppet master, must be created within a deployment environment in vCloud Application Director so that the Puppet master is always used when an application using Puppet based services is deployed in that deployment environment.

You can create only one Puppet master instance per a deployment environment. You can also use the same Puppet master in another deployment environment.

Prerequisites

- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
Depending on your cloud provider, you must have at least one vCloud Director, vCloud Automation Center, or Amazon EC2 deployment environment mapped in vCloud Application Director. See “Create a vCloud Director Deployment Environment,” on page 79, “Create a vCloud Automation Center Deployment Environment,” on page 59, or “Create an Amazon EC2 Deployment Environment,” on page 89.

Make sure that the Puppet master is configured to work with the vCloud Application Director server. See “Prepare the Puppet Master Environment for Registration,” on page 63 or http://kb.vmware.com/kb/2068342.

If you are using a custom policy, verify that it is available in the catalog. See “Add a Policy to the Catalog,” on page 124.

Procedure

1. Log in to the vCloud Application Director Web interface.
2. On the vCloud Application Director title bar, click the drop-down menu and select Clouds > Deployment Environments.
3. Select an existing deployment environment.
4. Select the Solution Instances tab and click Create Solution Instance.
5. Complete the solution instance information for the deployment environment.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name and Description</td>
<td>Include the solution name. You can add property or agent configuration information for the solution in the description section.</td>
</tr>
<tr>
<td>Solution</td>
<td>Select the available Puppet Master v1.0.0 solution from the drop-down list.</td>
</tr>
</tbody>
</table>

The solution properties and corresponding agents are populated.

6. In the Properties section, type the required property values.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>puppet_server</td>
<td>Qualified domain name of the Puppet master server. For the agent to trust the Puppet master certificate, use one of the valid DNS names you set when you installed the Puppet master. The vCloud Application Director server must be able to contact Puppet master at this DNS name, unless the puppet_server_ip property is also supplied.</td>
</tr>
<tr>
<td>puppet_server_ip</td>
<td>Puppet master server IP address. The IP address must be specified if the host name of the Puppet master server is not resolvable in the vCloud Application Director server or agent nodes.</td>
</tr>
<tr>
<td>identity_private_key</td>
<td>Private key of the certificate issued to the vCloud Application Director server by the Puppet master. vCloud Application Director uses this certificate for authentication with Puppet master and MCollective. The format of the certificate is PEM encoded including the header and trailer. The header for example is formatted as, ‘-----BEGIN RSA PRIVATE KEY--‘. Copy contents of the file TempCredsDir/private_keys/Name.pem you created to prepare the Puppet master, as the value for this property.</td>
</tr>
<tr>
<td>identity_public_cert</td>
<td>The X509 certificate corresponding to the identity_private_key property. The format of the certificate is PEM encoded including the header and trailer. The header for example is formatted as, ‘-----BEGIN CERTIFICATE--‘. Copy contents of the file TempCredsDir/certs/Name.pem you created to prepare the Puppet master, as the value for this property.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ca_cert</td>
<td>The X509 certificate of the certificate authority that issues digital certificates used to authenticate with Puppet master and MCollective. The format of the certificate is PEM encoded including the header and trailer. The header for example is formatted as, ‘-----BEGIN CERTIFICATE-----’. Copy contents of the file $TempCredsDir/certs/ca.pem you created to prepare the Puppet master, as the value for this property.</td>
</tr>
<tr>
<td>mc_servers_shared_cert</td>
<td>Shared server certificate or public key for the MCollective cluster. The format of the certificate is PEM encoded including the header and trailer. The header for example is formatted as, ‘-----BEGIN CERTIFICATE-----’ or ‘-----BEGIN PUBLIC KEY-----’. Copy contents of the file $TempCredsDir/certs/pe-internal-mcollective-servers.pem for Puppet enterprise or $TempCredsDir/certs/mcollective-servers.pem for Puppet OpenSource you created to prepare the Puppet master, as the value for this property. The trailer for example is formatted as, ‘-----END CERTIFICATE-----’ or ‘-----END RSA PUBLIC KEY-----’.</td>
</tr>
<tr>
<td>mc_messaging_server_password</td>
<td>Password corresponding to the config setting plugin.activemq.pool.1.password in the MCollective server config file. For example, the MCollective server config file in the Puppet enterprise is located at /etc/puppetlabs/mcollective/server.cfg.</td>
</tr>
<tr>
<td>identity_cert_name</td>
<td>Name of the identity certificate. This name is the string which you set when you generated the vCloud Application Director certificate to correspond to the Puppet master during the preparation of the Puppet master.</td>
</tr>
<tr>
<td>puppet_version</td>
<td>Puppet enterprise suite version for a Puppet enterprise and Puppet package version for a Puppet open source. For example, the Puppet enterprise version is 3.0.1 and the Puppet open source version is 3.2.4.</td>
</tr>
<tr>
<td>mc_messaging_server_username</td>
<td>User name corresponding to the config setting plugin.activemq.pool.1.user in the MCollective server config file. For example, the MCollective server config file in the Puppet enterprise is located at /etc/puppetlabs/mcollective/server.cfg.</td>
</tr>
<tr>
<td>mc_messaging_server_port</td>
<td>Port corresponding to the config setting plugin.activemq.pool.1.port in the MCollective server config file. For example, the MCollective server config file in the Puppet enterprise is located at /etc/puppetlabs/mcollective/server.cfg.</td>
</tr>
<tr>
<td>node_manifest_dir</td>
<td>Directory where the node definition manifests are located for virtual machines deployed using vCloud Application Director. The Puppet master site.pp must be able to import *.pp files from this directory.</td>
</tr>
<tr>
<td>mc_appd_agent_ddl</td>
<td>Location of the MCollective vCloud Application Director agent DDL file. You do not need to override this value.</td>
</tr>
<tr>
<td>mc_client_setup_script</td>
<td>Location of the MCollective Client setup Beanshell script. You do not need to override this value.</td>
</tr>
<tr>
<td>global_conf</td>
<td>URL to download the Darwin global configuration for each node.</td>
</tr>
</tbody>
</table>
7 In the Agents section, type the required values.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pe_installer_payload_base_path</td>
<td>Base URI where the operating system and architecture specific Puppet enterprise tar files are hosted. The script detects the URI based on the pe_installer_payload_base_path and the current version of operating system and architecture on the machine. You can skip this auto detection by specifying a value for the pe_installer_payload property. Automatic detection assumes that the specific directory structure mirrors the structure of the Puppet Labs hosted tar files that are located at <a href="https://s3.amazonaws.com/pe-builds/">https://s3.amazonaws.com/pe-builds/</a>.</td>
</tr>
<tr>
<td>is_enterprise</td>
<td>Indicates if the Puppet master being registered is a Puppet enterprise with a value set to true or a Puppet open source with a value set to false.</td>
</tr>
</tbody>
</table>

8 In the Agents section, type the optional value.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pe_installer_payload</td>
<td>Link to download tar file for Puppet Enterprise. Leave this value empty unless you want to explicitly provide a complete URI of an OS-neutral Puppet Enterprise tar file. If the value is not specified, the script detects the URI based on the pe_installer_payload_base_path and current version of operating system and architecture on the machine. An example URI is, <a href="https://s3.amazonaws.com/pe-builds/released/3.0.1/puppet-enterprise-3.0.1-all.tar.gz">https://s3.amazonaws.com/pe-builds/released/3.0.1/puppet-enterprise-3.0.1-all.tar.gz</a></td>
</tr>
</tbody>
</table>

9 Click Save when you finish.

vCloud Application Director checks the property and agent values you typed and displays an error message if the required values are missing.

10 Click Validate Connection to check the connection to the server.

11 Click the arrow next to the solution instance name to return to the previous page.

12 (Optional) On the client machine that has Puppet installed, type the command to remove the temporary credential directory.

   `rm -rf TempCredsDir`

**What to do next**

Import Puppet content into the vCloud Application Director catalog as services. See “Import a Puppet Service to the Catalog,” on page 112.
Setting Up Application Provisioning for the vCloud Director Environment

To use vCloud Application Director to deploy applications in a vCloud Director environment, you must install vCloud Director 5.1.2 or 5.5.

To set up application provisioning you must create a vCloud Director organization with one or more user accounts and an organization vDC to allow vCloud Application Director to create cloud providers and register them to the organization vDC. See “Set Up vCloud Director for vCloud Application Director,” on page 23. You must also create custom virtual machine templates in vCloud Director.

In vCloud Application Director, you must register a cloud provider and template. Registering a cloud provider involves specifying a vCloud Director organization. When you register a cloud template, you map a vCloud Director vApp template to a logical template in the vCloud Application Director catalog. You must then create a deployment environment and map this deployment environment to a contained vCloud Director organization vDC.

Familiarize yourself with the key concepts that relate to setting up and configuring a cloud environment for application provisioning. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Virtual Machine Requirements for Creating vCloud Director Custom Templates,” on page 70
- “Creating Windows Virtual Machine Templates in vCloud Director,” on page 71
- “Create Linux Virtual Machine Templates in vCloud Director,” on page 75
- “Verify Cloud Template Configuration from the vCloud Director Catalog,” on page 77
- “Updating Existing Virtual Machine Templates in vCloud Director,” on page 77
- “Exporting Virtual Machine Templates with OVF Format,” on page 78
- “Register the vCloud Director Cloud Provider and Template,” on page 78
- “Create a vCloud Director Deployment Environment,” on page 79
Virtual Machine Requirements for Creating vCloud Director Custom Templates

In vCloud Application Director, you map logical templates to cloud templates created in vCloud Director. These cloud templates must meet certain requirements to work properly in vCloud Application Director.

**Table 7-1. Virtual Machine Requirements for Custom Templates**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
</table>
| Operating system             | Use one of the following operating systems:  
  - Supported operating systems with 32-bit Red Hat Enterprise Linux 6.4, Ubuntu 10.04.3, and CentOS 6.4  
  - Supported operating systems with 64-bit Red Hat Enterprise Linux 6.4, Ubuntu 10.04.2, CentOS 6.4, and Windows Server 2008 R2 Enterprise with Service Pack 1 |
| VMware Tools                 | VMware Tools must be installed and the version must be from vCloud Director 5.1.2 or 5.5 or vCenter Server 5.0, 5.1, or 5.5.                                                                                       |
| CD/DVD drive                 | At least one CD/DVD drive must be available on the vCloud Director virtual machine. See the vCloud Director documentation.                                                                                   |
| JRE                          | JRE 1.7.0 Update 45 must be installed. The preferred and supported JRE can be installed from one of these packages, which are available in the vCloud Application Director virtual appliance:  
  - JRE for Windows  
    - [http://Application_Director_IP/agent/jre-1.7.0_45-win64.zip](http://Application_Director_IP/agent/jre-1.7.0_45-win64.zip) See “Configure Windows Virtual Machine Template in vCloud Director,” on page 72.  
  - JRE for Linux  
    - [http://Application_Director_IP/agent/jre-1.7.0_45-lin32.zip](http://Application_Director_IP/agent/jre-1.7.0_45-lin32.zip)  
    - [http://Application_Director_IP/agent/jre-1.7.0_45-lin64.zip](http://Application_Director_IP/agent/jre-1.7.0_45-lin64.zip)  
To install JRE on a virtual machine template, see “Create Linux Virtual Machine Templates in vCloud Director,” on page 75. |
| Linux agent bootstrap service| Download the Linux agent bootstrap package [vmware-appdirector-agent-service](http://vmware.com/web/vmware/downloads) from the VMware product download site [http://vmware.com/web/vmware/downloads](http://vmware.com/web/vmware/downloads). Install the Linux agent bootstrap script from one of the following packages:  
  - [http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_x86_64.rpm](http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_x86_64.rpm)  
  - [http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.rpm](http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.rpm)  
  - [http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_amd64.deb](http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_amd64.deb)  
  - [http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.deb](http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.deb)  
To install the agent bootstrap service on a virtual machine template, see “Create Linux Virtual Machine Templates in vCloud Director,” on page 75. |
| Windows agent bootstrap service on vCloud Director | Download the [http://Application_Director_IP/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip](http://Application_Director_IP/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip) file from the vCloud Application Director virtual appliance.  
To install the agent bootstrap service on a Windows-based virtual machine template, see “Configure Windows Virtual Machine Template in vCloud Director,” on page 72. |
| Supported Windows scripting  | vCloud Application Director supports scripting with Windows CMD or PowerShell 2.0.                                                                                                                        |
| Supported Linux scripting    | vCloud Application Director supports scripting with Bash.                                                                                                                                                |
Table 7-1. Virtual Machine Requirements for Custom Templates (Continued)

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux commands</td>
<td>The following Linux commands must be available on the virtual machine:</td>
</tr>
<tr>
<td></td>
<td>- wget</td>
</tr>
<tr>
<td></td>
<td>- md5sum</td>
</tr>
<tr>
<td></td>
<td>- grep</td>
</tr>
<tr>
<td></td>
<td>- sed</td>
</tr>
<tr>
<td></td>
<td>- setsid</td>
</tr>
<tr>
<td></td>
<td>- awk</td>
</tr>
<tr>
<td></td>
<td>- ifconfig</td>
</tr>
<tr>
<td></td>
<td>- apt-get</td>
</tr>
<tr>
<td></td>
<td>- yum</td>
</tr>
<tr>
<td>Optional services</td>
<td>If you plan to remotely access the virtual machine using Linux ssh logging or Windows remote desktop for troubleshooting or for other reasons, the OpenSSH server and client for Linux or Remote Desktop Services (RDS) for Windows must be installed and running properly.</td>
</tr>
</tbody>
</table>

IMPORTANT Because the boot process must not be interrupted, configure the virtual machine so that nothing causes the virtual machine’s boot process to pause before reaching the final operating system login prompt. For example, verify that no processes or scripts prompt for user interaction when the virtual machine starts. This requirement applies only to virtual machine templates created for the vCloud Application Director catalog.

Creating Windows Virtual Machine Templates in vCloud Director

With Windows virtual machine templates, you can use vCloud Application Director on vCloud Director 5.1.2 or 5.5 to create an application blueprint and deploy Windows-based services and applications such as .NET applications.

Before you create a Windows virtual machine template, you must set up the vCloud Director environment.

- Create a vCloud Director vApp on page 72
  When you create the Windows-based vCloud Director vApp, you must install and configure VMware Tools in the virtual machine.

- Configure Windows Virtual Machine Template in vCloud Director on page 72
  The Java SE 7 Runtime Environment (JRE) packages and agent bootstrap file must be configured for the Windows virtual machine to work properly in vCloud Application Director.

- Enable SID Change and Domain Join for Windows Virtual Machine Templates on page 74
  You can use Change SID and enable Join Domain for a Windows virtual machine template. With the Change SID option, all of the deployed virtual machines can acquire a unique security identifier (SID). Enable the option when you prepare a Windows virtual machine template. With the Join Domain option, the deployed virtual machines can become members of a specified Windows Active Directory domain.

- Add Windows Virtual Machine Templates to the vCloud Director Catalog on page 75
  Add the Windows virtual machine template to the vCloud Director Catalog and register it so that it works with vCloud Application Director.
Create a vCloud Director vApp

When you create the Windows-based vCloud Director vApp, you must install and configure VMware Tools in the virtual machine.

Prerequisites

- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Verify that the vCloud organization that you plan to use with vCloud Application Director is created and configured with a user account that has privileges to create and share vCloud Director catalog templates.
- Familiarize yourself with how to create a vApp in vCloud Director. See vCloud Director documentation.
- For the recommended storage and memory values and Remote Desktop Services for set up instructions, see the Windows Server 2008 R2 Enterprise with Service Pack 1 system requirements.
- Familiarize yourself with how to install and configure VMware Tools.

Procedure

1. Log in to vCloud Director.
2. Create a vApp with one virtual machine
   Allocate the appropriate virtual storage and virtual memory to support the Windows Server 2008 R2 Enterprise with Service Pack 1 operating system and your application.
3. Install Windows Server 2008 R2 Enterprise with Service Pack 1 in the virtual machine, using the Full Installation option.
4. Install VMware Tools in the virtual machine.

What to do next


Configure Windows Virtual Machine Template in vCloud Director

The Java SE 7 Runtime Environment (JRE) packages and agent bootstrap file must be configured for the Windows virtual machine to work properly in vCloud Application Director.

The agent bootstrap is a daemon that runs as an NT service on Windows after a virtual machine starts for the first time. The vCloud Application Director agent bootstrap uses the PowerShell script to run the bootstrap code.

Prerequisites

- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Verify that at least one vApp is created in vCloud Director. See “Create a vCloud Director vApp,” on page 72.

Procedure

1. Log in to the Windows virtual machine as a user belonging to the local Administrators group.
2. Download and install the supported Java SE 7 Runtime Environment from http://Application_Director_IP/agent/jre-1.7.0_45-win64.zip.
3. Unzip the JRE file to the C:\opt\vmware-jre\ directory.

4. In a PowerShell command window, navigate to the C:\opt\vmware-jre\ directory and type java -version to verify the installation. The installed version of Java appears.


6. Unzip the vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip file to a new folder.

7. Right-click to view the agent_bootstrap.ps1 file properties and click Unblock to disable the security on the file.

   **IMPORTANT** If you do not disable this Windows security feature, you cannot use the vCloud Application Director agent bootstrap file.

8. Add the NTRights.exe utility to the newly created folder.

   The NTRights.exe utility is used in the install.bat command to configure the agent bootstrap service to run in the darwin user account. The utility is included in the Windows Server 2003 Resource Kit Tools (rktools.exe), which you can download from the Microsoft download Web site.

9. (Optional) If the NTRights.exe utility is not available, manually configure the agent bootstrap service to run in a specific user account after you run the install.bat command.

   The install.bat command creates a user account called darwin for the agent bootstrap service and uses the password you set. The Password must meet the Windows password requirements.

10. To install the agent bootstrap service, open a Windows CMD console, navigate to the new folder, and type install.bat password=Password cloudProvider=vcd.

11. From the command-line, type services.msc and open the VMware vCloud Application Director agent bootstrap service.

12. On the Log On tab, set the Log on as option to This account and type the login credentials. The user name for this user account is .\darwin.

13. In the PowerShell command window, complete the following service-related tasks.

   a. Type net start AppDAgentBootstrap to verify that the service starts successfully.

   b. Type net stop AppDAgentBootstrap to stop the service.

14. Delete the runtime log files generated during the template creation.

15. (Optional) Enable Remote Desktop to remotely access future instances of this virtual machine.

   You can also use the remote access for troubleshooting purposes.

16. Shut down the Windows virtual machine and stop the vApp in vCloud Director.

17. Navigate to C:\opt\vmware-appdirector\agent-bootstrap\agent_bootstrap.ps1, change the $gcStatus value to $True, and add the Write-Host "Skipping guest customization check for vCD 5.1.2" command to the file.

**What to do next**

Before you save your vApp to a catalog in vCloud Director, you can select the Change SID and Join Domain features to allow the Windows template to become a member of a Windows Active Directory domain. See “Enable SID Change and Domain Join for Windows Virtual Machine Templates,” on page 74.
Add the vApp template to the vCloud Director catalog so that you can register it for use in vCloud Application Director. See “Add Windows Virtual Machine Templates to the vCloud Director Catalog,” on page 75.

Enable SID Change and Domain Join for Windows Virtual Machine Templates

You can use Change SID and enable Join Domain for a Windows virtual machine template. With the Change SID option, all of the deployed virtual machines can acquire a unique security identifier (SID). Enable the option when you prepare a Windows virtual machine template. With the Join Domain option, the deployed virtual machines can become members of a specified Windows Active Directory domain.

If you need to deploy a large number of Windows virtual machines that must join a corporate managed domain, use Windows templates with preconfigured domain settings. For smaller deployments or deployments that need the flexibility of joining different domains, a more effective approach is to use Windows templates without the preconfigured domain settings and add the Join Domain Predefined Task in the execution plan for deployment. See “Add Join Domain Predefined Task,” on page 187.

Prerequisites

- Verify that you have correctly created a Windows vApp and configured the virtual machine template. See “Create a vCloud Director vApp,” on page 72 and “Configure Windows Virtual Machine Template in vCloud Director,” on page 72.
- Verify that the Windows domain name, domain account, and account password are readily available.
- Verify that the virtual machine can connect to the designated domain controller.

  A DHCP service must be present on the network and the DNS server specified by the DHCP can resolve the domain name specified in the guest customization settings.

Procedure

1. Instantiate a vApp from the source template and locate the associated Windows virtual machine.
2. Log in to the Windows virtual machine as a user belonging to the local Administrators group.
3. Configure all of the virtual machine network interfaces to use DHCP.
   a. Open the Network control panel.
   b. Select Network and Sharing Center > Change Adapter Settings.
   c. Open each network connection and select Properties.
   d. Set the Internet Protocol Version 4/6 to Obtain an IP address Automatically and Obtain DNS server address automatically.
4. Shut down the Windows virtual machine and stop the vApp in vCloud Director.
5. In vCloud Director, open the properties for the virtual machine and select the Guest OS Customization tab to enable Guest Customization.
6. On the Guest OS Customization tab, select Change SID.
7. Select Join Domain and type the Windows domain name, domain account, and account password.
8. Select Add to Catalog to create a new vApp template from the vApp.

During deployment, Active Directory authenticates the login credentials and allows the Windows virtual machine to become a member of the domain.

What to do next

Add the vApp template to the vCloud Director catalog. See “Add Windows Virtual Machine Templates to the vCloud Director Catalog,” on page 75.
Add Windows Virtual Machine Templates to the vCloud Director Catalog

Add the Windows virtual machine template to the vCloud Director Catalog and register it so that it works with vCloud Application Director.

Prerequisites
- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Verify that the Windows template meets the vCloud Application Director virtual machine requirements. See “Virtual Machine Requirements for Creating vCloud Director Custom Templates,” on page 70.
- Complete the preparation tasks required to create a Windows vApp and configure the virtual machine template. See “Create a vCloud Director vApp,” on page 72 and “Configure Windows Virtual Machine Template in vCloud Director,” on page 72.

Procedure
1. Log in to the vCloud Director Web user interface.
2. Navigate to the vCloud Org location where the Windows vApp resides.
3. Select the vApp and click Add to Catalog to create a vApp template in the catalog.

The Windows template is added to the list of cloud templates that you can choose from when you map a logical template to a cloud template.

What to do next
Register the Windows template for use in vCloud Application Director. See “Register the vCloud Director Cloud Provider and Template,” on page 78.

Create Linux Virtual Machine Templates in vCloud Director

With Linux virtual machine templates, you can use vCloud Application Director on vCloud Director to create an application blueprint and deploy Linux-based services and applications.

Prerequisites
- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Verify that the vCloud organization that you plan to use with vCloud Application Director is created and configured with a user account that has privileges to create and share vCloud Director catalog templates.
- Familiarize yourself with how to create a vApp in vCloud Director. See vCloud Director documentation.

Creating multiple virtual machine vApp templates is not supported.
- Verify that the vApp meets the vCloud Application Director virtual machine requirements. See “Virtual Machine Requirements for Creating vCloud Director Custom Templates,” on page 70.
- Familiarize yourself with how to install and configure VMware Tools.
Procedure

1. Log in to vCloud Director and create a vApp with one virtual machine.
   Allocate the appropriate virtual storage and virtual memory to support the Linux operating system and your application.

2. Install the supported Linux operating system in the virtual machine.

3. Install VMware Tools in the virtual machine.

4. Log in the virtual machine with root privileges and open a terminal.

5. Download and install the supported JRE packages with the YUM package management tool or APT-GET packaging tool.
   If you have the supported JRE packages installed, skip this step.
   a. Download the JRE package from http://Application_Director_IP/agent/jre-1.7.0_45-lin-
      ArchitectureName.zip, where the ArchitectureName is 32 or 64.
   b. Create an /opt/vmware-jre directory.
   c. Unzip the JRE package.
      
      ```bash
      unzip -d /opt/vmware-jre
      ```
   d. Type /opt/vmware-jre/bin/java -version to verify the installation
      The installed version of Java appears.
   e. Remove any existing JRE packages from the /usr/java/default, /usr/java/latest, and /usr/java/1.7.0_21 directories.

6. Download and install the vCloud Application Director agent bootstrap service.
   a. Change directory to /tmp/ folder.
   b. For an RPM-based virtual machine, download the agent bootstrap from
      http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_ArchitectureName.rpm to the folder and type
      ```bash
      rpm -i vmware-appdirector-agent-service_6.0.0.0-0_ArchitectureName.rpm
      ```
      in the terminal to install the service, where the ArchitectureName is i386 for 32-bit and x86_64 for 64-bit.
   c. For a DEB-based virtual machine, download the agent bootstrap from
      http://Application_Director_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_ArchitectureName.deb to the folder and type
      ```bash
      dpkg -i vmware-appdirector-agent-service_6.0.0.0-0_ArchitectureName.deb
      ```
      in the terminal to install the service, where the ArchitectureName is i386 for 32-bit and amd64 for 64-bit.

7. Shut down the Linux virtual machine and stop the vApp in vCloud Director.

8. Navigate to the vCloud Org location where the Linux vApp resides.

9. Select the vApp and click Add to Catalog to create a vApp template in the catalog.

What to do next

Register the Linux template for use in vCloud Application Director. See “Register the vCloud Director Cloud Provider and Template,” on page 78.
Verify Cloud Template Configuration from the vCloud Director Catalog

Verify the cloud template configuration before you use the template in vCloud Application Director to avoid deployment failures.

If your cloud template has a customization script, vCloud Application Director overwrites the script with its own guest customization script. You use the guest customization script to set up the virtual machine so that it can communicate with the vCloud Application Director server to complete the deployment process.

Prerequisites
- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Verify that the virtual machine template is uploaded to the vCloud Director and registered as a vApp template in a vCloud Director catalog.
  
  For instructions, see the vCloud Director documentation.

Procedure

1. From the vCloud Director Web interface, manually instantiate a request to create a copy of the vApp to your cloud environment.

2. To verify that the guest customization process was successful, open the log file at /var/log/vmware-imc/customization.log and check that it shows that the customization completed with a status of success.

3. Check that an active and correct IP address exists for the IP assignment from vCloud Director.

4. Open and examine the agent bootstrap log file located at /opt/vmware-appdirector/agent/logs/agent_bootstrap.log.

5. Verify that the vCloud Application Director agent boot service ran and attempted to download the vCloud Application Director agent JAR file from the vCloud Application Director server.

  The attempt to download the JAR file is expected to fail at this point. The download process is successful when vCloud Application Director deploys the vApp from a vCloud Application Director execution plan.

What to do next

Verify that the items listed in “Virtual Machine Requirements for Creating vCloud Director Custom Templates,” on page 70 are present in the vApp and that they function properly.

Updating Existing Virtual Machine Templates in vCloud Director

To update the content of an existing Linux or Windows template, or to use it for creating a new template, you must run applicable commands to remove the agent bootstrap service.

For a Linux template, the agent_reset.sh command resets the vCloud Application Director agent bootstrap status and deletes existing runtime log files. You can log in to the virtual machine as root and run this command:

```
/opt/vmware-appdirector/agent-bootstrap/agent_reset.sh
```

For a Windows template, the agent_reset.bat command deletes existing runtime log files. In a PowerShell command window, type the following command.

```
\opt\vmware-appdirector\agent-bootstrap\agent_reset.bat
```

After you run the applicable command, you can place the vApp instance back in the catalog as a new vApp template.
Exporting Virtual Machine Templates with OVF Format

With vCloud Application Director you can create virtual machine templates in the OVF format and export the templates to a vCloud Director catalog.

If you plan to export virtual machine templates in the OVF format to a vCloud Director catalog, do not add custom OVF properties to the export packages. Custom OVF properties disable the Guest OS Customization feature in vCloud Director. With this feature disabled, vCloud Application Director cannot access virtual machine templates.

Register the vCloud Director Cloud Provider and Template

vCloud Application Director registers a cloud provider by connecting to a vCloud Director instance and organization. A catalog that is accessible to the vCloud Director user can provide access to virtual machine templates. The vDCs of the vCloud organization map to deployment environments.

For information about setting up vCloud Director for use with vCloud Application Director, see “Preparing to Install vCloud Application Director,” on page 19.

You cannot delete a cloud provider that is linked to a deployment environment. When you delete a cloud provider that is not linked to a deployment environment, all of the cloud template mappings from logical templates are removed.

Prerequisites

- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Create virtual machine templates that meet vCloud Application Director requirements. See “Virtual Machine Requirements for Creating vCloud Director Custom Templates,” on page 70.
- Verify that the vCloud organization that you plan to use with vCloud Application Director is created and configured with organization administrator privileges.
- When you map cloud templates for a cloud provider in vCloud Application Director, verify that the vCloud Director template that you are mapping to is a single virtual machine. Multiple virtual machine vApp templates are not supported.
- Templates with multiple virtual machines cannot be registered to a vCloud Application Director cloud provider.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Clouds > Cloud Providers.
2. Click New in the toolbar.
3. Complete the cloud provider information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name and Description</td>
<td>Include the vCloud Director organization name in either of these fields.</td>
</tr>
<tr>
<td></td>
<td>The text from these fields appears in the Deployment Profile wizard under</td>
</tr>
<tr>
<td></td>
<td>the Cloud Provider column.</td>
</tr>
<tr>
<td>Cloud Provider Type</td>
<td>vCloud Director 5.1.2 or 5.5 is the supported type.</td>
</tr>
<tr>
<td>Cloud IP/Hostname</td>
<td>The cloud IP or host name must match the IP address or host name of the</td>
</tr>
<tr>
<td></td>
<td>vCloud Director instance.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Organization Name</td>
<td>The organization name must match the name of an organization in vCloud Director.</td>
</tr>
</tbody>
</table>
| User Name and Password | Credentials for an organization user with organization administrator privileges for that organization. This user must not be the vCloud Director system administrator.  
**Note**: If the user name includes an @ symbol, replace it with %40 when you provide the login credentials. |

4 To test whether the values you entered are correct, click **Validate Connection**.

5 Click **Save** to register the cloud provider.

If required entries for the cloud provider were invalid or left blank, you are prompted to correct them when you click **Save**.

6 To register vCloud Director virtual machine templates, click **Edit** in the toolbar.

7 In the Templates section, click **New** to register templates.

8 Select a catalog from the drop-down menu and click **Get Templates**.

   Templates with multiple virtual machines cannot be registered to a vCloud Application Director cloud provider.

   Cloud templates with a single virtual machine appear in the drop-down menu.

9 Select the check boxes next to one or more templates in the list, and click **OK**.

   The templates you registered are added to the list of cloud templates that you can choose from when you map a logical template to a cloud template.

10 To change existing information for a vCloud Director cloud provider, click **Edit** in the toolbar, make your changes, and click **Save**.

The cloud provider is registered and its virtual machine templates and virtual datacenters are available for use in vCloud Application Director.

**What to do next**

Map these vCloud Director cloud templates to the vCloud Application Director logical templates. See “Add a Logical Template to the Catalog,” on page 121.

Create a deployment environment to deploy an application to the vCloud Director environment. See “Create a vCloud Director Deployment Environment,” on page 79.

---

**Create a vCloud Director Deployment Environment**

You must map a deployment environment to a vCloud Director organization vDC before you can deploy an application.

vCloud Director vDCs provide an environment where virtual systems can be stored, deployed, and operated. For example, you might have separate deployment environments for development, testing, staging, and production. For vCloud Director, a deployment environment is mapped to an organization vDC.

If a deployment environment is currently being used in a deployment profile or a current deployment, you cannot delete it.

**Prerequisites**

- Verify that your user account has the **ROLE_CLOUD_ADMIN** cloud administrator role assigned to it.
- Verify that at least one vCloud Director cloud provider is registered in vCloud Application Director. See “Register the vCloud Director Cloud Provider and Template,” on page 78.

- Verify that an organization vDC is created and configured in the vCloud Director organization. The vCloud Application Director deployment environment is equivalent to the organization vDC in vCloud Director.
  
  For information about setting up vCloud Director for use with vCloud Application Director, see “Preparing to Install vCloud Application Director,” on page 19.

- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Clouds > Deployment Environments**.
2. Click **New** in the toolbar.
3. Complete the deployment environment information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deployment Environment name and Description</td>
<td>Include the vDC name in either of these fields. The text from these fields appears in the Deployment Profile wizard under the Deployment Environment column.</td>
</tr>
<tr>
<td>Cloud Provider</td>
<td>If the cloud provider does not appear in the list, cancel the dialog box and select <strong>Clouds &gt; Cloud Providers</strong> to add the vCloud Director cloud provider.</td>
</tr>
<tr>
<td>Organization vDC URI</td>
<td>Click <strong>Select</strong> to select from a list of vDCs that the cloud provider you selected provides.</td>
</tr>
</tbody>
</table>

4. Select an organization vDC and click **OK**.
   
   A vCloud Director organization vDC is mapped to the vCloud Application Director deployment environment name.

5. Click **Save**.

6. (Optional) To change existing information for a vCloud Director deployment environment, click **Edit** in the toolbar, make your changes, and click **Save**.

The deployment environment you created is added to the list of deployment environments that you can select from when you create a deployment profile.

**What to do next**

You can map an external service instance, create a policy instance, or register a solution instance with the deployment environment. See “Map an External Service Instance,” on page 61, “Create a Policy Instance,” on page 61 or “Create a Solution Instance,” on page 64.
To use vCloud Application Director to deploy applications in an Amazon EC2 environment, you must set up one or more cloud environments.

**IMPORTANT** To deploy an application to Amazon EC2, you must install the vCloud Application Director for Release Automation edition.

Setting up application provisioning in an Amazon EC2 cloud environment involves signing up for an Amazon Web Services (AWS) user account and creating and configuring a VPC in an Amazon Region. With vCloud Application Director, you can create cloud providers and link them to a VPC in an Amazon Region. You must create virtual machine templates in the Amazon EC2 environments. These templates are used during application deployment to create virtual machines.

In vCloud Application Director, you must register a cloud provider and template. Registering a cloud provider involves specifying an Amazon EC2 Region. When you register a cloud template, you map an AMI from the cloud environment to a logical template in the vCloud Application Director catalog. You must then create a deployment environment and map this deployment environment to a combination of an Amazon Virtual Private Cloud (VPC) and Availability Zone.

To deploy a vCloud Application Director application to Amazon EC2, you must create a special virtual machine in the VPC, called an Endpoint VM. The Endpoint VM serves as the destination of a secure channel, called a cloud tunnel, from the vCloud Application Director appliance in your corporate network to your Amazon VPC environment. Application deployments in the VPC communicate with the vCloud Application Director appliance through the Endpoint VM. You can create and manage the cloud tunnels using the CLI. See “Managing Cloud Tunnels,” on page 207.
Familiarize yourself with the key concepts that relate to setting up and configuring a cloud environment for application provisioning. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Configure Amazon EC2 Environment for vCloud Application Director,” on page 82
- “Virtual Machine Requirements for Creating Amazon EC2 Custom Templates,” on page 86
- “Create Amazon EC2 Virtual Machine Templates or AMIs,” on page 87
- “Register the Amazon EC2 Cloud Provider and Template,” on page 88
- “Create an Amazon EC2 Deployment Environment,” on page 89

**Configure Amazon EC2 Environment for vCloud Application Director**

Setting up an Amazon EC2 environment requires you to create a VPC as a target location in an Amazon Region for application deployments. You must configure this VPC for vCloud Application Director applications to be deployed.

For deployments to Amazon EC2, NICs on the external networks receive Elastic IP addresses. In addition, a new security group is created for each deployment to allow communication between the instances in the deployment. This security group allows external access to 80, 8080, 8081, 8443, and 22 ports. For any other ports that you need to open, use the Amazon EC2 management console to locate the new security group and add the appropriate rules.

**Prerequisites**

- Set up an Amazon AWS user account.
Amazon defines the default limits for the number of Amazon EC2 instances, the number of Elastic IP addresses for an account, and the number of API calls. Contact Amazon support to request an increase in the instance, Elastic IP address, or API call limit. For more information on Amazon EC2, see Amazon AWS Documentation.

Procedure

1. Log in to the AWS Management Console using your AWS user account.
2. On the AWS Identity and Access Management (IAM) Dashboard, set up a new user with either an Administrator Access or Power User Access user privilege and generate an access key for this user. You can also set up an access key with the master AWS user account without setting up a new IAM user.
3. Download and save this access key for later use with vCloud Application Director.
4. To log in the deployed virtual machines in Amazon EC2, generate an SSH key-pair called titan_keypair and save the corresponding private key.
5. Create a VPC: On the VPC Dashboard of the AWS Management Console, create a VPC with a public subnet and a private subnet.
   a. Use the wizard option to create a VPC with public and private subnets.
      Two subnets are created whose instances have access to the Internet. The subnets are useful for downloading content during the application deployment.
   b. Specify an Availability Zone for the subnets.
      You can use the same Availability Zone for both subnets.
   c. Use the titan_keypair SSH key-pair.
      You can also use your own key-pair.
6. In the VPC, create a security group called EndpointAccess and set inbound and outbound port access rules.

<table>
<thead>
<tr>
<th>Port Access</th>
<th>Source or Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound 22</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>Inbound ALL</td>
<td>SubnetofVPC</td>
</tr>
<tr>
<td>Outbound ALL</td>
<td>0.0.0.0/0</td>
</tr>
<tr>
<td>(Optional) Inbound 2222</td>
<td>0.0.0.0/0</td>
</tr>
</tbody>
</table>

What to do next

Create a Linux virtual machine in your VPC. See “Create an Endpoint VM,” on page 83.

Create an Endpoint VM

Create a Linux virtual machine in Amazon EC2 called Endpoint VM in your VPC to deploy applications using vCloud Application Director.

A vCloud Application Director appliance can work with multiple Endpoint VMs. However, an Endpoint VM can only be designated as an endpoint of an Amazon Region, VPC, and Availability Zone for a single vCloud Application Director appliance.

Prerequisites

- Verify that an Amazon AWS user account is set up.
Verify that a VPC is available as a target location in an Amazon Region for application deployments. See “Configure Amazon EC2 Environment for vCloud Application Director,” on page 82.

Procedure

1. Install one Endpoint VM in each VPC to manage using vCloud Application Director.
2. Create an Amazon-based CentOS virtual machine on the external subnet.
   a. Use either the amazon/ami-vpc-nat-1.0.0-beta.i386-ebs AMI or the amazon/ami-vpc-nat-1.0.0-beta.x86_64-ebs AMI. The AMI is preconfigured to act as an IP masquerade device.
   b. Use the EndpointAccess security group for this virtual machine.
3. Set an Elastic IP address for use in the VPC and assign the IP address to the Endpoint VM.
4. Use an OpenSSH client to log in to the Endpoint VM with the Elastic IP address and the private key.
   ssh -i PathToPrivateKeyFile ec2-user@ApplicationDirectorEndpointVM
5. In the CLI, open the /etc/ssh/sshd_config configuration file, add the GatewayPorts yes, ClientAliveInterval 30, and ClientAliveCountMax 3 lines.
6. (Optional) If the outbound SSH is blocked by your corporate firewall, ask your firewall administrator to set an alternative port to 2222 for SSH traffic.
   Add the line Port 2222 to the /etc/ssh/sshd_config configuration file.
7. Restart the SSH daemon.
   sudo service sshd restart
8. Create an iptable rule to reroute the internal port 80 access to 8080.
   sudo iptables -t nat -I PREROUTING --source 0/0 --destination internal-ip-address-of-endpoint-vm -p tcp --dport 80 -j REDIRECT --to-ports 8080
9. Log out of the Endpoint VM.

What to do next

Set up a cloud tunnel for Amazon EC2 deployments to communicate with the vCloud Application Director server. See “Create a Cloud Tunnel to Connect to Amazon EC2,” on page 84.

Create a Cloud Tunnel to Connect to Amazon EC2

You must create and enable a cloud tunnel instance so that deployments in the Amazon EC2 VPC can communicate with the vCloud Application Director server.

IMPORTANT To deploy an application to Amazon EC2, you must install the vCloud Application Director for Release Automation edition.

Your network from vCloud Application Director to the Endpoint VM should have a minimum upload bandwidth of 1Mbps for every Amazon EC2 instance that is deployed. For faster downloads, store your applications on the Amazon Simple Storage Service instead of downloading them from the vCloud Application Director appliance over the cloud tunnel.

Prerequisites

- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
- Verify that the Endpoint VM is properly set up and configured. See “Configure Amazon EC2 Environment for vCloud Application Director,” on page 82.
Verify that the elastic IP address and the private IP address of the Endpoint VM are readily available.

Verify that the private key for the Endpoint VM is available to establish a cloud tunnel from the corporate network to the Endpoint VM.

Determine whether a proxy server is required to access Amazon EC2 from the network where vCloud Application Director is running.

The proxy server or the network must permit access to the standard SSH port 22 outside the network.

Start the vCloud Application Director CLI. See “Start the CLI Remotely,” on page 36.

Procedure

1. Use the SSH client to copy the downloaded private key file for the Endpoint VM to the vCloud Application Director appliance and copy the file to the /tmp directory.

2. In the root shell, create a secure cloud tunnel instance.

   create-cloud-tunnel --name TunnelName --description "TunnelDescription" --enabled false --
   externalAddress EndpointVMExternalIP
   --sshPort 22 --internalAddress EndpointVMPrivateIP --proxyUrl ProxyURL --username ec2-user --
   privateKeyPath PrivateKeyFilePath

   You can use the --sshPort parameter to designate a port other than 22. The --proxyUrl is an optional parameter that you can specify the proxy server to use to connect to the Endpoint VM.

3. Enable the secure cloud tunnel connection.

   enable-cloud-tunnel --name TunnelName

4. Confirm that the secure cloud tunnel connection is established.

   test-cloud-tunnel --name TunnelName

   Verifying the secure cloud tunnel connection might take a few minutes.

vCloud Application Director establishes a connection to the Endpoint VM on Amazon EC2.

What to do next

Meet the virtual machine requirements and create Amazon EC2 AMIs. See “Virtual Machine Requirements for Creating Amazon EC2 Custom Templates,” on page 86 and “Create Amazon EC2 Virtual Machine Templates or AMIs,” on page 87.

You can also manage the existing cloud tunnel connection in CLI. See “Managing Cloud Tunnels,” on page 207.
Virtual Machine Requirements for Creating Amazon EC2 Custom Templates

You can use supported Linux operating systems to create custom virtual machine templates or Amazon Machine Images (AMIs) for your applications. You can also customize existing AMIs or virtual machine templates and use them in vCloud Application Director.

Table 8-1. Virtual Machine Requirements for Custom AMIs

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating system</td>
<td>CentOS 6.3</td>
</tr>
</tbody>
</table>
| JRE                          | JRE 1.7.0 must be installed. The preferred and supported JRE can be installed from one of these packages, which are available in the vCloud Application Director virtual appliance: JRE for Linux  
  - [Application_Director_IP/agent/jre-7u45-linux-amd64.rpm](http://Application_Director_IP/agent/jre-7u45-linux-amd64.rpm)  
  - [Application_Director_IP/agent/jre-7u45-linux-i586.rpm](http://Application_Director_IP/agent/jre-7u45-linux-i586.rpm)  
  To install JRE on an RPM-based AMI, see “Create Amazon EC2 Virtual Machine Templates or AMIs,” on page 87. |
| Linux agent bootstrap service| Download the Linux agent bootstrap package [vmware-appdirector-agent-service-ec2](http://vmware.com/web/vmware/downloads) from the VMware product download site http://vmware.com/web/vmware/downloads. Install the Linux agent bootstrap script from one of the following packages:  
  - [Application_Director_IP/agent/vmware-appdirector-agent-service-ec2_5.2.0.0-0_x86_64.rpm](http://Application_Director_IP/agent/vmware-appdirector-agent-service-ec2_5.2.0.0-0_x86_64.rpm)  
  - [Application_Director_IP/agent/vmware-appdirector-agent-service-ec2_5.2.0.0-0_i386.rpm](http://Application_Director_IP/agent/vmware-appdirector-agent-service-ec2_5.2.0.0-0_i386.rpm)  
  To install the agent bootstrap service on an RPM-based AMI, see “Create Amazon EC2 Virtual Machine Templates or AMIs,” on page 87. |
| Supported Linux scripting    | vCloud Application Director supports scripting with Bash.                                                                                                                                                    |
| Linux commands               | The following Linux commands must be available on the virtual machine:  
  - `wget`  
  - `md5sum`  
  - `grep`  
  - `sed`  
  - `setsid`  
  - `awk`  
  - `ifconfig`                                                                                                                                 |
| Optional services            | If you plan to remotely access the virtual machine using Linux ssh logging for troubleshooting or for other reasons, the OpenSSH server and client for Linux must be installed and running properly. |

**IMPORTANT** Because the boot process must not be interrupted, configure the virtual machine so that nothing causes the virtual machine’s boot process to pause before reaching the final operating system login prompt. For example, verify that no processes or scripts prompt for user interaction when the virtual machine starts. This requirement applies only to virtual machine templates created for the vCloud Application Director catalog.
Create Amazon EC2 Virtual Machine Templates or AMIs

To use Amazon AMIs in vCloud Application Director you must create an AMI by customizing an existing AMI.

**IMPORTANT** To deploy an application to Amazon EC2, you must install the vCloud Application Director for Release Automation edition.

**Prerequisites**

- Familiarize yourself with the AWS management console and the steps to launch, configure, and terminate an Amazon EC2 instance or virtual machine.
- Verify that you have the proper credentials to access the AWS management console.
- Identify the AWS Region where the vCloud Application Director applications will run and confirm that the AMIs are accessible to the Region.
- Verify that the Amazon EC2 AMI meets the vCloud Application Director virtual machine requirements. See “Virtual Machine Requirements for Creating Amazon EC2 Custom Templates,” on page 86.

For more information about using AMIs, see *AWS Documentation*.

**Procedure**

1. Log in to the AWS console and locate an EBS-backed AMI in the appropriate Amazon Region to deploy applications.
2. Use the AMI to start an Amazon EC2 instance, log in as a root user, and open a terminal.
3. Download and install the supported JRE packages for an RPM-based AMI.
   a. On a machine in the same network as vCloud Application Director, download the JRE package from http://Application_Director_IP/agent/jre-7u45-linux-ArchitectureName.rpm, where the ArchitectureName is i586 for 32-bit and amd64 for 64-bit.
   b. Upload the JRE package to the Amazon EC2 instance.
   c. From the Amazon EC2 instance, type `rpm -i jre-7u45-linux-ArchitectureName.rpm` to install the package.
4. Download and install the vCloud Application Director agent bootstrap service for an RPM-based AMI.
   a. On a machine in the same network as vCloud Application Director, download the agent bootstrap service from http://Application_Director_IP/agent/vmware-appdirector-agent-service-ec2_5.2.0.0-0_ArchitectureName.rpm, where the ArchitectureName is i386 for 32-bit and x86_64 for 64-bit.
   b. Upload the agent bootstrap service to the Amazon EC2 instance.
   c. From the Amazon EC2 instance, type `rpm -i vmware-appdirector-agent-service-ec2_5.2.0.0-0_ArchitectureName.rpm` to install the service.
5. (Optional) Add a second Ethernet device eth1 on the CentOS operating system.
   a. Locate the `ifcfg-eth0` file.
   b. Copy the contents of the file to the `ifcfg-eth1` file.
   c. In the `ifcfg-eth1` file, replace all of the `DEVICE=eth0` strings with `DEVICE=eth1` and save your changes.
The second Ethernet device lets you deploy a virtual machine-based on this AMI with more than one NIC.

6 (Optional) Erase the Linux command history to secure the AMI you will create.
   a Delete the Linux command history saved in the .bash_history file.
      ```
      cat /dev/null > ~/.bash_history
      ```
   b Delete the Linux command history in the current Bash login session.
      ```
      history -c
      ```
7 In the AWS console, stop the Amazon EC2 instance.

**CAUTION** If you terminate the instance, you lose all of your changes.

8 To create an Amazon EBS-backed AMI, select the stopped Amazon EC2 instance, provide the Image Name, and initiate the process to convert to an EBS-backed image.

   The AMI creation might take several minutes.

   After the AMI is available, you can end the Amazon EC2 instance.

**What to do next**

Map the Amazon EC2 Region to a vCloud Application Director cloud provider and provide access to the AMIs or virtual machine templates. See “Register the Amazon EC2 Cloud Provider and Template,” on page 88.

### Register the Amazon EC2 Cloud Provider and Template

In vCloud Application Director, registering a cloud provider means using access keys established in a valid AWS account to connect to an Amazon EC2 Region.

**IMPORTANT** To deploy an application to Amazon EC2, you must install the vCloud Application Director for Release Automation edition.

**Prerequisites**

- Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.
- Verify that an AWS user account is set up with access keys.
- Create virtual machine templates that meet vCloud Application Director requirements. See “Virtual Machine Requirements for Creating Amazon EC2 Custom Templates,” on page 86.

**Procedure**

1 On the vCloud Application Director title bar, click the drop-down menu and select Clouds > Cloud Providers.

2 Click New in the toolbar.

3 Complete the cloud provider information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name and Description</td>
<td>Include detailed information in the Description text box. The text from the text box appears in the Deployment Profile wizard under the Cloud Provider column.</td>
</tr>
<tr>
<td>Cloud Provider Type</td>
<td>Amazon EC2 is the supported type.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Amazon Region</td>
<td>Specify the Region name. For example, if your Region is US East (N. Virginia) the Amazon Region is <strong>us-east-1</strong>.</td>
</tr>
<tr>
<td>Proxy URL</td>
<td>If your corporate network requires a proxy server for external connections, add the Web proxy URL to enable the vCloud Application Director server to connect to Amazon EC2.</td>
</tr>
<tr>
<td>User Name and Password</td>
<td>Access keys set up on the AWS account.</td>
</tr>
</tbody>
</table>

4. To test whether the values you entered are correct, click **Validate Connection**.

5. Click **Save** to register the cloud provider.

   If any of the required entries for the cloud provider are invalid or left blank, you are prompted to correct them when you click **Save**.

6. To register Amazon AMIs, click **Edit** in the toolbar.

7. In the Templates section, click **New** to continue and register AMIs or virtual machine templates.

8. Select a category from the drop-down menu and click **Get Templates**.

   You can select AMIs owned by the AWS account or those that are private to the AWS account.

   Filtered AMIs or cloud templates that you created or that are private to the AWS account appear in the drop-down menu.

9. Select the check boxes next to one or more templates in the list, and click **OK**.

   The templates you registered are added to the list of cloud templates that you can choose from when you map a logical template to a cloud template.

10. To change existing information for an Amazon EC2 cloud provider, click **Edit** in the toolbar, make your changes, and click **Save**.

    The Amazon EC2 Region is registered and its AMIs are available for use in vCloud Application Director.

**What to do next**

Create a deployment environment to deploy an application to the cloud environment. See “Create an Amazon EC2 Deployment Environment,” on page 89.

Map these AMIs to the vCloud Application Director logical templates. See “Add a Logical Template to the Catalog,” on page 121.

## Create an Amazon EC2 Deployment Environment

You must map a deployment environment to an Amazon EC2 VPC and associated Availability Zone before you can deploy an application to the cloud.

With a VPC, you can provision a logically isolated area of the AWS Cloud where you can deploy applications from vCloud Application Director.

If a deployment environment is currently being used in a deployment profile or a current deployment, it cannot be deleted.

**IMPORTANT** To deploy an application to Amazon EC2, you must install the vCloud Application Director for Release Automation edition.

**Prerequisites**

- Verify that your user account has the **ROLE_CLOUD_ADMIN** cloud administrator role assigned to it.
- Verify that the Amazon EC2 environment is configured to establish a secure connection with the vCloud Application Director appliance. See “Configure Amazon EC2 Environment for vCloud Application Director,” on page 82.

- Verify that at least one Amazon cloud provider is registered in vCloud Application Director. See “Register the Amazon EC2 Cloud Provider and Template,” on page 88.

- Verify that a secure cloud tunnel connection is available.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Clouds > Deployment Environments**.
2. Click **New** in the toolbar.
3. Complete the deployment environment information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment Environment and Description</strong></td>
<td>Include the name of the VPC and the Availability Zone name in either of these fields. The text from these fields appears in the Deployment Profile wizard under the Deployment Environment column.</td>
</tr>
<tr>
<td><strong>Cloud Provider</strong></td>
<td>If the cloud provider does not appear in the list, cancel the dialog box and select <strong>Clouds &gt; Cloud Providers</strong> to add the Amazon EC2 cloud provider.</td>
</tr>
<tr>
<td><strong>Virtual Private Cloud</strong></td>
<td>Click <strong>Select</strong> to view a list of available Amazon VPCs in the Region.</td>
</tr>
<tr>
<td><strong>Availability Zone</strong></td>
<td>Click <strong>Select</strong> to view a list of Availability Zones in the Amazon Region.</td>
</tr>
<tr>
<td><strong>Cloud Tunnel</strong></td>
<td>Select the cloud tunnel instance that connects the vCloud Application Director server to an Endpoint VM. This Endpoint VM must reside in the Amazon EC2 VPC already set in the Virtual Private Cloud section for this deployment environment. Click <strong>Validate</strong> to check the secure SSH connection status of the cloud tunnel.</td>
</tr>
</tbody>
</table>

4. Click **Save**.

5. To change existing information for an Amazon EC2 deployment environment, click **Edit** in the toolbar, make your changes, and click **Save**.

The deployment environment is added to the list of deployment environments that you can select from when you create a deployment profile.

**What to do next**

You can map an external service instance, create a policy instance, or register a solution instance with the deployment environment. See “Map an External Service Instance,” on page 61, “Create a Policy Instance,” on page 61 or “Create a Solution Instance,” on page 64.
vCloud Application Director provides an open framework to create and develop components that can be installed in a virtual machine.

The vCloud Application Director components in the catalog include predefined sample services such as JBoss and MySQL, predefined tasks such as scripts to configure the APT repository, and blueprint application components such as WAR and SQL_SCRIPT.

In this information, components include services, application components, and custom tasks. All of these components use actions and properties as their common underlying framework.

**Note** Only use the sample components in the vCloud Application Director catalog in a test environment.

Familiarize yourself with the key concepts that appear frequently in topics about developing deployable components. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Defining Component Actions,” on page 91
- “Configuring Component Properties,” on page 93
- “vCloud Application Director Predefined Properties,” on page 100
- “Secured Component Properties,” on page 101
- “Required Component Properties,” on page 101
- “Best Practices for Developing Components,” on page 101

**Defining Component Actions**

Each component includes the predefined life cycle stages or actions to install, configure, start, update, rollback, and teardown scripts for a service or application component.

The catalog administrator must provide a Bash, Windows CMD, PowerShell, or BeanShell script for at least one of the INSTALL, CONFIGURE, START, UPDATE, ROLLBACK, or TEARDOWN life cycle stages. These scripts are customized to use the component properties.

For example, to deploy an Apache Tomcat server in a virtual machine, you might add the following scripts:

**INSTALL**

Download the Tomcat server installation bits and install the Tomcat service.

**CONFIGURE**

Set the JAVA_OPTS, CATALINA_OPTS, and any other required configuration.

**START**

Start the Tomcat service using the start command in the Tomcat server.
UPDATE
Modify the configuration of Tomcat service using the update script or change the cluster size to scale a deployed application and manage the clustered nodes using a load balancer.

ROLLBACK
Modify the rollback script if the update process of the Tomcat service is unsuccessful because of performance or security problems or the update process is successful but the deployed application has errors.

TEARDOWN
Modify the teardown script of the Tomcat service for example, to perform specific actions in the application before a deployment is torn down.

The application architect can parameterize the script by declaring, for example, the installer location, installation path, and Tomcat start command as properties in the script. The parameters render the scripts generic. You can deploy the service on different environments without modifying these generic scripts.

You can also modify parameter values from the action script. These modified properties can be referred to as property values for other components. See VMware vCloud Application Director Catalog Services guide.

The scripts defined for an action are run in the /tmp/ directory. The Linux script is located at /tmp/runId/ComponentName-LifecycleStageName. The runId is the unique job identifier for each deployment, which is available on the Task Details status window of the deployment summary page. The Windows script is located at \Users\darwin\AppData\Local\Temp.

NOTE Verify that no processes are prompting for user interaction when the action script is running. Interruptions pause the script, causing it to remain in an idle state indefinitely, and eventually fail. In addition, if a Windows CMD script exits with a non-zero exit status, the vCloud Application Director agent stops the deployment and marks it as Failed Deployment. Use exit /b 0 to indicate success status and exit /b non-zero for error status.

See “Supported Action and Custom Task Scripts,” on page 92.

You can add a service or custom task in an application deployment and define the component action. See “Add a Service to the Catalog,” on page 105 and “Add a Custom Task to the Catalog,” on page 129.

Supported Action and Custom Task Scripts
vCloud Application Director supports authoring in Bash or BeanShell script for a Linux-based application and authoring in Windows CMD, PowerShell, or BeanShell scripts for a Windows-based application.

To select the appropriate action script type for a life cycle stage, in the blueprint canvas, double-click the Script Type column and select the script from the drop-down menu. For a custom task script, select one or more operating systems. You can set supported scripts from the drop-down menu.

Depending on the script type you select, you can author code and access the relevant properties through the variables in the script. For an action script of a life cycle stage, you can also use different script types in the same operating system family for each life cycle in the same component. For example, you can use the Windows CMD script for the INSTALL stage and a PowerShell script for the CONFIGURE stage. See “Types of Properties,” on page 93.

When you author an action script, the exit and return codes vary between script types. The application architect should set proper exit codes in the script that are applicable to the application deployment. If the script lacks exit and return codes, the last command that ran in the script becomes the exit status. See “Understanding the Deployment and Update Process,” on page 159.
Table 9-1. Action Script Exit and Return Codes

<table>
<thead>
<tr>
<th>Script Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bash</td>
<td>You can use <code>return 0</code> or <code>exit 0</code> codes in action scripts to indicate success status. To indicate error status, you can use <code>return non-zero</code> or <code>exit non-zero</code>.</td>
</tr>
<tr>
<td>Windows CMD</td>
<td>Do not use <code>exit 0</code> and <code>exit non-zero</code> codes in the action script. If you use these codes in the script, the computed properties task processing is stopped prematurely. Use <code>exit /b 0</code> to indicate success status and <code>exit /b non-zero</code> for error status.</td>
</tr>
<tr>
<td>Windows PowerShell</td>
<td>You can use <code>exit 0</code> to indicate success status and <code>exit non-zero</code> for error status.</td>
</tr>
<tr>
<td>BeanShell</td>
<td>You can use <code>System.exit(0);</code> to indicate success status and <code>System.exit(1);/non-zero</code> for error status.</td>
</tr>
</tbody>
</table>

When you use Windows PowerShell to author a script, you cannot use the `warning`, `verbose`, `debug`, and `host` calls in an action script.

**Configuring Component Properties**

Component properties are used to parameterize scripts so that vCloud Application Director can pass the defined properties as environment variables to scripts running in a virtual machine.

Before running a script from the life cycle stage, the vCloud Application Director agent in the virtual machine communicates with the vCloud Application Director server to resolve the properties. The agent then proceeds to create script-specific variables from these properties and passes them to the scripts.

- **Types of Properties** on page 93
  vCloud Application Director supports string, array, content, and computed properties.

- **Defining Property Values** on page 97
  A catalog administrator can define properties in the catalog for services and custom tasks.

- **Binding to Other Properties** on page 97
  In several deployment scenarios, a component needs the property value of another component to customize itself. In vCloud Application Director, this process is called binding to other properties.

- **Auto-Binding to Other Properties** on page 98
  You can Auto-bind to other properties by using the Auto-Bind meta data setup in a service and default meta data in nodes, node arrays, services, and the WAR application component.

- **Passing Property Values Between Life Cycle Stages and Components** on page 99
  Properties defined for each component can be used in an action script for each of the life cycle stages.

**Types of Properties**

vCloud Application Director supports string, array, content, and computed properties.

*Note* The names of properties are case-sensitive and can contain only alphabetic, numeric, hyphen (-), or underscore (_) characters.

**String Property**

The string property value can be a string or the value bound to another string property. A string value can contain any ASCII characters. For a bound property, use the Properties tab in the blueprint canvas to select the appropriate property for binding. The property value is then passed to the action scripts as raw string data.
**Sample String Property**

<table>
<thead>
<tr>
<th>Property</th>
<th>Bash - $admin_email</th>
<th>Sample Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin_email = &quot;<a href="mailto:admin@email987.com">admin@email987.com</a>&quot;</td>
<td>echo $admin_email</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Windows CMD - %admin_email%</td>
<td>echo %admin_email%</td>
</tr>
<tr>
<td></td>
<td>Windows PowerShell - $admin_email</td>
<td>write-output $admin_email</td>
</tr>
<tr>
<td></td>
<td>BeanShell - admin_email</td>
<td>print(admin_email);</td>
</tr>
</tbody>
</table>

**Array Property**

The array property value can be an array of strings defined as ["value1", "value2", "value3"...] or the value bound to another array property. When you define values for an array property you must enclose the array of strings in square brackets. For an array of strings, the value in the array elements can contain any ASCII characters. To properly encode a backslash character in an Array property value, add an extra backslash, for example, ["c:\test1\test2"]. For a bound property, use the Properties tab in the blueprint canvas to select the appropriate property for binding.

For example, consider a load balancer virtual machine that is balancing the load for a cluster of application server virtual machines. In such a case, an array property is defined for the load balancer service and set to the array of IP addresses of the application server virtual machines.

These load balancer service configure scripts use the array property to configure the appropriate load balancing scheme on the Red Hat, Windows, and Ubuntu operating systems.

<table>
<thead>
<tr>
<th>Sample Array Property</th>
<th>Script Syntax</th>
<th>Sample Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>operating_systems = [&quot;Red Hat&quot;,&quot;Windows&quot;,&quot;Ubuntu&quot;]</td>
<td>Bash -${operating_systems[0]} for the entire array of strings ${operating_systems[N]} for the individual array element</td>
<td>for (( i = 0 ; i &lt; ${#operating_systems[0]}; i++ )); do echo ${operating_systems[$i]} done</td>
</tr>
<tr>
<td></td>
<td>Windows CMD - %operating_systems_%N% where N represents the position of the element in the array</td>
<td>for /F &quot;delims== tokens=2&quot; %%A in (&quot;set operating_systems_&quot;) do ( echo %%A )</td>
</tr>
<tr>
<td></td>
<td>Windows PowerShell - $operating_systems for the entire array of strings $operating_systems[N] for the individual array element</td>
<td>foreach ($os in $operating_systems){ write-output $os }</td>
</tr>
<tr>
<td></td>
<td>BeanShell - $operating_systems[N] where N represents the position of the element in the array</td>
<td>for(index=0;index &lt; operating_systems.length; index++) { print(operating_systems[index]); }</td>
</tr>
</tbody>
</table>

**Content Property**

The content property value is a URL to a file to download content. vCloud Application Director agent downloads the content from the URL to the virtual machine and passes the location of the local file in the virtual machine to the script.

Content properties must be defined as a valid URL with the HTTP or HTTPS protocol. For example, the sample Hyperic HQ agent has a property value HQ_PACKAGE_32 set to http://DarwinServerIP/artifacts/services/hyperic/hyperic-hq-agent-4.6-x86-linux.tar.gz. The Hyperic artifacts are hosted in the vCloud Application Director appliance and the URL points to that location in the appliance. The vCloud Application Director agent downloads the artifacts from the specified location into the deployed virtual machine.
<table>
<thead>
<tr>
<th>Sample String Property</th>
<th>Script Syntax</th>
<th>Sample Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>HQ_PACKAGE = &quot;<a href="http://DarwinServerIP/hyperic/hyperic-hq-agent-linux.tar.gz">http://DarwinServerIP/hyperic/hyperic-hq-agent-linux.tar.gz</a>&quot;</td>
<td>Bash -</td>
<td></td>
</tr>
<tr>
<td>$HQ_PACKAGE</td>
<td>tar -zxvf $HQ_PACKAGE</td>
<td></td>
</tr>
<tr>
<td>Windows CMD</td>
<td>start /wait c:\unzip.exe</td>
<td></td>
</tr>
<tr>
<td>-%HQ_PACKAGE%</td>
<td>%HQ_PACKAGE%</td>
<td></td>
</tr>
<tr>
<td>Windows PowerShell</td>
<td>&amp; c:\unzip.exe</td>
<td></td>
</tr>
<tr>
<td>$HQ_PACKAGE</td>
<td>$HQ_PACKAGE</td>
<td></td>
</tr>
</tbody>
</table>
| BeanShell -                                                                               | import java.io.BufferedReader;
| $HQ_PACKAGE                                                                             | import java.io.File;
|                                                                                         | import java.io.FileInputStream;
|                                                                                         | import java.io.FileOutputStream;
|                                                                                         | import java.util.zip.ZipEntry;
|                                                                                         | import java.util.zip.ZipInputStream;
|                                                                                         | destDir = new File(bsh.cwd);
|                                                                                         | if (!destDir.exists()) {|
|                                                                                         |   destDir.mkdir();
|                                                                                         | }|
|                                                                                         | zipIn = new ZipInputStream(new FileInputStream(HQ_PACKAGE));
|                                                                                         | entry = zipIn.getNextEntry();
|                                                                                         | // iterates over entries in the zip file
|                                                                                         | while (entry != null) {|
|                                                                                         |   String filePath = bsh.cwd + File.separator + entry.getName();
|                                                                                         |   if (!entry.isDirectory()) {|
|                                                                                         |     // if the entry is a file, extracts it
|                                                                                         |     bos = new BufferedReader(new FileInputStream(entry));
|                                                                                         |     bytesIn = new byte[4096];
|                                                                                         |     read = 0;
|                                                                                         |     while ((read = zipIn.read(bytesIn)) != -1) {
|                                                                                         |       bos.write(bytesIn, 0, read);
|                                                                                         |     }|
|                                                                                         |     bos.close();
|                                                                                         | } else {|
|                                                                                         |   // if the entry is a directory, make the directory
|                                                                                         |     dir = new File(filePath);
|                                                                                         |     dir.mkdir();
|                                                                                         | }|
|                                                                                         | zipIn.closeEntry();
|                                                                                         | entry = zipIn.getNextEntry();
|                                                                                         | }|
|                                                                                         | zipIn.close();          |
Computed Property

The computed property type does not allow an initial value when it is being defined. Instead, the computed property type takes the value from the INSTALL, CONFIGURE, or START life cycle scripts. The assigned value is propagated to the subsequent available life cycle and dependent components.

**NOTE** Computed properties that are defined in the action scripts are not available to the UPDATE, ROLLBACK, or TEARDOWN life cycle script. You must set the value for the computed property in the respective life cycle scripts.

### Sample String Property

<table>
<thead>
<tr>
<th>Script Syntax</th>
<th>Sample Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bash - $my_unique_id</td>
<td>export my_unique_id=&quot;0123456789&quot;</td>
</tr>
<tr>
<td>Windows CMD - %my_unique_id%</td>
<td>set my_unique_id=0123456789</td>
</tr>
<tr>
<td>Windows PowerShell - $my_unique_id</td>
<td>$my_unique_id = &quot;0123456789&quot;</td>
</tr>
<tr>
<td>BeanShell - my_unique_id</td>
<td>my_unique_id = &quot;0123456789&quot;</td>
</tr>
</tbody>
</table>

### Property Type Reference

Property type references show which property types you can refer to for configuration purposes.

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Type to Bind</th>
<th>Auto-Bind Expose</th>
<th>Auto-Bind Consume</th>
<th>Accept Initial Property Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>String, Array, Content, and Computed</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Content</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Yes</td>
</tr>
<tr>
<td>Array</td>
<td>String, Content, and Computed</td>
<td>N/A</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Computed</td>
<td>N/A</td>
<td>Yes</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Array properties can also bind to a predefined node array property all in a cluster. The all property, in a regular expression, is a method to collect all of the values for a given property in a clustered node.

For more information about binding properties, see “Binding to Other Properties,” on page 97. To understand the Auto-Bind expose and Auto-Bind consume concepts, see “Auto-Binding to Other Properties,” on page 98.

Example of a string property value when binding to different types of properties.

<table>
<thead>
<tr>
<th>Sample Property Type</th>
<th>Property Type to Bind</th>
<th>Binding Outcome (A binds to B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>String (property A)</td>
<td>String (property B=&quot;Hi&quot;)</td>
<td>A=&quot;Hi&quot;</td>
</tr>
<tr>
<td>String (property A)</td>
<td>Content (property B=&quot;<a href="http://my.com/content">http://my.com/content</a>&quot;)</td>
<td>A=&quot;<a href="http://my.com/content">http://my.com/content</a>&quot;</td>
</tr>
<tr>
<td>String (property A)</td>
<td>Array (property B=[&quot;1&quot;,&quot;2&quot;])</td>
<td>A=[&quot;1&quot;,&quot;2&quot;]</td>
</tr>
<tr>
<td>String (property A)</td>
<td>Computed (property B=&quot;Hello&quot;)</td>
<td>A=&quot;Hello&quot;</td>
</tr>
</tbody>
</table>

Example of an array property value when binding to different types of properties.

<table>
<thead>
<tr>
<th>Sample Property Type</th>
<th>Property Type to Bind</th>
<th>Binding Outcome (A binds to B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array (property A)</td>
<td>String (property B=&quot;Hi&quot;)</td>
<td>A=&quot;Hi&quot;</td>
</tr>
<tr>
<td>Array (property A)</td>
<td>Content (property B=&quot;<a href="http://my.com/content">http://my.com/content</a>&quot;)</td>
<td>A=&quot;<a href="http://my.com/content">http://my.com/content</a>&quot;</td>
</tr>
<tr>
<td>Array (property A)</td>
<td>Computed (property B=&quot;Hello&quot;)</td>
<td>A=&quot;Hello&quot;</td>
</tr>
</tbody>
</table>
Defining Property Values

A catalog administrator can define properties in the catalog for services and custom tasks.

To allow for customization of a component script, the default value for a property in the catalog can be overridden in a blueprint to accommodate the needs of a specific application in which the service is used. The property value can be further overridden in the deployment profile to comply with the deployment environment where the service is deployed.

**Note** Property values are case-sensitive. A new property value does not take effect if the value is typed incorrectly.

For example, to deploy an Apache Tomcat server, the catalog administrator might have configured the value of the JVM HEAP_SIZE to 512MB in the service. The application architect can redefine the value to 1024MB for a larger application. The deployer might override the value to 2048MB when deploying the application in a production deployment environment to handle large loads.

Property values are non-overrideable by default, but the catalog administrator can choose to enable the overrideable feature. For properties that can be overridden, the application architect might force a value for the property in some regulated environments, and disable the overrideable feature for the deployer.

You can define property values when you add a service or custom task in an application deployment. See “Add a Service to the Catalog,” on page 105 and “Add a Custom Task to the Catalog,” on page 129.

Binding to Other Properties

In several deployment scenarios, a component needs the property value of another component to customize itself. In vCloud Application Director, this process is called binding to other properties.

The catalog administrator can modify property definitions in the script. For example, a WAR component might need the installation location of the Apache Tomcat server. The WAR component can set the server_home property value to the Apache Tomcat server install_path property value.

The Bash script for a component can use only its own properties. In addition to setting a property to a hard-coded value, vCloud Application Director allows a property to be bound to another property in the blueprint. When you bind to another property, you can customize a script based on the value of another component's properties and virtual machine properties such as IP addresses. To bind a property to another property, select the property value from the **Blueprint Value** drop-down menu in the Edit Property dialog box.

For a single virtual machine node, the referenced properties in the **Blueprint Value** drop-down menu are, **NodeName:ComponentName:PropertyName**.

For clustered virtual machines node, the referenced properties in the **Blueprint Value** drop-down menu are, all(**NodeName:ComponentName:PropertyName**). When another property refers to this cluster property, it gets the **PropertyName** property values from all of the virtual machines in the cluster. The predefined all(**NodeName:node_array_index**) property for clustered virtual machines gets the collection of node array indexes in the cluster. See “Predefined Node Array Index Property,” on page 100.

For cluster and single nodes, the self:**ComponentName:PropertyName** value is used to indicate a component property from the virtual machine where the target component is running. For instance, if a WAR component is deployed in an Apache Tomcat server, the WAR server_home property can be set to self:tomcat:install_path to refer to the Apache Tomcat server running in the current virtual machine.

Node level properties such as IP address display as, **NodeName:ip** or self:**ip**. These properties belong exclusively to the virtual machine and not to any specific component in the virtual machine.

For IP addresses of virtual machines with multiple NICs, vCloud Application Director provides either a **NodeName:NICx_ip** or self:**NICx_ip** property. Where x reflects the NIC number. See “Predefined IP Address Property,” on page 100.
Auto-Binding to Other Properties

You can Auto-bind to other properties by using the Auto-Bind meta data setup in a service and default meta
data in nodes, node arrays, services, and the WAR application component.

Based on the Auto-Bind meta data defined in a service by the catalog administrator, vCloud Application Director displays the Auto-Bind type as consume or expose to allow other properties to bind to it in the blueprint. In the blueprint canvas, an Expose icon appears next to the Auto-Bind Expose properties and a Consume icon appears next to the Auto-Bind Consume properties. The tooltips for the icons display the Auto-Bind type and Auto-Bind tag information. For example, the Consume icon next to the webapps_dir property shows the Auto-Bind Consume: Servlet Container, DIR information in the tooltip. The catalog administrator sets the service Auto-Bind type and tag parameters when creating a service or editing an existing service. See “Add a Service to the Catalog,” on page 105.

Preconfigured Components for Automatic Binding

WAR is the only application component that is preconfigured for auto-binding. You cannot edit the predefined Auto-Bind type and tag parameters.

Adding an application component onto a service implies that properties can be set on either. In this case, auto-binding happens twice. The source properties on the application component are set from target properties on the service and the parent node or node array. Then, the source properties on the service are set from target properties on the application component.

<table>
<thead>
<tr>
<th>Application Component</th>
<th>Property</th>
<th>Auto-Bind Type</th>
<th>Auto-Bind Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux WAR file</td>
<td>Context</td>
<td>Expose</td>
<td>Context</td>
</tr>
<tr>
<td>Linux WAR file</td>
<td>service_start</td>
<td>Consume</td>
<td>Servlet Container, Start</td>
</tr>
<tr>
<td>Linux WAR file</td>
<td>service_stop</td>
<td>Consume</td>
<td>Servlet Container, Stop</td>
</tr>
<tr>
<td>Linux WAR file</td>
<td>webapps_dir</td>
<td>Consume</td>
<td>Servlet Container, DIR</td>
</tr>
</tbody>
</table>
Some nodes and node arrays have predefined Auto-Bind type and tag parameters for properties. You cannot edit these parameters. When a service or application component is added to a node or node array, the source properties are taken from the service or application component and the target properties are taken from the node or node array.

Table 9-3. Predefined Node and Node Array Properties

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Property</th>
<th>Auto-Bind Type</th>
<th>Auto-Bind Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node, Node Array</td>
<td>hostname</td>
<td>Expose</td>
<td>Hostname</td>
</tr>
<tr>
<td>Node, Node Array</td>
<td>ip</td>
<td>Expose</td>
<td>IP Address</td>
</tr>
<tr>
<td>Node Array</td>
<td>node-array-index</td>
<td>Expose</td>
<td>Node Array Index</td>
</tr>
</tbody>
</table>

When dependencies exist, the source properties are taken from the component that the link originates. Target properties are taken from the component from which the link is drawn, and from parent service or node components.

**Property Compatibility**

Auto-binding depends on the compatibility between Source and Target property types and Node type.

Table 9-4. Property Type and Node Type Compatibility

<table>
<thead>
<tr>
<th>Source Property Type</th>
<th>Target Property Type</th>
<th>Node Type</th>
<th>Compatible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>String</td>
<td>Node</td>
<td>Yes</td>
</tr>
<tr>
<td>Array</td>
<td>Computed</td>
<td>Node</td>
<td>Yes</td>
</tr>
<tr>
<td>Array</td>
<td>String</td>
<td>Node Array</td>
<td>Yes</td>
</tr>
<tr>
<td>Array</td>
<td>Computed</td>
<td>Node Array</td>
<td>Yes</td>
</tr>
<tr>
<td>String</td>
<td>String</td>
<td>Node</td>
<td>Yes</td>
</tr>
<tr>
<td>String</td>
<td>Computed</td>
<td>Node</td>
<td>Yes</td>
</tr>
<tr>
<td>String</td>
<td>String</td>
<td>Node Array</td>
<td>No</td>
</tr>
<tr>
<td>String</td>
<td>Computed</td>
<td>Node Array</td>
<td>No</td>
</tr>
</tbody>
</table>

**Passing Property Values Between Life Cycle Stages and Components**

Properties defined for each component can be used in an action script for each of the life cycle stages.

For a computed property, you can modify the value of a property and pass the value to the next life cycle stage of the action script. For example, if component A has the progress_status value defined as staged, in the INSTALL and CONFIGURE life cycle stage you change the value to progress_status=installed in the respective action scripts. If component B is bound to component A, the property values of progress_status in the life cycle stages of the action script are the same as component A.

Define in the blueprint that component B depends on A. This dependency defines the passing of correct property values between components whether they are in the same node or across different nodes.

For example, you can update a property value in an action script by using the supported scripts.

- **Bash**
  
  ```bash
  progress_status="completed"
  ```

- **Windows CMD**

  ```cmd
  set progress_status=completed
  ```

- **Windows PowerShell**

  ```powershell
  $progress_status="completed"
  ```
vCloud Application Director Predefined Properties

vCloud Application Director provides some commonly used properties as predefined properties. These properties are available for clustered nodes.

Usually, most services need the IP address of the virtual machines they are running in. For clustered nodes, a service might need the IP addresses of all the virtual machines in the cluster. Therefore, the virtual machine IP address is designated as a predefined property.

Predefined IP Address Property

A node can have multiple NICs, with each NIC assigned one IP address in the deployed virtual machine. These IP addresses are available in the `NodeName:NICx_ip` properties, where `-x-` is the NIC number.

In the sample Clustered Duke's Bank application, the Load Balancer node properties are shown as `load_balancer:NIC0_ip` and `load_balancer:NIC1_ip` because the Load Balancer node has two NICs defined. It is not guaranteed that NIC0 and NIC1 will be assigned to eth0 and eth1, respectively, in the virtual machine. The NICs are logical names in the blueprint for the network interfaces. These NICs are mapped to logical networks, which are mapped to specific cloud networks. The property NIC0_ip returns the IP address assigned to the virtual machine as defined in the blueprint, not the eth0 IP address in the virtual machine.

The `load_balancer:ip` property is also provided. This property refers to the IP address of the first NIC and NIC0. If a component needs IP addresses of all virtual machines in a cluster, it uses the `all(NodeName:NICx_ip)` property. To refer to the IP address of the current virtual machine, `self:ip` property is available. This property is useful for clustered nodes, as a component might need to know the virtual machine it is in, rather than the IP addresses of all the virtual machines in the cluster. To get the properties for a specific virtual machine in a cluster, you can use the `all(NodeName:node_array_index)` property. See “Predefined Node Array Index Property,” on page 100.

Predefined Node Array Index Property

For clustered nodes, a special property called `node_array_index` identifies the position of the current virtual machine in the cluster.

The `node_array_index` property value of the first virtual machine in the cluster is 0, the `node_array_index` property value of the second virtual machine in the cluster is 1, and so on.

For example, in a deployment with a clustered node, if the first virtual machine is the master virtual machine, then it manages other virtual machines in the cluster as subordinate machines. This master virtual machine must be configured differently. For example, if one of the properties for the component is `myPosition`, then it must be bound to the predefined `self:node_array_index` property. The component script can verify whether the value of `myPosition` property is 0, and if so it can perform an additional configuration.

For clustered nodes, the predefined node array property is `all(NodeName:node_array_index)`. The property gives the indexes of all the virtual machines in the cluster and can be used to identify the size of the cluster.
Predefined Disk Layout Info Property

Flexible Disks enhances storage flexibility and lets you add additional disks to a node.

To use additional disk information, you must create bindable properties in the service definition and bind to the disk_layout_info property. The disk_layout_info property contains information of all the disks added to the application blueprint.

For example, in any application blueprint, the disk information is captured by the blueprint as Disk0, Disk1, Disk2, and so on.

You can get the following information:
- name
- sizeInGb
- metaTags
- seq
- mountPoint
- fileSystem

Secured Component Properties

Properties are used to configure deployed components. In some cases, they are used to store sensitive data, such as passwords.

For example, a WAR component might need to store sensitive data such as passwords to access the database. These properties can be marked as secured. Values of secured properties are masked and shown as asterisks in vCloud Application Director.

If a property is changed from secured to unsecured, for security purposes, vCloud Application Director resets the property value, for security purposes. You must set a new value for the property.

IMPORTANT If secured properties are printed in the script using the echo command or other similar commands, these values appear in plain text in the log files. The values in the log files are not masked.

Required Component Properties

You can set properties as they are required. For example, a deployment might fail if properties are not defined for scripts that rely on them during the deployment.

For example, to run an Apache Tomcat server, Java is required and the JAVA_HOME property value must be set.

When a property is marked as required, a value must be provided in at least one of the life cycle stages of the property, such as the catalog, blueprint, or deployment profile. For example, a catalog administrator can define a required property, mark it as overridable, and not set any value for the property in the catalog. The application architect must provide a value for this property in the blueprint or mark it as overridable in the deployment profile. If the application architect has not set a value for this property, the deployer is required to set a value for this property in the deployment profile before deploying the application.

Best Practices for Developing Components

Familiarize yourself with the sample components in the catalog, as they include a number of examples of how to define properties and action scripts.

NOTE Use the sample components in the vCloud Application Director catalog only in a test environment.
Follow these best practices when developing components in vCloud Application Director.

- Some installers might need access to the tty console. Redirect the input from `/dev/console`.
  
  For example, the predefined RabbitMQ service uses the `./rabbitmq_rhel.py --setup-rabbitmq < /dev/console` command in its install script.

- When a component uses multiple life cycle stages, the property value can be changed in the INSTALL life cycle stage. The new value is sent to the next life cycle stage. Action scripts can compute the value of a property during deployment to supply the value to other dependent scripts.

**Note** You cannot change the content property value for a component that uses multiple life cycle stages.

For example, in the Clustered Duke's Bank sample application, JBossAppServer service computes the JVM_ROUTE property during the install life cycle stage. This property is used by the JBossAppServer service to configure the life cycle. Apache load balancer service then binds its JVM_ROUTE property to the `all(appserver:JbossAppServer:JVM_ROUTE)` property to get the final computed value of node0 and node1.

If a component requires a property value from another component to complete an application deployment successfully, you must state explicit dependencies in the application blueprint.

- For a script to run without any interruptions, the return value must be set to zero (0).
  
  This setting allows the agent to capture all of the properties and send them to the vCloud Application Director server.
With the prepopulated, extensible catalog of standard logical templates, sample services, task scripts, operating systems, external services, policies, and tags, application architects can quickly create a blueprint of a multitier enterprise application. A cloud administrator can enable predefined policy definitions on deployment environments for a deployer to apply to the multitier application during deployment.

**IMPORTANT** vCloud Application Director includes many predefined logical templates, services, and external services. A catalog administrator can use or modify these predefined templates and services to avoid having to create them themselves. A best practice is to create a copy of the service to preserve the original for future reference.

The predefined services, tasks, operating systems, and tags in the catalog are available to all user groups in vCloud Application Director. You must add logical templates for each group outside the Default group.

You can use the vCloud Application Director catalog to create custom tasks, which are customized scripts that you can add to the execution plan in a deployment profile.

Familiarize yourself with the key concepts relating to managing the catalog. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Add Operating Systems to the Catalog,” on page 104
- “Add Tags to the Catalog,” on page 104
- “Add a Service to the Catalog,” on page 105
- “Import a Puppet Service to the Catalog,” on page 112
- “Add an External Service to the Catalog,” on page 113
- “Add a Logical Template to the Catalog,” on page 121
- “Add a Policy to the Catalog,” on page 124
- “Add a Custom Task to the Catalog,” on page 129
Add Operating Systems to the Catalog

You might have to add operating systems to the list of preinstalled operating systems that vCloud Application Director includes.

You can create a definition of an operating system in the catalog and specify which operating system a logical template uses or which operating systems are supported for a particular service. vCloud Application Director supports Linux and Windows operating systems.

NOTE: You cannot delete an operating system from the catalog.

Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Familiarize yourself with the preinstalled operating systems on the Operating Systems page.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Operating Systems.

2. Click New in the toolbar.

3. Set the operating system name and description.
   As a best practice, use the OS name and version number, and add the bit information in the description.

4. Select the architecture from the drop-down menu.

5. Select the operating system family from the drop-down menu and click Save.
   When you create an action script for a service, the script type depends on the designated operating system family. You use a Bash or BeanShell script for a Linux-based application and a Windows CMD, PowerShell, or BeanShell script for a Windows-based application.

6. Click Create OS Version to create an operating system version.
   You can create multiple versions for an OS.

7. Complete the OS version information and click Save.
   The OS name remains the same. You can specify major and minor versions and add service pack details for the OS.

The new operating system is added to the Operating Systems page. You can select the operating system when you create other components to add to the catalog or to an application blueprint.

Add Tags to the Catalog

When you create a logical template, service, or external service, you can associate one or more tags with the component.

NOTE: You cannot delete a tag from the catalog.
Tag types are not interchangeable. Changing the tag type for example, from Server Type to Property Descriptor might cause rendering issues in the blueprint canvas or generate error messages when you attempt to use a service.

**Prerequisites**

- Verify that your user account has the **ROLE_CATALOG_ADMIN** catalog administrator role assigned to it.
- Familiarize yourself with the preinstalled tags listed on the Tags page.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Catalog > Tags**.
2. Click **New** in the toolbar.
3. Type the tag name and description in the text box.
4. Select a tag type from the drop-down menu.
   - The Property Descriptor tag type appears as an option in the **Auto-Bind Tags** drop-down menu for a service property. The Server Type tag appears as an option in the drop-down menu when you create a service or logical template.
5. Click **Save** when you are finished.

The newly created tag is listed on the Tags page. You can select the tag when you add a service, task, or logical template to the catalog or to an application blueprint.

**Add a Service to the Catalog**

A service comprises scripts for installing, configuring, starting, and updating the software services that your application requires. You can add custom services to the vCloud Application Director catalog.

**Procedure**

1. **Create a Service Version in the Catalog** on page 106
   - Use the vCloud Application Director catalog to create software service versions. The vCloud Application Director catalog also contains predefined services.
2. **Define Service Version Properties** on page 107
   - vCloud Application Director passes defined properties as environment variables to scripts running in a virtual machine. The service version properties define the variables used in the scripts for the service.
3. **Add Action Scripts to the Service Version** on page 109
   - The catalog administrator must provide an action script for at least one of the life cycle stages. For deploying an application, you can create an install, configure, start, rollback, and teardown script, or create an update script to update an existing deployment. These scripts are customized to use the component properties.
4. **Maintain Service Versions** on page 111
   - You can copy or edit a service from an existing service version.
5. **Delete a Service from the Catalog** on page 111
   - You can delete a service from the Services page.
Create a Service Version in the Catalog

Use the vCloud Application Director catalog to create software service versions. The vCloud Application Director catalog also contains predefined services.

Sometimes, rather than creating a new service, you might prefer to edit the scripts and variables in the predefined service. As a best practice, create a copy of the predefined service before you make changes. See “Maintain Service Versions,” on page 111.

You can share a service publicly or privately in a group. Sharing is supported at the component level and not at the individual version level.

Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Verify whether you must add items to the operating systems and descriptive tag lists. See “Add Operating Systems to the Catalog,” on page 104 and “Add Tags to the Catalog,” on page 104.
- If you plan to add an operating system with SELinux enabled, verify that the permissive mode is enabled or specific exceptions are in place for the applications being installed to avoid any installation failures.
- If a service is used in a blueprint or included as a preinstalled service in a logical template, the service cannot be deleted. Before you delete a service, delete the referenced blueprint or logical template with the preinstalled service.
- To view examples about how to name or describe a service, select Catalog > Services to view the predefined services.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Services.
2. Click New in the toolbar.
3. Set the name of the service, add a description, and click Save.
   Use a name that corresponds to the software component to which the action scripts relate and append qualifiers for tracking purposes. For example, if you use the service on a particular operating system, you can set the service name as ServiceName_Windows or ServiceName_Linux.
4. Click Create Service Version to create a service version.
   You can create multiple versions for a service.
   A page for creating a service version opens.
5. Complete the service version information.
   The service version name stays the same.
   a. Specify major, minor, or micro release versions, with or without qualifiers.
      For example, you might use version numbers such as 1.0, 1.5, or 1.0.1-Linux.
   b. (Optional) In the Description section, if you are creating a service version for a specific operating system, describe the operating system version used and any applicable required configuration.
6 Set the tags to organize the list of services you see when you create a deployment blueprint for an application

**NOTE** Only tags designated as Server Type appear in the drop-down menu.

You can add multiple tags.

7 To create a tag that is not in the list, click **Cancel**, and select **Catalog > Tags**.

8 (Optional) In the Supported OSes section, if the scripts used in this service can run only on particular operating systems, select those operating systems here.

   In the blueprint editor, vCloud Application Director prevents the service from being added to a template unless the template contains one of these operating systems. Leave this field blank if the service can be used in any operating system.

9 To use an operating system name that is not in the list, click **Cancel**, and select **Catalog > Operating Systems** to create an operating system name.

10 (Optional) In the Supported Components section, if only certain types of application component can run in this service, specify those components here.

   For example, only WAR and JAR components can run in a vFabric tc Server instance. Only SQL scripts can run in a database server. The components that you select restrict what application components can be added to this service in an application blueprint. Leave this field blank if you can add components to the service.

11 Select the Pre-install in a Template check box to list the service in the Services Included section while you create or edit a logical template.

   Selecting the checkbox Indicates that the service is already installed in a template.

   If you did not select this check box, the service appears in the list of services that you can include when you create a deployment blueprint for an application.

   Logical templates with preinstalled services also appear in the OS Templates section of the application blueprint and are available as part of the template.

What to do next


**Define Service Version Properties**

vCloud Application Director passes defined properties as environment variables to scripts running in a virtual machine. The service version properties define the variables used in the scripts for the service.

**Prerequisites**

- Verify that your user account has the **ROLE_CATALOG_ADMIN** catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- To view examples about how to define a service, select **Catalog > Services** to view the predefined services.
- Verify that a service version is available in vCloud Application Director. See “Create a Service Version in the Catalog,” on page 106.

**Procedure**

1 Open an existing service version and click **Edit**.
2. For the Property Name and Description, define a variable and add an optional description. The property name cannot begin with a digit. After a variable is added and defined, you can create an install, configure, start, update, rollback, or teardown script for the service version.

3. To change the type, select a property type from the drop-down menu. You cannot add types to the menu.

4. Type the value to substitute for this property when the script runs. For example, for a property called http_port, you might type 80 in this field.

   **Note** For vCloud Director or vCloud Automation Center, if you add a value to the http_proxy, https_proxy, or ftp_proxy property with service scripts that use the darwin_global.conf file as a file source, when the script runs, these properties override any existing proxy information in the deployed application. Amazon EC2 does not require a proxy to deploy an application.

   If the property type is set to Computed, you cannot type a value.

   If the Required check box is selected and the Overridable in Blueprint check box is not selected, the property must have a value.

5. From the Auto-Bind Type drop-down menu, select an Auto-Bind type. The Auto-Bind Type describes the role that a property is assigned to for autobinding in the blueprint canvas.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>The property does not Auto-Bind.</td>
</tr>
<tr>
<td>Consume</td>
<td>The property value is set with autobinding.</td>
</tr>
<tr>
<td>Expose</td>
<td>The property can be used to customize another property value with autobinding.</td>
</tr>
</tbody>
</table>

   **Note** A property cannot have the Consume and Expose roles.

   An Auto-Bind type of Consume or Expose requires at least one Auto-Bind tag to be defined for the property. Auto-Bind tags on the Consume property must be a subset of the tags on the Expose property. The tag sets do not have to be equal.

6. Define the property type. The property type you define affects the Auto-Bind type you can set. The following combination of property type and Auto-Bind types are not allowed.

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Cannot Set Auto-Bind Type to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Array</td>
<td>Expose</td>
</tr>
<tr>
<td>Computed</td>
<td>Consume</td>
</tr>
<tr>
<td>Content</td>
<td>Expose or Consume</td>
</tr>
</tbody>
</table>
From the Auto-Bind Tags drop-down menu, click **New** to select an Auto-Bind type to add the Property Descriptor tags to a property.

**NOTE** Tags designated only as Property Descriptor appear in the drop-down menu.

You can add multiple Auto-Bind tags.

For example, if the Auto-Bind type for the Apache 2.2.0 service appsrv_routes property is assigned as Consume, and the Auto-Bind type for the vFabric tc Server 2.1.0 service JVM_ROUTE property is assigned as Expose. The appsrv_routes property uses the JVM_ROUTE property values to customize itself. To allow Auto-Bind to connect to these service properties, you can add the Servlet Container and Route tags to the appsrv_routes and JVM_ROUTE properties, respectively. When you create an application that includes the vFabric tc Server and Apache services, the properties Auto-Bind to each other in the blueprint editor if there is a dependency between the two nodes.

Select the **Required** check box for properties that are required to deploy an application.

If any of the required property values are left blank, you are prompted to complete them before an application deployment.

If the property type is Computed, the **Required** check box is not applicable. If the property type is Content, the **Required** check box is selected by default and the Secured check box is not applicable.

For example, for an http_proxy_port property using the Apache JServ Protocol (AJP), you must set the value to 8889.

Select the **Secured** check box for passwords you define or to obscure the values of other properties.

If a property is changed from Secured to Unsecured, vCloud Application Director resets the property value, for security purposes. You must set a new password value for the property.

For example, the db_password property in the sample MySQL service is secured.

Select the **Overridable in Blueprint** check box to allow users such as an application architect to override the value for the property in an application blueprint.

If the property type is Computed, the **Overridable in Blueprint** check box is not applicable.

If a property is not overridable in the blueprint, you cannot set the Auto-Bind type to Consume.

For example, the catalog administrator might configure the vFabric tc Server service to have a JVM heap size of 512MB. But for large deployments, the application architect might change the setting to 1024MB.

Click **Delete** to remove the selected row from the Properties section.

(Optional) Click **Reset** to revert to the original property value.

Click **Upload** to import the property values to a selected CSV file.

Save your service property definitions.

**What to do next**


**Add Action Scripts to the Service Version**

The catalog administrator must provide an action script for at least one of the life cycle stages. For deploying an application, you can create an install, configure, start, rollback, and teardown script, or create an update script to update an existing deployment. These scripts are customized to use the component properties.

You are not required to add scripts for all the life cycle stages. If you do not need a particular stage, ignore it.
Prerequisites

- Verify that your user account has the \texttt{ROLE_CATALOG_ADMIN} catalog administrator role assigned to it.
- If you plan to use a script that downloads software from an external Web site, verify that the virtual machine you use for deploying the application has access to an external network.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- To view examples about how to add scripts to a service, select \texttt{Catalog > Services} to view the predefined services.
- Verify that properties are defined for the service version. See “Define Service Version Properties,” on page 107.

Procedure

1. Open an existing service version and click \texttt{Edit}.
2. From the drop-down menu, select a script type for your action script.
   You can author in Bash or BeanShell script for Linux-based applications, or Windows CMD, PowerShell, or BeanShell script for Windows-based applications.
3. In the Script column, click the hyperlink to open the Edit Script dialog box.
   You can write the script or copy a script into the dialog box.
   A catalog administrator can parameterize the installation and configuration of services. The properties that are defined for a service can be used inside the script.
4. Click the down arrow in the script type field to select an action script type for a life cycle stage.
5. To insert the properties you defined, click the down arrow in the Select a property to insert list.
6. (Optional)
   Select the \texttt{Reboot} check box to restart the virtual machine after the script runs successfully, during an application deployment.
7. Click \texttt{OK} when you are finished.
8. In the Reboot column, select the check box so that the agent bootstrap can restart the virtual machine after an action script runs successfully.
   After the virtual machine is restarted, the agent proceeds to the next life cycle stage script defined in the service version.
9. Click \texttt{Reset} to clear the script.
   This operation does not remove the row for the life cycle stage.
10. When you are finished creating the service version, click \texttt{Save} and \texttt{OK}.

The service you created is added to the page.

What to do next

Create a copy of an existing service version, or edit the service. See “Maintain Service Versions,” on page 111.
Maintain Service Versions

You can copy or edit a service from an existing service version.

Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- If you plan to edit the properties and scripts contained in services that you add to the application blueprint, familiarize yourself with the tasks described in “Add a Service to the Catalog,” on page 105.
- If you plan to use a script that downloads software from an external Web site, verify that the virtual machine you use for deploying the application has access to an external network.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- Create a copy of the service to preserve the original for future reference and edit the copied version.
- Verify that the current group you are logged in to owns the service version.
  
  If you are not part of the group that owns the service version, the Copy and Edit buttons are disabled.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Services.
2. Copy an existing service version.
   a. Open a service version and click Copy in the toolbar.

   When you copy a service, all of the properties, action scripts, operating systems, a list of descriptive tags, and supported application components are retained. You cannot change the name of the existing service version.
   b. Change the version number, update the description, and click Save.

   The version number and description differentiate the new service from the parent version.
3. To modify the variables or update the action scripts, click Edit in the toolbar, make the changes, and click Save.

What to do next

You can delete the services that you do not use to model an application blueprint. See “Delete a Service from the Catalog,” on page 111.

Delete a Service from the Catalog

You can delete a service from the Services page.

Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Verify that the service is not used in a blueprint or included as a preinstalled service in a logical template.
  
  Delete the referenced blueprint or logical template that contains the preinstalled service.
- Verify that the current group you are logged in to owns the service version.
  
  If you are not part of the group that owns the service version, the Delete button is disabled.
Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Services.
2. Open the service.
3. Select Delete to remove the existing service versions and confirm your selection.
4. Click the arrow next to the service name to return to the Services page.
5. Delete the service and confirm your selection.

Import a Puppet Service to the Catalog

Puppet services can be used to build application blueprints. You must import Puppet classes and defined resources into the vCloud Application Director catalog as services.

You can import multiple Puppet classes at the same time.

After you import the Puppet classes, these services have a special tag called Puppet Services. The newly imported services do not have action scripts. vCloud Application Director delegates their life cycle activities to the registered Puppet master instance in the corresponding deployment environment.

The imported Puppet content cannot be updated. You can remove the service or change the existing Puppet service version and import the updated Puppet content into the catalog.

Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Start the vCloud Application Director CLI. See “Start the CLI Remotely,” on page 36.
- The remote machine that you use to run the CLI must have Puppet modules installed and the Puppet CLI available.
  You can also use a Puppet master machine to maintain the Puppet modules on two different machines.
- Verify that the Puppet services you are importing are available on the Puppet master.

Procedure

- In the root shell, type the command to import the Puppet class such as Apache.

  ```bash
  import-puppet-manifest --typeFilter "^apache$"
  ```

  The parameters in the command are case sensitive. If you use the incorrect case, the command is ignored.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>typeFilter</td>
<td>Limits the import to types matching the specified search expression.</td>
</tr>
<tr>
<td></td>
<td>For details about expression syntax refer to the Puppet documentation or</td>
</tr>
<tr>
<td></td>
<td>type puppet man resource_type to view the man pages. For example, to</td>
</tr>
<tr>
<td></td>
<td>import the mysql class, type import-puppet-manifests --typeFilter &quot;^mysql$&quot;</td>
</tr>
<tr>
<td>osfamily</td>
<td>Specifies the operating system for the imported type.</td>
</tr>
<tr>
<td></td>
<td>Linux is the default operating system.</td>
</tr>
<tr>
<td>shared</td>
<td>Indicates whether the artifact is shared publicly or private.</td>
</tr>
<tr>
<td></td>
<td>You can define true or false in the command. The default setting false</td>
</tr>
<tr>
<td></td>
<td>indicates that the artifact is private to the user’s group performing the</td>
</tr>
<tr>
<td></td>
<td>import operation.</td>
</tr>
</tbody>
</table>

The command imports the apache Puppet class to the catalog.
What to do next

Add the Puppet content to a new or existing application blueprint from the Puppet Services section of the blueprint canvas and deploy the application. See “Create an Application,” on page 134 and “Set Up and Configure a Deployment Profile,” on page 147.

Add an External Service to the Catalog

In the vCloud Application Director catalog, an external service is a service that is installed external to the deployment of the application. An external service and the application need to be configured to work with each other.

Procedure

1. Create an External Service in the Catalog on page 113
   With vCloud Application Director you can create basic or advanced external services.
2. Define External Service Properties on page 115
   vCloud Application Director passes defined properties as environment variables to scripts running in a virtual machine. The external service version properties define the variables used in the scripts for the external service.
3. Create Provider Specification for an Advanced External Service Version on page 117
   An advanced external service requires you to create a provider specification. This specification is a set of one or more scripts that defines how an external service can be configured, updated, rolled back, and torn down in various service provider environments such as VMware Data Director or Amazon RDS. The provider specification also has properties that you can define to connect to the provider and the provider specific parameters.
4. Maintain External Service Versions on page 119
   You can copy a basic or advanced external service from an existing version or modify the service.
5. Delete an External Service on page 120
   To manage your external services you can delete duplicate or outdated services.

Create an External Service in the Catalog

With vCloud Application Director you can create basic or advanced external services.

A basic external service does not have support for multiple service providers. You must define the logical template and have the option to add scripts for the CONFIGURE, UPDATE, ROLLBACK, and TEARDOWN life cycle stages. Some examples of a basic external service include, an existing database with the application schema installed, a SAAS application such as Workday, an LDAP server, or a Single Sign-On (SSO) server.

An advanced external service includes options for provider specifications that add support for multiple providers of the external service. After the advanced external service property values are defined in the provider specification, these values are populated in the external service instance when you map that instance to a deployment environment. For example, an external Oracle database can be provisioned and configured through VMWare Data Director or Amazon RDS. The external load balancer in an application can be implemented with an F5 load balancer instance or a software load balancer such as an Apache server.

You can share an external service publicly or privately in a group. Sharing is supported at the component level and not at the individual version level. When you share an external service, the external service instance profile are not shared. Each group must define an external service instance profile in its deployment environment.
Prerequisites

- Verify that your user account has the **ROLE_CATALOG_ADMIN** catalog administrator role assigned to it.
- Verify whether you must add items to the operating systems and descriptive tag lists. See “Add Operating Systems to the Catalog,” on page 104 and “Add Tags to the Catalog,” on page 104.
- To view examples about how to name or describe an external service, select Catalog > External Services to view the predefined external services.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > External Services.
2. Click **New** in the toolbar.
3. Set the name of the external service, add a description, and click **Save**.
   Use a name that corresponds to the server or database that you are
4. Click **Create External Service Version** to create an external service version.
   You can create multiple versions for an external service.
5. In the dialog box, select **Basic** or **Advanced** and click **OK**.
6. Complete the external service version information.
   The external service version name stays the same.
   a. Specify major, minor, or micro release versions, with or without qualifiers.
      For example, you might use version numbers such as 1.0, 1.5, or 1.0.1-Dev.
   b. (Optional) In the Description section, if you are creating an external service version for a specific database, describe the database version used and any applicable required configuration.
7. Click **New** to set the tags from the drop-down menu.
   The tags group the external services and you can view them in their corresponding groups when you create a deployment blueprint for an application.
   You can add multiple tags.
8. To create a tag that is not in the list, click **Cancel**, and select Catalog > Tags.
9. (Optional) In the Supported Components section, if only certain types of application component can run in this advanced external service, click **New** to specify those components here.
   For example, only SQL scripts can run in a database server. The components that you select restrict what application components can be added to this advanced external service in an application blueprint. Leave this field blank if you can add components to the advanced external service.
10 Designate an operating system for the external service.
   a For the basic external service, select a Windows or Linux-based template from the drop-down menu in the Logical Template section.
      The supported OS associated to the logical template is populated.
   b For the advanced external service, if the scripts used in the advanced external service can run only on particular operating systems, click New to add those operating systems in the Supported OSes section.
      In the blueprint editor, vCloud Application Director prevents the external service from being added to a template unless the template contains one of these operating systems. Leave this field blank if the advanced external service can be used in any operating system.

11 To use an operating system name that is not in the list, click Cancel, and select Catalog > Operating Systems to create an operating system name.

What to do next
Configure the external service version properties. See “Define External Service Properties,” on page 115.

Define External Service Properties
vCloud Application Director passes defined properties as environment variables to scripts running in a virtual machine. The external service version properties define the variables used in the scripts for the external service.

When you deploy an application with a preconfigured or custom external service, you cannot edit the property definitions or provider specification properties for that external service while the deployment is in progress.

Prerequisites
- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- To view examples about how to define an external service, select Catalog > External Services to view the predefined services.
- Verify that an external service version is available in vCloud Application Director. See “Create an External Service in the Catalog,” on page 113.

Procedure
1 Open an existing external service version and click Edit.
2 For the Property Name and Description, define a variable and add an optional description.
   - The property name cannot begin with a digit.
3 To change the type, select a property type from the drop-down menu.
   - You cannot add types to the menu.
4 Type the value to substitute for this property when the script runs.

For example, for a property called http_port, you might type 80 in this field.

**Note** For vCloud Director or vCloud Automation Center, if you add a value to the http_proxy, https_proxy, or ftp_proxy property with service scripts that use the darwin_global.conf file as a file source, when the script runs, these properties override any existing proxy information in the deployed application. Amazon EC2 does not require a proxy to deploy an application.

If the property type is set to Computed, you cannot type a value.

If the **Required** check box is selected and the **Overridable in Blueprint** check box is not selected, the property must have a value.

5 From the Auto-Bind Tags drop-down menu, click **New** to select an Auto-Bind type to add the Property Descriptor tags to a property.

**Note** Tags designated only as Property Descriptor appear in the drop-down menu.

You can add multiple Auto-Bind tags.

For example, if the Auto-Bind type for the Apache 2.2.0 service appsrv_routes property is assigned as Consume, and the Auto-Bind type for the vFabric tc Server 2.1.0 service JVM_ROUTE property is assigned as Expose. The appsrv_routes property uses the JVM_ROUTE property values to customize itself. To allow Auto-Bind to connect to these service properties, you can add the Servlet Container and Route tags to the appsrv_routes and JVM_ROUTE properties, respectively. When you create an application that includes the vFabric tc Server and Apache services, the properties Auto-Bind to each other in the blueprint editor if there is a dependency between the two nodes.

6 From the Auto-Bind Type drop-down menu, select an Auto-Bind type.

The Auto-Bind Type describes the role that a property is assigned to for autobinding in the blueprint canvas.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>The property does not Auto-Bind.</td>
</tr>
<tr>
<td>Consume</td>
<td>The property value is set with autobinding.</td>
</tr>
<tr>
<td>Expose</td>
<td>The property can be used to customize another property value with autobinding.</td>
</tr>
</tbody>
</table>

**Note** A property cannot have the Consume and Expose roles.

An Auto-Bind type of Consume or Expose requires at least one Auto-Bind tag to be defined for the property. Auto-Bind tags on the Consume property must be a subset of the tags on the Expose property. The tag sets do not have to be equal.

7 Select the **Required** check box for properties that are required to deploy an application.

If any of the required property values are left blank, you are prompted to complete them before an application deployment.

If the property type is Computed, the **Required** check box is not applicable. If the property type is Content, the **Required** check box is selected by default and the Secured check box is not applicable.

For example, for an http_proxy_port property using the Apache JServ Protocol (AJP), you must set the value to 8009.
8. Select the **Secured** check box for passwords you define or to obscure the values of other properties.

If a property is changed from Secured to Unsecured, vCloud Application Director resets the property value, for security purposes. You must set a new password value for the property.

For example, the db_password property in the sample MySQL service is secured.

9. From the Override drop-down menu, select a property override option.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blueprint</td>
<td>Property is overridable in the application blueprint.</td>
</tr>
<tr>
<td></td>
<td>The cloud admin cannot view this property when registering the external</td>
</tr>
<tr>
<td></td>
<td>service instance to a deployment environment.</td>
</tr>
<tr>
<td>Deployment Environment</td>
<td>Cloud admin can override the property in the deployment environment.</td>
</tr>
<tr>
<td></td>
<td>The property appears in the application blueprint canvas and user can</td>
</tr>
<tr>
<td></td>
<td>Auto-Bind to a property. The property value is not editable in the</td>
</tr>
<tr>
<td></td>
<td>application blueprint.</td>
</tr>
<tr>
<td>Not Overridable</td>
<td>Property is not overridable.</td>
</tr>
</tbody>
</table>

10. Select the **Overridable in Blueprint** check box to allow users such as an application architect to override the value for the property in an application blueprint.

If the property type is Computed, the **Overridable in Blueprint** check box is not applicable.

If a property is not overridable in the blueprint, you cannot set the Auto-Bind type to Consume.

For example, the catalog administrator might configure the vFabric tc Server service to have a JVM heap size of 512MB. But for large deployments, the application architect might change the setting to 1024MB.

11. Click **Delete** to remove the selected row from the Properties section.

12. (Optional) Click **Reset** to revert to the original property value.

13. Click **Upload** to import the property values to a selected CSV file.

14. Save your external service property definitions.

**What to do next**

Specifying external service definitions has no impact on deployments unless you create an external service instance in a deployment environment to enable the external service definitions on all of the deployments under the deployment environment. See “Map an External Service Instance,” on page 61.

Create provider specification for the custom external service version. See “Create Provider Specification for an Advanced External Service Version,” on page 117.

**Create Provider Specification for an Advanced External Service Version**

An advanced external service requires you to create a provider specification. This specification is a set of one or more scripts that defines how an external service can be configured, updated, rolled back, and torn down in various service provider environments such as VMware Data Director or Amazon RDS. The provider specification also has properties that you can define to connect to the provider and the provider specific parameters.

You can change only the group membership settings for the external service. The new membership settings are then propagated to the service provider specifications of the external service.

User cannot update group membership of the service provider spec separately.
Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- If you plan to use a script that downloads software from an external Web site, verify that the virtual machine you use for deploying the application has access to an external network.
- To view examples about how to create a provider specification for an external service, select Catalog > External Services to view the predefined services.
- Verify that properties are defined for the external service version. See vCloud Application Director. See “Create an External Service in the Catalog,” on page 113.

Procedure

1. Open an existing service version and click Create Provider Spec Version.
2. Type the provider specification name and select Create Version.
3. Specify major, minor, or micro release versions, with or without qualifiers.
   The provider specification name stays the same.
4. (Optional) In the Description section, if you are creating a provider specification for a specific configuration, describe the configuration used and applicable requirements.
5. In the Logical Templates section, if you set the supported OSes to Windows or Linux-based operating systems, the corresponding logical templates are available in the drop-down menu.
6. Add applicable properties.
7. From the drop-down menu, select a script type for your action script.
   You can author in Bash or BeanShell script for Linux-based applications, or Windows CMD, PowerShell, or BeanShell script for Windows-based applications.
8. In the Script column, click the hyperlink to open the Edit Script dialog box.
   You can write the script or copy a script into the dialog box.
   A catalog administrator can parameterize the installation and configuration of services. The properties that are defined for a service can be used inside the script.
9. Click the down arrow in the script type field to select an action script type for a life cycle stage.
10. To insert the properties you defined, click the down arrow in the Select a property to insert list.
11. (Optional)
   Select the Reboot check box to restart the virtual machine after the script runs successfully, during an application deployment.
12. Click OK when you are finished.
13. In the Reboot column, select the check box so that the agent bootstrap can restart the virtual machine after an action script runs successfully.
   After the virtual machine is restarted, the agent proceeds to the next life cycle stage script defined in the service version.
14 Click Reset to clear the script.
   This operation does not remove the row for the life cycle stage.

15 When you are finished creating the provider specification version, click Save.

The provider specification version you created is added to the advanced external service.

What to do next
Specifying external service definitions has no impact on deployments unless you create an external service instance in a deployment environment to enable the external service definitions on all of the deployments under the deployment environment. See “Map an External Service Instance,” on page 61.

Add the external service to your application blueprint. See “Add an External Service to an Advanced Blueprint,” on page 142.

Update external services or copy an existing external service. See “Maintain External Service Versions,” on page 119.

Maintain External Service Versions
You can copy a basic or advanced external service from an existing version or modify the service.

Prerequisites
- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- To view examples about how to define an external service, select Catalog > External Services to view the predefined services.
- Verify that an external service version is available in vCloud Application Director. See “Create an External Service in the Catalog,” on page 113.
- Create a copy of the external service to preserve the original for future reference and edit the copied version.
- Verify that the current group you are logged in to owns the external service version.
   If you are not part of the group that owns the external service version, the Copy and Edit buttons are disabled.

Procedure
1 On the vCloud Application Director title bar, click the drop-down menu and select Catalog > External Services.

2 Copy an existing external service version.
   a Open an external service version and click Copy in the toolbar.
      When you copy an external service, all of the properties, operating systems, a list of descriptive tags, supported application components, and provider specification scripts and properties are retained. You cannot change the name of the existing external service version.
   b Change the version number, update the description, and click Save.
      The version number and description differentiate the new external service from the parent version.
3 Modify the existing properties.
   a Click **Edit** in the toolbar.
   b Make your changes and click **Save**.

4 Update the provider specifications.
   a Open the provider specification version.
   b Click **Edit** in the toolbar.
   c Make your changes and click **Save**.

5 Copy the existing provider specifications.
   a Open the provider specification version.
   b Click **Copy** in the toolbar.
   c Change the version number, update the description, and click **Save**.
   
   The version number and description differentiate the new provider specification from the parent version.

**What to do next**

Delete an older version of an external service you do not use in an application blueprint. See “Delete a Service from the Catalog,” on page 111.

**Delete an External Service**

To manage your external services you can delete duplicate or outdated services.

**Prerequisites**

- Verify that your user account has the **ROLE_CATALOG_ADMIN** catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- Verify that the external service is not used in a blueprint. See “Add an External Service to an Advanced Blueprint,” on page 142.
- Verify that the current group you are logged in to owns the external service version.
  
  If you are not part of the group that owns the external service version, the **Delete** button is disabled.

**Procedure**

1 On the vCloud Application Director title bar, click the drop-down menu and select **Catalog > External Services**.

2 Open the existing external service.

3 Select **Delete** to remove the existing service versions and confirm your selection.

4 Click the arrow next to the external service name to return to the External Services page.

5 Delete the external service and confirm your selection.
Add a Logical Template to the Catalog

You can add compatible services and scripts to custom or sample logical templates when you model an application blueprint. You can map the logical templates to actual cloud templates from vCloud Director, vCloud Automation Center, or Amazon EC2. Logical templates allow an application blueprint to remain cloud agnostic.

As part of the logical template definition, you can describe which services are already installed in the template with the operating system. Typically, in IT organizations, a few performance monitoring agents or virus scanners are installed in a template. Also, for example, vFabric tc Server might be preinstalled in the logical template to accelerate deployments. If you always use a particular service when you deploy a logical template, you can preinstall it to avoid having to add it for every deployment.

You can share a logical template publicly or privately in a group. Sharing is supported at the component level and not at the individual version level.

Save your changes frequently. The vCloud Application Director user interface session expires after 30 minutes of inactivity. If the session expires, any changes that were not saved are lost.

Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Create cloud templates in vCloud Director, vCloud Automation Center, or Amazon EC2 that meet the requirements for working properly in vCloud Application Director. See “Virtual Machine Requirements for Creating vCloud Director Custom Templates,” on page 70, “Virtual Machine Requirements for Creating vCloud Automation Center Custom Templates,” on page 48, or “Virtual Machine Requirements for Creating Amazon EC2 Custom Templates,” on page 86.
- Verify that at least one cloud provider is registered in vCloud Application Director. See “Register the vCloud Director Cloud Provider and Template,” on page 78, “Register the vCloud Automation Center Cloud Provider and Template,” on page 57, or “Register the Amazon EC2 Cloud Provider and Template,” on page 88.
- If the application requires access to URLs from outside the corporate firewall, configure the applicable services and application components to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- Verify whether you must add items to the operating systems and descriptive tag lists. See “Add Operating Systems to the Catalog,” on page 104 and “Add Tags to the Catalog,” on page 104.
- If you plan to have application or services preinstalled in a template, create and configure the services before you add them to the template. See “Add a Service to the Catalog,” on page 105.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Logical Templates.
2. Click New in the toolbar.
3. Set the name of the logical template, add a description, and click Save.
   To keep track of which cloud template or operating system you are using, include the name of the cloud template or operating system.
4. Click Create LT Version to create a logical template version.
   You can create multiple versions for a logical template.
   A page opens for creating a logical template version.
5 Complete the template version information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Logical template version name stays the same.</td>
</tr>
<tr>
<td>Version</td>
<td>Add a qualifier to denote major and minor versions such as 1.2.1 or 1.0.1-CentOS56-32bit.</td>
</tr>
<tr>
<td>Description</td>
<td>Add detailed information about the logical template. For example, describe a script you are using or the amount of disk size required.</td>
</tr>
</tbody>
</table>
| Tags   | Categorize logical templates based on the functions that they provide. vCloud Application Director organizes the templates based on tags you see when you create a deployment blueprint for an application. 

**NOTE** Only tags designated as Server Type appear in the drop-down menu. 
You can add multiple tags. 
To use a tag that is not in the list, click Cancel, and select Catalog > Tags to create a tag. |
| Supported OS | Specifies the operating system installed in the logical template. This OS information is used in the application blueprint to limit which services you can add to this template. Not all services run on all operating systems. For example, if you specify an Ubuntu operating system, when you use this template in a blueprint and try to add a service that is not compatible with Ubuntu, vCloud Application Director prevents you from adding that service. 
To use an operating system name that is not in the list, click Cancel, and select Catalog > Operating Systems to create an operating system name. |

6 Map a cloud template to the logical template.

You can add multiple cloud templates to one logical template version or select different cloud templates for different clouds. Even if you are using the same cloud provider, you might need to select from different cloud templates at deployment time to allow for different template configurations.

Duplicate cloud template and logical template mappings and empty rows are not saved.

For example, with multiple cloud templates, you can use the same logical template. If you are deploying to a production environment, you can select a cloud template that has a large amount of disk space. For a test or staging environment, you can select a cloud template with a small amount of disk space.

a In the Cloud Provider Name column, click the down arrow to select a cloud provider. 
A list of cloud providers that you created appears.

b In the Cloud Template column, click the down arrow to designate a cloud template. 
Cloud templates that belong to the same group as the user appear in the drop-down menu. If the list of cloud templates is empty, the existing cloud templates do not belong to your group or a cloud template was not registered.

7 (Optional) To map multiple cloud templates to a logical template, repeat Step 6.

8 (Optional) Define a preinstalled service.

a In the Service Name column, click the down arrow to select a preinstalled service.

b To use a service that is not in the list, click Cancel, and select Catalog > Services to create a service and the action scripts it includes.

If any preinstalled services are added to a logical template after creating a blueprint, the new preinstalled services are not added to the node.
9 (Optional) Add the new preinstalled services to the node.
   a Drag the logical template with the preinstalled service to the application blueprint.
   b Transfer the services and components to the new logical template.
   c Recreate any applicable dependencies and delete the old template.
10 When you finish creating the template, click Save.

The logical template that you created is added to the Logical Templates page. The template also appears in the list of logical templates that you can include when you create a deployment blueprint for an application.

What to do next
Copy an existing logical template version, edit, or delete the template. See “Maintain Logical Template Versions,” on page 123.

Maintain Logical Template Versions
To map an existing logical template to another cloud template or add preinstalled services, you can edit a logical template version.

Prerequisites
- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- If the application requires access to URLs from outside the corporate firewall, configure the applicable services and application components to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- Verify whether you must add items to the operating systems and descriptive tag lists. See “Add Operating Systems to the Catalog,” on page 104 and “Add Tags to the Catalog,” on page 104.
- If you plan to have application or services preinstalled in a template, create and configure the services. See “Add a Service to the Catalog,” on page 105.
- Verify that the current group you are logged in to owns the logical template version.
  - If you are not part of the group that owns the logical template version, the Copy and Edit buttons are disabled.

Procedure
1 On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Logical Templates.
2 Copy a logical template from an existing logical template version.
   a Open the logical template version and click Copy in the toolbar.
      All of the cloud templates, services, operating systems, and a list of descriptive tags to identify
      application components are retained. You cannot change the name of the existing logical template
      version.
   b Change the version number, update the description, and if needed update the cloud provider,
      cloud template, or services.
      The version number and description differentiates the new logical template from the parent
      version.
      A best practice is to create a copy of the template to preserve the original for future reference.
   c Click Save.
3 To change existing cloud provider, cloud template, or services for a logical template, click Edit in the
   toolbar, make your changes, and click Save.

What to do next
If you have an older version of a logical template, you can delete it from the catalog. See “Delete a Logical
Template from the Catalog,” on page 124.

Delete a Logical Template from the Catalog
You can delete a logical template from the Logical Templates page.

Prerequisites
- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned
  to it.
- Verify that the logical templates in application blueprints and all of the other objects referencing the
  template are deleted.
- Verify that the current group you are logged in to owns the logical template version.
  If you are not part of the group that owns the logical template version, the Delete button is disabled.

Procedure
1 On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Logical
   Templates.
2 Open the logical template.
3 Select Delete to remove the existing logical template versions and confirm your selection.
4 Click the arrow next to the logical template name to return to the Logical Templates page.
5 Delete the logical template and confirm your selection.

Add a Policy to the Catalog
You can create policy definitions to capture a custom set of rules that assess the compliance state of an
application deployment.

vCloud Application Director includes the following predefined policy definitions.
- Blacklist services - checks if an unwanted service is used in the blueprint node or logical template.
- Memory policy - compares the node memory values against the defined minimum and maximum
  memory values.
CPU policy - compares the node CPU count values against the specified minimum and maximum memory values.

Max VM count policy - checks for the total number of virtual machine instances, with the exclusion of temporary virtual machines for external services, across all of the nodes in a deployment.

Mandatory services - checks if a blueprint node is missing the specified mandatory services.

**Prerequisites**

- Verify that your user account has the **ROLE_CATALOG_ADMIN** catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- To view examples about how to name or define a policy, select **Catalog > Policies** to view the predefined policy.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Catalog > Policies**.
2. Click **New** in the toolbar.
3. Set the name of the policy, add a description, and click **Save**.
   Use a name associated to the policy definition for tracking purposes. For example, if the policy checks whether specific services are always applied to applications during deployment, you can list the service names in the description.
4. Click **Create Policy Version** to create a policy version.
   You can create multiple versions for a policy.
   A page for creating a policy version opens.
5. Complete the policy version information.
   The policy version name stays the same.
   a. Specify major, minor, or micro release versions, with or without qualifiers.
      For example, you might use version numbers such as 1.0, 1.5, or 1.0.1-CPU-Policy.
   b. (Optional) In the Description section, describe the definitions in the policy. If the policy is based on a previous version, detail the differences between the two policy versions.
6. For the Property Name and Description, define a variable and add an optional description.
   The property name cannot begin with a digit.
7. To change the type, select a String or Array property type from the drop-down menu.
   You cannot add types to the menu.
8. Type the value to substitute for this property when the script runs.
   For example, for a property called max_cpu_count, you might type **10** in this field to specify the maximum number of CPU allowed in an application.
9. Select the **Secured** check box for passwords you define or to obscure the values of other properties.
   If a property is changed from Secured to Unsecured, vCloud Application Director resets the property value, for security purposes. You must set a new password value for the property.
10. Click **Delete** to remove the selected row from the Properties section.
11. (Optional) Click **Reset** to revert to the original property value.
12 Click OK and save your changes when you are finished.

The policy you created is added to the page.

**What to do next**

Add a policy definition script to assess the compliance state of a deployment. See “Create a Policy Definition Script,” on page 126.

**Create a Policy Definition Script**

Each policy definition has a SCAN script for the life cycle stage to assess the compliance state of a deployment. vCloud Application Director calls the policy scan action script prior to performing operations, except the teardown process, on the deployment or when a user explicitly initiates a policy scan on the deployment.

The scan action script includes a model of deployment as defined in vCloud Application Director REST API specification. The scan action script also receives additional components used in blueprint for the deployment.

You must create policy instances in specific deployment environments to enable policies. If a policy violation occurs during deployment, it is flagged and you can view the violation details in the compliance view summary page.

**NOTE** Java Script is the only supported language for authoring policy definition scripts.

Policy properties defined in a policy definition are supplied to the script as individual variables. The script can access them by declaring a variable with same name as the property name.

<table>
<thead>
<tr>
<th>Script Input Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>var min_cpu_count</td>
<td>Corresponds to min_cpu_count property and the value for the script is set to consume.</td>
</tr>
</tbody>
</table>
| eventPayload          | Includes the details of the deployment assessed for policy compliance. For regular properties, var eventPayload must be added to access the eventPayload object. The policy script might regard the eventPayload variable as a java object with the following properties:
  - deploymentProfile of type DeploymentProfile as defined in V2 API Represents the deployment profile capturing latest details of deployment. In the case of updates, this includes all of the changes that are part of update profile.
  - blueprint of type Blueprint as defined in V2 API Represents the actual blueprint object that’s referenced from deploymentProfile.
  - logicalTemplates of type ListLogicalTemplate where LogicalTemplate is as defined in V2 API Represents the list of logical templates referenced from various nodes inside the application blueprint.
  - serviceVersions of type ListServiceVersion where ServiceVersion is as defined in V2 API Represents the list of service versions references from various nodes inside the application blueprint.
Policy scripts are expected to output the following properties to communicate the result of the compliance assessment. Scripts must declare them as variables.

<table>
<thead>
<tr>
<th>Script Output Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>complianceResult</td>
<td>Type of string that is mandatory. If script fails to set it then the compliance result is assumed as an Error. The valid values for the variable are:</td>
</tr>
<tr>
<td>Compliant</td>
<td>Indicates that a deployment is compliant against the policy being assessed.</td>
</tr>
<tr>
<td>Non_Compliant</td>
<td>Indicates that a deployment violates the policy being assessed.</td>
</tr>
<tr>
<td>Error</td>
<td>Indicates failure to produce an assessment result.</td>
</tr>
<tr>
<td>complianceMessage</td>
<td>Type of string. This optional value provides a high-level summary of the reason behind policy violation. Value can be any string with less than 2048 characters.</td>
</tr>
</tbody>
</table>

The scripts can create log messages with the standard println function available in Java Script. The log is captured by vCloud Application Director, which is useful to diagnose errors in policy scripts or provide details for a policy assessment result.

**Prerequisites**
- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Verify that at least one policy is created in the catalog. See “Add a Policy to the Catalog,” on page 124.

**Procedure**
1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Policies.
2. Open a policy to add a policy definition script.
3. In the Script column, click the hyperlink to open the Edit Script dialog box.
   - You can refer to the existing predefined policy definitions and create a script in the dialog box.
4. Click OK and save your changes when you are finished.

**What to do next**
Specifying a policy definition has no impact on deployments unless you create a policy instance in a deployment environment to enable that policy definition on all of the deployments under the deployment environment. See “Create a Policy Instance,” on page 61.

**Maintain Policy Versions**
You can edit a policy to add or remove policy definitions, edit the action script, or change the scan parameters. When you edit policy definitions, it does not affect existing policy instances created from it. The changes are implemented only to new policy instances.

**Prerequisites**
- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- To view examples about policy definition and action, select Catalog > Policies to view the predefined policies.
Verify that a policy version is available in the vCloud Application Director catalog. See “Add a Policy to the Catalog,” on page 124.

Create a copy of the policy to preserve the original for future reference and edit the copied version.

Procedure
1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Policies.
2. Click New in the toolbar.
3. Copy an existing policy version.
   a. Open a policy version and click Copy in the toolbar.
      When you copy a policy, the action script and all of the properties are retained. You cannot change the name of the existing policy version.
   b. Change the version number, update the description, and click Save.
      The version number and description differentiate the new policy from the parent version.
4. Modify the existing policy properties and action script.
   a. Click Edit in the toolbar.
   b. Make your changes and click Save.

What to do next
Delete a duplicate or outdated policy. See “Delete a Policy,” on page 128.

Delete a Policy
If the policy definitions are not applicable to your application deployment, you can delete the policy from the vCloud Application Director catalog.

Prerequisites
- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- Verify that the policy does not have existing policy instances in the deployment environment. See “Create a vCloud Director Deployment Environment,” on page 79, “Create a vCloud Automation Center Deployment Environment,” on page 59, or “Create an Amazon EC2 Deployment Environment,” on page 89.

Procedure
1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Policies.
2. Click New in the toolbar.
3. Open the existing policy.
4. Remove the existing policy version and confirm your selection.
5. Click the arrow next to the policy name to return to the Policies page.
6. Delete the policy and confirm your selection.
Add a Custom Task to the Catalog

With vCloud Application Director, you can create a custom task to perform customized tasks in the application deployment such as run security patches.

The vCloud Application Director catalog contains predefined tasks to configure APT or YUM repositories, a script to register a machine to the Red Hat network, or a script to enable a virtual machine to connect to the designated Windows Active Directory domain. You can add these customized tasks to the execution plan in a deployment profile. In some cases, rather than creating a custom task, you might prefer to edit a predefined task.

Save your changes frequently. The vCloud Application Director user interface session expires after 30 minutes of inactivity. If the session expires, any changes that were not saved are lost.

Prerequisites

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
- Verify whether you must add items to the operating systems and descriptive tag lists. See “Add Operating Systems to the Catalog,” on page 104 and “Add Tags to the Catalog,” on page 104.
- If a script in a task requires access to URLs from outside the corporate firewall, configure the applicable services and application components to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Tasks.
2. Click New in the toolbar.
3. Set the name of the custom task, add a description, and click Save.
   As a best practice, use a name that corresponds to the task the script performs.
4. Click Create Task Version to create a custom task version.
   You can create multiple versions for a custom task.
   A page appears for creating a custom task version.
5. Complete the custom task version information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Task version name stays the same.</td>
</tr>
<tr>
<td>Description</td>
<td>Add detailed information about the custom task. For example, describe what</td>
</tr>
<tr>
<td></td>
<td>the script does when added to a node in the execution plan.</td>
</tr>
<tr>
<td>Version</td>
<td>You can specify major, minor, or micro releases, with or without qualifiers.</td>
</tr>
<tr>
<td></td>
<td>For example, you might use unique version numbers such as 1.0 or 1.5.</td>
</tr>
<tr>
<td>Supported OSes</td>
<td>If the scripts used in this task can run only on particular operating systems, select those operating systems here. In the execution plan, vCloud Application Director prevents the custom task from appearing in the Catalog Task Name list unless it contains one of these operating systems. You must add at least one operating system in the text box. To create an operating system name that is not in the list, click Cancel, and select Catalog &gt; Operating Systems.</td>
</tr>
</tbody>
</table>
6  In the Properties section, define the variables used for a custom task.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Property Name</strong></td>
<td>Click the first row of the Property Name column to define a variable. For example, you might create a custom task to configure a repository or create a custom email task to send a notification email when the deployment task for a service or application component successfully finishes.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Include details about the custom task.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>To change the type, select a type from the drop-down menu. The available property types are String, Content, and Array. You cannot add types to the menu.</td>
</tr>
<tr>
<td><strong>Value</strong></td>
<td>Type the value to substitute for this property when the custom task runs. For example, for a property called remove_all, you might type either true or false in this text box.</td>
</tr>
<tr>
<td><strong>Secured</strong></td>
<td>Select the check box for passwords you define or other properties whose values you want to obscure. For example, the JBOSS_JMX_PWD property in the JBoss service is secured. If a property is changed from Secured to Unsecured, vCloud Application Director resets the property value, for security purposes. You must set a new value for the property.</td>
</tr>
<tr>
<td><strong>Delete button</strong></td>
<td>Removes the selected row from the Properties section.</td>
</tr>
</tbody>
</table>

7  In the Script Editor section, add a script for the custom task.

A catalog administrator can parameterize the installation and configuration of services. The properties that are defined for a service can be used inside the script.

a  Expand the Edit script dialog box, set the script type from the drop-down menu to write a script or copy a script.

b  Select the Reboot check box to restart the virtual machine after the script runs successfully, during an application deployment.

c  Click the down arrow in the Select a property to insert list to add custom task properties.

d  Click OK when you are finished.

8  When you finish creating the custom task, click Save.

The custom task you created is added to the Tasks page.

**What to do next**

You can add a custom task in the execution plan and deploy it to a deployment environment. See “Review the Execution Plan and Add Custom Tasks,” on page 153. Modify the custom task to support your current needs. See “Maintain Custom Task Versions,” on page 130.

**Maintain Custom Task Versions**

You can configure the script or redefine existing properties for a custom task version. To preserve the original custom task version you can create a copy and modify the copied version.

**Note**  If a custom task is used in a deployment profile to customize an application deployment, the custom task cannot be deleted.

**Prerequisites**

- Verify that your user account has the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.
Verify whether you must add items to the operating systems and descriptive tag lists. See “Add Operating Systems to the Catalog,” on page 104 and “Add Tags to the Catalog,” on page 104.

If a script in a task requires access to URLs from outside the corporate firewall, configure the applicable services and application components to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.

Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Catalog > Tasks**.
2. Copy a custom task from an existing custom task version.
   a. Open the custom task version and click **Copy** in the toolbar.
      All of the properties, script, and operating systems are retained. You cannot change the name of the existing custom task version.
   b. Change the version number, update the description, and if needed update the operating system.
      The version number and description differentiates the new custom task from the parent version.
   c. Click **Save**.
3. To change existing properties or an existing script, click **Edit** in the toolbar, make your changes, and click **Save**.
vCloud Application Director provides a drag-and-drop canvas that application architects can use to model an application blueprint. With this blueprint, you can create applications for deployment on a cloud.

You can drag the following items to the blueprint.

- Logical templates from the catalog
  
  You can use the vCloud Application Director sample templates or create custom templates. The cloud templates that contain operating system images and might have services preinstalled and configured are mapped to vCloud Application Director logical templates.

- Application infrastructure components and scripts from the catalog
  
  The included sample services are reusable software components. vCloud Application Director includes installation and configuration scripts that follow best practices for services.

- Application components
  
  These applications operate on top of the services. After you configure an application component, you can deploy it on a compatible service or an operating system image.

NOTE For information about deleting an application deployment from vCloud Application Director, see “Delete an Application Deployment from vCloud Application Director,” on page 199.

Familiarize yourself with the key concepts that appear frequently in topics about creating applications. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Create an Application,” on page 134
- “Creating an Advanced Blueprint,” on page 138
- “Copy an Application Version,” on page 142
- “Copy an Application,” on page 143
- “Delete an Application Version,” on page 143
- “Working with Marketplace,” on page 144
Create an Application

With vCloud Application Director, you can model your application deployment, create dependencies, and edit the application configurations.

Procedure

1 Create an Application Version on page 134
   You can create multiple application versions of an application. When you create an application version, the application name remains the same, but the blueprint contents are not copied from the existing application. You must create a blueprint for the new application version.

2 Model an Application Blueprint on page 135
   The application blueprint provides detailed control over installation dependencies, configuration changes, and editable scripts. vCloud Application Director generates execution plans from the blueprint, which you can revise and use to deploy applications on supported cloud environments.

3 Configure an Application Blueprint on page 137
   You can configure the properties and action scripts of the services and application components to customize your application deployment.

Create an Application Version

You can create multiple application versions of an application. When you create an application version, the application name remains the same, but the blueprint contents are not copied from the existing application. You must create a blueprint for the new application version.

You can share applications publicly or privately in a group. When you share an application, all of the components in the application blueprint such as logical templates, services, and external services must be explicitly shared. Deployment profiles associated with the application are not shared. They are always private to a group.

**NOTE** To create an application version, your current group must be the owning group.

Prerequisites

- Verify that your user account has the ROLE_APP_ARCHITECT application architect role assigned to it.

- For ideas about how to name the application and what text to put in the description text boxes, see the predefined applications already included on the Applications page. To see these applications, you must log in as a user that is a member of the Default group.

Procedure

1 On the vCloud Application Director title bar, click the drop-down menu and select Applications.

2 Click New > Create New Application in the toolbar.

3 Set the name of the application, add an optional description, and click Save.

4 Click Create App Version to create an application version, add the major and minor version of the application.
   For example, you might also add a qualifier to the major and minor versions such as 1.0.0-SNAPSHOT.

5 Click Save when you finish.
   The application is saved and the new application version appears in the Application Versions section of the original application.
What to do next

Model an application blueprint for the application version. See “Model an Application Blueprint,” on page 135.

Model an Application Blueprint

The application blueprint provides detailed control over installation dependencies, configuration changes, and editable scripts. vCloud Application Director generates execution plans from the blueprint, which you can revise and use to deploy applications on supported cloud environments.

You model and create an application in the blueprint canvas. On the left side of the blueprint canvas are the logical templates from the vCloud Application Director catalog. On the right is a list of the services available from the catalog and application components. You can select a logical template and drag it to center of the blueprint canvas to start modeling your application.

Save your changes frequently. The vCloud Application Director user interface session expires after 30 minutes of inactivity. If the session expires, any changes that were not saved are lost.

Prerequisites

- Verify that your user account has the ROLE_APP_ARCHITECT application architect role assigned to it.
- Verify that at least one application version is created in vCloud Application Director. See “Create an Application Version,” on page 134.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Select an existing application and open an application version.
3. Click Create Blueprint to create an application blueprint.
   The canvas for modeling the application blueprint appears.
4. Select and drag one or more logical templates to the canvas to create nodes.
   For example, to create a three-tiered application, you might drag three items from the OS Templates list to the canvas, or drag one template from the OS Templates list, one item from the Database Servers list, and one from the Application Servers list. The names of the lists correspond to the tags associated with a template.
5. (Optional) Select one of the nodes on the canvas and change the name of the node on the Details tab below the blueprint.
   The name of the blueprint node must meet the following naming conventions:
   - Limited to 15 characters.
   - If the node name contains a character other than a letter, digit, underscore, or hyphen, that character is replaced with a hyphen.
   For example, for a three-tiered application, you might rename each of the three nodes as Application_Server, Database_Server, and Load_Balancer.
6. (Optional) Set a host name on the Details tab below the blueprint if you plan to deploy the application to vCloud Director or vCloud Automation Center.
   This host name serves as an identifier for the virtual machine computer name in its network. If the host name text box is left blank, vCloud Application Director generates a host name with random characters.
   The host name must meet the following naming conventions:
   - Limited to 15 characters.
Must begin with a letter.

- Can contain a letter, digit, or hyphen, but cannot end with a hyphen.
- Cannot have the same host name as another node in the application blueprint.
- Can include a \${random} expression at the end of a host name to generate unique characters.

For example, an Apache\${random} host name might generate characters such as Apache9INOIK3YT after an application is deployed. You can view the host name with unique characters from the deployment summary page.

For a clustered node, the host name cannot exceed 15 characters with the appended node array index.
For example, a clustered node called AppServer has the host names of the virtual machines in the cluster as AppServer-1, AppServer-2, and so on.

7 (Optional) To change the default number of CPUs or amount of memory for a deployed virtual machine, select the relevant node on the canvas and edit the values on the Details tab below the blueprint.

The tabs that appear below the blueprint correspond to the node selected.

The CPU and memory values in the blueprint might not match the corresponding values in the virtual machines created in Amazon EC2. Because Amazon EC2 allows virtual machines with specific CPU and memory combinations, it uses the values you assign in a blueprint to determine the closest possible match.

8 Select and drag one or more services or application components to the nodes.

For example, you might drag the MySQL service onto a database server node, drag the JBoss service onto an application server node, and drag an Apache service onto the Load Balancer node.

If a service or application component is not compatible with a particular node, you cannot drop it on the node. For example, you can drag the application component called SQL SCRIPT onto a MySQL service, but you cannot drag the SQL SCRIPT component onto a JBoss service.

Compatibility restrictions are created when the catalog administrator sets the supported operating systems and components for a catalog service. The catalog administrator can also add to the list of operating systems and tags that are already available in the catalog. For example, the MySQL service in the catalog has the supported components listed as SQL SCRIPT. Only the SQL SCRIPT application component type can be added to the MySQL service.

In addition, MySQL service has the supported operating systems set to CentOS32 6.3.0, CentOS64 6.3.0, CentOS32 6.4.0, CentOS64 6.4.0, and Ubuntu32 12.4.2. The MySQL service can be added to logical templates that include one of the operating systems.

**Note** If any preinstalled services are added to a logical template after you create a blueprint, the new preinstalled services are not added to the node. In this case, you must recreate the node and add the preinstalled services.

You can add the application components SCRIPT and Other to a node or any service.

9 Click Save when you finish.

**What to do next**

Configure the services and application components you added to the application blueprint. See “Configure an Application Blueprint,” on page 137.
Configure an Application Blueprint

You can configure the properties and action scripts of the services and application components to customize your application deployment.

Prerequisites

- Verify that your user account has the ROLE_APP_ARCHITECT application architect role assigned to it.
- To examine the sample applications and their components, log in to vCloud Application Director with a user account that belongs to the Default group.
- If the application requires access to URLs from outside the corporate firewall, configure the applicable services and application components to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- Familiarize yourself with the basic concepts of binding to another property if you plan to customize your application. See “Binding to Other Properties,” on page 97.
- Make sure that there are services and application components in the application blueprint. See “Model an Application Blueprint,” on page 135.

Procedure

1. Select a service or application component and edit the information on the Details and Actions tabs below the blueprint.

   Only those properties that the catalog administrator designated as overridable can be changed on the application blueprint.

   On the Actions tab, scripts are accessible for all stages of the component's life cycle, including install, configure, start, update, rollback, and teardown. A catalog administrator can edit a service script from Catalog > Services.

2. To edit a property, click the property in the table.

   If you have a single or clustered node, you must provide a value for the required property in at least one of the life cycle stages of the property. For example, to run an Apache Tomcat server, Java is required and the JAVA_HOME property value must be set.

   The Edit Property dialog box appears.

3. (Optional) To bind a property to another property, select the property value from the Blueprint Value drop-down menu in the Edit Property dialog box.

   Binding to another property lets you customize a script based on the value of other node's run time property values such as IP addresses.

4. When you are finished creating the blueprint for the application, click Save.

   vCloud Application Director checks the application topology you created and displays a message box listing any errors. For example, you see a message if a property type that you selected is not compatible with a script type, or if a service or component is missing a required script. Some errors require correction before you can save the application.

What to do next

Create a deployment profile. See “Create a Deployment Profile,” on page 148.
Creating an Advanced Blueprint

With vCloud Application Director, you can create an advanced application blueprint with dependencies between components, clustered nodes, and multiple networks.

**Figure 11-1. Clustered Windows Application Example**

- **Create a Dependency Between Components** on page 139
  Dependencies are added in the blueprint to define an order in which the deployment tasks must be performed. Creating a dependency link from one item such as a service or application component to another service or application component guarantees that the task of creating the first item finishes successfully before a second task begins.

- **Specify a Node as a Cluster** on page 139
  For scaling deployments, you might need to deploy multiple virtual machines or a cluster for a particular node and use a load balancer to manage them.

- **Define Multiple NICs for a Node** on page 140
  In most deployments, some servers are deployed to a DMZ zone and some servers are deployed to a network protected by a firewall.

- **Add and Manage a Flexible Disk Layout** on page 140
  Flexible disk layout enhances the storage flexibility and lets you add additional disks to a node. You can also add the disks when you create an application blueprint. The disks are created dynamically during provisioning and added to the node.

- **Add an External Service to an Advanced Blueprint** on page 142
  You can add basic or advanced external services to the blueprint that are installed external to the deployment of the vCloud Application Director application. External services are commonly used because these services are not provisioned as part of the blueprint, but the application needs the external service in order to work.
Create a Dependency Between Components

Dependencies are added in the blueprint to define an order in which the deployment tasks must be performed. Creating a dependency link from one item such as a service or application component to another service or application component guarantees that the task of creating the first item finishes successfully before a second task begins.

Procedure

1. To create dependencies between services or application components, click the Add Relation button in the toolbar above the canvas.
2. Select the first component, then the component on which it depends.
   
   For example, because a load balancer usually cannot be configured until the application is up and running, you might add a dependency from an Apache service to a WAR component.
   
   A blue dotted line appears and points to the dependent component.
3. When you are finished, click Save in the toolbar above the canvas.

Example: Create Dependency from JBoss to MySQL

For example, you might create a dependency line from the JBoss service to the MySQL service.

This blue line that appears indicates that the application server JBoss, is dependent on having the MySQL database, created and configured.

What to do next

Deploy the application. See Chapter 12, “Deploying Applications,” on page 147.

Specify a Node as a Cluster

For scaling deployments, you might need to deploy multiple virtual machines or a cluster for a particular node and use a load balancer to manage them.

Prerequisites

Familiarize yourself with the basic concepts of binding to another property, node array index property, and defining component actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.

Procedure

1. To specify a cluster of virtual machines, click the Convert to Node Array icon in the node.
2. Below the blueprint, set the cluster size.
3. Bind to a property like node_array_index to identify in which virtual machine the current script is running.
   
   You can find the IP addresses of all the virtual machines in a cluster by binding a property to all(node:ip).
4. (Optional) If other properties refer to a cluster property, define the component properties to access the array of property values from the clustered nodes.
5. Click Save in the toolbar above the canvas.
Example: Specifying the Load Balancer

To understand how a cluster is used, see the Clustered Duke’s Bank sample application. In the Load Balancer node, click **Apache_LB** and note that the `http_node_ips` property refers to all(appserver:ip). The `https_node_ips` property references all of the IP addresses for each node within the specified cluster size.

**What to do next**

Deploy the application. See Chapter 12, “Deploying Applications,” on page 147.

Define Multiple NICs for a Node

In most deployments, some servers are deployed to a DMZ zone and some servers are deployed to a network protected by a firewall.

In the Clustered Duke’s Bank sample application, the Load Balancer node is the only node that you should access from a public network. The Database and Appserver nodes must be deployed in a private network behind a firewall. The Load Balancer node must also have access to the Database and AppServer nodes.

In vCloud Application Director, to resolve this situation, you can define two NICs on the Load Balancer. Each NIC must specify a logical network name. At deployment time, the logical network is mapped to an actual cloud network. When a virtual machine is created, the number of NICs for the virtual machine are derived from the node.

In the Duke’s Bank sample application, the Load Balancer node has two NICs, NIC0 pointing to ServiceNetwork and NIC1 pointing to MgmtNetwork. Database and Appserver nodes have one NIC pointing to the ServiceNetwork. At deployment time, ServiceNetwork can be mapped to a cloud network protected by firewall and MgmtNetwork can be mapped to a public cloud network.

**Prerequisites**

Familiarize yourself with the predefined IP address property concept when you have multiple NICs in a virtual machine. See “Predefined IP Address Property,” on page 100.

**Procedure**

1. To add multiple NICs, select a node and click the **NICs** tab below the node. You can add up to 10 NICs to a node.
2. To add a NIC, click the **Add** icon (+) and specify a logical network name.
3. To be part of the same network as another node, pick the network name from the drop-down menu or type a new network.
4. When you are finished, click **Save** in the toolbar above the canvas.

**What to do next**

Deploy the application. See Chapter 12, “Deploying Applications,” on page 147.

Add and Manage a Flexible Disk Layout

Flexible disk layout enhances the storage flexibility and lets you add additional disks to a node. You can also add the disks when you create an application blueprint. The disks are created dynamically during provisioning and added to the node.

You can also manage placing the disks on different datastores. Once the user defines the flexible disk layout for nodes in the blueprint, application deployer can further customize the deployment:

- Map each individual disk to specific datastore
Set the disk size for each additional disk

The additional disks are part of virtual machine node. Both the service that uses the disks and the user who uses the service, define the disks. Because of this, some assumptions are made between the service and the user.

For example, if an Oracle service requires at least one data disk and one redo log disk to work, the user must add two disks to the node else the service script does not run correctly.

**Note** You cannot specify different disk settings for an individual node in a same node array.

Databases can leverage flexible disk layout. The new Oracle 11g database service is added in vCloud Application Director as out-of-the-box service. This service is modified to leverage flexible disk layout.

**Prerequisites**

Familiarize yourself with the predefined disk layout Info property when you plan to add multiple disks in a virtual machine. See “Predefined Disk Layout Info Property,” on page 101

**Procedure**

1. Select the node and click **New** on the **Disks** tab below the blueprint.

Consider the following options when you add a disk to your application blueprint:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The unique disk name within the node.</td>
</tr>
<tr>
<td>Mount Path</td>
<td>The path for the disk to be mounted. VCloud Application Director does not mount the disk if the mount path is not specified or if the disk is not formatted.</td>
</tr>
<tr>
<td>File System</td>
<td>File systems such as ext3 or NTFS can be specified.</td>
</tr>
<tr>
<td>Size (GB)</td>
<td>The size of the disk to be provisioned.</td>
</tr>
<tr>
<td>Tags</td>
<td>A string specified that you can leverage to determine the mapping between disk and storage.</td>
</tr>
<tr>
<td>Description</td>
<td>Specify more information for the disk.</td>
</tr>
</tbody>
</table>

2. (Optional) Assign each additional disk with multiple disk tags.

Each disk tag can be used for multiple disks in the disk layout information. The disk tag is used to indicate the disk usage purpose. There are four predefined disk tags optimized for database services:

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archive</td>
<td>Stores database archive files.</td>
</tr>
<tr>
<td>Binary</td>
<td>Stores binary files, such as Oracle binary.</td>
</tr>
<tr>
<td>Data</td>
<td>Stores data files, such as Oracle table space files.</td>
</tr>
<tr>
<td>Log</td>
<td>Stores database online logs.</td>
</tr>
</tbody>
</table>

3. When you are finished, click **Save** in the toolbar above the canvas.

**What to do next**

Deploy the application. See Chapter 12, “Deploying Applications,” on page 147
Add an External Service to an Advanced Blueprint

You can add basic or advanced external services to the blueprint that are installed external to the deployment of the vCloud Application Director application. External services are commonly used because these services are not provisioned as part of the blueprint, but the application needs the external service in order to work.

**Prerequisites**

- Familiarize yourself with the basic concepts of basic or advanced external services. See “Add an External Service to the Catalog,” on page 113.
- Understand the basic concepts of defining a service in vCloud Application Director. See “Define Service Version Properties,” on page 107.
- Familiarize yourself with how to create dependencies among nodes. See “Create a Dependency Between Components,” on page 139.

**Procedure**

1. Open an existing application blueprint.
2. Select and drag an external service to the blueprint canvas.
   For example, you can add the vPostgres external services in the blueprint.
3. (Optional) Create dependencies among applicable components in the blueprint.
   A blue dotted line appears and points to the dependent component.
4. To edit a property, click the property in the table below the blueprint.
   You can edit or define the external service properties just like any other service in the catalog.
   The Edit Property dialog box appears.
5. When you are finished, click **Save** in the toolbar above the canvas.

**What to do next**

Deploy the application. See Chapter 12, “Deploying Applications,” on page 147.

Copy an Application Version

When you copy an existing application version, all of the blueprint contents are copied except deployment profiles, and the application name remains the same. You can only edit the application version and version description.

**Prerequisites**

- Verify that your user account has the **ROLE_APP_ARCHITECT** application architect role assigned to it.
- If you are not familiar with the process of adding components to an application blueprint, creating dependencies between components, or modifying services and scripts, see “Create an Application,” on page 134.
- Verify that the current group you are logged in to owns the application version.
  If you are not part of the group that owns the application version, the **Copy** button is disabled.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Applications**.
2. Open an application and select an application version.

3. Click the Copy button in the toolbar to copy the application version.

4. Change the application version, add a description, and click Save.

   The copied application version opens.

What to do next

Create a deployment profile. See “Create a Deployment Profile,” on page 148.

Copy an Application

Copying an application allows you to keep all of the components of an application version, set a new application name, and change the application version into a standalone application.

Prerequisites

- Verify that your user account has the ROLE_APP_ARCHITECT application architect role assigned to it.
- If you are not familiar with the process of adding components to an application blueprint, creating dependencies between components, or modifying services and scripts, see “Create an Application,” on page 134.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.

2. From the Applications page, point to the application card view.

3. Click the Copy button to copy the application.

4. Select an existing application version from the drop-down menu.

5. Set a new name for the application and add a description.
   You might include the details about the configured properties and services, dependencies, node configuration, or custom tasks in the execution plan.

6. Click OK.
   The new application appears on the Applications page.

What to do next

Create a deployment profile. See “Create a Deployment Profile,” on page 148.

Delete an Application Version

If your application version becomes obsolete, you can remove it from the vCloud Application Director appliance.

When you delete an application version, the blueprints are removed. You cannot delete an application version with deployment profiles.

Prerequisites

- Verify that your user account has the ROLE_APP_ARCHITECT application architect role assigned to it.
If you are not familiar with the process of adding components to an application blueprint, creating dependencies between components, or modifying services and scripts, see “Create an Application,” on page 134.

Verify that the current group you are logged in to owns the application version.

If you are not part of the group that owns the application version, the Delete button is disabled.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Open an application and select an application version.
3. Click Delete and OK to confirm.

The application version is removed and the Application Versions list appears.

Working with Marketplace

Marketplace lets you browse, view, and import blueprints for applications or services from VMware Solution Exchange. You must create a VMware Solution Exchange account to use this feature with Application Director.

Marketplace is disabled by default. To enable Marketplace, run the following command in the command-line interface: `update-global-prop --name VSX_MARKET_PLACE_ENABLE --value true`. After you enable Marketplace for the first time, logout and login to view the changes. You can access Marketplace from the Applications landing page.

A proxy configuration is required for Marketplace if there is any access restriction to connect to internet from within the enterprise. Marketplace requires you to enable proxy settings from the command-line interface to access external servers:

`update-global-prop --name VSX_PROXY_ENABLE --value true`.

You must also run the following commands to configure proxy settings:

- `APPD_HTTP_PROXY_HOST`
- `APPD_HTTP_PROXY_PORT`
- `APPD_HTTP_PROXY_USER`: Required only if the proxy setting needs additional authentication.
- `APPD_HTTP_PROXY_PASSWORD`: Required only if the proxy setting needs additional authentication.

For example, if the proxy server is enabled, to run the host command, use: `update-global-prop --name APPD_HTTP_PROXY_HOST --value proxy.abcd.com`.

From Marketplace you can search for solutions (blueprints for applications or services) by using keywords such as solution title or publisher name.

Import Solutions from Marketplace

Marketplace lets you import blueprints for applications or services from VMware Solution Exchange. Once you enable the Marketplace feature from command-line, you can access it from the Applications landing page. You can import only solutions that contain supported files in Application Director.

Prerequisites

- Verify that you have enabled Marketplace from the command-line interface.
- Verify that you have an user account created for VMware Solution Exchange.
- Verify that you have internet access by enabling proxy settings.
Procedure

1. On the vFabric Application Director title bar, click the drop-down menu and select **Applications**.
   The Applications landing page appears.

2. Click **New** in the toolbar and select **Import From Marketplace**.
   The Marketplace page appears with the Most Recent and Featured solutions.

3. Click the appropriate solution from the list.
   The solution appears with the details of blueprint for an application or a service.

4. Click **Import** if you wish to import this solution. The VMware Solution Exchange login screen appears.
   a. Enter the Solution Exchange **Username** and **Password**. Click **Login**.
   b. The EULA screen appears. EULA is applicable to certain solutions only. Check the box to accept the license agreement and click **Next**.
   c. The blueprint for application or service is validated. It lists any applications or services that are conflicting. Click the drop-down menu for **Conflict Resolution**. Choose any option from following choices to resolve the conflict:
      - **Use Existing**: Any conflicting artifact in the blueprint is skipped and the existing definition is used.
      - **Import as New**: A new copy of each artifact in the blueprint is created. Enter any alphanumeric value in the **Suffix** field that you want to append to the application name.
      - **Overwrite**: Any artifact that is imported overwrites the existing artifact. You cannot overwrite any publicly shared artifacts.

   **NOTE**: If the existing artifacts are shared publicly then the import fails due to sharing level mismatch.
   d. Click **Next**. The Review page appears. Confirm that the information on this page is as desired.

5. Click **Import** to import the blueprint or service.

**What to do next**

Deploy the application. See Chapter 12, Deploying Applications, and Page 135.
vCloud Application Director simplifies and automates deployments of multitier enterprise applications in hybrid cloud environments.

With vCloud Application Director, you can create different deployment profiles for a life cycle, such as development, testing, and production. The deployment profile settings are saved in vCloud Application Director. You can reuse this saved deployment profile to deploy an application version to a supported cloud environment.

If you update a deployed application, vCloud Application Director creates an update profile based on the modified values. This update profile includes an update script generated from the defined update life cycle stage.

Familiarize yourself with the key concepts relating to deploying applications. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Set Up and Configure a Deployment Profile,” on page 147
- “Deploy an Application,” on page 155
- “Publish a Deployment Profile to the vCloud Automation Center Service Catalog,” on page 157
- “Request a vCloud Automation Center Service Catalog Item,” on page 158
- “Understanding the Deployment and Update Process,” on page 159
- “Using the Deployment Summary Page,” on page 160

**Set Up and Configure a Deployment Profile**

Deployment profiles let you configure settings such as cloud templates, networks, and application configuration values that are allowed for use in specific deployment environments.

In the deployment profile you can also review the execution plan, add custom tasks to the execution plan, review the deployment profile settings and make changes before you deploy the application.

**Procedure**

1. **Create a Deployment Profile** on page 148

   Create a different deployment profile for each deployment environment. In a deployment profile, you can enter or override application properties for a specific deployment if the **Override at Deployment** option is enabled for the property.
Configure the Deployment Environment Tab on page 149

You use the deployment environment tab to retrieve the list of cloud templates and networks available in the deployment environment and map them to logical templates and logical networks. You can also configure custom properties in vCloud Application Director to override the vCloud Automation Center blueprint custom properties, mapping pre-defined disks to storage, or add to the existing properties.

Configure the Application Properties Tab on page 152

You can define new values for all the node properties such as host name or vCPU. You can define new values only for application component and service properties that have the Overridable at Deployment check box selected in the application blueprint.

Review the Execution Plan and Add Custom Tasks on page 153

The system generates deployment execution plans based on the application blueprint. You can review the execution plan and add custom tasks to perform additional customized tasks in the application deployment before deploying the application.

Use an Existing Deployment Profile on page 154

You can reuse an existing deployment profile for an application version.

Create a Deployment Profile

Create a different deployment profile for each deployment environment. In a deployment profile, you can enter or override application properties for a specific deployment if the Override at Deployment option is enabled for the property.

For example, for a particular deployment environment, you might change the database port to 3307.

Prerequisites

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that at least one application is created in vCloud Application Director. See Chapter 11, “Creating Applications,” on page 133.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Click the name of the application.
   A list of application versions appears.
3. Select an application version and click Create DP to create a deployment profile that you can save or deploy immediately.
4. (Optional) To create another deployment profile, click New.
5. Set a name for the deployment profile and click Deploy.
   You might name the profile so that the name indicates which type of environment is used, specific override property, or clustered node configuration.
   For example, for an application called MyTimecard, you might name the profile myTimecard-QA to indicate that this profile is for the QA environment rather than for the production environment.

The Deployment Profile wizard appears, with the Deployment Environment page highlighted.

What to do next

Map the logical templates and network templates for the application deployment. See “Configure the Deployment Environment Tab,” on page 149.
Configure the Deployment Environment Tab

You use the deployment environment tab to retrieve the list of cloud templates and networks available in the deployment environment and map them to logical templates and logical networks. You can also configure custom properties in vCloud Application Director to override the vCloud Automation Center blueprint custom properties, mapping pre-defined disks to storage, or add to the existing properties.

The nodes listed in the VM Templates section correspond to the components of the application, as shown in the application blueprint. If the application includes custom external services that require scripts to run, the VM template is mapped to the temporary virtual machine.

The Networking section lists the logical networks defined in the blueprint. The network you select and map depends on the deployment environment. The deployment profile also contains a network list of vCloud Director and Amazon EC2 networks. For vCloud Director, vCloud Application Director supports external networks and vCloud routed networks with or without DHCP. If the network list is empty, contact your vCloud Director administrator. For Amazon EC2, vCloud Application Director supports the NAT-routed, public, and private networks. NAT-routed and public networks can access the Internet. A private network that is not NAT-routed can access only the vCloud Application Director appliance. To deploy to Amazon EC2, you must carefully determine the virtual machine you put on an external network. Every NIC on an external network gets an Elastic IP address, which puts that interface on the Internet. Put a NIC on an external network only when it is absolutely required.

**Note** For vCloud Automation Center, the networking information is not available. vCloud Automation Center uses the network connection specified in the vCenter Server template.

vCloud Automation Center blueprint has custom properties that are defined in the build profiles. These custom properties are applied to a virtual machine when it is created. vCloud Application Director lets you override the vCloud Automation Center blueprint custom properties or add to the existing properties. For example, to override the existing vCloud Automation Center network information you can specify network or static IP addresses in the custom properties for a specific node in the application blueprint. This defined custom property is applied whenever a virtual machine is created.

Custom properties are key-value pairs. You can define these properties as key=value.

**Note** Do not use the vCloud Application Director reserved and internal properties as your custom properties. If you use these properties, you receive an error message.

See “vCloud Application Director Reserved and Internal Properties,” on page 151.

**Prerequisites**

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that at least one cloud template is mapped to each logical template used in the blueprint. See “Add a Logical Template to the Catalog,” on page 121.
- Verify that at least one application is created in vCloud Application Director. See Chapter 11, “Creating Applications,” on page 133.
- Depending on your cloud environment, you must have at least one vCloud Director or Amazon EC2 network available for the deployment environment. See “Create a vCloud Director Deployment Environment,” on page 79 or “Create an Amazon EC2 Deployment Environment,” on page 89.

**Note** For vCloud Automation Center, the networking information is not available. vCloud Automation Center uses the network connection specified in the vCenter Server template.
Register your cloud template to a vCloud Application Director cloud provider. See “Register the vCloud Director Cloud Provider and Template,” on page 78, “Register the vCloud Automation Center Cloud Provider and Template,” on page 57 and “Register the Amazon EC2 Cloud Provider and Template,” on page 88.

Familiarize yourself with custom properties for vCloud Automation Center deployments and reserved custom properties. See vCloud Automation Center Operating Guide.

Verify that a deployment profile is created. See “Create a Deployment Profile,” on page 148.

**Procedure**

1. Select a deployment environment from the list and click **Map Details**.

   You can view the corresponding list for the Deployment Environment section, click the title bar and select **Clouds > Deployment Environments** from the drop-down menu.

2. In the External Services section, map an external service to an external service instance in the deployment environment.

   If the list of external service instance is empty, the existing instance was not registered in the deployment environment. If you have access, register at least one external service instance or ask your cloud administrator to register an instance for you.

   Associated external service instances appear in the drop-down menu.

3. In the VM Templates section, map a logical template to a cloud template in the cloud environment.

   If the list of cloud templates is empty, the existing cloud templates do not belong to your group or a cloud template was not registered. If you have access, log in to the group that has existing cloud templates, or register at least one cloud template in vCloud Application Director. You can also ask your cloud administrator to register a cloud template for you.

   For vCloud Automation Center, the cloud templates that have the same reservation policy as the deployment environment appear in the drop-down menu.

   Cloud templates that belong to the same group as the user appear in the drop-down menu.

4. (Optional) For vCloud Automation Center deployments, click the **Extra Configuration** icon (🔍) to add custom properties to each node in the application blueprint.

   The custom properties should not be vCloud Application Director reserved and internal properties. If you use these properties, you receive an error message.

   a. Define custom properties to map the Management Network to a vCenter Server direct network and the Service Network to a vCenter Server routed network for the sample Clustered Dukes Bank application.

      The vCenter Server direct network is network1 and the vCenter Server routed network is network2.

   b. In the appserver node row, click the **Extra Configuration** icon (🔍), type `virtualmachine.network0.name=network2` in the Extra Configuration Information for appserver dialog box, and click **Save**.

   c. In the database node row, click the **Extra Configuration** icon (🔍), type `virtualmachine.network0.name=network2` in the Extra Configuration Information for database dialog box, and click **Save**.

   d. In the load_balancer node row, click the **Extra Configuration** icon (🔍), type `virtualmachine.network0.name=network1 virtualmachine.network1.name=network2` in the Extra Configuration Information for load_balancer dialog box, and click **Save**.
5 For vCloud Director and Amazon EC2 deployments, in the Networking section select a supported cloud network for each logical network in the catalog.

For example, for a load balancer, if you are deploying the application to a test environment, you might select an internal network for both load balancer network NICs. When you create a deployment profile for the production environment, you might select an internal network for one load balancer NIC and an external network for the other load balancer NIC.

6 In the Disks section, map each individual disk to a specific storage.

The Disks section might be empty in the following scenarios:

- The corresponding application blueprint might not define additional disks. In such a scenario, leave the section empty and continue the deployment.

- A non-vCloud Automation Center deployment environment is selected. In such a scenario, you can safely ignore the warning message, Selected Deployment Environment does not allow customization of disks in the VM. Disks defined in the blueprint will not be provisioned. The deployment might fail if the action scripts depend on customized disk settings in the blueprint and continue with the deployment.

Flexible disk layout lets you place the disks in specific datastores to gain maximum performance and minimum cost.

For example, create the operating system disk on a fast datastore to gain better performance and create an archive disk on a slow datastore to reduce the cost.

7 When you finish making your selections, click Next.

The Application Properties tab appears.

What to do next

Define the applicable property values for the node, service, and application components. See “Configure the Application Properties Tab,” on page 152.

vCloud Application Director Reserved and Internal Properties

There are reserved and internal properties that you cannot use when you create custom properties. If you use these properties, you receive an error message.

List of Reserved and Internal Properties

<table>
<thead>
<tr>
<th>Reserved Properties</th>
<th>Internal Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>agent.download.url</td>
<td>Hostname</td>
</tr>
<tr>
<td>agent.jar.md5sum</td>
<td>VirtualMachine.Cpu.Count</td>
</tr>
<tr>
<td>agent.max.logsize</td>
<td>VirtualMachine.Memory.Size</td>
</tr>
<tr>
<td>agent.root.folder</td>
<td>VirtualMachine.Disk0.Size</td>
</tr>
<tr>
<td>amqp.heartbeat</td>
<td>VirtualMachine.Admin.CustomizeGuestOSDelay</td>
</tr>
<tr>
<td>amqp.host</td>
<td>VirtualMachine.Admin.UseGuestAgent</td>
</tr>
<tr>
<td>amqp.port</td>
<td>VirtualMachine.Software0.Name</td>
</tr>
<tr>
<td>queue.name</td>
<td>VirtualMachine.Software0.ScriptPath</td>
</tr>
<tr>
<td>server.url</td>
<td></td>
</tr>
<tr>
<td>temp.key</td>
<td></td>
</tr>
</tbody>
</table>

APPD_REQUEST_ID

DEPLOYMENT_NAME
Reserved Properties | Internal Properties
---|---
DEPLOYMENT_URI | 
DEPLOYMENT_LOCATION_URI | 
VM_URI | 
VM_NAME | 
os.M_NAMEfamily | 
APPPD_REQUEST_ID | 

### Configure the Application Properties Tab

You can define new values for all the node properties such as host name or vCPU. You can define new values only for application component and service properties that have the **Overridable at Deployment** check box selected in the application blueprint.

For successful deployment, assign a value to the required node properties from the catalog, blueprint, or deployment profile. The system defines node properties, such as memory allocation and number of CPUs, but you can override them. For example, the vFabric tc Server service might have a JVM heap size of 512MB. But for a large deployment, you can override that setting and change the size to 1024MB.

You can set a host name so that the virtual machine can be easily identified in the vCloud Director or vCloud Automation Center deployments. To generate unique characters you can append the ${random} expression at the end of a host name.

The cluster size of a clustered node can also be defined with new values.

**Prerequisites**

- Verify that your user account has the **ROLE_DEPLOYER** deployer role assigned to it.
- Verify that at least one application is created in vCloud Application Director. See Chapter 11, “Creating Applications,” on page 133.
- Verify that the **Deployment Environment** tab is configured. See “Configure the Deployment Environment Tab,” on page 149.

**Procedure**

1. Verify that the properties you set in the blueprint appear accurately in the component tabs.
2. (Optional) Click the **Service**, **Application Component**, or **Node** tab.
3. Select a specific service, application component, or node property.
4. In the table, click the **New Value** column of a specific row and type the value to use in the deployment profile.
   
   For a deployment to Amazon EC2, you must change the global_conf property value to
   
   `https://DarwinServerIP:8443/darwin/conf/darwin_global_noproxy.conf`

5. To revert to the original value, click the **Reset Value** button (✏).
6. (Optional) In the Disk Mappings section, type a new integer value to increase or decrease the disk size.
7. When you finish making changes, click **Next**.

The **Execution Plan** tab appears.

**What to do next**

Review the provisioning tasks and dependencies in the execution plan. See “Review the Execution Plan and Add Custom Tasks,” on page 153.
Review the Execution Plan and Add Custom Tasks

The system generates deployment execution plans based on the application blueprint. You can review the execution plan and add custom tasks to perform additional customized tasks in the application deployment before deploying the application.

The blue dotted lines in the execution plan define a specific order in which the deployment tasks run. Host and agent bootstrap provisioning tasks appear next to components for each node. For applications deployed to the vCloud Automation Center environment, in addition to the host and agent bootstrap tasks, the network bootstrap provisioning task appears. These provisioning tasks display the processes that take place before the agent performs the installation and setup tasks for each component. When a deployment fails, you can see the provisioning task logs to troubleshoot the problem. You cannot add custom tasks between host, agent bootstrap, or network bootstrap provisioning tasks in an execution deployment plan.

If an application includes external services that require scripts to run, temporary virtual machines appear in the execution plan. vCloud Application Director removes these virtual machines after scripts run successfully in the host and agent bootstrap provisioning tasks. If the application fails before the temporary virtual machines are removed, you must identify the nodes that include the external services virtual machines and remove them from your cloud environment.

A blueprint helps to generate a common execution plan for an application on all of the deployment environments. Sometimes, you must customize the execution plan for each deployment environment. For example, when an application is deployed to the production deployment environment, you might need to send an email after deploying. In the test deployment environment, such checks might not be required. You can create an email custom task to send a notification email when the deployment task for a service or application component successfully finishes. You can add this task to the execution plan in the deployment profile, which deploys to the production deployment environment.

**CAUTION** Verify that no processes are prompting for user interaction when the custom task is running. Any interruption pauses the task, causing it to remain in an idle state indefinitely. You can cancel the application deployment after an hour or vCloud Application Director fails the deployment in an idle state after three days.

**Prerequisites**

- Verify that your user account has the **ROLE_DEPLOYER** deployer role assigned to it.
- Verify that at least one application is created in vCloud Application Director. See Chapter 11, “Creating Applications,” on page 133.
- Verify that the **Application Properties** tab is configured. See “Configure the Application Properties Tab,” on page 152.
- Verify that at least one custom task is created in the vCloud Application Director catalog. See “Add a Custom Task to the Catalog,” on page 129.
- Verify that all of the required node properties in the application are assigned a value for successful deployment.
- Depending on your cloud environment, you must have at least one vCloud Director or Amazon EC2 network available for the deployment environment. See “Create a vCloud Director Deployment Environment,” on page 79 or “Create an Amazon EC2 Deployment Environment,” on page 89.

**NOTE** For vCloud Automation Center, the networking information is not available. vCloud Automation Center uses the network connection specified in the vCenter Server template.

- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
Procedure

1. Review the provisioning tasks, components, and dependencies in the execution plan.

2. Click the Expand Cluster button to expand the node, if the node is clustered.

   If the clustered node is not expanded, the custom task is added only to the first virtual machine in the cluster. If an application architect modifies a node to a clustered node, an existing custom task is applied to only the first virtual machine in the cluster. A deployer should check during deployment whether the custom task applies to the first virtual machine or to all the virtual machines in the cluster, and perform the appropriate steps.

3. For services and components that have scripts associated with them, click the down arrow next to the component or service name to view the script or the variable definitions used in the script.

4. (Optional) Select the Add Script Task button and drag the custom task to the node.

   When you drag the Add Script Task button, you see anchors that indicate where you can insert the custom task. For a clustered node, add the custom task to each node.

   For example, you can drag one or more custom tasks to the Application Server, Database Server, or Load Balancer node.

   After you drop a custom task to a node, the Add Custom Task dialog box opens.

5. (Optional) Select a task from the Catalog Task Name drop-down menu.

   The custom task supported for that node’s operating system appears. For example, if a custom task is supported on the CentOS 6.3 operating system and the operating system of the node is Ubuntu 12.4.2, the task is not listed in the menu.

   When you select a custom task, the task, script, and property details appear in the dialog box.

6. (Optional) To override a property value on the Properties tab, click the property.

   For example, in a send email custom task, one of the properties is the recipient’s email address. You can set the property email address value to the recipient email address.

   a. In the Edit Property dialog box, type the new value for the property or select an existing property from the drop-down menu to bind the property to one of the properties in the application blueprint.

   b. Click Save.

7. (Optional) In the Add Custom Task dialog box, review the script and property details of the custom task, and click OK.

8. Click Next to review the deployment profile settings and click Save.

9. Click OK to confirm.

   The deployment profile is listed for the application version.

What to do next

Use the deployment profile to deploy the application. See “Deploy an Application,” on page 155.

Use an Existing Deployment Profile

You can reuse an existing deployment profile for an application version.

Prerequisites

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
Register the uploaded CentOS 6.3 32-bit template to a cloud provider. See “Register the vCloud Director Cloud Provider and Template,” on page 78, “Register the vCloud Automation Center Cloud Provider and Template,” on page 57, or “Register the Amazon EC2 Cloud Provider and Template,” on page 88.

Verify that at least one cloud template is mapped to each logical template used in the blueprint. See “Add a Logical Template to the Catalog,” on page 121.

Verify that a deployment profile is available in vCloud Application Director. See “Set Up and Configure a Deployment Profile,” on page 147.

Depending on your cloud environment, you must have at least one vCloud Director or Amazon EC2 network available for the deployment environment. See “Create a vCloud Director Deployment Environment,” on page 79 or “Create an Amazon EC2 Deployment Environment,” on page 89.

**NOTE** For vCloud Automation Center, the networking information is not available. vCloud Automation Center uses the network connection specified in the vCenter Server template.

### Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select **Applications**.
2. Click the name of the application.
   
   A list of application versions appears.
3. Select an application version and click the existing deployment profile.
   
   The Deployment Profile wizard opens.
4. (Optional) If you created new nodes in the blueprint, map the new nodes to a cloud template.

   Property overrides saved in an existing deployment profile remain overridden in the new value column even if the blueprint value is updated. Deleted properties no longer appear in the deployment profile.

### What to do next

Complete the deployment tasks to deploy the application. See “Create a Deployment Profile,” on page 148 and “Deploy an Application,” on page 155.

### Deploy an Application

After you save a deployment profile, you can deploy the application from the vCloud Application Director user interface.

You can also deploy an application from the command-line interface. See “Deploying and Updating an Application Using CLI,” on page 208.

### Prerequisites

- Verify that your user account has the **ROLE_DEPLOYER** deployer role assigned to it.
- Verify that at least one deployment profile is available in vCloud Application Director. See “Create a Deployment Profile,” on page 148.

### Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select **Applications**.
2. Click the name of the application.

   A list of application versions appears.
3. Select an application version and create a deployment profile or use an existing profile.
Follow the prompts in the Deployment Profile wizard and make any applicable changes.

Review the deployment profile settings.
You can make necessary changes to the deployment profile before deploying the application.

Click Deploy.
A deployment summary page appears and refreshes in approximately 30 seconds to display the deployment status.

(Optional) Click Refresh in the toolbar to update the real-time status of the deployment.

What to do next
Learn about the processes that take place in the background when an application is deployed to the cloud. See “Understanding the Deployment and Update Process,” on page 159.
You can also check the status of an in-progress deployment. See “Using the Deployment Summary Page,” on page 160.

Quick Deploy an Application
When you tear down a deployed application from the cloud environment, you can quickly redeploy the application without configuring the elements in the Deployment Profile wizard.

With quick deploy, you can also update required properties and overridable properties outside the Deployment Profile wizard and deploy the application blueprint.

When you quick deploy an application, the latest application version and associated deployment profiles are available for deployment. To quick deploy an older application version, open the application and quick deploy the specific application version.

NOTE For a vCloud Automation Center deployment, if you add another NIC to an application and use the existing deployment profile to quick deploy the application you receive an error. You must open the Deployment Profile wizard and select Map Details to map the cloud network to the newly added NIC.

Prerequisites
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that at least one deployment profile is available in vCloud Application Director. See “Create a Deployment Profile,” on page 148.
- Verify that the current group you are logged in to owns the associated deployment profile.
  If you are not part of the group that owns the associated deployment profile, the Quick Deploy button is disabled.

Procedure
1 On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2 From the Applications page, select the application to deploy.
3 Select an application version to Quick Deploy.
4 Click the Quick Deploy button to initiate the deployment process.
  If an application version is not selected, by default the latest application version is deployed.
5 In the Destination drop-down menu, select the associated deployment profile.
  The properties that are required for deployment and overridable at deployment appear.
6  (Optional) Set a new value for a required or overridable property and click **Deploy**. The modified value for a required or overridable property is not saved in the deployment profile for future deployments.

The deployment summary page appears and displays the status of the in-progress deployment.

**What to do next**

Use the status windows on the deployment summary page to track the deployment status. See “Using the Deployment Summary Page,” on page 160.

---

**Publish a Deployment Profile to the vCloud Automation Center Service Catalog**

With vCloud Application Director you can publish a deployment profile of an application to the vCloud Automation Center service catalog. vCloud Automation Center users can request the catalog item to start using the application in that environment.

**Note** You cannot delete a published deployment profile from the vCloud Automation Center service catalog.

**Prerequisites**

- Make sure that you log in to the vCloud Application Director appliance as an SSO user that has **APPLICATION PUBLISHER** and **DEPLOYER** roles assigned to it from the vCloud Automation Center user interface.
- Verify that a deployment profile is available in vCloud Application Director. See “Set Up and Configure a Deployment Profile,” on page 147.
- Verify that vCloud Application Director is registered as a vCloud Automation Center endpoint. See “Register vCloud Application Director to vCloud Automation Center,” on page 30.
- The vCloud Automation Center 6.0 IaaS environment and entitlements must be properly set up and configured. See vCloud Automation Center Installation and Configuration Guide and vCloud Automation Center Virtual Provisioning Guide.
- Make sure that if there is a custom task in the deployment profile, all of the required property values are defined so that the SSO user can request the item from the vCloud Automation Center service catalog or quick deploy the latest application version and associated deployment profile.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Applications**.
2. Click the name of the application.
   A list of application versions appears.
3. Select an application version and click the existing deployment profile or create a deployment profile.
   The Deployment Profile wizard opens.
4. (Optional) If you created new nodes in the blueprint, map the new nodes to a cloud template.
   Property overrides saved in an existing deployment profile remain overridden in the new value column even if the blueprint value is updated. Deleted properties no longer appear in the deployment profile.
5. Review the deployment profile settings.
   You can make necessary changes to the deployment profile before publishing the application.
6. Click **Publish**.
7 In the Publish dialog box, append the name of the deployment profile with more information and click OK.

For example, you can append Publish_vCAC to the ClusteredDukesBankDP deployment profile. In the Description section, add an optional description such as, the Dev, QE, or IT environment where the deployment profile is going to be used.

The published name and description appears in the vCloud Automation Center service catalog. This name and description is different from the deployment profile name and description.

8 Log in to your tenant in vCloud Automation Center.

The vCloud Automation Center Web interface opens.

9 Select the Administration > Catalog Management > Catalog Items tab and open the published item in the catalog.

10 Assign an entitled service to the catalog item from the Service drop-down menu and click Update.

The catalog item list appears.

11 (Optional) In the right-hand navigation, to entitle the service to a catalog item select Entitlements and open the entitlement.

a In the Details tab, type a name for the entitlement and click Next.

b In the Items & Approval tab, assign the published item to the entitled catalog items from the drop-down menu.

c In the Entitled Resource Action section, type the Application Deployment Details resource action and select it from the drop-down menu.

12 Click Update.

What to do next

Request a vCloud Automation Center catalog item for development. See “Request a vCloud Automation Center Service Catalog Item,” on page 158.

Request a vCloud Automation Center Service Catalog Item

When you request a service catalog item you provision the item to the designated cloud environment.

A typical user in vCloud Automation Center does not have direct access to the vCloud Application Director environment. Therefore, the user can access the vCloud Automation Center service catalog, which is a self-service portal and request vCloud Application Director application deployments.

Prerequisites

- Make sure that you log in to the vCloud Application Director appliance as an SSO user that has APPLICATION PUBLISHER and DEPLOYER roles assigned to it from the vCloud Automation Center user interface.

- The vCloud Automation Center 6.0 IaaS environment and entitlements must be properly set up and configured. See vCloud Automation Center Installation and Configuration Guide and vCloud Automation Center Virtual Provisioning Guide.

- Verify that vCloud Application Director is registered as a vCloud Automation Center endpoint. See “Register vCloud Application Director to vCloud Automation Center,” on page 30.

- Verify that at least one vCloud Automation Center catalog item has an entitlement and a service assigned to it. See “Publish a Deployment Profile to the vCloud Automation Center Service Catalog,” on page 157.
Procedure

1. Log in to your tenant with APPLICATION PUBLISHER and DEPLOYER roles in vCloud Automation Center. The vCloud Automation Center Web interface opens.
2. Select the Catalog tab and locate the catalog item you configured with a service and an entitlement.
3. Select the Requests tab and complete the description.
   For example, you can request an item for development and testing purposes.
4. Click Next to review the properties of the item.
5. Click Submit to request the catalog item.

When the deployment is complete, vCloud Application Director creates resources in vCloud Automation Center that appear in the Items tab.

What to do next

Select the vCloud Automation Center resource to track the progress. You can also use the vCloud Application Director deployment summary page to check the status in real times See “Using the Deployment Summary Page,” on page 160.

Understanding the Deployment and Update Process

When you deploy an application to the cloud or update a deployed application in the cloud, several processes take place in the background. Virtual machines are created and software is provisioned in the virtual machines. It is important to understand the deployment and update process so that you can easily identify and troubleshoot any deployment failures.

In vCloud Application Director, you can view the processes during a deployment in the execution plan. The host and agent bootstrap provisioning tasks appear next to components for each node. For applications deployed to the vCloud Automation Center environment, in addition to the host and agent bootstrap tasks, the network bootstrap provisioning task appears.

The process of deploying an application to the cloud and updating a deployed application includes the following steps:

1. vCloud Application Director runs policy assessment to determine compliance of the deployment process. If there is violation of a policy instance marked as critical, the deployment or update process stops.

2. vCloud Application Director provisions the virtual machines by instantiating the cloud templates that were mapped in the deployment profile.
   For the update process to scale a deployed application, vCloud Application Director provisions the scaled clustered virtual machines by instantiating the cloud templates from the vCloud Director catalog or vCloud Automation Center that were mapped in the deployment profile. The scaled clustered virtual machines use the same templates that were initially applied when deploying the application to the cloud. You cannot change the templates when you update the deployed application.
   The rest of the update profile processes are the same as the deployment process.

3. vCloud Application Director requests the cloud service to establish network connections and receive IP addresses for all of the virtual machines in the deployment. After the IP addresses are assigned, the virtual machines restart to make sure the setup process is completed properly. The host names are derived from the application blueprint and assigned as the node names.
   If host name is not assigned, then the host name takes the logical name and a sequence of randomly generated characters are appended to it.
4 Bootstrap scripts included in each virtual machine download the agent from the vCloud Application Director server to the virtual machine. The agent is a JAR file that runs in a Java virtual machine. Bootstrap scripts must be installed on the physical templates.

5 The bootstrap script starts the agent process.

6 The agent authenticates with the vCloud Application Director server.

7 The agent in each virtual machine downloads the execution plan from vCloud Application Director to the virtual machine.

8 The agent performs the installation and setup tasks for each component in the order specified in the deployment execution plan.

9 For each script, the agent waits for the dependent tasks to finish successfully, and downloads all of the content to the virtual machine directory at `/tmp/runid/content/ComponentName/PropertyName` and the task scripts to the directory at `/tmp/runid/TaskName`. The agent runs the tasks according to the parameter values sent by the server. When a task is complete, the agent informs the server about the status of the task.

   If a script finishes with a nonzero exit status, the agent marks that task as failed. Otherwise, the agent marks the script as completed and proceeds to the next task. When a task fails, the entire deployment is stopped, marked as Failed Deployment, and no future tasks are run. The reason for failure is available on the Details tab. When all of the tasks pass, the deployment is marked as Deployed Successfully.

   **NOTE** For a script to run without interruptions, the return value must be set to zero (0). This value allows the agent to capture all of the computed properties and send them to the vCloud Application Director server.

For troubleshooting purposes, you can access the deployment details or view the provisioning task logs included in the execution plan. The task information is captured in log files for the install, configure, start, update, rollback, and teardown scripts used on each component of the deployment. These logs capture all of the information that is sent to the stdout and stderr log files.

### Using the Deployment Summary Page

vCloud Application Director provides a graphical user interface for checking the status of an application deployment in real time on the deployment summary page.

You can also use the vCloud Director and vCloud Automation Center user interfaces to check the status of the deployed virtual machines. To view the status of a vCloud Application Director deployment in an Amazon VPC and associated Availability Zone, see Amazon AWS Documentation.

On the deployment summary page, when a deployment is running, the overall deployment status of the deployment, update, or teardown process appears in the toolbar. After the deployment finishes, the status bar turns red or green depending on the success or failure of the tasks in the deployment. Above the task status windows, a task timeline contains a time stamp that shows when the application deployment was initiated, any subsequent update deployments, such as scaled deployments or updates to modify configuration, or if a teardown process was run.
You can expand a window in the deployment summary page to view details and status of an application deployment. See “View Deployment Task and Blueprint Details for an Application,” on page 193 and “View Deployed VM Details and Execution Plan of an Application,” on page 195.
When you update an existing application deployment, you create an update profile that captures new values for the changes required for that update. You can also rollback updates to restore to the previous valid update and continue to initiate additional update processes.

You can deploy a saved update profile multiple times to update existing deployments. With vCloud Application Director 6.0, you can initiate an update process for existing deployments in vCloud Director and vCloud Automation Center multiple times by scaling in and out clusters of a node or modifying the configurations of existing services and application components.

When you scale node clusters of an existing application, make sure that you have enough resources in the cloud to support the additional nodes in the application.

If a deployment includes an external service and the configuration was modified in the external service instance, a notification icon appears on the deployment card. Click the icon and update the external service properties before you initiate an update or a rollback process.

**IMPORTANT** vCloud Application Director 6.0 does not support updating existing deployments in Amazon EC2.

To help you identify the changed and impacted properties in the application with dependencies, when you update the cluster size of the node, vCloud Application Director highlights the scaled in or out node as changed and the dependent property is highlighted as impacted in the Blueprint window on the Update wizard and review page. The component that contains the dependent property is highlighted as the impacted component.

When you update a property to modify a configuration, vCloud Application Director highlights the property as changed and the dependent property is highlighted as impacted in the Blueprint window on the Update wizard and review page. The component that contains the new property value is highlighted as the changed component. The component that contains the dependent property is highlighted as the impacted component.

When you rollback an update, vCloud Application Director highlights the changed and rollback to properties in the Blueprint window on the Rollback wizard and review page. The component that contains the new property value is highlighted as the rolled back to component. The component that contains the dependent property is highlighted as the changed component.

This chapter includes the following topics:

- “Initiate an Update Process to Scale Out Deployments,” on page 164
- “Initiate an Update Process to Scale In Deployments,” on page 166
- “Initiate an Update Process to Modify Configurations,” on page 169
- “Use an Existing Update Profile,” on page 171
Initiate an Update Process to Scale Out Deployments

With vCloud Application Director, you can create an update profile for an existing deployment to scale out the clusters of a node for example, to improve the performance and maintain the needs of the scalable application deployment. When you scale out a clustered node of a deployed application, you can configure only the cluster size of the node that were modeled as clusters in the application blueprint.

During an update process to scale out a deployed application, based on the update settings, virtual machines are created and required action scripts are run on the new virtual machines. In a multitiered application, if a node depends on the scaled out clustered node other than the external services, an update script must run on the dependent node.

For example, in a deployed Clustered Dukes Bank App, you can scale the AppServer node to handle additional load. During the update process, the AppServer install, configuration, and start scripts run on the newly scaled out virtual machine. Because the http_node_ips and appsrv_routes properties of the Apache_LB service are dependent on the AppServer node, changes in the AppServer cluster size affect the Apache_LB service and initiate the update script to run.

You define the UPDATE life cycle stage script for a dependent service or application component during the initial application deployment. You can also add or modify the update script during the update process. When you configure the update script during the update process, the script is saved for future deployments in the update profile.

**NOTE** You cannot modify the install, configure, or start scripts during an update process. You can configure only the update script.

You can also initiate an update process to scale out a deployed application from the command-line interface. See “Deploying and Updating an Application Using CLI,” on page 208. With REST APIs in vCloud Application Director, you can automate the scale out of a deployed application. See VMware vCloud Application Director API Programming guide.

An update process to scale a deployment might fail sometimes. A cleanup to delete the new virtual machines is required following the scaleout update failure. Deprovisioning the newly created virtual machines manually is a tedious task and is difficult for external services users. To avoid the manual effort, automatic deletion of virtual machines is considered. Set the following flag to true, to enable automatic cleanup of virtual machines after a scaleout failure:

- **VM_CLEANUP_AFTER_UPDATE_FAILURE**

You can set the flag post deployment and scaleout or rollback at a later time. If you do not wish to deprovision the virtual machines automatically, set the flag to False.

**Prerequisites**

- Verify that you installed the VMware vCloud Application Director for Release Automation edition to update a deployed application.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- The deployed application must include at least one clustered node. See “Specify a Node as a Cluster,” on page 139.
If you plan to add a custom task, verify that at least one custom task is created in the vCloud Application Director catalog. See “Add a Custom Task to the Catalog.” on page 129.

Verify that the initial deployment is successfully deployed to a cloud environment.

You cannot scale clustered nodes from a failed deployment to modify configuration.

Contact your cloud administrator to get information about the deployment environment storage space limit.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select **Deployments**.
2. Select an application deployment that is successfully deployed.
3. From the **Operations** drop-down menu in the toolbar, select **Update**.
   
   The updates summary page opens.
4. Select **Create Update Profile**.
   
   The Update Profile dialog box opens.
5. In the Update Type drop-down menu, select **Scale Out**.
6. Name the scaled update process, add an optional description, and click **OK**.
   
   In the description, you can add information about the changes included in this update.
7. (Optional) Click the blueprint image to review the highlighted dependencies between services or application components.
   
   Note all of the dependent components so that you can create an update script if one does not exist or modify an existing one.
8. From the **Node** tab, increase the cluster size value for one or more clustered nodes.
9. (Optional) From the **Service** or **Application Component** tab, create an update script for all of the available dependent components with a property bound to the clustered node.
   
   If the UPDATE life cycle stage script are defined, then the scripts appear with the associated service or application component. If the scripts are not defined, create the applicable script for the update process.
   
   For example, if a node has the all(node_array:ip) property bound to a clustered node, then it must run an update script.
10. (Optional) Select the **Reboot** check box if the agent must restart the virtual machine after the update script runs successfully and click **Next**.
   
   The scaled out clustered node and the update tasks of the dependent components appear in the execution plan. The original execution plan does not appear during an update process.
11. (Optional) Add an APPD_UPDATE_PROPS property in the update script to view a list of all the changed properties.
   
   The update script is saved in the update profile and can be used for multiple updates. You cannot use APPD_UPDATE_PROPS as a property name or as a qualifier for a property name.
   
   For example, the sample MySQL service update script includes the APPD_UPDATE_PROPS property to update the database port and password.
12 (Optional) Add an APPD_PREV property in the update script to view the previous value of a property.

The update script is saved in the update profile and can be used for multiple updates. You cannot use APPD_PREV as a property name or as a qualifier for a property name.

For example, the sample MySQL service update script includes the APPD_PREV property to view the value of the database password.

13 (Optional) Click the **Expand Cluster** button to expand the clustered node, select a custom task, and drag the task to each node.

You can configure the task properties in the Add Custom Task dialog box and save your changes.

14 In the execution plan, review the scaled out clustered node, and update script if applicable and click **Next**.

The blue dotted lines in the execution plan define a specific order in which the deployment tasks run.

Click the down arrow next to the service update script to view script details or the variable definitions used in the script.

15 Review the modified properties and actions in the update.

The dependent properties with defined update scripts are highlighted.

16 Click **Update** to deploy the updated application.

The update process deploys the scaled out update to the cloud.

**What to do next**

You can check the status of the deployment from the deployment summary page. See “Using the Deployment Summary Page,” on page 160.

Learn about the various processes that take place in the background when the deployed application is updated in the cloud. See “Understanding the Deployment and Update Process,” on page 159.

### Initiate an Update Process to Scale In Deployments

With vCloud Application Director, you can create an update profile for an existing deployment to scale in clusters of one or more nodes for example, to free unused resources and accommodate the changing load in the scalable application. When you scale in a clustered node of a deployed application, you can configure only the cluster size of the node that were modeled as clusters in the application blueprint.

During an update process to scale in a deployed application, based on the update settings, the update script runs on the dependent components followed by the teardown action script to remove the nodes. In a multitiered application, if a node depends on the scaled in clustered node, an update script must run on the dependent node.

For example, in a deployed Clustered Dukes Bank App, you can scale in the appServer node to reduce the load. During the update process, the update script runs on dependent load_balancer node. The teardown action script removes the appServer_2 and appServer_3 nodes from the deployment. The action script also calls the cloud provider API to remove the nodes from the cloud environment.

You define the UPDATE and the TEARDOWN life cycle stage scripts for a dependent service or application component during the initial application deployment. The update scripts run on dependent nodes and the teardown scripts run on the scaled in nodes. You can also add or modify the update and teardown script during the update process. When you configure the update or teardown script during the update process, the script is saved for future deployments in the update profile.

**Note** You cannot modify the install, configure, or start scripts during an update process. You can configure only the update and teardown script.
With REST APIs in vCloud Application Director, you can automate the scale in of a deployed application. See VMware vCloud Application Director API Programming guide.

An update process to scale in a deployment might fail sometimes. A cleanup to retry deprovisioning of the new virtual machines is required following the scale in update failure. Deprovisioning the newly created virtual machines manually is a tedious task and is difficult for external services users. To avoid the manual effort, automatic deletion of virtual machines is considered. Set the following flag to True, to retry the cleanup of virtual machines if deletion fails in the first attempt:

- `UPDATE_RETRY_VM_DEPROVISIONING_AFTER_FAILURE_FLAG`

Set the following flag to True to retry deletion of virtual machines with a delay in milliseconds:

- `UPDATE_RETRY_VM_DEPROVISIONING_AFTER_FAILURE_DELAY_INTERVAL`

**Prerequisites**

- Verify that you installed the VMware vCloud Application Director for Release Automation edition to update a deployed application.
- Verify that your user account has the `ROLE_DEPLOYER` deployer role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- The deployed application must include at least one clustered node. See “Specify a Node as a Cluster,” on page 139.
- If you plan to add a custom task, verify that at least one custom task is created in the vCloud Application Director catalog. See “Add a Custom Task to the Catalog,” on page 129.
- Verify that the initial deployment is successfully deployed to a cloud environment. You cannot scale clustered nodes from a failed deployment to modify configuration.
- Contact your cloud administrator to get information about the deployment environment storage space limit.

**Procedure**

1. On the vCloud Application Director title bar, click the drop-down menu and select *Deployments*.
2. Select an application deployment that is successfully deployed.
3. From the *Operations* drop-down menu in the toolbar, select *Update*.
   
The updates summary page opens.
4. Select *Create Update Profile*.
   
The Update Profile dialog box opens.
5. In the Update Type drop-down menu, select *Scale In*.
6. Name the scaled update process, add an optional description, and click *OK*.
   
   In the description, you can add information about the changes included in this update.
7. (Optional) Click the blueprint image to review the highlighted dependencies between services or application components.
   
   Note all of the dependent components so that you can create an update script if one does not exist or modify an existing one.
8 From the **Node** tab, decrease the cluster size value.

vCloud Application Director selects the nodes that were last created.

For example, if a cluster includes appserver_0, appserver_1, appserver_2, and appserver_3 nodes and you decrease the cluster size to two nodes, the appserver_2 and appserver_3 nodes are selected for removal.

9 (Optional) Click the **Choose Nodes to remove** button to list all of the nodes in the cluster and manually select the nodes for removal.

10 (Optional) From the **Service** or **Application Component** tab, create an update script for all of the available dependent components with a property bound to the clustered node.

If the UPDATE and TEARDOWN life cycle stage are defined, then the scripts appear with the associated service or application component.

For example, if a load balancer node has the all(node_array:ip) property bound to a clustered node, then it must run an update script.

11 (Optional) Select the **Reboot** check box if the agent must restart the virtual machine after the update script runs successfully and click **Next**.

The scaled in clustered node and the update tasks of the dependent components appear in the execution plan. The original execution plan does not appear during an update process.

12 (Optional) Add an **APPD_UPDATE_PROPS** property in the update script to view a list of all the changed properties.

The update script is saved in the update profile and can be used for multiple updates. You cannot use **APPD_UPDATE_PROPS** as a property name or as a qualifier for a property name.

For example, the sample MySQL service update script includes the **APPD_UPDATE_PROPS** property to update the database port and password.

13 (Optional) Add an **APPD_PREV** property in the update script to view the previous value of a property.

The update script is saved in the update profile and can be used for multiple updates. You cannot use **APPD_PREV** as a property name or as a qualifier for a property name.

For example, the sample MySQL service update script includes the **APPD_PREV** property to view the value of the database password.

14 (Optional) Click the **Expand Cluster** button ( ) to expand the clustered node, select a custom task, and drag the task to each node.

You can configure the task properties in the Add Custom Task dialog box and save your changes.

15 Review the scaled in clustered node and update script in the execution plan and click **Next**.

The blue dotted lines in the execution plan define a specific order in which the deployment tasks run.

Click the down arrow next to the service update script to view script details or the variable definitions used in the script.

16 Review the modified properties and actions in the update.

The dependent properties with defined update scripts are highlighted.

17 Click **Update** to deploy the updated application.

The update process deploys the scaled in update to the cloud.
What to do next

You can check the status of the deployment from the deployment summary page. See “Using the Deployment Summary Page,” on page 160.

Learn about the various processes that take place in the background when the deployed application is updated in the cloud. See “Understanding the Deployment and Update Process,” on page 159.

Initiate an Update Process to Modify Configurations

You can initiate an update process to modify the configurations and code of existing services in a deployed application such as Tomcat and MYSQL, or application components such as WAR and SQL. When you modify the configurations of a deployed application, you can configure only the application property values.

After the initial deployment, if you modify existing application components or services or modify components that have dependent properties, an update script must run on all of the impacted and dependent components. You define the UPDATE and the ROLLBACK life cycle stage scripts for a service or application component during the initial application deployment or in the update process.

For example, in the Clustered Dukes Bank App, if you modify the db_port property of the MYSQL service, during the deployment update process the update script runs on the MYSQL service. The Dukes_Bank_App application component on the Appserver node also runs the update script because the db_port property is bound to the db_port property of MYSQL service.

You cannot modify the install, config, or start scripts during an update process. You can configure only the update or rollback script.

You can also initiate an update process to modify the configuration and code of services or application components from the command-line interface. See “Deploying and Updating an Application Using CLI,” on page 208. With REST APIs in vCloud Application Director, you can automate the modification of configurations of a deployed application. See VMware vCloud Application Director API Programming guide.

Prerequisites

- Verify that you installed the VMware vCloud Application Director for Release Automation edition to update a deployed application.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- If you plan to add a custom task, verify that at least one custom task is created in the vCloud Application Director catalog. See “Add a Custom Task to the Catalog,” on page 129.
- Verify that the initial deployment is successfully deployed to a cloud environment.
  
  You cannot start an update process to modify the configuration and code of an existing service from a failed deployment to scale a clustered node.
- The deployed application must have at least one service property or application component property that is Overridable at Deployment.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Select an application deployment that is successfully deployed.
3. From the Operations drop-down menu in the toolbar, select Update.

The updates summary page opens.
Select Create Update Profile.

The Update Profile dialog box opens.

In the Update Type drop-down menu, select Configuration.

Name the configuration update process, add a description, and click OK.

In the description, you can add information about the changes included in this update.

Click the blueprint image to review the highlighted dependencies between services or application components.

Note all of the dependent components so that you can create an update script if one does not exist or modify an existing one.

From the Service or Application Component tab, modify the applicable properties and update scripts.

If the UPDATE and ROLLBACK life cycle stage are defined, then the scripts appear with the associated service or application component. You must create an update script for the changed and dependent component.

In the case of a clustered node, if you modify a property of a service or an application component, the changes are applied to all the nodes of the cluster.

For example, in the Clustered Dukes Bank App if you change the db_password property in the initialize_db_script application component, the db_password property in the Dukes_Bank_App depends on the initialize_db_script application component. The update scripts run on both because initialize_db_script is the changed component and Dukes_Bank_App is the impacted component. The update script for Dukes_Bank_App runs on all of the nodes of the Appserver cluster.

(Optional) Select the Reboot check box if the agent must restart the virtual machine after the update script runs successfully and click Next.

The update tasks of the changed and affected components appear in the execution plan. The original execution plan does not appear during an update process.

(Optional) Add an APPD_UPDATE_PROPS property in the update script to view a list of all the changed properties.

The update script is not saved and it applies to that particular update process only. You cannot use APPD_UPDATE_PROPS as a property name or as a qualifier for a property name.

For example, the sample MySQL service update script includes the APPD_UPDATE_PROPS property to update the database port and password.

(Optional) Add an APPD_PREV property in the update script to view the previous value of a property.

The update script is not saved and it applies to that particular update process only. You cannot use APPD_PREV as a property name or as a qualifier for a property name.

For example, the sample MySQL service update script includes the APPD_PREV property to view the value of the database password.

(Optional) Click the Expand Cluster button to expand the clustered node, select a custom task, and drag the task onto each node.

If you modify a service property or an application component property on a clustered node, the changes are applied to all of the nodes. You can configure the task properties in the Add Custom Task dialog box and save your changes.
13 Review the modified clustered node and update script in the update execution plan.

The blue dotted lines in the execution plan define a specific order in which the deployment tasks will run.

Click the down arrow next to the configuration update script to view script details or the variable definitions used in the script.

14 Review the modified clustered node and update script in the rollback execution plan and click Next.

The blue dotted lines in the execution plan define a specific order in which the deployment tasks will run.

Click the down arrow next to the configuration update script to view script details or the variable definitions used in the script.

15 Review the changed and impacted components in the update.

The modified properties and dependent properties are highlighted.

16 Click Update to deploy the modified configuration for the application.

The update process deploys the configured deployment to the cloud.

What to do next

Review the status of the deployment from the deployment summary page. See “Using the Deployment Summary Page,” on page 160.

Learn about the processes that take place in the background when the deployed application is updated in the cloud. See “Understanding the Deployment and Update Process,” on page 159.

Use an Existing Update Profile

A deployer can use an existing update profile to update a previously deployed application in the same cloud environment multiple times. The update profile saves the common properties and scripts used in the initial update process.

You can save and apply an update profile only to an application version of a deployment.

CLI does not support saving new or existing update profiles. If you apply an existing update profile using CLI, you cannot override scripts or properties when you initiate an update process to modify configuration or scale clustered nodes.

When you deploy an application using CLI, if you select an update profile the auto complete option for the update profile name is not displayed.

Prerequisites

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that at least one deployment is successfully updated in vCloud Application Director.

Procedure

1 On the vCloud Application Director title bar, click the drop-down menu and select Deployments.

2 Select a deployment that is updated successfully.

   The deployment summary page opens.

3 In the toolbar above the deployment summary, select Deployment View > Updates View.

   All of the details associated with the deployment such as available update profiles, application name and version appear on the updates summary page.
Find the type of update profile to use and click it.

The update profile can be for a scale out, scale in, or configuration update process. Each update profile shows the number of times the profile was used to successfully and unsuccessfully update a deployment.

The Deployment Profile wizard opens.

What to do next

Promote an Update Profile
A deployer can promote an existing update profile to update any deployment across multiple deployment environments. You can apply the same type of an update process to more than one deployment originating from same application version.

For example, a deployer can apply a security related configuration update on one or more components of an application on different deployments originating from same application version.

Promoting update profiles across different deployment environments through REST API is not supported.

Prerequisites
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.

Procedure
1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Select a deployment that is updated successfully.
   - The deployment summary page opens.
3. In the toolbar above the deployment summary, select Deployment View > Updates View.
   - All of the details associated with the deployment such as available update profiles, application name and version appear on the updates summary page.
4. Locate an existing update profile to promote.
5. To promote the update profile to one or multiple deployment environments, click the Promote Update Profile icon.
   - The Manage Update Promotion dialog box opens with a list of the available deployment environments.
6. In the Available column, select the deployment environment and click the Add to Selected arrow.
   - The deployment environment appears in the Selected column.
7. (Optional) To withdraw a deployment environment from the Selected column, select the deployment environment and click the Remove from Selected arrow.
   - The deployment environment appears in the Available column.
8. Review the deployment environments in the Selected column and click Submit.

When the update profile is successfully promoted to the selected deployment environments, a green thumbs up icon appears on the update profile summary card.
A notification icon ( ) appears on deployment in the single or multiple deployment environments. For example, if the Clustered Dukes Bank update profile is promoted, then the notification appears on all of the existing Clustered Dukes Bank application deployments within the selected deployment environment.

What to do next

Apply the promoted update profile to a deployment which belongs to one of the selected deployment environments. See “Accept or Discard a Promoted Update Profile,” on page 173.

Accept or Discard a Promoted Update Profile

A deployer can click the notification to review the components of the promoted update profile and decide whether to accept or discard the changes. When you accept the changes, vCloud Application Director performs a compatibility check between the promoted update profile and the deployment before applying the changes.

Prerequisites

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that at least one update profile is promoted to one or multiple deployment environments. See “Promote an Update Profile,” on page 172.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Locate a deployment with a notification icon ( ) and click the icon.
   The updates summary page opens.
3. Click Apply to accept the promoted update profile.
   vCloud Application Director checks the compatibility between the promoted update profile and the current updated deployment. If severe incompatibilities are found, you can cancel or discard the notification. If non-severe incompatibilities such as an update profile existing with same name in the deployment is found, then you can continue to apply, discard or cancel the notification.
4. In the wizard, review the new values or configurations in the Application Properties tab and Execution Plan tab and select Apply Update.
5. (Optional) If you do not want to apply the new values or configurations in the promoted update profile, select Discard Update.
   The promoted update profile is discarded and the notification icon ( ) is removed from the deployment.

What to do next

You can select another deployment with a notification and continue to review and apply or discard the promoted update profile.
Rollback an Update Process

With vCloud Application Director if an update process to modify a configuration fails or if the updated deployment has performance or availability problems, you can use the rollback feature to restore that update to the last successful state in the timeline.

The rollback script runs on the components whose properties are changing and the other dependent nodes. For example, in a deployed Clustered Dukes Bank App, when you modify the configuration of the MySQL port number from 3306 to 3307, the rollback script runs on the mysql service and dukes_bank_app component. If this update process failed, you can rollback the configuration update to revert to the previous port value 3306. In the Rollback wizard, you can view the Current Value as 3307 and the Rollback to value as 3306. The rollback script runs on the mysql service and dukes_bank_app component to complete the process.

You define the ROLLBACK life cycle stage script for a dependent service or application component during the initial application deployment. You can also add or modify the rollback scripts when you rollback an update process to modify a configuration.

Prerequisites

- Verify that you installed the VMware vCloud Application Director for Release Automation edition to update a deployed application.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- Verify that the initial deployment is successfully deployed to a cloud environment.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Select an application deployment to rollback to a previous successful state.
3. From the Operations drop-down menu in the toolbar, select Rollback.
   The Rollback wizard page opens.
4. Click the blueprint image to review the highlighted dependencies between services or application components.
   Note all of the dependent components so that you can create a rollback script if one does not exist or modify an existing one.
5. From the Service or Application Component tab, modify the applicable properties and rollback script.
   If the ROLLBACK life cycle stage is defined, then the script appears with the associated service or application component. You must create a rollback script for the changed and rollback to component.
   In the case of a clustered node, if you modify a property of a service or an application component, the changes are applied to all the nodes of the cluster.
   If you update the configuration properties of a deployment that was triggered by a change in the external service instance, you cannot rollback these updated properties.
6. (Optional) Select the Reboot check box if the agent must restart the virtual machine after the rollback script runs successfully and click Next.
   The rollback tasks of the changed and rollback to components appear in the execution plan. The original execution plan does not appear during an rollback process.
7 Review the modified clustered node and rollback script in the execution plan and click Next. The blue dotted lines in the execution plan define a specific order in which the deployment tasks will run.

Click the down arrow next to the configuration rollback script to view script details or the variable definitions used in the script.

8 Review the changed and rollback to components in the update.

The changed and rollback to properties are highlighted.

9 Click Rollback to start the rollback process.

The rollback process reverts the deployment to the last successful state in the cloud.

What to do next

Review the status of the deployment from the deployment summary page. See “Using the Deployment Summary Page,” on page 160.

Troubleshoot Failed Update Process to Scale Deployments

Known solutions and recommendations can help you when your update process to scale a deployment fails.

Problem

An update process to scale deployments fails.

Cause

You might be attempting to initiate an update process to scale a deployment that previously failed to scale.

Solution

1 The cloud administrator must delete the new virtual machines for all the clustered nodes of the deployment.

2 The cloud administrator must delete the virtual machines that were not removed during scale in update process.

3 Initiate another scale out update process on the previously failed deployment.

vCloud Application Director prepopulates the previously applied successful property values. The new values are taken from the previous failed deployment.

For example, if you initiate an update process to scale an AppServer1 clustered node from 1 to 2 nodes and scale an AppServer2 clustered node from 1 to 3 nodes and the update process failed for AppServer2 node. In the second update process, the value for AppServer1 is populated to 2 and for AppServer2 node you can specify a value greater than the previous value.

4 Initiate another scale in update process if the deployment failed before the virtual machines were removed.

5 Initiate another update process to scale out all the failed nodes, so that the update scripts run on all of the dependent components such as load balancer, to make the components valid.

Troubleshoot Failed Update Process to Modify Configuration

You can apply some solutions when your update process to modify configuration fails.

Problem

An update process to modify configuration fails.
Cause
You might be attempting to initiate an update process to modify configuration on a previously failed update by resetting the failed property such as port number, and trying to proceed.

Solution
- Initiate another update process to modify configuration on the previously failed update.
  
vCloud Application Director prepopulates the previously applied successful property values. The new values are taken from the previous failed deployment.
  
  You must manually update all of the failed properties when you initiate another update process to modify configuration.

- Create or modify an update script to ignore a failed property.
  
  If you do not want to update all of the failed properties, you can author an update script to ignore the failed properties and allow you to proceed with the update process.

  The modified update script is not saved and it applies to only that particular update process.
vCloud Application Director catalog includes predefined sample services, logical templates, application components, and tasks.

A sample application consists of either a Windows or Linux-based logical template, services, and application components configured specifically for that application. You must configure only the action scripts of the services in a predefined application and deploy it to the supported cloud environment. When you create a deployment profile to deploy a sample application, you can add a compatible customized task to the execution plan.

**IMPORTANT** Use the predefined sample templates, application components, services, and tasks only in a test environment. If you plan to use the sample templates in a production environment, make sure that you apply the latest security patches to the operating system in the template before deployment.

All user accounts that are assigned to the vCloud Application Director Default group can access and deploy the sample applications.

All of the icons rendered for the sample templates and services in the catalog are based on the name of the template or service. For example, the official MySQL icon appears next to the MySQL service in the catalog. If the name of the service or template is unique, then a generic icon appears.

Familiarize yourself with the key concepts that appear frequently in topics about deploying sample catalog components. See “Key Concepts,” on page 12.

This chapter includes the following topics:

- “Using the Sample Templates,” on page 177
- “Available Sample Applications for Deployment,” on page 178
- “Update Deployed Sample Applications,” on page 183
- “Sample vCloud Application Director Catalog Tasks,” on page 186
- “Sample vCloud Application Director Catalog Services,” on page 191

**Using the Sample Templates**

vCloud Application Director provides the CentOS and Ubuntu sample templates with the operating system installed and all of the necessary libraries to deploy an application.

For example, to use the CentOS sample template, download the following files from the VMware product download site.

- CentOS 6.3 32-bit OVF
- CentOS 6.3 32-bit VMDK
For a list of the available sample templates, see the vCloud Application Director Readme file.

Upload the CentOS 6.3 32-bit OVF package to the vCloud Director catalog or vCloud Automation Center. For more information on the vCloud Director catalog or vCloud Automation Center, see the respective documentation.

**Note**  For CentOS 32-bit sample templates, Physical Address Extension (PAE) is not enabled, so you can allocate up to 3.25GB of RAM for a virtual machine. PAE is enabled for Ubuntu 32-bit sample templates, so you can allocate more than 4GB of RAM for a virtual machine.

---

### Add libpython Package to the Ubuntu Template

For the sample Ubuntu template to work properly, you must add the libpython XML package.

**Procedure**

1. Download the python package from the Ubuntu mirror site.
   
   ```bash
   wget http://UbuntuMirrorSite/libpython2.7_2.7.3-0ubuntu3.4_amd64.deb
   ```

2. Install the python package to the Ubuntu template.
   
   ```bash
   dpkg -i libpython2.7_2.7.3-0ubuntu3.4_amd64.deb
   ```

### Enable an SSH Connection for Ubuntu Virtual Machines

Ubuntu virtual machines must have SSH host keys generated to work properly.

**Prerequisites**

Verify that you have the vCloud Director VMRC plug-in installed to work with your browser. For information about compatible Web browsers, see “vCloud Application Director System Requirements,” on page 20.

**Procedure**

1. Upload the Ubuntu OVF package to the vCloud Director catalog.

2. Click in the vCloud Director VMRC console.

3. Log in to the Ubuntu virtual machine.

4. Enable the SSH connection.
   
   ```bash
   ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key
   "ssh-keygen -t ecdsa -f /etc/ssh/ssh_host_ecdsa_key"
   ```

5. Restart the SSH server.
   
   ```bash
   service ssh restart
   ```

The SSH connection to the Ubuntu virtual machine is enabled.

---

### Available Sample Applications for Deployment

You can use the predefined single-tier or three-tier applications to deploy either a simple Web application or a complex email suite. The application blueprint of a sample application includes logical templates, services, and application components that are fully functional when deployed to the cloud.

An application architect can use the vCloud Application Director sample applications as a base to model and create custom applications.
Deploy Nanotrader Application

The sample Nanotrader application is a three-tier Web application that deploys the NanoTrader application using the vFabric Web Server, vFabric tc Server, vFabric RabbitMQ, and vFabric SQLFire components. Use the predefined sample application only in a test environment.

Prerequisites

- Register the uploaded CentOS 6.3 32-bit template to a cloud provider. See “Register the vCloud Director Cloud Provider and Template,” on page 78, “Register the vCloud Automation Center Cloud Provider and Template,” on page 57, or “Register the Amazon EC2 Cloud Provider and Template,” on page 88.
- Map the cloud template to the CentOS63 32bit logical template. See “Add a Logical Template to the Catalog,” on page 121.
- Verify that your user account has the ROLE_APP_ARCHITECT application architect role and ROLE_DEPLOYER deployer role assigned to it.
- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Understand the basic concepts of creating a deployment profile and deploying an application. See Chapter 12, “Deploying Applications,” on page 147.
- Verify that the spring-nanotrader-asynch-services.war, spring-nanotrader-services.war, spring-nanotrader-web.tgz, and spring-nanotrader-web.war files are downloaded to an HTTPD server.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Select Nanotrader.
3. Click the Blueprint screenshot at the top of the screen.
4. In the blueprint editor, select the service and SCRIPT component and add URL values to download the applicable files.

<table>
<thead>
<tr>
<th>Service and Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vFabric_Web_Server</td>
<td>spring-nanotrader-web.tgz file for the deployment_archive property</td>
</tr>
<tr>
<td>vFabric_tc_Server</td>
<td>spring-nanotrader-web.war file for the external_template property</td>
</tr>
<tr>
<td>spring-nanotrader-services</td>
<td>spring-nanotrader-services.war file for the war_file property</td>
</tr>
</tbody>
</table>

5. Click Save.
6. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
7. Open the Nanotrader application that you configured.
8. Create a deployment profile and deploy the application.
9. To access the deployed application, open a supported Web browser and type the http://Load_Balancer_IP/#login URL.
   The Load_Balancer_IP is the IP address of the deployed load balancer.
10. Create a NanoTrader account to set a user name and password.
What to do next

Initiate an update process to scale or modify configuration of the Nanotrader application. See “Initiate an Update Process to Scale Sample Applications,” on page 184 and “Initiate an Update Process to Modify Configurations in Sample Applications,” on page 185.

Deploy Clustered DotShoppingCart Application

The Clustered DotShoppingCart is a sample three-tier Windows-based application that uses Microsoft SQL Server as the database server, Microsoft IIS Server as the application server, and Apache HTTP Server as the load balancer.

The Create_DB script creates and initializes the DotShoppingCart database in the SQL Server database. The DotShoppingCart_App script installs the DotShoppingCart application in the Microsoft IIS server and uses the DotShoppingCart database. The Microsoft IIS AppServer node is defined as a cluster that allows scaling out to multiple nodes for handling a larger load. The Apache HTTP server handles the load balancing.

Use the predefined sample application only in a test environment.

Prerequisites

- Register the Windows Server 2008 R2 SP1 vApp template to a cloud provider. See “Register the vCloud Director Cloud Provider and Template,” on page 78.

- Map the cloud template to the Windows Server 2008 R2 SP1 logical template. See “Add a Logical Template to the Catalog,” on page 121.

- Verify that your user account has the ROLE_APP_ARCHITECT application architect role and ROLE_DEPLOYER deployer role assigned to it.

- Verify that the DotShoppingCart 3.0 (Open Source Edition) package OpenSourceEdition.V3.zip file is downloaded. The Web server must be configured to transfer the file using the HTTP protocol. Identify the URL used to access the file in a Web browser.

- Understand the basic concepts of creating a deployment profile and deploying an application. See Chapter 12, “Deploying Applications,” on page 147.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.

2. Select Clustered DotShoppingCart.

3. Click the Blueprint screenshot at the top of the screen.

4. In the blueprint editor, select the DotShoppingCart_App application component and type a valid URL for the DSC_ZIP property value.

   During deployment, the vCloud Application Director agent in the AppServer Windows virtual machine must be able to access the valid URL for the OpenSourceEdition.V3.zip file.

5. (Optional) If you plan to have more than one AppServer node, select AppServer and increase the cluster size.

6. Create a deployment profile and deploy the application.

7. To access the deployed application, open a supported Web browser and type the http://LoadBalancer_IP:8081 URL.

   The LoadBalancer_IP is the IP address of the deployed load balancer.
What to do next

Initiate an update process to scale or modify configuration of the Clustered DotShoppingCart application. See “Initiate an Update Process to Scale Sample Applications,” on page 184 and “Initiate an Update Process to Modify Configurations in Sample Applications,” on page 185.

Deploy Clustered Dukes Bank Application

The Clustered Dukes Bank application is a sample three-tier vCloud Application Director application that uses MySQL or Microsoft SQL Server as its database, JBoss Server as its application server, and Apache HTTP server as its load balancer that you can deploy to your test environment.

The initialize_db_script script creates and initializes the dukes_db database in the MySQL or SQL Server. The Dukes_Bank_App EAR application component uses the dukes_db database, which is deployed in the JBoss server. The JBoss appserver node is defined as a cluster that allows scaling out to multiple nodes to handle a higher load. The Apache HTTP server handles the load balancing.

Use the predefined sample application only in a test environment.

Prerequisites

- Register the uploaded CentOS 6.3 32-bit Linux template or Windows Server 2008 R2 SP1 vApp template to a cloud provider. See “Using the Sample Templates,” on page 177 and “Register the vCloud Director Cloud Provider and Template,” on page 78.
- Map the cloud template to the CentOS63 32bit or Windows Server 2008 R2 SP1 logical template. See “Add a Logical Template to the Catalog,” on page 121.
- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Verify that your user account has the ROLE_APP_ARCHITECT application architect role and ROLE_DEPLOYER deployer role assigned to it.
- For the Linux-based Clustered Dukes Bank, if your system requires a proxy to access the Internet, verify that your proxy connections are properly configured. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- For the Windows-based Clustered Dukes Bank, verify that the Windows virtual machine has a valid JRE installation path that you can use in the template.
- Understand the basic concepts of creating a deployment profile, deploying an application, and updating a deployed application. See Chapter 12, “Deploying Applications,” on page 147.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Click Clustered Dukes Bank App.
3. Select the application version.

<table>
<thead>
<tr>
<th>Option</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux-based Clustered Dukes Bank</td>
<td>Select version 2.1.0.</td>
</tr>
<tr>
<td>Windows-based Clustered Dukes Bank</td>
<td>Select version 3.0.0.</td>
</tr>
</tbody>
</table>

4. Click the Blueprint screenshot at the top of the screen.
In the blueprint editor, select the JBossAppServer service and type the appropriate property value.

- For the Windows-based Clustered Dukes Bank, update the JAVA_INSTALL_DIR property.
- For the Linux-based Clustered Dukes Bank, update the JAVA_HOME property.

During deployment, the vCloud Application Director agent in the JBossAppServer service should be able to access the installed JRE.

For the Linux-based Clustered Dukes Bank, if the cloud template is in a private network without a proxy, when you deploy the application, add a YUM configuration task to each virtual machine in the deployment profile, and set the repository_url property to use a CentOS 6.3 32-bit repository hosted in the private network.

Create a deployment profile and deploy the application.

To access the deployed application, open a supported Web browser and type the http://LoadBalancer_IP:8081/bank/main.faces URL.

The LoadBalancer_IP is the IP address of the deployed load balancer.

Use the log in 200 and password foobar to access the application.

What to do next

Initiate an update process to scale or modify configuration of the Clustered Dukes Bank 3.0.0 or 2.1.0 application version. See “Initiate an Update Process to Scale Sample Applications,” on page 184 and “Initiate an Update Process to Modify Configurations in Sample Applications,” on page 185.

Deploy jPetStore Application

vCloud Application Director includes two versions of the jPetStore sample application that you can deploy. The sample application deploys the jPetStore App WAR file on tc Server using SQLFire as a database.

jPetStore 1.0.0 is a single-tier sample Web store application that deploys the jPetStore App WAR file on vFabric tc Server using SQLFire as a database that you can deploy to your test environment.

Use the predefined sample application only in a test environment.

Prerequisites

- Register the uploaded CentOS 6.3 32-bit template for jPetStore 1.0.0 to a cloud provider. See “Using the Sample Templates,” on page 177 and “Register the vCloud Director Cloud Provider and Template,” on page 78.
- Map the cloud template to the CentOS63 32bit disk logical template jPetStore 1.0.0. See “Add a Logical Template to the Catalog,” on page 121.
- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- If your system requires a proxy to access the Internet, verify that your proxy connections are properly configured. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- Understand the basic concepts of creating a deployment profile and deploying an application. See Chapter 12, “Deploying Applications,” on page 147.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Click the jPetStore version.

   All the services and application components for this application are predefined and do not need additional configuration.
3 Click the Blueprint screenshot at the top of the screen.
4 Create a deployment profile and deploy the application.
5 To access the deployed application, open a supported Web browser and type the 
   http://jPetStore_VM_IP:8080/jpetstore-1.0.0 URL.

Deploy Radiant CMS Application

Radiant CMS is a single-node application that deploys a Ruby On Rails service and a MySQL database, and installs the Radiant Application configured to use the MySQL database.

Use the predefined sample task only in a test environment.

Prerequisites

- Register the uploaded CentOS 6.3 32-bit template to a cloud provider. See “Register the vCloud Director Cloud Provider and Template,” on page 78, “Register the vCloud Automation Center Cloud Provider and Template,” on page 57, or “Register the Amazon EC2 Cloud Provider and Template,” on page 88.
- Map the cloud template to the CentOS63 32bit logical template. See “Add a Logical Template to the Catalog,” on page 121.
- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that vCloud Application Director is configured to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- Understand the basic concepts of creating a deployment profile and deploying an application. See Chapter 12, “Deploying Applications,” on page 147.

Procedure

1 On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2 Click Radiant CMS.
   All the services and application components for this application are predefined and do not need additional configuration.
3 Click the Blueprint screenshot at the top of the screen.
4 Create a deployment profile and deploy the application.
5 To access the application, add the Radiant CMS virtual machine IP address http://VM_IP/ in a supported Web browser and log in using the Radiant default admin username and Radiant password.

Update Deployed Sample Applications

Updating an existing sample application deployment initiates a process that captures new values for the changes required for that update.

IMPORTANT To update deployed applications, you must install the vFabric Application Director for Release Automation edition.

You can initiate an update process for a single-tier or three-tier sample application by scaling clusters of a node or modifying the configurations and code of existing services and application components.
Initiate an Update Process to Scale Sample Applications

You can initiate an update process for predefined sample application deployments multiple times by scaling the clusters of a node. When you scale out or scale in a clustered node of a deployed application, you can configure only the cluster size of the nodes that are modeled as clusters in the application blueprint.

You can scale the following predefined sample applications.

- Nanotrader application
- Clustered DotShoppingCart application
- Clustered Dukes Bank 3.0.0 or 2.1.0 application version

Prerequisites

- Verify that you installed the VMware vCloud Application Director for Release Automation edition to update a deployed application.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- The deployed application must include at least one clustered node. See “Specify a Node as a Cluster,” on page 139.
- Verify that the initial deployment is successfully deployed to a cloud environment.
  You cannot scale clustered nodes from a failed deployment to modify configuration.
- Contact your cloud administrator to get information about the deployment environment storage space limit.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Select a predefined sample application that deployed successfully.
3. From the Operations drop-down menu in the toolbar, select Update.  
   The updates summary page opens.
4. Select Create Update Profile.
   The Update Profile dialog box opens.
5. In the Update Type drop-down menu, select Scale Out or Scale In.
6. Name the scaled update process, add an optional description, and click OK.
   In the description, you can add information about the changes included in this update.
7. From the Node tab, increase the cluster size value for one or more clustered nodes and click Next.
8. Review the scaled clustered node in the execution plan and click Next.
   The blue dotted lines in the execution plan define a specific order in which the deployment tasks run.
9. Review the modified properties in the update.
   The impacted properties are highlighted.
10. Click Update to deploy the updated application.
### What to do next

You can check the status of the deployment from the deployment summary page. See “Using the Deployment Summary Page,” on page 160.

### Initiate an Update Process to Modify Configurations in Sample Applications

You can initiate an update process to modify the configurations and code of existing services and application components in a deployed sample application. When you modify the configurations of a deployed application, you can configure only the application property values.

You can modify the configurations and code of the following predefined sample applications.

- Nanotrader application
- Clustered DotShoppingCart application
- Clustered Dukes Bank 3.0.0 or 2.1.0 application version

### Prerequisites

- Verify that you installed the VMware vCloud Application Director for Release Automation edition to update a deployed application.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See Chapter 9, “Developing vCloud Application Director Components,” on page 91.
- Verify that the initial deployment is successfully deployed to a cloud environment.
- You cannot start an update process to modify the configuration and code of an existing service from a failed deployment to scale a clustered node.
- The deployed application must have at least one service property or application component property that is Overridable at Deployment.

### Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select **Deployments**.
2. Select a predefined sample application that has deployed successfully.
3. From the **Operations** drop-down menu in the toolbar, select **Update**.
   The updates summary page opens.
4. Select **Create Update Profile**.
   The Update Profile dialog box opens.
5. In the **Update Type** drop-down menu, select **Configuration**.
6. Name the configuration update process, add a description, and click **OK**.
   In the description, you can add information about the changes included in this update.
7. From the **Service** tab, modify the applicable properties for your sample application.

<table>
<thead>
<tr>
<th>Sample Application</th>
<th>Service Version</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nanotrader</td>
<td>vFabric_Web_Server 5.1.1</td>
<td>http_port, deployment_archive, and</td>
</tr>
<tr>
<td></td>
<td></td>
<td>webserver_conf_file</td>
</tr>
<tr>
<td></td>
<td>vFabric_tc_Server 2.7.1</td>
<td>port, db_ip, db_port, and jdbc_url</td>
</tr>
<tr>
<td></td>
<td>vFabric_SQLFire_Server 1.0.3</td>
<td>schema_file and dataload_file</td>
</tr>
<tr>
<td>Clustered DotShoppingCart</td>
<td>Apache_LB 2.2.22</td>
<td>http_port</td>
</tr>
</tbody>
</table>
### Sample Application  

<table>
<thead>
<tr>
<th>Service Version</th>
<th>Properties</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL _Server_2008 1.0.0</td>
<td>SA_PWD</td>
</tr>
<tr>
<td>Clustered Dukes Bank 3.0.0</td>
<td>Apache_LB 2.2.22 http_port and http_proxy_port</td>
</tr>
<tr>
<td>JBossAppServer 5.1.0</td>
<td>JBOSS_JMX_USER and JBOSS_JMX_PWD</td>
</tr>
<tr>
<td>SQL _Server_2008 1.0.0</td>
<td>SA_PWD</td>
</tr>
<tr>
<td>initialize_db_script</td>
<td>db_username and db_password</td>
</tr>
<tr>
<td>Dukes_Bank_App</td>
<td>JAR_FILE, EAR_FILE, db_password, and db_user</td>
</tr>
<tr>
<td>Clustered Dukes Bank 2.1.0</td>
<td>Apache_LB 2.2.22 http_port and http_proxy_port</td>
</tr>
<tr>
<td>JBossAppServer 5.1.0</td>
<td>JBOSS_JMX_USER and JBOSS_JMX_PWD</td>
</tr>
<tr>
<td>MySQL 5.0.0</td>
<td>db_port and db_root_password</td>
</tr>
<tr>
<td>initialize_db_script</td>
<td>db_username, db_password, init_db_username, and init_db_password</td>
</tr>
<tr>
<td>Dukes_Bank_App</td>
<td>JAR_FILE, EAR_FILE, db_port, db_password, and db_user</td>
</tr>
</tbody>
</table>

8 Review the changed and impacted components in the update. The modified properties and dependent properties are highlighted.

9 Click **Update** to deploy the modified configuration for the application.

### What to do next

You can check the status of the deployment from the deployment summary page. See “Using the Deployment Summary Page,” on page 160.

## Sample vCloud Application Director Catalog Tasks

You can add predefined tasks in an execution plan during deployment. These tasks can be required or optional during the application deployment process.

For example, you must add and configure the RHN Repository predefined task in the execution plan to install or update the YUM repositories, for a Red Hat Linux-based application to run properly.

You can also create a custom task, add it to the predefined application, and deploy it to the cloud. For the custom task to be available in the Tasks page, you must add it to the catalog. See “Add a Custom Task to the Catalog,” on page 129.

- **Add Join Domain Predefined Task** on page 187
  The Join Domain predefined task allows a Windows virtual machine to join an Active Directory domain as part of the deployment process. The predefined task starts the Windows Domain Manager *Netdom.exe* utility, to automate this operation.

- **Add APT Repository Config Predefined Task** on page 188
  The APT Repository Config predefined task is a script used to update the APT repositories to install or update software on Ubuntu or other DEB-based operating systems.

- **Add a YUM Repository Config Predefined Task** on page 189
  The YUM Repository Config predefined task is a script used for updating the YUM repositories to install or update software on CentOS or other RPM-based operating systems.

- **Add a RHN Registration Predefined Task** on page 190
  The RHN Registration predefined task is a script used to update the YUM repositories to install Red Hat Enterprise Linux or other Red Hat operating systems that have the Red Hat Network with YUM.
Add Join Domain Predefined Task

The Join Domain predefined task allows a Windows virtual machine to join an Active Directory domain as part of the deployment process. The predefined task starts the Windows Domain Manager `netdom.exe` utility, to automate this operation.

With the predefined task, you do not need to manually configure a Windows virtual machine template with static domain settings and you can customize the task. The predefined task requires an extra restart cycle to complete the process of joining an Active Directory domain.

Prerequisites

- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Verify that your user account has the `ROLE_DEEPLOYER` deployer role assigned to it.
- Verify that vCloud Application Director is configured to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.
- For multiple deployments, manually enable the Change SID and Domain Join options for a Windows virtual machine template before you deploy an application. See “Enable SID Change and Domain Join for Windows Virtual Machine Templates,” on page 74.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Click the name of a Windows-based application.
   A list of application versions appear.
3. Select an application version.
4. Create a deployment profile.
5. In the Deployment Profile wizard, follow the prompts to Execution Plan.
6. If the node is clustered, click the Expand Cluster button ( ).
   If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.
7. Click the Add Script Task button ( ) and drag a predefined task in the blueprint.
   When you drag a predefined task, you see anchors ( *) that indicate where you can drop the predefined task.
   After you drag and drop a predefined task to a node, the Add Custom Task dialog box opens.
8. Select the predefined task from the Catalog Task Name drop-down menu.
   The supported operating systems, predefined task details, script, and property details appear in the dialog box.
9. On the Properties tab, configure the properties.
   a. Select the domain_name property, type a new name for the Windows domain in the Edit Property dialog box, and click Save.
   b. Select the domain_user property, specify the name of the domain user who can join the Active Directory in the Edit Property dialog box, and click Save.
   c. Select the domain_password property, type the domain user password in the Edit Property dialog box, and click Save.
d) (Optional) Select the apply_ou property, change the new value to Yes in the Edit Property dialog box to allow a specific organization unit in the Active Directory domain to join a Windows virtual machine, and click Save.

e) (Optional) Select the domain_ou property, specify the organization unit in the Active Directory domain in the Edit Property dialog box, and click Save.

10 Click OK to close the Edit Property dialog box.

The Join Domain predefined task is added to the execution plan.

11 Review the deployment profile settings and deploy the application.

What to do next

Explore whether to add a customized task to the vCloud Application Director catalog. See “Add a Custom Task to the Catalog,” on page 129.

Add APT Repository Config Predefined Task

The APT Repository Config predefined task is a script used to update the APT repositories to install or update software on Ubuntu or other DEB-based operating systems.

You can configure the APT Repository Config properties to add a new repository or remove all of the existing repositories. If you need more than one repository, you can create multiple tasks and link them in the execution plan by adding one task next to the other.

Use the predefined sample task only in a test environment.

Prerequisites

- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that the predefined task is added to the execution deployment plan before you add any services or application components that require APT for installing or updating software packages.
- Verify that vCloud Application Director is configured to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.

Procedure

1 On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2 Click the name of the application.
   A list of application versions appears.
3 Select an application version.
4 Create a deployment profile.
5 In the Deployment Profile wizard, follow the prompts to Execution Plan.
6 If the node is clustered, click the Expand Cluster button ( ).
   If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.
7 Click the Add Script Task button ( ) and drag a predefined task in the blueprint.
   When you drag a predefined task, you see anchors ( # ) that indicate where you can drop the predefined task.
   After you drag and drop a predefined task to a node, the Add Custom Task dialog box opens.
8 Select the predefined task from the **Catalog Task Name** drop-down menu.
   The supported operating systems, predefined task details, script, and property details appear in the dialog box.

9 On the **Properties** tab, configure the properties.
   a Select the `repository_name` property, type a new unique value identifying the repository in the Edit Property dialog box, and click **Save**.
   b Select the `source_str` property, type `http://site.example.com/debian distribution component1 component2 ...` in the **URL value** text box of the Edit Property dialog box, and click **Save**.
   A sample Ubuntu URL is `deb http://us.archive.ubuntu.com/ubuntu/ lucid main`.
   c Select the `remove_all_repos` property and define an appropriate value in the Edit Property dialog box.
   Set the value to `true` to remove all other repositories before you add the new configuration. You can also accept the default `false` value to add a new repository.

10 Click **Save** and **OK** to close the Edit Property dialog box.
   The APT Repository Config predefined task is added to the execution plan.

11 Review the deployment profile settings and deploy the application.

**What to do next**
Determine whether to add a customized task to the vCloud Application Director catalog. See “Add a Custom Task to the Catalog,” on page 129.

**Add a YUM Repository Config Predefined Task**

The YUM Repository Config predefined task is a script used for updating the YUM repositories to install or update software on CentOS or other RPM-based operating systems.

You can configure the YUM Repository Config properties to add a new repository or remove all of the existing repositories. If you need more than one repository, you can create multiple tasks and link them in the execution plan by adding one task next to the other.

Use the predefined sample task only in a test environment.

**Prerequisites**
- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Verify that your user account has the **ROLE_DEPLOYER** deployer role assigned to it.
- Verify that the predefined task is added to the execution deployment plan before you add any services or application components that require YUM for installing or updating software packages.
- Verify that vCloud Application Director is configured to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.

**Procedure**
1 On the vCloud Application Director title bar, click the drop-down menu and select **Applications**.
2 Click the name of the application.
   A list of application versions appears.
3 Select an application version.
4 Create a deployment profile.
5 In the Deployment Profile wizard, follow the prompts to **Execution Plan**.

6 If the node is clustered, click the **Expand Cluster** button ( ).
   If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.

7 Click the **Add Script Task** button ( ) and drag a predefined task in the blueprint.
   When you drag a predefined task, you see anchors ( ) that indicate where you can drop the predefined task.
   After you drag and drop a predefined task to a node, the Add Custom Task dialog box opens.

8 Select the predefined task from the **Catalog Task Name** drop-down menu.
   The supported operating systems, predefined task details, script, and property details appear in the dialog box.

9 On the **Properties** tab, configure the properties.
   a Select the repository_name property, type a new unique value identifying the repository in the Edit Property dialog box, and click **Save**.
   b Select the repository_url property, type a URL value in the Edit Property dialog box, and click **Save**.
      An example of a CentOS 6.3 32-bit URL is http://vault.centos.org/6.3/os/i386/.
   c Select the remove_all_repos property and define an appropriate value in the Edit Property dialog box.
   d Set the value to **true** to remove all other repositories before adding the new configuration.
   e Accept the default **false** value to add a new repository.

10 Click **Save** and **OK** to close the Edit Property dialog box.
   The YUM Repository Config predefined task is added to the execution plan.

11 Review the deployment profile settings and deploy the application.

**What to do next**
Consider adding a customized task to the vCloud Application Director catalog. See “**Add a Custom Task to the Catalog**,” on page 129.

**Add a RHN Registration Predefined Task**

The RHN Registration predefined task is a script used to update the YUM repositories to install Red Hat Enterprise Linux or other Red Hat operating systems that have the Red Hat Network with YUM.

The predefined task registers the virtual machine with the Red Hat Network using the credentials provided with a machine name VMware_AppDirector_$RANDOM, where $RANDOM is a short string that makes the virtual machine registration unique.

Use the predefined sample task only in a test environment.

**Prerequisites**

- Log in to vCloud Application Director with a user account that belongs to the Default group.
- Verify that your user account has the **ROLE DEPLOYER** deployer role assigned to it.
- Verify that the predefined task is added to the execution deployment plan before you add any services or application components that require YUM for installing or updating software packages.
Verify that vCloud Application Director is configured to use a proxy. See “Configure vCloud Application Director to Use a Proxy for External URLs,” on page 28.

Procedure
1. On the vCloud Application Director title bar, click the drop-down menu and select Applications.
2. Click the name of the application.
   A list of application versions appears.
3. Select an application version.
4. Create a deployment profile.
5. In the Deployment Profile wizard, follow the prompts to Execution Plan.
6. If the node is clustered, click the Expand Cluster button (▶).
   If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.
7. Click the Add Script Task button (▶) and drag a predefined task in the blueprint.
   When you drag a predefined task, you see anchors (‡) that indicate where you can drop the predefined task.
   After you drag and drop a predefined task to a node, the Add Custom Task dialog box opens.
8. Select the predefined task from the Catalog Task Name drop-down menu.
   The supported operating systems, predefined task details, script, and property details appear in the dialog box.
9. On the Properties tab, configure the properties.
   a. Select the rhn_username property, type the username value used to register the virtual machine with the Red Hat Network, and click Save.
   b. Select the rhn_password property, type the password value used to register the virtual machine with the Red Hat Network, and click Save.
10. Click OK to close the Edit Property dialog box.
    The RHN Registration predefined task is added to the execution plan.
11. Review the deployment profile settings and deploy the application.

What to do next
If you have a customized task, you can add it to the vCloud Application Director catalog. See “Add a Custom Task to the Catalog,” on page 129.

Sample vCloud Application Director Catalog Services
vCloud Application Director includes predefined components, such as services, in its catalog that are reusable components in several applications. These services are available to all user groups in vCloud Application Director. Logical templates must be added for each group outside the Default group.

Catalog Services
On the vCloud Application Director title bar, click the drop-down menu and select Catalog > Services to view the available sample services. The Catalog menu also includes standard logical templates, tasks, operating systems, tags, policies and external services.
An application architect can create an application blueprint and add the sample services to the applicable nodes and configure them. The sample services can also be configured when deploying a predefined application.

In the application blueprint, these sample services are grouped into Application Servers, Database Servers, Web Servers, Windows Services, Monitoring, and Other.

For recommended property configurations, supported operation systems, and application components information, see the VMware vCloud Application Director Catalog Services guide.

**NOTE** Use the predefined sample catalog services only in a test environment.
When you deploy an application, an item is added to the Deployments page in vCloud Application Director.

You can use the Deployments page to view the following items:

- List of all the deployments
- Deployment details of an individual deployment such as cloud provider, deployment environment, deployment profile, type and version of the application, or update process
- Deployments with policies and available updates

You can also start a policy scan, an update process, tear down a deployed application from the cloud, and remove an application deployment record from vCloud Application Director.

From the deployment summary page, you can navigate to the updates and compliance summary pages.

This chapter includes the following topics:

- “View Deployment Task and Blueprint Details for an Application,” on page 193
- “View Deployed VM Details and Execution Plan of an Application,” on page 195
- “Start a Policy Scan,” on page 197
- “Tear Down an Application from the Cloud,” on page 197
- “Delete an Application Deployment from vCloud Application Director,” on page 199
- “Cancel a Deployment or an Update Process,” on page 199
- “View Policy Compliance Summary,” on page 200

**View Deployment Task and Blueprint Details for an Application**

You can view details about the progress, success, or failure of a particular application deployment or an update process on the deployment summary page. You can view the IP addresses that were assigned, the cloud networks chosen, and the logs for each installation, configuration, and startup or update scripts that were run.

The deployment summary page shows the overall status of the deployment. Each deployment listed on the summary page has a snapshot of the application blueprint and deployment profile at the time that the deployment was started. If you make changes to the actual application blueprint or deployment profile, those changes do not affect the blueprint or deployment shown in a specific deployment. See “Using the Deployment Summary Page,” on page 160.

All of the user roles can view deployments that are associated with their user’s group.
Prerequisites
Verify that an application is deployed or an update process is initiated. See “Deploy an Application,” on page 155 or Chapter 13, “Updating Application Deployments,” on page 163.

Procedure
1. On the vCloud Application Director title bar, click the drop-down menu and select **Deployments**.
2. Click the name of the deployment.
   The deployment summary page with status windows appears. The VM Details status window provides quick access to each virtual machine, and lets you log in to the virtual machine deployed on vCloud Director.
3. To view the status of a task, a dependant task, or virtual machines acquiring IP addresses, click the **Expand** icon (eligible) to open the applicable window in the deployment summary page.
4. Click the **Minimize** icon (eligible) to view the entire deployment status summary.
5. For the overall deployment status, look at the Task Details status window.
6. To find the name of the deployment task, look at the Task Summary section in the Task Details status window.
   The Task Summary section lists details about the deployment or update profile, user role, policy compliance information, start and end time of the deployment, the last time the application was updated, run ID number of the subfolder containing logs, and the assigned to the application name.
   You need the vApp, machine, or instance name to find information about the deployment in vCloud Director, vCloud Automation Center, or Amazon EC2. You can also check the progress of a deployment from the vCloud Director, vCloud Automation Center, and Amazon EC2 user interface.
7. For deployment environment, cloud provider name, cloud provider description, host IP address, vCloud Director Org name, provisioning group and reservation policy, or Amazon VPC and associated Availability Zone, and User name details, look at the Deployment Location section in the Task Details status window.
   The deployment information in this section is a snapshot of the application blueprint and deployment profile at the time of the deployment. If you make changes to the actual application blueprint, those changes do not affect the blueprint shown in a specific deployment. The cloud provider and deployment information reflects the cloud provider mapping and deployment environment details that were created for the application deployment.
8. To view the application blueprint name and version information, look at the Application Details section in the Task Details status window.
9. To view the property overrides in the blueprint and deployment profile, expand the Blueprint status window.
   This status windows shows a snapshot that reflects the settings and definitions the blueprint contained at the time of the deployment. If you make changes to the actual application blueprint, those changes do not affect the blueprint shown in a specific deployment. You can delete a deployment without affecting components or services in the application.
   a. Select a service or application component and click the **Properties** tab.
   b. For an update process, in the **Properties** tab, the Previous Value column shows the catalog, blueprint, or deployment profile values of a previous update process.
      On the same tab, the New Value column shows the values added in the current update process.
10 View failed tasks in the expanded Blueprint status window.

If a task fails on a service or application component, the task failure icon (●) appears on the service or application component. When a task fails, the entire deployment is stopped, marked as Failed Deployment, and no future tasks are run. The task did not run icon appears on all the rest of the tasks.

If an update deployment occurs, the updated node application components and properties are highlighted. If the update deployment fails, the application component is marked as failed and the impacted properties are highlighted in red.

What to do next

Review the virtual machine details and the tasks available in the execution plan. See “View Deployed VM Details and Execution Plan of an Application,” on page 195.

View Deployed VM Details and Execution Plan of an Application

From the deployment summary page, you can expand the VM Details status window for virtual machine-specific information. You can also expand the Execution Plan status window to view the provisioning tasks running based on the dependencies defined in the application.

Virtual machine-related information, such as instance name, logical template, and cloud template in vCloud Director, vCloud Automation Center, or Amazon EC2, memory allocation, number of CPUs, and network connection details are available. You can also view the host name to easily identify the virtual machine in the vCloud Director or vCloud Automation Center deployments.

Prerequisites

- Verify that an application is deployed or that an update process is initiated. See “Deploy an Application,” on page 155 or Chapter 13, “Updating Application Deployments,” on page 163.
- For vCloud Director, verify that you installed the supported VMRC plug-in and supported Web browser. See “vCloud Application Director System Requirements,” on page 20.

Procedure

1 To locate the IP addresses of virtual machines, look at the VM Details status window.

IP addresses for each virtual machine appear in the IP address column when they are assigned.

For example, in a three-tiered application that includes a load balancer node, you might find the IP address of the load balancer and give that IP address to your end users.

2 Locate virtual machine-specific agent bootstrap logs in the table of virtual machines of the VM Details status window.

Agent bootstrap logs for each virtual machine are updated in the Log column. If the bootstrap process fails, the deployment is labeled Failed Deployment and the reason appears in the Task Details window.

3 Expand the window and click the ellipses button (…) in the Cloud Template column to retrieve the detailed virtual machine-specific cloud template information from either vCloud Director, vCloud Automation Center, or Amazon EC2.

This information includes information such as disk size, CPU, memory.

4 Expand the window and click the ellipses (…) in the Network Information column to retrieve the network details from either vCloud Director or Amazon EC2.

   **NOTE** For vCloud Automation Center, the networking information is not available. vCloud Automation Center uses the network connection specified in the vCenter Server template.

5 To view the custom properties defined for each node to override properties in the vCloud Automation Center blueprint, click the Extra Configuration icon (⋯).
6 For vCloud Director, click the virtual machine to open the console in a Web browser that supports the VMRC plug-in and log in to the virtual machine. After the IP address is acquired, the check mark under the Console column indicates that the virtual machine in the cloud is powered on, the agent bootstrap and install processes have succeeded.

7 For a deployment in progress, click **Refresh** in the upper-right corner of the page to update the status. The page refreshes in approximately 30 seconds.

8 For specific task details such as start time, end time, and last updated time, expand the Execution Plan status window. For each task, after the IP addresses are acquired and agents are bootstrapped, the components are deployed, installed, and configured. These processes occur according to the dependencies of the application shown by the arrows between the tasks in the execution plan.

9 Review the status of each task. Depending on the status of the task, an icon appears next to it.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🌧️</td>
<td>Task has not yet begun or did not run.</td>
</tr>
<tr>
<td>⬝</td>
<td>Task is in progress. The status icon appears when a task successfully completes. This icon also appears to show the overall completion status of the host, agent bootstrap, and network bootstrap provisioning tasks.</td>
</tr>
<tr>
<td>🌪️</td>
<td>Expand and view the details of each provisioning task.</td>
</tr>
<tr>
<td>🔥</td>
<td>Task requires reboot.</td>
</tr>
<tr>
<td>🌧️</td>
<td>Task has failed. The status icon appears when a task fails. This icon also appears to show the overall failed status of the host, agent bootstrap, or network bootstrap provisioning task.</td>
</tr>
<tr>
<td>🌦️</td>
<td>Expand and view the logs of one or more failed provisioning tasks.</td>
</tr>
<tr>
<td>☁️</td>
<td>Task is waiting for one of the dependencies to finish running.</td>
</tr>
</tbody>
</table>

10 For details about a component or action script and its properties, expand the Execution Plan status window, click the gear icon (⚙️) next to the task, and select **View Component Properties**. The details listed there are the settings and definitions that the blueprint contained at the time of deployment.

11 To view an action script failure, expand the Execution Plan status window, click the gear icon (⚙️) next to the task, and select **View Action Script**.

12 To access the virtual machine log files, click the gear icon (⚙️) next to the task and select **View Virtual Machine Logs**.

13 To view the property values for the script, click the gear icon (⚙️) next to the task and select **View Component Properties**.

14 For details about updated deployments, expand the Execution Plan status window and examine the updated nodes. For a scaled deployment, the execution plan displays the clustered nodes that were modified and the update scripts of the impacted nodes. The host, agent bootstrap, and network bootstrap provisioning tasks appear only on the scaled out node. The dependent nodes do not have any provisioning tasks. For an update process to modify the configuration of a deployed application, the execution plan displays the update scripts of the changed and impacted nodes.
What to do next

To tear down a deployed application from the cloud, see “Tear Down an Application from the Cloud,” on page 197.

To delete a deployment record from vCloud Application Director, see “Delete an Application Deployment from vCloud Application Director,” on page 199.

Start a Policy Scan

You can start a policy scan on a deployment to evaluate all of the policy instances defined under the deployment environment that corresponds to a deployment. vCloud Application Director runs a policy assessment as the first step when you initiate a deployment or update process.

Policy scans can be performed only on active deployments. For example, if an update process fails, you must restore the state of the current deployment to active before retrying the policy scan.

If a policy used in the deployment environment for a policy instance is updated, the user must delete the existing policy instance and create a policy instance based on the updated policy.

Prerequisites

- Verify that your user account has the ROLE_DEPLODER deployer role assigned to it.
- Verify that a policy or policy instance has changed.
- Verify that deployments with the obsolete policy definitions are available in vCloud Application Director.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Type the name of the existing deployment to scan in the search text box.
3. From the search result list, select the deployment to scan.
   The deployment summary page opens.
4. In the toolbar above the deployment summary, select Operations > Scan.
5. Confirm the scan deployment process.

The compliance summary page opens to display the overall compliance status and policy details. The policy scan appears in the time line with the time stamp of when the scan was initiated.

What to do next

Track the policy compliance details of the deployment scan. See “View Policy Compliance Summary,” on page 200.

Tear Down an Application from the Cloud

In vCloud Application Director, you can start the teardown process to remove the vCloud Director vApp and associated virtual machines, vCloud Automation Center virtual machine and the associated virtual machine in vCenter Server, or Amazon EC2 instances from the cloud environment.

You can tear down a deployed application from the vCloud Application Director user interface or the command-line interface. See “Using CLI to Tear Down a Deployment,” on page 212. In the user interface you can define a TEARDOWN life cycle stage script for a service version and application component to remove an application and associated virtual machines or an updated application.
If a deployment fails after installing one or more virtual machines that are part of an application, or if an application is not used, you can use vCloud Application Director to tear down the application. All virtual machines in the application are removed from their hosts in the cloud.

Deployment tear down from the cloud does not remove the deployment record from the Deployments page in vCloud Application Director. To remove the deployment record from the Deployments page, see “Delete an Application Deployment from vCloud Application Director,” on page 199.

Prerequisites

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that the virtual machines that are part of the application still exist in the cloud. If, for example, your environment has a policy of deleting virtual machines after a certain number of days, the virtual machines might already be deleted.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Type the name of the deployment to tear down in the search text box.
3. From the search result list, select the deployment to tear down.
   The deployment summary page opens.
4. In the toolbar above the deployment summary, select Operations > Teardown.
   - In the Teardown Deployment dialog box, select Quick Teardown to remove the virtual machines for this deployment and click Teardown.
   - In the Teardown Deployment dialog box, select Assisted Teardown to open the Teardown wizard and click Teardown.
     For the quick tear down process, the TEARDOWN life cycle stage script does not run.
     For the assisted tear down process, the Teardown wizard displays the TEARDOWN life cycle stage script that you can configure. You can also view the order in which the tasks in the script run, and review the changes in the script before you run the script to remove the virtual machines for this deployment.
5. (Optional) If the teardown process fails, repeat the process.

Above the task status windows, the task time line shows the status of the teardown process. After the virtual machines are deleted successfully from their hosts in the cloud, the status appears in the Task Summary section and in the overall deployment status.

After a teardown process starts, you cannot update the deployed virtual machines even if the teardown process fails and the virtual machines exist in the cloud environment. Ask your cloud administrator to tear down the deployment from the cloud.

What to do next

To delete the record of a deployment from the Deployments page, see “Delete an Application Deployment from vCloud Application Director,” on page 199.
Delete an Application Deployment from vCloud Application Director

After you remove an application from the cloud, or if you do not need details for a particular deployment, you can delete the deployment from the Deployments page in vCloud Application Director.

Deleting a deployment from the Deployments page in vCloud Application Director does not delete the deployed application, vApp, and its virtual machines for vCloud Director, vCloud Automation Center virtual machine and the associated virtual machine in vCenter Server, or Amazon EC2 instance from the cloud. To delete an application from the cloud, see “Tear Down an Application from the Cloud,” on page 197.

Prerequisites

Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select Deployments.
2. Type the name of the deployment to delete in the search text box.
3. From the search result list, select the deployment to delete.
   The deployment summary page opens.
4. In the toolbar above the deployment summary, select Operations > Delete.
5. Confirm the deletion.
6. If you delete a deployment from the Deployments page without first tearing down the deployed application from the cloud, you must use vCloud Director, vCloud Automation Center, or Amazon EC2 to remove the residual components in the cloud.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vCloud Director</td>
<td>Delete the vApp and associated virtual machines.</td>
</tr>
<tr>
<td>vCloud Automation Center</td>
<td>Delete the vCloud Automation Center virtual machine and the associated virtual machine in vCenter Server.</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>Stop the instances of the deployment, delete the Security Group corresponding to the deployment, and release the Elastic IP address assigned to the instances of the deployment.</td>
</tr>
</tbody>
</table>

The deployment record is removed from the Deployments page.

Cancel a Deployment or an Update Process

In some cases, if a deployment or update process is in progress indefinitely and does not show either a pass or fail deployment status, you can mark the deployment or update process as failed without stopping provisioning in the cloud environment.

**Note** When you cancel a deployment or an update process, vCloud Application Director does not delete the deployment or update process.

Prerequisites

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that you have a deployment or update process in progress indefinitely for more than an hour.
Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select **Deployments**.
2. Type the name of the deployment or update process to cancel in the search text box.
3. From the search result list, select the deployment or update process that was in progress for more than an hour.
   The deployment summary page opens.
4. Click **Cancel** in the toolbar above the deployment summary.
5. Confirm your selection.

What to do next

When you mark a deployment as failed, you can tear down the deployment from the cloud or delete the application deployment record from vCloud Application Director. See Chapter 15, “Managing Deployments,” on page 193.

For an update process, you can continue to interact with the deployment by starting another update process. See Chapter 13, “Updating Application Deployments,” on page 163.

View Policy Compliance Summary

You can view the overall compliance status and associated policy details of a deployment.

During deployment, the SCAN life cycle script in the policy definition assesses the compliance state of a deployment. The overall compliance status icon turns red or green, depending on the violation or compliance of policy definitions in the deployment. The status displayed on the compliance summary page is a snapshot of the application blueprint at the time the deployment was started. The number of compliant and violated policy definitions and critical and noncritical definitions appear in the overall compliance status. The associated deployment profile is located next to the overall status.

Above the overall compliance status, a policy scan timeline contains a time stamp that shows when the deployment scan was initiated and any subsequent deployment compliance scans. For example, a subsequent scan checks whether an updated policy definition in the catalog is compliant with the existing deployment.

Prerequisites

- Verify that your user account has the ROLE_DEPLOYER deployer role assigned to it.
- Verify that a policy instance is applied to the deployment.

Procedure

1. On the vCloud Application Director title bar, click the drop-down menu and select **Deployments**.
2. Type the name of a deployment that has a policy definition in the search text box.
3. From the search result list, select the deployment.
   The deployment summary page opens.
4. In the toolbar above the deployment summary, select **Deployment View > Compliance View**.
   The overall compliance status appears. The detailed policies status is listed in a table.
In the Policy column, click a policy name that violates compliance to view details in the compliance log. Based on the information in the compliance log you can fix the violation in the policy definition and deploy the application.

For example, a memory limit policy shows two virtual machines out of 10 virtual machines in the application deployment are violating the policy. You can view the Details column of the policy and the Compliance Log section to identify the violation and apply a fix to the application.

**What to do next**

If the deployment is in progress, track the deployment status from the deployment summary page. See “Using the Deployment Summary Page,” on page 160.
The vCloud Application Director CLI is a Spring Roo-based client that communicates to the vCloud Application Director server over HTTPS using REST APIs.

User accounts with the system administrator role can use the vCloud Application Director CLI to create and manage users, groups, and LDAP configurations. User accounts with the deployer roles can use the CLI to deploy or tear down applications from the cloud. See “Predefined Users, Groups, and Roles,” on page 36.

**NOTE** The predefined user accounts are disabled by default. If you do not enable these accounts, you can use only the admin user. The password for the admin user is the admin password that was set the first time the appliance was started.

This chapter includes the following topics:

- “General CLI Options,” on page 203
- “Managing Users and Groups,” on page 204
- “Managing LDAP Configurations,” on page 206
- “Managing Cloud Tunnels,” on page 207
- “Deploying and Updating an Application Using CLI,” on page 208
- “Using CLI to Tear Down a Deployment,” on page 212

### General CLI Options

With the vCloud Application Director command-line interface options, you can create and manage users and groups, deploy an application, update a deployed application, or tear down an application from the cloud.

After you log in to the CLI program, in the roo shell prompt, press the **Tab** key to display the list of available command options in vCloud Application Director. If you use multiple words with spaces between the words, enclose the words in quotation marks. Often after you enter a command, the system displays many details, in addition to indicating whether the command was successful.

**NOTE** Do not use command options that are not available in CLI.

The general CLI options are available to all user groups in vCloud Application Director.

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>Lists use information.</td>
</tr>
<tr>
<td>cliversion</td>
<td>Displays the CLI version information.</td>
</tr>
</tbody>
</table>
### Table 16-1. General CLI Commands (Continued)

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>login</td>
<td>Logs in the current user to the root shell.</td>
</tr>
<tr>
<td>logout</td>
<td>Logs out the current user without closing the root shell prompt. You can log out and log in as a different user.</td>
</tr>
<tr>
<td>status</td>
<td>Indicates whether a user is logged in. If a user is logged in, user name is displayed.</td>
</tr>
<tr>
<td>exit</td>
<td>Exits the CLI program.</td>
</tr>
</tbody>
</table>

### Managing Users and Groups

To manage users and groups from the CLI, you must create users and groups.

Verify that your user account has the **ROLE_SYSTEM_ADMIN** system administrator role assigned to it.

See “Create Users and Groups with vCloud Application Director CLI,” on page 37.

**Note** Unknown CLI options are ignored by the Roo shell.

### Table 16-2. Manage Users and Groups in the CLI

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>change-password</td>
<td>Changes the password of the current user.</td>
</tr>
<tr>
<td>change-user-password <strong>--username UserName</strong></td>
<td>Changes a user's password. If you run the <strong>--password</strong> parameter with the login command or a command that lets you add a password, your password is saved as plain text in the <em>darwin-cli-history.log</em> file located in the current directory. By default, the CLI deletes the log file. For added security, delete this log file.</td>
</tr>
<tr>
<td>create-group <strong>--name GroupName --description Description</strong></td>
<td>Creates a group. For example, to create a group called Test Group, use the command <code>create-group --name &quot;Test Group&quot;</code>.</td>
</tr>
<tr>
<td>CLI Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>create-user --username UserName --firstName FirstName --lastName LastName --enabled true --roles ROLE_SYSTEM_ADMIN, ROLE_CATALOG_ADMIN, ROLE_CLOUD_ADMIN, ROLE_SYSTEM_INTEGRATOR, ROLE_DEPLOYER, ROLE_APP_ARCHITECT --group GroupName --email EmailAddress</td>
<td>Creates a user. If you specify all the roles shown in this example, the user is the equivalent of the admin user. Specify only the roles the specific user must have. The valid values for the enabled option are true, yes, 1, false, no, and 0. If you run the --password parameter with the login command or a command that lets you add a password, your password is saved as plain text in the darwin-cli-history.log file located in the current directory. By default, the CLI deletes the log file. For added security, delete this log file.</td>
</tr>
<tr>
<td>update-user --username UserName --roles ROLE_SYSTEM_ADMIN, ROLE_CLOUD_ADMIN</td>
<td>Changes the roles assigned to a user. In this example, the roles ROLE_SYSTEM_ADMIN, ROLE_CLOUD_ADMIN are used. Substitute a comma-separated list of the roles you want the user to have. Do not add a space between the comma-separated list.</td>
</tr>
<tr>
<td>update-user --username UserName --group GroupName</td>
<td>Changes a user's group.</td>
</tr>
<tr>
<td>enable-user --username UserName</td>
<td>Enables a user account.</td>
</tr>
<tr>
<td>disable-user --username UserName</td>
<td>Disables a user account. CAUTION The disable option also allows you to disable your user account. If you are the only user with system administrator privileges, you cannot reenable your user account.</td>
</tr>
<tr>
<td>list-user</td>
<td>Lists information about all users.</td>
</tr>
<tr>
<td>list-user --username UserName</td>
<td>Displays information about a specific user.</td>
</tr>
<tr>
<td>list-roles</td>
<td>Lists information about all the roles.</td>
</tr>
<tr>
<td>list-group</td>
<td>Shows information about all the groups.</td>
</tr>
</tbody>
</table>
Managing LDAP Configurations

With vCloud Application Director you can perform various operations such as activating, importing, updating, or deleting an LDAP configuration.

When you run the import command, LDAP completes the following processes.

1. Verifies the existence of an LDAP user or group name in the LDAP server.
2. Creates an entry in the vCloud Application Director database for that LDAP user or group.
3. Adds the user or group SID information in the database.
4. Associates the LDAP user or group in the database with the specified vCloud Application Director group and roles.

**Note** While importing existing LDAP users and groups, LDAP does not collect or transfer any secure information from the directory to vCloud Application Director.

The existing LDAP users or groups that are imported to vCloud Application Director and assigned roles can log in to the vCloud Application Director Web interface using their LDAP credentials.

Verify that your user account has the **ROLE_SYSTEM_ADMIN** system administrator role assigned to it.

See “Create and Activate an LDAP Configuration,” on page 38.

**Table 16-3. Using CLI to Manage LDAP Configurations**

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create-ldap-config</td>
<td>Creates an LDAP configuration and saves the configuration in the vCloud Application Director database.</td>
</tr>
<tr>
<td>activate-ldap-config --configname</td>
<td>Activates an LDAP configuration in the vCloud Application Director server to authenticate against an LDAP configuration. Note: The LDAP configuration name should not include any periods.</td>
</tr>
<tr>
<td>update-ldap-config --configname</td>
<td>Updates an existing LDAP configuration in the system. Note: The LDAP configuration name should not include any periods.</td>
</tr>
<tr>
<td>import-ldap-group --name LDAPGroup --configname LDAPConfigName --group GroupName --roles ROLE_CLOUD_ADMIN</td>
<td>Imports an existing LDAP group, configures the group to become a member of a vCloud Application Director group, and assigns the vCloud Application Director cloud administrator role to the LDAP group. Note: The LDAP configuration name should not include any periods because an error might occur when you import a LDAP group. For example, to import an LDAP group called Admin Group to the vCloud Application Director group called Default and assign this group the ROLE_CLOUD_ADMIN role, use the command import-ldap-group --name &quot;Admin Group&quot; --group Default --configname LDAPConfigTest --roles ROLE_CLOUD_ADMIN.</td>
</tr>
</tbody>
</table>
### Table 16-3. Using CLI to Manage LDAP Configurations (Continued)

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>import-ldap-user</code> <strong>--name</strong> LDAPUser <strong>--configname</strong> LDAPConfigName <strong>--group</strong> GroupName <strong>--roles</strong> ROLE_CLOUD_ADMIN</td>
<td>Imports an existing LDAP user, configures the user to become a member of a vCloud Application Director group, and assigns the vCloud Application Director cloud administrator role to the LDAP user. In this command, the GroupName is the name of the vCloud Application Director group that the LDAP user will be associated with. <strong>Note</strong> You must have an active LDAP configuration before you can import LDAP users.</td>
</tr>
<tr>
<td><code>update-ldap-user</code> <strong>--name</strong> UserName <strong>--group</strong> GroupName <strong>--roles</strong> ROLE_APP_ARCHITECT</td>
<td>Updates an existing user that was imported from the LDAP directory and assigns the user the vCloud Application Director application architect role.</td>
</tr>
<tr>
<td><code>update-ldap-group</code> <strong>--name</strong> LDAPGroupName <strong>--group</strong> GroupName <strong>--roles</strong> ROLE_CATALOG_ADMIN, ROLE_CLOUD_ADMIN</td>
<td>Updates an existing group that was imported from the LDAP directory. In this example, the group has the ROLE_CATALOG_ADMIN and ROLE_CLOUD_ADMIN roles assigned.</td>
</tr>
<tr>
<td><code>list-ldap-configs</code></td>
<td>Lists existing LDAP configurations in the system.</td>
</tr>
<tr>
<td><code>list-ldap-principals</code></td>
<td>Lists all of the users and groups imported from the LDAP directory.</td>
</tr>
<tr>
<td><code>print-active-ldap-config</code></td>
<td>Lists the details of the active LDAP configurations.</td>
</tr>
<tr>
<td><code>print-named-ldap-config</code> <strong>--configname</strong> LDAPConfigName</td>
<td>Displays the details of the named LDAP configuration. <strong>Note</strong> The LDAP configuration name should not include any periods.</td>
</tr>
<tr>
<td><code>delete-ldap-user</code> <strong>--name</strong> UserName <strong>--configname</strong> LDAPConfigName</td>
<td>Removes a user from the local database. <strong>Note</strong> The LDAP configuration name should not include any periods.</td>
</tr>
<tr>
<td><code>delete-ldap-group</code> <strong>--name</strong> GroupName <strong>--configname</strong> LDAPConfigName</td>
<td>Removes a group from the local database. <strong>Note</strong> The LDAP configuration name should not include any periods.</td>
</tr>
<tr>
<td><code>disable-ldap</code></td>
<td>Deactivates the current LDAP configuration and removes the LDAP authentication from the authentication chain in the system, so that only the local authentication is available. The deactivated LDAP configurations are available on the system and can be reactivated.</td>
</tr>
</tbody>
</table>

### Managing Cloud Tunnels

You can perform operations such as creating, updating, enabling, or deleting a secure cloud tunnel connection between the vCloud Application Director appliance and an Endpoint VM in an Amazon EC2 VPC.

Verify that your user account has the ROLE_CLOUD_ADMIN cloud administrator role assigned to it.

See “Create a Cloud Tunnel to Connect to Amazon EC2,” on page 84.
Table 16-4. Using the CLI to Manage Cloud Tunnels

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>create-cloud-tunnel --name</td>
<td>Creates a secure cloud tunnel. For example, you can use the following command to create a cloud tunnel called EC2 Tunnel.</td>
</tr>
<tr>
<td>TunnelName --description &quot;TunnelDescription&quot; --enabled false --externalAddress EndpointVMElasticIP --sshPort 22 --internalAddress EndpointVMPrivateIP --proxyUrl ProxyURL --username UserName --privateKeyPath PrivateKeyFilePath</td>
<td></td>
</tr>
<tr>
<td>list-cloud-tunnels --name TunnelName</td>
<td>Retrieves a list of all the available cloud tunnels or a particular cloud tunnel, when specified. To view the details of a particular cloud tunnel, use the command list-cloud-tunnels --name EC2 Tunnel.</td>
</tr>
<tr>
<td>update-cloud-tunnel --name PreviousTunnelName --new-name NewTunnelName --description &quot;TunnelDescription&quot; --externalAddress EndpointVMElasticIP --sshPort 22 --internalAddress EndpointVMPrivateIP --proxyUrl ProxyURL --username UserName --privateKeyPath PrivateKeyFilePath</td>
<td>Changes the values of various parameters of an existing cloud tunnel. For most of the parameters, this command works only if the cloud tunnel is disabled.</td>
</tr>
<tr>
<td>enable-cloud-tunnel --name TunnelName</td>
<td>Enables an existing cloud tunnel. This command initiates a background operation that establishes the secure cloud tunnel connection.</td>
</tr>
<tr>
<td>disable-cloud-tunnel --name TunnelName</td>
<td>Disables an existing cloud tunnel. This command initiates a background operation that disconnects the secure cloud tunnel connection.</td>
</tr>
<tr>
<td>delete-cloud-tunnel --name TunnelName</td>
<td>Deletes an existing cloud tunnel. The cloud tunnel must be disabled for it to be deleted. Removes the cloud tunnel connection between the vCloud Application Director appliance and the Amazon EC2 environment.</td>
</tr>
</tbody>
</table>

Deploying and Updating an Application Using CLI

To deploy an application, your user account must have the ROLE_DEPLOYER deployer role assigned to it. Before you deploy an application, verify that you have a deployment profile that is complete, saved, and free of validation errors. You also need the name of the application. Verify that the deployed application includes a clustered node before you initiate an update process to scale a clustered node.
You can perform a quick deployment of an application, check the status of an application deployment, initiate a scale out process for a deployed application, or use the CLI to modify service and application component configurations of deployed applications using the CLI. You can use the vCloud Application Director user interface to perform all of these tasks. See “Quick Deploy an Application,” on page 156, “Using the Deployment Summary Page,” on page 160, “Initiate an Update Process to Scale Out Deployments,” on page 164, or “Initiate an Update Process to Modify Configurations,” on page 169.

You can also view the deployed virtual machine status from the vCloud Director, vCloud Automation Center, and Amazon EC2 user interfaces.

**IMPORTANT** To update deployed applications, install the vCloud Application Director for Release Automation edition.

<table>
<thead>
<tr>
<th>Table 16-5. Deploy or Update an Application in the CLI</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLI Command</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>deploy-application --destination ApplicationName-ApplicationVersion-DeploymentProfileName</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>deployment-status --deploymentName DeploymentName</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>


**Table 16-5. Deploy or Update an Application in the CLI (Continued)**

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>deploy-application --destination ApplicationNameApplicationVersion --deploymentProfileName --propertiesFile FileName.xml</td>
<td>Deploys the latest application version with new property values. You can create a properties file that defines new property values for required properties and properties that are overridable at deployment. The following code example defines new values for the global_conf, http_node_port, and http_server_port properties in the Apache LB and AppServer nodes.</td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="UTF-8"?>
<ns1:config-update-properties xmlns:ns1="http://www.test.com/darwin/schema/bea ns/api">
    <node>
        <name>load_balancer</name>
        <node-component>
            <name>Apache_LB</name>
            <property>
                <key>http_proxy_port</key>
                <value>9001</value>
            </property>
        </node-component>
    </node>
    <node>
        <name>appserver</name>
        <node-component>
            <name>Dukes_Bank_App</name>
            <property>
                <key>EAR_FILE</key>
                <value><![CDATA[http://192.10.1.113/Share/UPRConfig/CDB/d ukesbank-2b-Update1-IndexPage.ear]]></value>
            </property>
            <property>
                <key>JAR_FILE</key>
                <value><![CDATA[http://192.10.1.113/Share/UPRConfig/CDB/m ysql-connector-java-5.1.8.jar]]></value>
            </property>
        </node-component>
    </node>
</ns1:config-update-properties>
```

For example, to deploy the Clustered Dukes Bank application version 2.1.0, with the staging-dep deployment profile, and new.props properties file, use the command deploy-application --destination "Clustered Dukes Bank App-2.1.0-DP_MAN_VCD" --propertiesFile ~/new.props.xml.

Type `deploy` and press the Tab key to display the possible options for the command. If the list does not display quotation marks for multiple words with spaces between the words, enclose the words in quotation marks.
### Table 16-5. Deploy or Update an Application in the CLI (Continued)

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| `update-scaleout --deploymentName DeploymentName --propertiesFile FileName.xml` | Initiates an update process to scale a deployed application. You must create a properties file that defines a new cluster size value for a node. The following code example defines cluster size values for the AppServer1 and DBServer1 nodes.  
  ```xml
  <?xml version="1.0" encoding="UTF-8"?>
  <ns1:scaleout-properties xmlns:ns1="http://www.test.com/darwin/schema/bea ns/api">
    <cluster-node>
      <name>appserver</name>
      <scale-out-by>1</scale-out-by>
    </cluster-node>
  </ns1:scaleout-properties>
  ```  
  For example, to scale out the Clustered Dukes Bank application, version 2.1.0 with `scaleout` properties file, use the command `update-scaleout --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-6-cc8e3b20-43a0-4c22-be3e-49d4f31ab8e8" --propertiesFile ~/scaleout.xml`.  
  Type `update` and press the Tab key to display the possible options for the command. If the list does not display quotation marks for multiple words with spaces between the words, enclose the words in quotation marks. |
| `update-config --deploymentName DeploymentName --configUpdatePropertiesFile ConfigUpdatePropertiesFileName.xml` | Initiates an update process to modify configurations of existing services or application components in a deployed application. You must create a config update properties file to configure an existing service. The following code example modifies the `global_conf` and `db_port` properties in the vFabric tc Server service of the AppServer node.  
  ```xml
  <?xml version="1.0" encoding="UTF-8"?>
  <ns1:config-update-properties xmlns:ns1="http://www.test.com/darwin/schema/bea ns/api">
    <node>
      <name>load_balancer</name>
      <node-component>
        <name>Apache_LB</name>
        <property>
          <key>http_proxy_port</key>
          <value>9001</value>
        </property>
      </node-component>
    </node>
  </ns1:config-update-properties>
  ```  
  For example, to modify a configuration in the Clustered Dukes Bank application, version 2.1.0 with `configupdate` properties file, use the command `update-config --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-2-19d63535-673e-4766-b380-de4e6ec3676d" --configUpdatePropertiesFile ~/configupdate.xml`.  
  Type `update` and press the Tab key to display the possible options for the command. If the list does not display quotation marks for multiple words with spaces between the words, enclose the words in quotation marks. |
Using CLI to Tear Down a Deployment

To tear down a deployment, your user account must have the deployer role (ROLE_DEPLOYER) assigned to it.

Before you tear down a deployed application from the cloud, verify that you have the deployment name of the deployed application. You can monitor the status of the teardown process from the vCloud Application Director user interface. See “Tear Down an Application from the Cloud,” on page 197.

Table 16-6. Remove Deployment in the CLI

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>teardown --deploymentName DeploymentName</td>
<td>Tear down a deployed application from the vCloud Director. For example, the <code>teardown --name appd-TestApp-1.0.0-admin-3-a99309aa-aa5b-45c2-b6a8-bac4a421178e</code> command removes the application from the cloud. Type <code>teardown</code> and press the Tab key to display the possible options for the command. If the list does not display quotation marks for multiple words with spaces between the words, enclose the words in quotation marks.</td>
</tr>
</tbody>
</table>
You can import and export preconfigured application blueprints and their associated deployment profiles, services, and available custom tasks across vCloud Application Director instances to help you further customize your applications.

You can import and export packages created in vCloud Application Director 5.0 and 5.2 to 6.0. You can also import and export packages between different vCloud Application Director 6.0 instances.

For example, if you decide to switch from an instance with vCloud Application Director for Provisioning to a new instance with vCloud Application Director for Release Automation edition, you can transfer your preconfigured application blueprints and associated components to the new instance by using the import and export features.

You can start the CLI from a vCloud Application Director remote machine. See “Start the CLI Remotely,” on page 36.

Verify that your user account has the ROLE_APP_ARCHITECT application architect role and the ROLE_CATALOG_ADMIN catalog administrator role assigned to it.

If the import or export package is larger than the memory available on the instance, an error message informs you to reallocate the memory. For example, you can use the java -Xmx6000m -jar darwin-cli.jar command to allocate 6GB of memory for an instance.

**Using the CLI Export Command on page 213**

The CLI export command creates a package that consists of applications and their associated blueprints and deployment profiles, external services, policies, services, logical templates, and available custom tasks to export between different vCloud Application Director instances.

**Using the CLI Import Command on page 216**

The CLI import command imports a package that includes applications and their associated blueprints and deployment profiles, external services, policies, services, logical templates, and available custom tasks between different vCloud Application Director instances.

**Using the CLI Export Command**

The CLI export command creates a package that consists of applications and their associated blueprints and deployment profiles, external services, policies, services, logical templates, and available custom tasks to export between different vCloud Application Director instances.

You cannot export application, service version, external service, script task, and policy version names that contain characters other than space, tab, underscore, new line, or carriage return.

You can export only those applications, services, and external services that the current group of the logged in user owns. When you run the import command, the entire package is imported to the designated server. You cannot selectively import elements from the export package.
When you export an application, service, external service, script task, or policy version, all of the secure properties are removed by default to avoid exporting sensitive information such as passwords to another vCloud Application Director instance.

**Table 17-1. CLI Export Commands**

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>export-package --exportFilePath ExportFilePath</code></td>
<td>Exports an application and service version. For example, you can specify the <code>exportFilePath</code> as <code>/Users/UserName/tmp/test.xml</code>. If the specified directory in the <code>exportFilePath</code> is not available when you run the export command, the operation fails. The <code>ApplicationVersionSpecification</code> command option includes <code>ApplicationName: major . minor . micro [-qualifier]</code>. Specifying <code>ServiceVersionSpecification</code> is optional because, during the export process all of the application dependencies are exported. The command option includes <code>ServiceName: major . minor . micro [-qualifier]</code>. You can export multiple application and service versions at the same time. For example, <code>export-package --exportFilePath /Users/test_user/tmp/test.xml --applicationVersion &quot;Clustered Dukes Bank App:2.1.0, Clustered Zimbra App:7.1.0, App_with_Qualifier:11.22.33-RELEASE&quot; --serviceVersion &quot;Hyperic HQ Agent: 4.6.0, Hyperic HQ Server:4.6.0, Preinstalled vFabric GemFire Server for Ubuntu:6.6.0&quot;</code> If the application version contains spaces, you must put the application version in double quotation marks.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath ExportFilePath</code></td>
<td>Exports the external service version. The <code>ExternalServiceVersion</code> command option includes <code>ExternalServiceName: major . minor . micro [-qualifier]</code>. Specifying <code>substituteSecuredProperties</code> is optional because, during the export process if the option is not specified, all of the secured property values are removed. If the command option is specified, all of the secured property values are encrypted with the <code>VirtualMachine</code> string. <code>export-package --exportFilePath /Users/test_user/tmp/test.xml --externalServiceVersion LoadBalancer:1.0.0</code> If the external service name contains spaces, you must put the external service name in double quotation marks.</td>
</tr>
<tr>
<td>CLI Command</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>export-package --exportFilePath ExportFilePath</code></td>
<td>Exports the policy version. The <code>PolicyVersion</code> command option includes <code>PolicyName: major.minor.micro[-qualifier]</code>. Specifying <code>substituteSecuredProperties</code> is optional because, during the export process if the option is not specified, all of the secured property values are removed. If the command option is specified, all of the secured property values are encrypted with the VirtualMachine string.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath /Users/test_user/tmp/test.xml --policyVersion &quot;Memory Policy:1.0.0&quot;</code></td>
<td>If the policy name contains spaces, you must put the policy name in double quotation marks.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath allApps.pkg --applicationVersion ALL</code></td>
<td>Exports all of the applications and associated versions.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath allServiceVersions.pkg --serviceVersion ALL</code></td>
<td>Exports all of the services and associated versions.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath allAVSV.pkg --applicationVersion ALL --serviceVersion ALL</code></td>
<td>Exports all of the application and service versions.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath allExternalServiceVersions.pkg --externalServiceVersion ALL</code></td>
<td>Exports all of the external service versions.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath allPolicied.pkg --policyVersion ALL</code></td>
<td>Exports all of the policy versions.</td>
</tr>
<tr>
<td><code>export-package --exportFilePath /Users/test_user/tmp/mixed_package.xml --applicationVersion &quot;DukesBank:1.0.0,SpringTravel:2.0.0,Zimbra:1.0.0&quot; --serviceVersion &quot;MySQL:1.0.0,RabbitMQ:2.0.0,Postgres:1.0.0&quot; --externalServiceVersion &quot;LoadBalancer:1.0.0,vPostgres 9.x:9.2.0&quot; --scriptTaskVersion ALL</code></td>
<td></td>
</tr>
</tbody>
</table>
Using the CLI Import Command

The CLI import command imports a package that includes applications and their associated blueprints and deployment profiles, external services, policies, services, logical templates, and available custom tasks between different vCloud Application Director instances.

Before you import a deployment profile, save the custom task you added to the execution deployment plan. See “Review the Execution Plan and Add Custom Tasks,” on page 153.

The export package includes the applications, services, external services, script task, or policies you selected to import. When you run the import command, the entire package is imported to the designated server. You cannot selectively import elements from the export package.

When you import packages between different vCloud Application Director instances, group permissions are not transferred and groups are not created. During the import process, the current group of the logged in user is used to add group permissions to applications, services, or logical templates created. Importing external services is supported, provided that the external service you are importing from the package is owned by the current group of the logged in user. You cannot import an external service on its own or as a dependency if it is owned by another group.

You cannot import external services with the same name into different groups. If you attempt to perform this import process, the operation fails with an error.

For the import process to finish successfully, the elements in the package cannot include non-ASCII characters.
### Table 17-2. CLI Import Commands

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| `import-package --importFilePath`  
`ImportFilePath`  
`--conflictResolutionAction CHECK` | Compares the package and publicly available repository elements that are in the group you are logged in to, lists the package elements that match and do not match the repository. Elements are not imported to the repository when you run this command. The match is based on the name and version qualifiers such as major, minor, micro, and optional. You must set one of the following settings for the `conflictResolutionAction` command option.  
- CHECK  
- SKIP  
- OVERWRITE  
- IMPORTASNEW  
For example, you can specify the `importFilePath` as, `/Users/UserName/tmp/test.xml`. If the specified directory in the `ImportFilePath` is not available when you run the import command, the operation fails. You can import multiple application versions at the same time. For example,  
`import-package --importFilePath`  
`/Users/test_user/tmp/test.xml`  
`--applicationVersion`  
"Clustered DotShoppingCart App:1.1.0, Apache Hadoop App:2.0"  
`--conflictResolutionAction CHECK`  
If the application version contains spaces, you must put the application version in double quotation marks. |
| `import-package --importFilePath`  
`ImportFilePath`  
`--conflictResolutionAction SKIP` | Compares the package and publicly available repository elements that are in the group you are logged in to, reuses the elements that match from the repository elements, and imports the package. The import process adds versions to elements that the logged in user's current group owns. If the current group does not own the element, the import process is cancelled. The process also matches the group membership information from the package. If such a match is not found, then the import process matches elements from the current group of logged in user. The `SKIP` option compares only the application, service, external service, policy, and task name and version. This option does not compare properties or action scripts in an application. For example, if you change a property in an existing application and use this option to import the package, the revised property is not imported. In this case, you must use the `OVERWRITE` or `IMPORTASNEW` option. If a package element does not match the repository element, then the `SKIP` option creates them in the repository. The `SKIP` option does not work if an application, service, external service, policy, or task name in the repository differs only by an upper or lower case letter. |
<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>import-package</strong> <strong>--importFilePath</strong> <strong>ImportFilePath</strong> <strong>--conflictResolutionAction</strong> <strong>OVERWRITE</strong></td>
<td>For example, if the repository already contains an Apache 2.0.0 service, you cannot import a package called APACHE 2.0.0, apache 2.0.0, or ApAche 2.0.0. In this case, change the name of the import package or use the IMPORTASNEW option. You can add the <strong>--shared</strong> option to make the new elements in the package publicly available. If the <strong>--shared</strong> option is not added, by default the new elements in the package are available to only the logged in user's current group. Custom sharing is not available for the import command.</td>
</tr>
</tbody>
</table>

Comparing the package and repository elements, overwrites the repository elements that match with the package elements, and imports the package. The process does not overwrite the group information. For example, you cannot change publicly available repository items to private and the reverse. The import process adds versions to elements that the logged in user's current group owns. If the current group does not own the element, the import process is cancelled. The process also matches the group membership information from the package. If such a match is not found, then the import process matches elements from the current group of logged in user. If a package element does not match the repository element, then the **OVERWRITE** option creates an element. When the **OVERWRITE** option overwrites an existing logical template in the repository, all of the cloud templates that were assigned to the logical template are removed. After the **OVERWRITE** operation successfully finishes, you must remap cloud templates to this logical template. The **OVERWRITE** option does not work if an application, service, external service, policy, or task name in the repository differs only by an upper or lower case letter. For example, if the repository contains an Apache 2.0.0 service, you cannot import a package called APACHE 2.0.0, apache 2.0.0, or ApAche 2.0.0. In this case, change the name of the import package or use the IMPORTASNEW option. You can add the **--shared** option to make the new elements in the package publicly available. If you do not add the **--shared** option, by default the new elements in the package are available to only the logged in user's current group. Custom sharing is not available for the import command. |
Table 17-2. CLI Import Commands (Continued)

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>import-package --importFilePath</td>
<td>Imports the package and repository elements as new elements. The --suffix option renames the element with the specified suffix. It must be used with the IMPORTASNEW option. If the NewSuffix contains spaces, you must enclose the suffix in double quotation marks. When you import an application version using the IMPORTASNEW option, an application and associated application version is created based on the specified suffix. This option does not create an application version in an existing application. As part of the import process, it also creates a logical template version that you must map to a cloud provider to deploy the newly imported application. For example, to import the Clustered DotShoppingCart application as a new application called test Windows app, use the command <code>import-package --importFilePath ./ClusteredDotShoppingCart 1.0.0.xml --conflictResolutionAction IMPORTASNEW --suffix &quot;test Windows app&quot;</code>. If you attempt to import a package element twice with the same suffix, the operation fails. You can add the --shared option to make the new elements in the package publicly available. If you do not add the --shared option, by default the new elements in the package are available to only the logged in user's current group. Custom sharing is not available for the import command.</td>
</tr>
</tbody>
</table>

Complete the tasks provided in the description section, before using this command.

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>import-package --importFilePath</td>
<td>IMPORTANT Before you use this command, use the import with CHECK option to review the existing logical templates in the repository. Then use the import with SKIP option to import the package. From the vCloud Application Director user interface, map the logical template to a cloud template. The SKIP option is valid only with this command. When the import command is set to SKIP, all the deployment profiles associated with the exported application versions are imported to the repository. Imports a deployment profile. The DeploymentEnvironmentSpecification includes CloudProviderName: DeploymentEnvironmentName For example, you can use the --withDeploymentProfilesUsingDeploymentEnvironment command to import a deployment profile. When a deployment profile is imported, existing cloud network mappings are removed. In the Deployment Profile wizard, click Map Details to map to a cloud network and verify cloud template mapping before you can use the imported deployment profile to deploy an application blueprint.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CLI Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImportFilePath</td>
<td>IMPORTANT Before you use this command, use the import with CHECK option to review the existing logical templates in the repository. Then use the import with SKIP option to import the package. From the vCloud Application Director user interface, map the logical template to a cloud template. The SKIP option is valid only with this command. When the import command is set to SKIP, all the deployment profiles associated with the exported application versions are imported to the repository. Imports a deployment profile. The DeploymentEnvironmentSpecification includes CloudProviderName: DeploymentEnvironmentName For example, you can use the --withDeploymentProfilesUsingDeploymentEnvironment command to import a deployment profile. When a deployment profile is imported, existing cloud network mappings are removed. In the Deployment Profile wizard, click Map Details to map to a cloud network and verify cloud template mapping before you can use the imported deployment profile to deploy an application blueprint.</td>
</tr>
</tbody>
</table>
Symbols
$(random) 135

A
access key 82
actions
  component 91
  life cycle stages 91
activate LDAP configuration 38
Active Directory domain 74, 187
add, custom task 129, 153
add Windows template 75
advanced application blueprint 138
agent bootstrap service requirements for custom templates 48, 70, 86
Amazon EC2
  application provisioning 81
custom AMI 86
definition profile 148
network 149
Amazon EC2 AMI 87
Amazon EC2 templates 88
AMI 87
Application Director
  deploying the Application Director OVF 22
  deploying the Application Director vApp 24
disk space requirements 20
memory requirements 20
ports opened 20
See also Application Director
application blueprint 135
application provisioning
Amazon EC2 81
vCloud Automation Center 47
vCloud Director 69
Application Director shuts down 28
applications
  configure 137
  creating 133
deleting virtual machines from the cloud 197
deploying 147, 155
deployment details 193
definition profile 147, 154, 156
publish 157
quick deploy 156
share 134
version 134
APT Repository Config 188
architectural principles 10
Auto-Bind consume 98, 107, 115
Auto-Bind expose 98, 107, 115
Auto-Bind tag 98, 107, 115
Auto-Bind type 98, 107, 115
Availability Zone 88, 89
B
basic application blueprint 135
bind property 137
binding properties 97
blueprint, vCloud Automation Center 49
blueprint for an application 135, 138
C
cancel
deployment 199
update process 199
catalog
  Application Director 12, 103, 121
  Puppet service 112
catalog external service properties 115
catalog policy properties 124
catalog service properties 107
certificate
  import 39
  LDAP 39
  local 39
  SSL 39
check list to set up vCloud Application Director 17
CLI (command-line interface)
  creating users with 37
  creating groups with 37
deploy 208
deployment status 208
export 213
general commands 203
import 213, 216
managing cloud tunnels 207
managing LDAP configurations with 206
managing users with 204
quick deploy 208
register 30
tear down deployment 212
update process 208
clone deployment 49, 54
cloud providers, register 88
cloud abstraction layer (CAL) 9, 11
cloud templates 121
cloud tunnel 81
cluster node 97
command-line interface 203
compliance, policy definition 200
compliance summary 193
compliance summary page 200
configure
Amazon EC2 82
component properties 93
DHCP 74
Linux template 52, 75
vCloud Application Director 84
Windows template 50, 72
configuring
storage options 21
network options 21
vCloud Director 23
connect Amazon EC2 84
connection problems 27
copy
application version 142
application 143
create
advanced external service 113
basic external service 113
external service version 113
action script 109
cloud tunnel 84
dependency 139
LDAP configuration 38
Linux template 52, 75, 87
policy definition script 126
provider specification 117
service version 106
user groups 203
Windows vApp 72
Windows template 50
creating groups with the CLI 37
creating users with the CLI 37
custom task
copy 130
delete 130
edit 130
customization specification 50, 54
D
Default group 41
define
external service properties 115
multiple NICs 140
policy properties 124
property values 97
service properties 107
service 107
delete
application deployments from Application Director 199
application version 143
applications from the cloud 197
external service 120
policy 128
service 111
deploy
Clustered DotShoppingCart 180
Clustered Dukes Bank App 181
jPetStore 182
Nanotrader 179
Radiant CMS 183
deploy sample application 179–183
deployment, update profile 171
deployment profile 147
deployment summary 193
deployment environment 59, 79, 89
deployment process 159
deployment status 160, 193, 195
deployment of applications 147
Deployment Profile wizard 155
deployment summary page 160, 193, 195
deployments 193, 199
details, virtual machine 195
develop components, catalog 91
developing components, best practice 101
Disk Layout Info 101
download, libpython package 178
download JRE 72
download agent bootstrap 50, 52, 72, 75
E
Elastic IP address 82
enable
Join Domain 74
SID 74
Endpoint VM 81, 89
environment variables 93
execution plan 155
existing deployment profile 154
existing update profile 171
export
  application 213
  custom tasks 213
external service version 213
policy version 213
service version 213
templates 78
external service
  add 113, 142
  blueprint 142
  copy 119
  delete 120
  edit 119
external service instance 59, 61, 89
Extra Configuration 149
F
failed task 193
firewalls, accessing URLs outside the firewall 28
flexible disk layout 149
flexible disks 140
G
global_conf 152
groups 33
guest customization 77
H
host name 135, 152, 159
I
import
  CHECK option 216
  IMPORTASNEW option 216
  Puppet service 112
  shared option 216
  SKIP option 216
import deployment profile 216
install, libpython package 178
install JRE 72
install agent bootstrap 50, 52, 72, 75
installation prerequisites 19
installing vCloud Application Director 19
instance, Puppet Master 64
J
Join Domain 187
JRE requirements for custom templates 48, 70, 86
L
linked clone deployment 49, 55, 56
locked account 27
logging in 41
logical templates
  copy 123
  delete 124
  edit 123
  virtual machine 121
login attempts 27
M
manage, user groups 203
manage deployments 193
manage LDAP configuration 206
manage users with the CLI 204
managing cloud tunnels 207
map, logical template 149
modify configuration 169, 185
multicloud 11
multiple hosts in a cluster 21
N
network bandwidth 84
O
OOB (out-of-the-box), services 11, 105
operating systems, for custom templates 48, 70, 86
operating systems in the vCloud Application Director catalog 104
out-of-the-box users, groups, and roles 36
P
passing property values 99
policy
  add 124
  copy 127
  delete 128
  edit 127
policy definition, variables 126
policy instance 59, 61, 89
policy scan timeline 200
predefined
  IP Address property 100
  node array property 100
predefined service 105
predefined users, groups, and roles 36
predefined properties 100
predefined users and roles 33
prepare, Puppet master 63
profile, application deployment 147, 154
promote update profile 172
promoted update profile
accept 173
discard 173
properties
array 93
computed 93
content 93
custom 149
internal 151
required 101
reserved 151
secured component 101
string 93
provider specification, edit 119
provisioning task
agent bootstrap 153, 159, 195
host 153, 159, 195
network bootstrap 153, 159
proxy configuration 28
publish deployment profile 157

R
reboot
agent bootstrap 109, 117
virtual machine 109, 117
register
automated 62
manual 62
Puppet master 62, 63
solution instance 62
request, service catalog 158
REST API 203
restart Application Director 28
RHN Registration 190
roles 33
rollback deployments 163, 174
run ID for a deployment 193

S
sample
applications 177, 178, 183
catalog services 191
CentOS template 177
services 177
tasks 177, 186–190
template 177
Ubuntu template 177
virtual machine templates 177
scale in deployments 166
scale out deployments 164
script
Bash 92
BeanShell 92
PowerShell 92
Windows CMD 92
search criteria 42
security settings 12
service
copy 111
delete 111
edit 111
services, adding 105
set
application component property 152
host name 152
node property 152
service property 152
set host name 135
set up, provisioning environment 47
shut down appliance 25
single host in a cluster 21
single node 97
single-tier application 178
snapshot 56
solution instance 64
special update property 164, 166, 169
specify clustered node 139
SSH connection, Ubuntu VM 178
start, policy scan 197
start appliance 25
start CLI program, remote 36
supply the correct password 27
supported action script 92
supported custom task script 92
system requirements for Application Director 20

T
tags in the vCloud Application Director
catalog 104
task status 160, 193, 195
task status window 160, 193
task timeline 160
tear down an application 197
templates
OVF format 22, 78
virtual machine 11, 77, 88
temporary VM, external service 149, 153
terminology 12
three-tier application 178
troubleshoot, update process 175
troubleshooting 27

U
understand
deployment process 159
update process 159
unlock your darwin_user account 27
update
  application deployment 163
delete code 169, 185
delete configuration 169, 185
Clustered DotShoppingCart 184, 185
Clustered Dukes Bank 184, 185
Nanotrader 184, 185
policy 197
policy instance 197
rollback 174
sample applications 183
scale in 166, 184
scale out 164, 184
update profile, deployment 171, 172
update deployment 160, 193, 195
update process 159, 193
update existing VM template 57, 77
update Linux VM template 57, 77
update Windows VM template 57, 77
updates summary 193
upgrading vCloud Application Director 31
user accounts 33
user interface
  buttons 43
  icons 44
users, groups and roles 36

V
vCloud Director
  application provisioning 69
cloud providers 78
custom template 70
network 149
vCloud Application Director
  applications 134, 138
catalog 103
custom task 129
deployment profiles 147
external service 113
logical templates 121
overview of 9
policy 124
services 105
setup 17
terminology 12
Web interface 41
vCloud Automation Center
  application provisioning 47
  blueprint 47
  business group 57
  cloud providers 57
custom template 48
deployment profile 148
entitlement 157
guest agent 50, 52
 gutent 50, 52
network 149
provisioning group 57
reservation 59
reservation policy 59
SSO 157
template 57
vCloud Director catalog 75
vCloud Director templates 78
verify vCloud Director setup 23
violation, policy definition 200
virtual machine
  Endpoint 84
  Linux 82
  NAT 82
virtual machine template
  Amazon EC2 87
  vCenter Server 49
virtual datacenters (vDCs) 79
virtual private cloud (VPC) 88, 89
VM, Endpoint 83
VMware Solution Exchange 144
VMware Tools 72, 75
VMware Tools requirements for custom templates 70

W
Web interface for vCloud Application Director 41
Web user interface 42
Windows, virtual machine template 71, 74, 75
withdraw update profile 172

Y
YUM Repository Config 189