

Using Application Services

vRealize Automation 6.2



vmware®

You can find the most up-to-date technical documentation on the VMware website at:

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

If you have comments about this documentation, submit your feedback to

docfeedback@vmware.com

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

Copyright © 2012–2015 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

Using Application Services	9
Updated Information	10
1 Introducing Application Services	11
Application Services Overview	11
Core Architectural Principles	13
Deploying Any Application and Middleware Service	14
Multicloud Support	14
Application Services Extensibility and Open Architecture	14
Standardization in Application Services	15
User Security	15
Key Concepts	15
2 Install and Configure Application Services	20
3 Installing Application Services	22
Preparing to Install Application Services	22
Application Services System Requirements	23
Set Up vCenter Server Cluster Configurations	24
Set Up vCloud Director for Application Services	26
Start the Application Services Appliance	28
Configure the Application Services Appliance	29
Register the Appliance and the vRealize Automation Server	30
Import Predefined Sample Content	31
Open the Application Services Web Interface	31
Troubleshooting Problems Connecting to the Application Services Web Interface	32
Unlock Your darwin_user Account	33
Restart Application Services	34
Configure Application Services to Use a Proxy for External URLs	34
Register Application Services to vRealize Automation	36
4 Upgrading Application Services	38
Upgrade Application Services	38
Roll Back an Application Services Upgrade	39
5 Setting Up Users and Groups	41

- 6 Using Tenants and Business Groups in Application Services 42**
- 7 Import Predefined Content to a Tenant 43**
- 8 Using the Application Services Web Interface 45**
 - Log In to Application Services 45
 - Using the Application Services Web Interface 46
 - Using the Application Services Buttons 47
 - Understanding the Application Services Icons 48
- 9 Setting Up Application Provisioning for the Application Services Environment 49**
 - Virtual Machine Requirements for Creating vRealize Automation Custom Templates 50
 - Creating Virtual Machine Templates in vRealize Automation 51
 - Prepare vCenter Server Windows Virtual Machine Templates 52
 - Prepare vCenter Server Linux Virtual Machine Templates 55
 - Create vRealize Automation Blueprints for Clone Deployment 56
 - Create and Configure vRealize Automation Blueprint for Linked Clone Deployment 57
 - Updating Existing Virtual Machine Templates in vRealize Automation 59
 - Register the vRealize Automation Cloud Provider and Template 60
 - Create a vRealize Automation Deployment Environment 62
 - Map an External Service Instance 64
 - Create a Policy Instance 65
 - Register a Puppet Solution Instance 66
- 10 Setting Up Application Provisioning for the vCloud Director Environment 75**
 - Virtual Machine Requirements for Creating vCloud Director Custom Templates 75
 - Creating Windows Virtual Machine Templates in vCloud Director 77
 - Create a vCloud Director vApp 78
 - Configure Windows Virtual Machine Template in vCloud Director 78
 - Enable SID Change and Domain Join for Windows Virtual Machine Templates 80
 - Add Windows Virtual Machine Templates to the vCloud Director Catalog 81
 - Create Linux Virtual Machine Templates in vCloud Director 82
 - Verify Cloud Template Configuration from the vCloud Director Catalog 84
 - Updating Existing Virtual Machine Templates in vCloud Director 84
 - Exporting Virtual Machine Templates with OVF Format 85
 - Register the vCloud Director Cloud Provider and Template 85
 - Create a vCloud Director Deployment Environment 87
- 11 Setting Up Application Provisioning for the Amazon EC2 Environment 89**
 - Configure Amazon EC2 Environment for Application Services 90
 - Create an Endpoint VM 92
 - Create a Cloud Tunnel to Connect to Amazon EC2 93

- Virtual Machine Requirements for Creating Amazon EC2 Custom Templates 94
- Create Amazon EC2 Virtual Machine Templates or AMIs 95
- Register the Amazon EC2 Cloud Provider and Template 97
- Create an Amazon EC2 Deployment Environment 99

12 Developing Application Services Components 101

- Defining Component Actions 101
 - Supported Action and Custom Task Scripts 102
- Configuring Component Properties 103
 - Types of Properties 104
 - Defining Property Values 108
 - Binding to Other Properties 109
 - Auto-Binding to Other Properties 110
 - Passing Property Values Between Life Cycle Stages and Components 111
- Application Services Predefined Properties 112
 - Predefined IP Address Property 112
 - Predefined Node Array Index Property 113
 - Predefined Disk Layout Info Property 113
- Secured Component Properties 114
- Required Component Properties 114
- Best Practices for Developing Components 114

13 Managing the Application Services Library 116

- Add Operating Systems to the Library 116
- Add Tags to the Library 118
- Add a Service to the Library 118
 - Create a Service Version in the Library 119
 - Define Service Version Properties 121
 - Add Action Scripts to the Service Version 124
 - Maintain Service Versions 125
 - Delete a Service from the Library 126
- Import a Puppet Service to the Library 126
- Add an External Service to the Library 128
 - Create an External Service in the Library 128
 - Define External Service Properties 130
 - Create Provider Specification for an Advanced External Service Version 133
 - Maintain External Service Versions 135
 - Delete an External Service 136
- Add a Logical Template to the Library 137
 - Maintain Logical Template Versions 140
 - Delete a Logical Template from the Library 141

- Add a Policy to the Library 141
 - Create a Policy Definition Script 143
 - Maintain Policy Versions 145
 - Delete a Policy 146
- Add a Custom Task to the Library 146
 - Maintain Custom Task Versions 148

14 Creating Applications 150

- Create an Application 151
- Create an Application Version 151
- Model an Application Blueprint 152
- Configure an Application Blueprint 154
- Creating an Advanced Blueprint 155
 - Create a Dependency Between Components 156
 - Specify a Node as a Cluster 157
 - Define Multiple NICs for a Node 158
 - Add and Manage a Flexible Disk Layout 159
 - Add an External Service to an Advanced Blueprint 160
- Copy an Application Version 161
- Copy an Application 161
- Delete an Application Version 162

15 Working with Artifacts 163

- Create an Artifact Repository Specification 164
- Create an Artifact Repository Instance 166
- Create an Artifact Specification 167
- Map an Artifact Specification to an Artifact Repository Instance 168
- Bind an Artifact to an Application Blueprint 169
- Deploy an Application with Artifacts 169

16 Deploying Applications 171

- Setting Up and Configuring a Deployment Profile 171
 - Create a Deployment Profile 172
 - Configure the Deployment Environment 173
 - Configure Application Properties 177
 - Review the Execution Plan and Add Custom Tasks 178
 - Use an Existing Deployment Profile 180
 - Create a Composite Deployment Profile 181
- Deploying with Deployment Profiles 184
 - Deploy with a Single Deployment Profile 184
 - Deploy with a Composite Deployment Profile 185
 - Quick Deploy an Application 186

- Publishing Deployment Profiles to the vCloud Automation Center Service Catalog 187
 - Publish a Single Deployment Profile 187
 - Publish a Composite Deployment Profile 189
- Request a vRealize Automation Service Catalog Item 190
- Using the Deployment Summary Page 191
- Using the Composite Deployment Summary Page 192
- Understanding the Deployment and Update Process 193
- Understanding Deployment Failures 194
 - Resolve Deployment Failure 195

- 17 Updating Application Deployments 196**
 - Initiate an Update Process to Scale Out Deployments 197
 - Initiate an Update Process to Scale In Deployments 200
 - Initiate an Update Process to Modify Configurations 203
 - Use an Existing Update Profile 206
 - Promote an Update Profile 206
 - Accept or Discard a Promoted Update Profile 208
 - Rollback an Update Process 208
 - Understanding Run Custom Task Update 210
 - Troubleshoot Failed Update Process to Scale Deployments 211
 - Troubleshoot Failed Update Process to Modify Configuration 211

- 18 Deploying Predefined Library Components 213**
 - Using the Sample Templates 213
 - Add libpython Package to the Ubuntu Template 214
 - Enable an SSH Connection for Ubuntu Virtual Machines 214
 - Available Sample Applications for Deployment 215
 - Deploy Nanotrader Application 215
 - Deploy Clustered DotShoppingCart Application 216
 - Deploy Clustered Dukes Bank Application 217
 - Deploy jPetStore Application 219
 - Deploy Radiant CMS Application 220
 - Update Deployed Sample Applications 221
 - Initiate an Update Process to Scale Sample Applications 221
 - Initiate an Update Process to Modify Configurations in Sample Applications 222
 - Sample Application Services Library Tasks 224
 - Add Join Domain Predefined Task 224
 - Add APT Repository Config Predefined Task 226
 - Add a YUM Repository Config Predefined Task 227
 - Add an RHN Registration Predefined Task 229
 - Sample Library Services 230

- 19 Managing Deployments 231**
 - [View Deployment Task and Blueprint Details for an Application 232](#)
 - [View Deployed VM Details and Execution Plan of an Application 234](#)
 - [Start a Policy Scan 236](#)
 - [Tear Down an Application from the Cloud 236](#)
 - [Scale In Deployments from vCloud Automation Center 238](#)
 - [Scale Out Deployments from vCloud Automation Center 239](#)
 - [Tear Down an Application from vCloud Automation Center 240](#)
 - [Delete an Application Deployment from Application Services 241](#)
 - [Cancel a Deployment or an Update Process 242](#)
 - [View Policy Compliance Summary 243](#)

- 20 Using the Application Services CLI 245**
 - [Start the CLI Remotely 245](#)
 - [General CLI Options 247](#)
 - [Managing Cloud Tunnels 247](#)
 - [Deploying and Updating an Application Using CLI 249](#)
 - [Using CLI to Tear Down a Deployment 252](#)

- 21 Using the CLI Import and Export Functions 253**
 - [Using the CLI Export Command 253](#)
 - [Using the CLI Import Command 256](#)

Using Application Services

vRealize Automation Application Services, formerly 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. The Application Services feature 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 use Application Services to create, deploy, manage, and update applications across virtual and cloud-based infrastructures.

Intended Audience

This information is intended for anyone who wants to use Application Services for application deployments. This audience includes application architects and application deployers who work in collaboration with application infrastructure administrators and cloud administrators.

Updated Information

This *Using Application Services* guide is updated with each release of the product or when necessary.

This table provides the update history of the *Using Application Services* guide.

Revision	Description
001654-02	<ul style="list-style-type: none">Removed version values and added a reference to the Support Matrix in vRealize Automation documentation at https://www.vmware.com/support/pubs/vcac-pubs.html to Virtual Machine Requirements for Creating vRealize Automation Custom Templates, Prepare vCenter Server Linux Virtual Machine Templates, Application Services System Requirements, and Set Up vCenter Server Cluster Configurations. Updated Upgrade Application Services .
001654-01	<ul style="list-style-type: none">Updated Create a Snapshot from a vRealize Automation Blueprint and Create and Configure vRealize Automation Blueprint for Linked Clone Deployment to better describe the sample procedure.Updated Creating Virtual Machine Templates in vRealize Automation.Updated Upgrade Application Services to state that vRealize Automation must be restarted after upgrade is finished.Updated the Register the Amazon EC2 Cloud Provider and Template topic to better list the support Amazon Region code values.Update the View Deployed VM Details and Execution Plan of an Application topic to include information for VMware Remote Console.
001654-00	Initial 6.2 release.

Introducing Application Services

1

VMware vRealize Automation Application Services, formerly 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.

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, vRealize Automation, and across public clouds such as VMware vCloud Air and Amazon EC2.

This chapter includes the following topics:

- [Application Services Overview](#)
- [Core Architectural Principles](#)
- [Key Concepts](#)

Application Services Overview

VMware vRealize Automation Application Services has a model-driven, open, and extensible architecture. With its library of standard components, or services, Application Services 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.

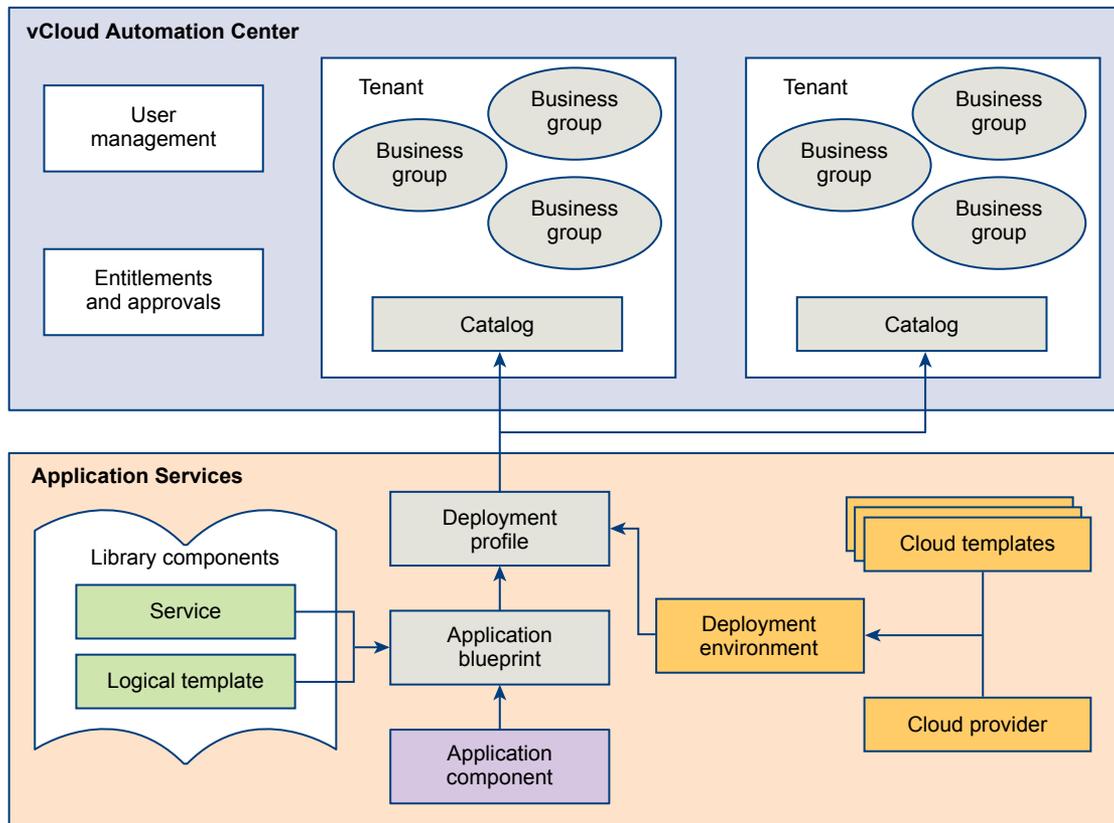
Application Services uses vRealize Automation for user and group support, access control, and catalog management. A vRealize Automation instance can have multiple tenants, a tenant can contain multiple business groups, and a business group can contain multiple users and objects such as applications and services. A user must be a member of a business group to add or edit objects in that business group. To view a private object in a business group, a user must belong to that group. To view a shared object, a user can belong to any business group in the same tenant. Applications deployed in Application Services become catalog items in vRealize Automation where users can request them for provisioning.

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 Application Services library. The cloud provider offers a cloud instance for deployment. The deployment environment provides a particular environment for the deployment in the cloud provider instance. Both the cloud provider and deployment environment map components from the cloud environment to Application Services.

Application architects can use the drag-and-drop interface to create visual application blueprints. Application architects can use the prepopulated and extensible library 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. For example, 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. Application Services and vRealize Automation Workflow

Core Architectural Principles

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

- [Deploying Any Application and Middleware Service](#)

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

- [Multicloud Support](#)

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

- [Application Services Extensibility and Open Architecture](#)

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

- [Standardization in Application Services](#)

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

- [User Security](#)

Users log in to Application Services with SSO credentials for the registered tenant and with at least one assigned application user role.

Deploying Any Application and Middleware Service

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

With Application Services, you can deploy applications on Windows and Linux operating systems in the vCloud Director and vRealize Automation 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

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

- Application Services 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 vRealize Automation provisioning infrastructure to access the vSphere private and public clouds.
- Application Services uses a CAL to plug in cloud providers. It automates deployments to the vCloud Director, the vSphere through vRealize Automation, and the Amazon EC2 cloud environment. vCloud Director uses open standards like the vCloud API and the Open Virtualization Format (OVF). Application Services 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.

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

Application Services Extensibility and Open Architecture

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

- The Application Services library 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 libraries. The Application Services library 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 Application Services

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

Application Services includes the following standardized configuration properties:

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

User Security

Users log in to Application Services with SSO credentials for the registered tenant and with at least one assigned application user role.

Application Services supports an authentication mechanism through vRealize Automation. All of the users in the identity stores (IDP) configured in vRealize Automation for the tenant registered with Application Services can log in with SSO credentials, if they have at least one application user role assigned. For information on how to assign user roles, see vRealize Automation *Tenant Administration*.

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 library 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 1-1. Cloud Template Mapping to Supported Cloud Environments

Application Services Environment	Cloud Environment
Cloud template	Application Services template
	vRealize Automation blueprint
	Amazon Machine Image (AMI)

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

Application Services Environment	Cloud Environment
Cloud provider	vCloud Director organization
	vRealize Automation 6.1 business group
	Amazon Region

cloud provider type

Type of cloud infrastructure on which deployments can be made. Application Services supports only vCloud Director, vRealize Automation, 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

Application Services Environment	Cloud Environment
Deployment environment	vCloud Director organization vDC
	vRealize Automation reservation policy
	Amazon VPC

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 vRealize Automation, a deployment environment maps to a reservation policy. If a deployment environment is not selected, vRealize Automation assigns a reservation policy depending on the resource requirements of the virtual machine in the vRealize Automation 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 library components.

library	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 Application Services. A logical template can be mapped to an actual cloud template in the cloud library 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 library and add it to an application deployment. Application Services also provides predefined tasks in the library 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 <code>installation_path</code> 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.
storage	Space provided by cloud provider to place the disk into different storages to meet the performance requirements.
deployment profile	Collection of deployment settings for a blueprint, including cluster size, CPU, memory, cloud templates, and networks.
logical network	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.
cloud network	A network defined within the cloud deployment environment.

execution plan

Task plan for viewing the order in which virtual machines are created and action scripts for library 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.

teardown

Removal of deployed virtual machines from the cloud. For example, you can tear down vCloud Director vApp and associated virtual machines, vRealize Automation 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.

Install and Configure Application Services

2

Application Services 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 Application Services, configure, and deploy a predefined sample application to the vCloud Director, vSphere through vRealize Automation, or Amazon EC2 environment.

Prerequisites

Familiarize yourself with the Application Services provisioning workflow described in [Application Services Overview](#).

Procedure

- 1 Install and set up the Application Services virtual appliance.

See [Preparing to Install Application Services](#) and [Start the Application Services Appliance](#).

- 2 Configure Application Services to use a proxy.

Perform this task when an application needs to download files from outside the corporate firewall.

See [Configure Application Services to Use a Proxy for External URLs](#).

- 3 Log in to the Application Services Web interface and familiarize yourself with the product features.

See [Log In to Application Services](#) and [Using the Application Services Web Interface](#).

- 4 Register a cloud provider.

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

Cloud Provider Types	Reference Topic
vCloud Director instance and organization	Register the vCloud Director Cloud Provider and Template
vRealize Automation business group	Register the vRealize Automation Cloud Provider and Template
Amazon EC2 Region	Register the Amazon EC2 Cloud Provider and Template

- 5 Create a deployment environment in the registered cloud provider.

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

Deployment Environment	Reference Topic
Organization vDC within vCloud Director	Create a vCloud Director Deployment Environment
Reservation policy within vCloud Automation Center	Create a vRealize Automation Deployment Environment
Amazon VPC and associated Availability Zone	Create an Amazon EC2 Deployment Environment

- 6 Map the cloud template to a logical template.

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

See [Add a Logical Template to the Library](#).

- 7 Deploy a predefined sample application from the **Deployment Profile** wizard.

See [Chapter 18 Deploying Predefined Library Components](#) and [Create a Deployment Profile](#).

- 8 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](#).

You can use the user interface to check the status of an application deployment in real time. See [Using the Deployment Summary Page](#).

- 9 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](#). To resolve the problem, see the *Application Services Troubleshooting* document.

Installing Application Services

To install Application Services 6.1, you can create and deploy the virtual appliance in either vCloud Director or in vSphere vCenter Server.

This chapter includes the following topics:

- [Preparing to Install Application Services](#)
- [Start the Application Services Appliance](#)
- [Troubleshooting Problems Connecting to the Application Services Web Interface](#)
- [Unlock Your darwin_user Account](#)
- [Restart Application Services](#)
- [Configure Application Services to Use a Proxy for External URLs](#)
- [Register Application Services to vRealize Automation](#)

Preparing to Install Application Services

Before you begin to install the Application Services appliance, verify that your computing environment meets the hardware and software system requirements.

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

- [Application Services System Requirements](#)

The virtual appliance on which you run Application Services must meet certain hardware and software requirements. In addition, you must open certain ports for Application Services.
- [Set Up vCenter Server Cluster Configurations](#)

You must set up the storage and network of your vCenter Server cluster configurations to meet certain requirements so that you can install Application Services in vCloud Director or vSphere.
- [Set Up vCloud Director for Application Services](#)

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

Application Services System Requirements

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

vCloud Director Disk Space and Memory Requirements

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

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

Port Requirements

You must open certain ports for the Application Services virtual machine.

Port	Connection Type
TCP Port 8443	External port for the Application Services user interface connection through HTTPS.
TCP Port 8080	Optional port for the Application Services user interface connection through HTTP.
TCP Port 80	External port for vami-lighttpd.
TCP Ports 5671	External port for vFabric RabbitMQ.
TCP Port 443	External port for the Application Services user interface to connect to a cloud environment.
TCP Port 22	Optional port for the external SSH connection.

For the most current port requirements for your release, see the Support Matrix in vRealize Automation documentation at <https://www.vmware.com/support/pubs/vcac-pubs.html>.

Web Interface Support

Application Services supports the following Web browsers. For current version requirements for your release, see the Support Matrix in vRealize Automation documentation at <https://www.vmware.com/support/pubs/vcac-pubs.html>.

- Internet Explorer
- Mozilla Firefox
- Chrome

Virtualization Software Requirements

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

- vSphere ESXi. See vSphere documentation at <https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html>.

- vCenter Server. See vSphere documentation at <https://www.vmware.com/support/pubs/vsphere-esxi-vcenter-server-pubs.html>.
- vCloud Director 5.5. See vCloud Director documentation at https://www.vmware.com/support/pubs/vcd_pubs.html.
- vRealize Automation 6.2. See vRealize Automation documentation at <https://www.vmware.com/support/pubs/vcac-pubs.html>.

For the most current system requirements for your release, see the Support Matrix in vRealize Automation documentation at <https://www.vmware.com/support/pubs/vcac-pubs.html>.

Supported Operating Systems for Virtual Machine Templates in the Application Services Library

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

- vCloud Director. See [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#).
- vRealize Automation. See [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#).
- Amazon EC2. See [Virtual Machine Requirements for Creating Amazon EC2 Custom Templates](#).

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

Set Up vCenter Server Cluster Configurations

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

Prerequisites

- For current version requirements for your release, see the Support Matrix in vRealize Automation documentation at <https://www.vmware.com/support/pubs/vcac-pubs.html>.
- Familiarize yourself with the procedures for creating vSphere resource pools and vCenter Server clusters that have DRS enabled. See the vSphere documentation center.
- Verify that you have one or more hosts running vSphere ESXi.
- Verify that you have a vCenter Server cluster with DRS enabled.
- Verify that you have the NTP client running on each of the vSphere ESXi hosts in your vCenter Server installation.

Synchronize the time on the host on which Application Services is deployed and on the hosts on which virtual machines are to be deployed. If the time is not synchronized, the Application Services 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.
- 2 Check the configuration settings for datastore and network requirements.

Option	Action
Multiple hosts in a cluster	<ul style="list-style-type: none"> ■ Verify that all of the hosts have at least one shared datastore, for example, shared LUN, NFS, and so on. ■ Verify that all of the hosts in the cluster have connectivity to at least one common network. ■ To support live virtual machine migration, verify that all of the hosts in the cluster are identical.
Network or SAN storage	<ul style="list-style-type: none"> ■ Verify that your host has shared storage. Although vCloud Director and Application Services can use local storage, shared storage ensures future scalability. ■ Verify that the host has at least one network.

What to do next

For vSphere, deploy the Application Services appliance. See [Deploy the Application Services Appliance in vSphere](#).

For vCloud Director, configure the vApp settings to install Application Services. See [Set Up vCloud Director for Application Services](#).

Deploy the Application Services Appliance in vSphere

To install Application Services 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 Application Services](#) for vSphere.
- A Application Services 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.

3 Configure the Cluster, Resource Pool, and Storage settings.

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 Application Services virtual machine deploys and powers on.

What to do next

Initialize the Application Services appliance. See [Start the Application Services Appliance](#).

Set Up vCloud Director for Application Services

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

Prerequisites

- Configure your vCenter Server settings for Application Services. See [Set Up vCenter Server Cluster Configurations](#).
- 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 Application Services performs application deployments, you must publish the catalog and the templates. Verify that 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 Application Services 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 Application Services.

What to do next

Verify your vCloud Director setup. See [Verify Your vCloud Director Environment](#).

To create or update custom virtual machine templates, see [Creating Windows Virtual Machine Templates in vCloud Director](#) and [Create Linux Virtual Machine Templates in vCloud Director](#).

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.

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

Note DHCP addressing for directly connected networks is not supported.

Prerequisites

- Configure your vCenter Server settings for Application Services. See [Set Up vCenter Server Cluster Configurations](#).
- 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 [Application Services System Requirements](#).
- Verify that the required TCP ports are open for the Application Services virtual machine. See [Application Services System Requirements](#).

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 Application Services cloud provider, verify that the virtual machines in the vCloud organization have a vCloud network configuration that allows them to connect to the Application Services appliance.
- 3 Deploy the vApp.
- 4 Use the VMRC plug-in to connect to the virtual machines and verify that they can ping an external IP.
- 5 Open TCP ports so that the deployed virtual machines can reach the Application Services appliance.

What to do next

Download and deploy the Application Services appliance. See [Deploy the Application Services Appliance in vCloud Director](#).

Deploy the Application Services Appliance in vCloud Director

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

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

Prerequisites

Verify that you completed the prerequisites listed in [Preparing to Install Application Services](#) for vCloud Director.

Procedure

- 1 Download the Application Services 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 *VersionNumber* with the build number of the appliance.
- 3 Log in to vCloud Director and select the organization vDC in which to deploy Application Services.
- 4 Select the Catalogs view and click the **vApp Templates** tab.

- 5 Click **Upload** ().

- 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**.

- 11 Right-click the newly added vApp and click **Start**.

What to do next

Initialize the Application Services appliance. See [Start the Application Services Appliance](#).

Start the Application Services Appliance

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

Procedure

- 1 [Configure the Application Services Appliance](#)

The first time you start the Application Services appliance, you must enter the Application Services serial number and create passwords for the root user and darwin_user.

2 Register the Appliance and the vRealize Automation Server

You register the Application Services appliance with the vRealize Automation server. Registration allows you to use the vRealize Automation users, business groups, and tenants in Application Services and to assign Application Services user roles.

3 Import Predefined Sample Content

After you register the Application Services appliance and the vRealize Automation server, you can optionally install predefined sample content in to a vRealize Automation tenant.

4 Open the Application Services Web Interface

After configuring the appliance and registering it with the vRealize Automation server, and optionally importing the predefined sample content, you can open the Web interface to explore and work.

Configure the Application Services Appliance

The first time you start the Application Services appliance, you must enter the Application Services serial number and create passwords for the root user and darwin_user.

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*.

As part of starting the Application Services appliance, you register the appliance with a vRealize Automation server and tenant and choose whether to import predefined sample content in to the tenant.

If you are upgrading to a new Application Services release, see [Chapter 4 Upgrading Application Services](#).

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 [Application Services System Requirements](#).
- Verify that the vSphere client meets the Web browser requirements. See VMware vSphere documentation.
- [Start the Application Services Appliance](#).

Procedure

- 1 Open the Application Services appliance.
 - a From the vCloud Director My Cloud view, click the image under the consoles column that corresponds to your Application Services 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.
 - b From the vSphere client, locate the powered-on virtual machine and click the **Console** tab.
- 2 At the prompt, enter the Application Services serial number and press Enter.
- 3 Enter a password for the root user account and press Enter.

- 4 At the prompt, re-enter the password to confirm it and press Enter.
- 5 Enter 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 enter the correct password.
- 6 At the prompt, re-enter the password to confirm it and press Enter.
- 7 If prompted, respond to the request to enable SSH login.

What to do next

[Register the Appliance and the vRealize Automation Server.](#)

Register the Appliance and the vRealize Automation Server

You register the Application Services appliance with the vRealize Automation server. Registration allows you to use the vRealize Automation users, business groups, and tenants in Application Services and to assign Application Services user roles.

Prerequisites

[Configure the Application Services Appliance.](#)

Procedure

- 1 Enter the vRealize Automation server URL at the prompt, and press Enter.
For example, enter a secure URL in the form `https://vcac.it.mycompany.com` or `https://192.0.2.115:443`.
- 2 Enter the user name of a vRealize Automation system administrator account, and press Enter.
- 3 Enter the password of the vRealize Automation system administrator account, and press Enter.
- 4 If registration with the vRealize Automation server fails, you are prompted to either skip registration or try again.
- 5 If registration succeeds, the system prompts you to import predefined sample content to Application Services. Enter **Y** for Yes or **N** for No and press Enter.

Enter	Description
Y	Yes, import the predefined sample content to a specified tenant and business group. After the import, all business groups in the tenant can share the content.
N	No, do not import the predefined sample content.

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.

What to do next

(Optional) [Import Predefined Sample Content](#).

Import Predefined Sample Content

After you register the Application Services appliance and the vRealize Automation server, you can optionally install predefined sample content in to a vRealize Automation tenant.

The system prompts you for all of the information required to log you in to the CLI before you can import the predefined sample content.

Prerequisites

- [Register the Appliance and the vRealize Automation Server](#).
- Verify the tenant user account has all Application Services roles assigned. For information on assigning roles to users, see vRealize Automation *Tenant Administration*.
- Verify that the tenant user belongs to the business group.

Procedure

- 1 Enter the tenant URL name, and press Enter.

A system administrator in vRealize Automation specifies the URL name when adding a tenant.

- 2 Enter the user name of an account with access to the tenant and press Enter.

The user account must also have all Application Services roles assigned.

- 3 Enter the password of the tenant user and press Enter.

- 4 Enter the name of a business group that includes the tenant user and press Enter.

If the business group name includes spaces, enclose the entire name in double quotes.

- 5 If the CLI login fails, the system prompts you to try again. Enter **Y** or **N** and press Enter.

Enter	Description
Y	Try to log in again. The system then prompts you for Step 1 through Step 4 .
N	Do not try to log in again. Do not import the predefined sample content.

- 6 If the CLI login succeeds, the system imports the predefined sample content and prompts you to press a key to continue. Press any key.

What to do next

[Open the Application Services Web Interface](#).

Open the Application Services Web Interface

After configuring the appliance and registering it with the vRealize Automation server, and optionally importing the predefined sample content, you can open the Web interface to explore and work.

Prerequisites

- [Register the Appliance and the vRealize Automation Server.](#)

Procedure

- 1 Enter a password for the Application Services admin user account and press Enter.
- 2 At the prompt, re-enter the password to confirm it and press Enter.

The bootup script starts the necessary services and displays the URL for accessing the Application Services server.

- 3 Navigate to the Application Services server with a supported browser.

The URL format is `https://Application_Services_IP_or_hostname:8443/darwin/org/tenantURL`, where *tenantURL* is the tenant URL name.

- 4 Log in as a vRealize Automation tenant administrator or tenant user who has one or more Application Services roles assigned. The user must belong to at least one business group in the tenant.

The browser opens the Web interface. See [Using the Application Services Web Interface](#).

Note If you want to shut down the Application Services appliance deployed in vCloud Director, do so from the vCloud Director user interface. Do not use vCenter Server to shut down the Application Services 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](#).

Log in to Application Services and familiarize yourself with the product features. See [Log In to Application Services](#) and [Using the Application Services Web Interface](#).

Register a cloud provider for your cloud environment. See [Register the vCloud Director Cloud Provider and Template](#), [Register the vRealize Automation Cloud Provider and Template](#), or [Register the Amazon EC2 Cloud Provider and Template](#).

Troubleshooting Problems Connecting to the Application Services Web Interface

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

Table 3-1. Common Connection Errors

Error	Possible Cause	Possible Solution
The Application Services virtual appliance does not have a working IP address or network.	The following problems might cause a networking error: <ul style="list-style-type: none"> 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 Application Services 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: <ul style="list-style-type: none"> 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: <pre>/home/darwin/tools/darwin_util.sh -a AUTO</pre>
The vFabric tc Server service encounters an error in the Application Services 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 <code>/home/darwin/tcserver/darwin/logs/catalina.out</code> 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 Application Services appliance. SSH is disabled for the root account.
- Verify that you have root privileges.

Procedure

- In the VMRC console, log in as the root user.
- Check the number of failed login attempts from the shell prompt.

```
faillog -u darwin_user
```
- Unlock the failed account.

```
faillog -u darwin_user -r
```

Restart Application Services

In some cases, you might have to restart Application Services.

Procedure

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

Configure Application Services to Use a Proxy for External URLs

Even if you use Application Services 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 Application Services 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.

Application Services 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 vRealize Automation Deployment Environment](#).

Prerequisites

- Log in to Application Services as an **application architect** and an **application catalog administrator**.
- Verify that you have access to the virtual machine where Application Services 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 Application Services Appliance](#).
- Familiarize yourself with the procedure for creating new services. See [Add a Service to the Library](#).

Procedure

- 1 For vCloud Director or vRealize Automation, configure the Application Services virtual appliance to use a proxy.
 - a Log in to the Application Services virtual appliance as **darwin_user**.
 - b Enter **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 library 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 Application Services as an application catalog administrator 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 library.
 - 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 Application Services as an application architect 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 Application Services and familiarize yourself with the product features. See [Log In to Application Services](#) and [Using the Application Services Web Interface](#).

Register a cloud provider for your cloud environment. See [Register the vCloud Director Cloud Provider and Template](#), [Register the vRealize Automation Cloud Provider and Template](#), or [Register the Amazon EC2 Cloud Provider and Template](#).

Register Application Services to vRealize Automation

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

Typically, registration is done as part of starting the Application Services appliance. Use this procedure if you need to reestablish the registration or if registration information has changed.

When the connection between Application Services and vRealize Automation is established, you can publish the deployment profile to the vRealize Automation Application Services catalog so that other users can request the deployment profile. You can also assign Application Services roles to users and user groups after registration.

Note You can have only one active registration. If you try to register more than once, you receive an error message.

Prerequisites

- Log in to Application Services as a **system administrator**.
- Verify that you know the password for the `darwin_user`. This password was set during installation. See [Start the Application Services Appliance](#).
- Start the Application Services CLI. See [Start the CLI Remotely](#).
- The vRealize Automation IaaS environment and endpoints must be properly set up and configured. See *Installation and Configuration* and *IaaS Configuration for Virtual Platforms*.
- You must have SSO admin credentials available.

Procedure

- 1 In the `roo` shell prompt, register Application Services to the vRealize Automation server.

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

The `componentRegistryUrl` is the URL for the vRealize Automation self-service catalog.

A sample URL for the `componentRegistryUrl` is `https://vcac-oci-071.eng.test.com`.

The registration might take approximately 30 seconds before the status message, `vCAC server registered successfully`, appears.

- 2 (Optional) View the vRealize Automation server information.

list-vcac-info

- 3 (Optional) If there are any changes in the vRealize Automation server information, register Application Services to the vRealize Automation server.

```
register-vcac-server --componentRegistryUrl VCACServerURL --ssoAdministratorUsername  
UserName --ssoAdministratorPassword Password --update
```

- 4 (Optional) Unregister the vRealize Automation server from Application Services to register the appliance to another vRealize Automation server.

unregister-vcac-server

Unregister the vRealize Automation server only if the server becomes unusable with Application Services. If you then register Application Services with a different vRealize Automation server, the existing cloud providers and deployment environments become unusable because the underlying templates are no longer valid. You also lose access to any existing Application Services content if the business groups in the new and old vRealize Automation servers do not match.

What to do next

After you register Application Services to vRealize Automation, register the cloud provider and template. See [Register the vRealize Automation Cloud Provider and Template](#).

4

Upgrading Application Services

You can upgrade your Application Services installation to a new release using a standard upgrade process.

You can upgrade from the previous Application Services release to the current release. To upgrade from previous versions, for example from Application Director 6.0 or 6.0.1 to Application Services 6.2, follow the upgrade path from 6.0 to 6.0.1 to 6.1 to 6.2 in that order. For more information, see previous release documentation.

Note You can migrate vCloud Application Director 6.0.1 to Application Services 6.1. You can then upgrade your Application Services installation to a next release. For details about migrating from Application Director 6.0.1 to Application Services 6.1, see the 6.1 version of *Using Application Services* documentation.

This chapter includes the following topics:

- [Upgrade Application Services](#)
- [Roll Back an Application Services Upgrade](#)

Upgrade Application Services

You can download and run the Application Services upgrade installation program to upgrade your Application Services installation.

Root permission is required to download and run the Application Services upgrade installation program.

Note You must upgrade to vRealize Automation 6.2.x before you upgrade Application Services.

During upgrade, install and uninstall operations stop the Application Services server.

Prerequisites

- Verify that the currently installed version of Application Services is 6.1.x.
- Verify that there are no operations in progress or running on the Application Services server.
- Verify that you are logged in on the Application Services server with root permission.
- Verify that you have upgraded to vRealize Automation 6.2.x.

Procedure

- 1 Navigate to the VMware download site.
- 2 Download the upgrade file named `ApplicationServices–version–build_Upgrade_Installer.tgz`.
- 3 Copy `ApplicationServices–version–build_Upgrade_Installer.tgz` to the machine where vCloud Automation Center Application Services 6.1.x is installed.
- 4 Create a new directory in which to untar the `ApplicationServices–version–build_Upgrade_Installer.tgz` file, for example `my_appservices_upgrade`.
- 5 Untar the `ApplicationServices–version–build_Upgrade_Installer.tgz` file to the new directory, for example `my_appservices_upgrade`.

```
tar xvfz ApplicationServices–version–build_Upgrade_Installer.tgz
```

- 6 Switch user credentials to root to have the correction permissions to run the upgrade installer.

```
su root
```

- 7 Run the upgrade install program from the directory in which the untarred upgrade installer resides.

```
./install.sh
```

- 8 Respond to any prompts.

Application Services is upgraded to the new release.

What to do next

When upgrade is finished, you must perform the following tasks:

- Restart the vRealize Automation server to complete the component registry refresh process.
- Clear the browser cache before using the Application Services Web user interface.

Roll Back an Application Services Upgrade

If you encounter a problem during or after upgrading your Application Services installation, you can roll back the installation to the version from which you started the upgrade procedure.

Prerequisites

Perform the upgrade procedure described in [Upgrade Application Services](#).

Procedure

- 1 Switch user credentials to root to have the correction permissions to run the uninstall upgrade program.

```
su root
```

- 2 Run the uninstall upgrade program from the directory in which the untarred upgrade installer resides.

```
uninstall.sh
```

- 3 Respond to any prompts.
- 4 When rollback is finished, clear your browser cache before using the Application Services Web user interface.

Application Services is rolled back to the release from which you initiated the upgrade process.

5

Setting Up Users and Groups

After you install Application Services, you must determine who can use what type of objects in Application Services and what tasks those users or groups of users are authorized to perform. You use vRealize Automation to create user groups and business groups and assign user roles, entitlements, and approvals in both Application Services and vRealize Automation.

You can selectively assign administrative permissions by assigning roles to specific users and user groups. You can limit access to specific deployment environments and cloud templates by assigning the objects and their users or groups to the same business group. User groups let you assign roles and entitlements to more than one user at a time. Business groups let you associate a set of services and resources in Application Services with a set of users.

User roles specify which functions a user can perform in Application Services. These functions include managing the library, managing the cloud providers and deployment environments, creating applications, and deploying applications.

Application Services has the following roles.

Role	Functions
Application Architect	Creates, modifies, and deletes applications.
Application Catalog Administrator	Defines services, templates, operating systems, tasks, and tags.
Application Cloud Administrator	Defines resources and deployment environments.
Application Publisher and Deployer	<ul style="list-style-type: none">Deploys applications into the vRealize Automation catalog.Creates, updates, and publishes services, library items, and actions.

You must log in to vRealize Automation as a tenant administrator to assign roles for Application Services. For more information on managing users, user groups, and business groups, see the vRealize Automation 6.1 documentation.

Using Tenants and Business Groups in Application Services

6

Application Services and vRealize Automation share tenants and business groups. A tenant is an organizational unit in a vRealize Automation deployment. A business group belongs to a tenant and associates a set of services and resources with a set of users.

An Application Services appliance is registered to a vRealize Automation deployment, also called an instance, and to a tenant in that instance. An Application Services system administrator can register the appliance to different tenants, but only one at a time.

A tenant has multiple business groups. Each business group within a tenant has members and objects such as applications, services, and external services. Users can be members of multiple business groups and multiple tenants. Every object is owned by a business group. Only users who belong to the business group can edit the objects in that group.

Applications, services, and other objects in Application Services are private or shared. Only users who are members of a particular business group can view or edit private objects in that group. All users in all business groups within a tenant can view shared objects.

As part of their function, user roles determine which objects users can create. For example, a user assigned the application architect role can create an application, among other objects. When creating the application, the user assigns the business group and specifies whether the application is private or shared. Thereafter, only members of the owning business group can edit the application. If the application is private, only members of the owning business group can view it. If the application is shared, all members of all business groups in the tenant can view the application. Regardless of whether the application is private or shared, only members of the owning business group can edit the application.

The different levels of sharing determine what you can do with the components within the blueprint, and with the application itself. As the logged-in user who constructs the application blueprint, you must belong to the owning business group of the application to edit the blueprint. Logical templates, services, external services, tasks, artifacts, and artifact repositories all belong to business groups and can be private or shared. If you select shared blueprint components that belong to a different business group, you can view those components but you cannot alter them. Only when you belong to the owning business group of a component can you change the component. 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.

Import Predefined Content to a Tenant



You can import predefined sample content from an Application Services appliance to a tenant in a registered vRealize Automation instance.

You use the `/home/darwin/tools/import_oob_content.sh` script to import predefined sample content from an Application Services appliance to a vRealize Automation instance. The script is located on the Application Services appliance. To import sample content in to multiple tenants, you must run the script for each tenant.

The log files are `/tmp/standalone_import.log` and `/tmp/standalone_import_all.log`.

To use non-English character input for the script, you must set the environment variable `LC_CTYPE` to `en_US.UTF-8`.

Prerequisites

- Verify the registration of the Application Services appliance with a vRealize Automation instance.
- Verify the tenant name and tenant user credentials for the vRealize Automation instance.
- Verify that the tenant user belongs to the targeted business group for the predefined import, and has all Application Services roles.
- Log in to the Application Services appliance with SSH.

Procedure

- 1 Set your current directory to `/home/darwin/tools`.

You must have write permission.

- 2 Run the `/import_oob_content.sh` script.

The script prompts you for your input.

- 3 Enter the tenant URL name, and press Enter.

A system administrator in vRealize Automation specifies the URL name when adding a tenant.

- 4 Enter the user name of an account with access to the tenant and press Enter.

The user account must also have all Application Services roles assigned.

- 5 Enter the password of the tenant user and press Enter.

- 6 Enter the name of a business group that includes the tenant user and press Enter.
If the business group name includes spaces, enclose the entire name in double quotes.
- 7 If the CLI login fails, the system prompts you to try again. Enter Y or N and press Enter.

Enter	Description
Y	Try to log in again. The system then prompts you for Step 3 through Step 6 .
N	Do not try to log in again. Do not import the predefined sample content.

- 8 If the CLI login succeeds, the system imports the predefined sample content.

What to do next

You can create an export package with the darwin CLI `export-package` command and import a specific application or service from that package to a tenant on another Application Services appliance with the complementary `import-package` command. For more information about the `export-package` and `import-package` commands, see *Using Application Services*.

Using the Application Services Web Interface



You can use the Application Services Web interface to register cloud providers, maintain the library of virtual machine templates and services, create applications, deploy applications, publish a deployment profile to the vRealize Automation service catalog, update deployed applications, and tear down deployed applications from the cloud.

Note You must use the CLI to register the vRealize Automation server 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](#).

This chapter includes the following topics:

- [Log In to Application Services](#)
- [Using the Application Services Web Interface](#)

Log In to Application Services

You log in to the Application Services Web interface with SSO credentials for the registered tenant and with an Application Services user role assigned.

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 business group associated with the user, and whether the objects are also associated with that business group. You can log in with any Application Services user role. See [Chapter 5 Setting Up Users and Groups](#).

Prerequisites

- Verify that the Application Services appliance has a registered vRealize Automation instance and tenant.
- Verify that your SSO credentials include access to the registered tenant and have at least one Application Services user role assigned.
- Verify that you have the URL for the Application Services Web interface. This Web URL appears in the console of the virtual machine that hosts Application Services when installation is finished. For the URL and instructions on starting Application Services, see [Start the Application Services Appliance](#).

- Verify that you have a supported version of a Web browser installed on your computer. See [Application Services System Requirements](#).

Procedure

- 1 Open a Web browser and enter the Application Services URL for a registered tenant.

You can access a registered tenant by changing the tenant ID at the end of the URL. If you do not specify a tenant ID, Application Services opens the default tenant `vsphere.local`.

For example, you can access the registered tenant `dev` with the URL `https://192.0.2.255:8443/darwin/org/dev`.

- 2 Enter 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 application architect role assigned to it, when you log in to the Application Services Web interface, the Applications 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 Application Services features. See [Using the Application Services Web Interface](#).

Using the Application Services Web Interface

With the 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, Application Services 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 Application Services appliance. All of the users can see the sample applications, services, tasks, logical templates, external services, operating systems, and tags included in the Application Services library.

The Application Services title bar includes a drop-down menu that lists the main categories Applications, Deployments, Clouds, and Library, and subcategories under Clouds and Library. From the Application Services title bar, you can select another category to navigate from one page to another.

The Applications page organizes all of the applications. From this page, you can create and edit 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 open a deployment summary page where 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 Library category lets you create services, external services, policies, logical templates, tasks, operating systems, tags, artifact repositories, and artifacts for your application. You can also use the sample services and other components in the library to model an application blueprint.

Using the Application Services Buttons

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

Table 8-1. Application Services Buttons

Icon	Name	Description
	Promote Update Profile	Available on an updated deployment. Indicates that the update profile is promoted to another deployment environment.
	Add Relation	Available in the blueprint. Creates relations between components in a blueprint.
	Convert to Node Array	Available in the blueprint. Creates a clustered node. You can also use the button to convert a clustered node to a single node.
	Add	Available in the blueprint. Adds a NIC to a node from the NICs tab or to add application component properties from the Properties tab.
	Reset	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.
	View Task Information	Available in execution plans. View action script text, properties, and logs.
	Expand	Available in execution plans. Displays all virtual machines in a cluster node.
	Collapse	Available in execution plans. Minimizes all virtual machines in a cluster node.
	Add Script Task	Available in execution plans. Adds custom tasks to an application deployment.
	Notification	Appears when a task is incomplete or a user performs a task that is not allowed.
	Expand	Maximizes a status window in the deployment summary page.
	Minimize	Collapses a status window in the deployment summary page.
	Expand	Maximizes the provisioning tasks in the execution plan status window of the deployment summary page.
	View Logs	Available on a deployment details page or an execution plan after deployment.
	Copy Application Version	Creates a new version of an existing application.

Table 8-1. Application Services Buttons (Continued)

Icon	Name	Description
	Quick Deploy	Available on the Applications page and application version card. Deploys an application without configuring any of the elements in the Deployment Profile wizard.
	Update Deployment	Initiates an update process to scale a clustered node or modify configuration of a deployed application.
	Teardown Deployment	Removes a deployed application from a cloud environment.

Understanding the Application Services Icons

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

Table 8-2. Application Services Icons

Icon	Name	Description
	Group name	Appears on the Applications, Services, External Services, Logical Templates, and Deployments pages next to the name of the group that owns the component.
	Shared Publicly	Appears on the Applications, Services, External Services, Logical Templates, and Deployments pages for components that are shared publicly within a given tenant.
	Shared Privately	Appears on the Applications, Services, External Services, Logical Templates, and Deployments pages for components that are private to a business group within the tenant.
	Policy Scan Success	Appears on the Deployments page for a deployment or deployed application with a successful policy scan.
	Policy Scan Fail	Appears on the Deployments page for a deployment or deployed application with a failed policy scan.
	Auto-Bind Consume	Appears in the blueprint canvas, next to properties that can Auto-Bind with another property, which is set to Expose.
	Auto-Bind Expose	Appears in the blueprint canvas, next to properties that can be used to customize another property, which is set to Consume.
	Extra Configuration	Appears in the Deployment Profile wizard and the VM details status window of the deployment summary page for a vRealize Automation deployment. In the Deployment Profile wizard, you can use Extra Configuration to override the vRealize Automation blueprint custom properties or add to the existing properties.

Setting Up Application Provisioning for the Application Services Environment

9

Application Services uses the virtual cloud infrastructure features in vRealize Automation to deploy applications to the vSphere environment.

As part of the setup process, you must create custom vRealize Automation blueprints. A vRealize Automation 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. vRealize Automation applies a set of custom properties defined in the build profile or in the blueprint to that requested machine. From the vRealize Automation interface, you can access the clone or linked clone blueprints and specify properties to override several properties in the vCenter Server template.

Important Application Services supports only the vRealize Automation virtual blueprint provisioning to vSphere.

In Application Services, you must register a cloud provider and template. When you register a cloud provider, you map a specific vRealize Automation blueprint in a business group to a cloud template in Application Services. This registration process makes this vRealize Automation blueprint available in the Application Services library. 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 vRealize Automation reservation policy.

Familiarize yourself with the key concepts that relate to setting up and configuring a cloud environment for application provisioning. See [Key Concepts](#).

This chapter includes the following topics:

- [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#)
- [Creating Virtual Machine Templates in vRealize Automation](#)
- [Updating Existing Virtual Machine Templates in vRealize Automation](#)
- [Register the vRealize Automation Cloud Provider and Template](#)
- [Create a vRealize Automation Deployment Environment](#)

Virtual Machine Requirements for Creating vRealize Automation Custom Templates

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

Table 9-1. Virtual Machine Requirements for Custom Templates

Requirement	Description
Operating system	<p>For current version requirements for your release, see the Support Matrix in vRealize Automation documentation at https://www.vmware.com/support/pubs/vcac-pubs.html.</p> <ul style="list-style-type: none"> ■ Red Hat Enterprise Linux ■ Windows Server ■ CentOS ■ SUSE Linux Enterprise ■ Ubuntu
JRE	<p>Install JRE 1.7.0.</p> <ul style="list-style-type: none"> ■ Windows. You can install JRE for Windows from the package http://Application_Director_IP/agent/jre-1.7.0_51-win64.zip located in the Application Services virtual appliance. See Prepare vCenter Server Windows Virtual Machine Templates. ■ Linux. You have the option to install JRE for Linux during template preparation. See Prepare vCenter Server Linux Virtual Machine Templates.
Windows agent bootstrap service	<p>Download the http://Application_Director_IP/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip file from the Application Services virtual appliance.</p> <p>To install the agent bootstrap service on a Windows-based virtual machine template, see Prepare vCenter Server Windows Virtual Machine Templates.</p>
vRealize Automation guest agent	<p>For Windows templates, download the following file from the vRealize Automation Appliance management console installation page.</p> <p>https://vcac-va-hostname_or_IPaddress:5480/installer/LinuxGuestAgentPkgs.zip</p> <p>The file includes the vRealize Automation guest agent for Windows and Linux, but you need to download the file only for a Windows template. For a Linux template, the file is downloaded automatically.</p> <p>To install the Windows and Linux guest agent in a template, see Prepare vCenter Server Windows Virtual Machine Templates and Prepare vCenter Server Linux Virtual Machine Templates.</p>
Supported Windows scripting	<p>Application Services supports scripting with Windows CMD, PowerShell 2.0, and BeanShell 1.3.0.</p>
Supported Linux scripting	<p>Application Services supports scripting with Bash and BeanShell 1.3.0.</p>

Table 9-1. Virtual Machine Requirements for Custom Templates (Continued)

Requirement	Description
Linux commands	The following Linux commands must be available on the virtual machine: <ul style="list-style-type: none"> ■ wget ■ unzip ■ sha256sum ■ grep ■ sed ■ setsid ■ awk ■ ifconfig ■ apt-get ■ yum ■ chkconfig ■ dmidecode ■ perl
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 Application Services library.

Creating Virtual Machine Templates in vRealize Automation

To start a clone deployment in vCenter Server with a vRealize Automation 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 *laas Configuration for Virtual Platforms*.

■ [Prepare vCenter Server Windows Virtual Machine Templates](#)

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

- [Prepare vCenter Server Linux Virtual Machine Templates](#)

You must prepare vCenter Server Linux virtual machine templates for vRealize Automation blueprints to create clone deployments. Application Services uses the vRealize Automation blueprints to deploy Linux-based applications and services to the vSphere environment.

- [Create vRealize Automation Blueprints for Clone Deployment](#)

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

- [Create and Configure vRealize Automation Blueprint for Linked Clone Deployment](#)

When you create a custom vRealize Automation blueprint you can allow a linked clone deployment to provision Application Services Windows-based or Linux-based applications and services to the vSphere environment. A linked clone is a copy of a virtual machine that is based on a snapshot. It occupies a small amount of disk space, is quick to provision, and is 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 vRealize Automation blueprints to create clone deployments. Application Services uses the vRealize Automation blueprints to deploy Windows-based applications and services to the vSphere environment.

Prerequisites

- Adhere to the correct version requirements for your release. See the Support Matrix in vRealize Automation documentation at <https://www.vmware.com/support/pubs/vcac-pubs.html>.
- Verify that vCenter Server 5, 5.1, or 5.5 is installed and configured.
- Verify that a vSphere Windows virtual machine is available. See [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#).
- Verify that the vSphere Windows virtual machine does not have an older version of either a vRealize Automation 5.2 or 6.0 guest agent or Application Services 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.
- Remove the existing vRealize Automation agent service and Application Services agent bootstrap service. See [Updating Existing Virtual Machine Templates in vRealize Automation](#).
- 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 Enter 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_72-win64.zip.
 - a Create a `\opt\vmware-jre` folder and unzip the JRE file to the folder.
 - b Open a PowerShell command window and enter `type \opt\vmware-jre\bin\java -version` to verify the installation.

The installed version of Java appears.
- 4 To install the vRealize Automation guest agent.
 - a Change directory to `C:\`.
 - b Download the file `GugentZip.zip` for 32-bit platforms or `GugentZip_64.zip` for 64-bit platforms from the vRealize Automation Appliance management console installation page at https://vcac-va-hostname_or_IPaddress:5480/installer/.
 - c Right-click the downloaded ZIP file and select Unblock from the pop-up menu.

The selection unblocks the contents of the ZIP file.
 - d Unzip the installation file to `C:\`.

For example, `C:\VRMGuestAgent\WinService.exe`.
- 5 Download the Application Services agent bootstrap file from http://Application_Director_IP/agent/vmware-appdirector-agent-bootstrap-windows_VersionNumber.zip.

- a Right-click the downloaded ZIP file and select Unblock from the pop-up menu.

Important If you do not disable this Windows security feature, you cannot use the Application Services agent bootstrap file.

- b Unzip the `vmware-appdirector-agent-bootstrap-windows_version.zip` file to the `\temp` folder.
- 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 vRealize Automation, complete the following steps.
 - a Open a Windows CMD console and navigate to the `\temp` folder.
 - b Enter the command to install the agent bootstrap.

```
install.bat password=Password cloudProvider=vcac vcacServer=IaaS_Server_FQDN httpsMode=true
```

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 Verify that the user **darwin_user** exists.
 - a Enter `lusrmgr.msc` at a command prompt.
 - b Verify that the user **darwin_user** exists and belongs to the administrator group.
 - c Set the password to never expire.

The setting ensures that the template remains usable after 30 days.

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

- 8 Verify that Application Services and VRM Agent services are installed.
- 9 Shut down the Windows virtual machine.
- 10 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 Click **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 click **Use the virtual machine name** and in the Administrator Password section, enter the administrator password of the Windows operating system.
- e Accept the default settings and click **Finish**.

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

- 11 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 vRealize Automation blueprint for clone or linked clone deployment. See [Create vRealize Automation Blueprints for Clone Deployment](#) or [Create and Configure vRealize Automation Blueprint for Linked Clone Deployment](#).

Prepare vCenter Server Linux Virtual Machine Templates

You must prepare vCenter Server Linux virtual machine templates for vRealize Automation blueprints to create clone deployments. Application Services uses the vRealize Automation blueprints to deploy Linux-based applications and services to the vSphere environment.

Prerequisites

- Verify that vCenter Server is installed and configured. For current version requirements for your release, see the Support Matrix in vRealize Automation documentation at <https://www.vmware.com/support/pubs/vcac-pubs.html>.
- Verify that a supported vSphere Linux virtual machine is available. See [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#).
- 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 Application Services virtual machine requirements. See [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#).
- 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 Download the installer script
`http://ApplicationServicesServerIP/tools/preparevCACTemplate.sh` to the virtual machine template.

You can use the `wget` or `scp` commands to download the file.

- 4 Ensure that the installer script has permission to execute.

```
chmod +x ./preparevCACTemplate.sh
```

- 5 Run the installer script.

```
./preparevCACTemplate.sh
```

- 6 Answer all prompts.

If the installer script succeeds, it displays a success message. If not, the installer script displays an error message and logs on to the console.

- 7 If an error occurs, resolve it and repeat [Step 4](#) through [Step 6](#).
- 8 Shut down the Linux virtual machine.
- 9 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 vRealize Automation Blueprints for Clone Deployment](#) or [Create and Configure vRealize Automation Blueprint for Linked Clone Deployment](#).

Create vRealize Automation Blueprints for Clone Deployment

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

Prerequisites

- Verify that vRealize Automation 6.1 or later is installed and configured.
- Verify that the vRealize Automation business group that you plan to use with Application Services is created.
- Verify that a vCenter Server virtual machine is available for the vRealize Automation blueprint to create a clone deployment.

You must have the customization specification name available for a Windows-based vRealize Automation blueprint. See [Prepare vCenter Server Windows Virtual Machine Templates](#).
- Familiarize yourself with how to create a blueprint in vRealize Automation. See *laas Configuration for Virtual Platforms*.

Procedure

- 1 Log in to the vRealize Automation console as a **tenant administrator**.
- 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 vRealize Automation.
- 6 For a Windows-based vRealize Automation blueprint, enter 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.

ThevRealize Automation blueprint appears in the **Blueprint** menu.

What to do next

Register the vRealize Automation blueprint configured for clone deployment for use in Application Services. See [Register the vRealize Automation Cloud Provider and Template](#).

Create and Configure vRealize Automation Blueprint for Linked Clone Deployment

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

Prerequisites

- Verify that vRealize Automation 6.1 or later is installed and configured.
- Verify that the vRealize Automation business group that you plan to use with Application Services is created.
- Verify that a vCenter Server virtual machine is available for thevRealize Automation blueprint to create a linked clone deployment.

You must have the customization specification name available for a Windows-based vRealize Automation blueprint. See [Prepare vCenter Server Windows Virtual Machine Templates](#).

- Familiarize yourself with how to create a blueprint in vRealize Automation. See *laas Configuration for Virtual Platforms*.

Procedure

- 1 Log in to the vRealize Automation console as a **tenant administrator**.
- 2 Select **New Blueprint > Virtual > vSphere (vCenter)** and name the blueprint. For example, 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 vRealize Automation.
- 6 For a Windows-based vRealize Automation blueprint, enter 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, as stated in the prerequisites section.

- 7 Click **OK** to save your changes.

- 8 Select the blueprint, request a machine from the drop-down menu, and accept the default settings. For example, select the blueprint named **vCAC_Clone**.

The request information is derived from the resources defined in the vCenter Server template.

When your machine request is processed, it appears in the **Infrastructure > Machines > Managed Machines** menu. The machine name, for example vCAC_Clone, is appended with a unique series of letters and numbers, for example as vCAC_Clone-VFF17Y0.

- 9 Log in to the machine, for example vCAC_Clone-VFF17Y0, and stop the Application Services and vRealize Automation services depending on your Windows or Linux operating system.

Service	Description
vmware-appdirector-agent-service-vcac	Application Services bootstrap service for Linux
vrmm-agent	vRealize Automation agent service for Linux
AppDAgentBootstrap	Application Services bootstrap service for Windows
VRMAgent or winservice	vRealize Automation 6.1 or later agent service for Windows

- 10 Run the Windows script `agent_reset.bat` or Linux script `agent_reset.sh` to remove runtime log files from the machine, for example from machine vCAC_Clone-VFF17Y0.
- 11 Power off the machine, for example power off machine vCAC_Clone-VFF17Y0.

What to do next

Create a snapshot from the blueprint for a linked clone deployment. See [Create a Snapshot from a vRealize Automation Blueprint](#).

Create a Snapshot from a vRealize Automation Blueprint

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

For information about creating a machine snapshot, see [Create a Snapshot from a vRealize Automation Blueprint](#). For related information about taking a snapshot of a VM template using vSphere Client, see VMware vSphere Documentation.

Prerequisites

- Verify that vRealize Automation 6.1 or later is installed and configured.
- Verify that the vRealize Automation business group that you plan to use with Application Services is created.
- Verify that a vRealize Automation blueprint is available from which to create a snapshot. See [Create and Configure vRealize Automation Blueprint for Linked Clone Deployment](#).

Procedure

- 1 Select **Infrastructure > Blueprints > Blueprints**.

- 2 Click **New Blueprint > Virtual > vSphere (vCenter)** and name the blueprint. For example, name the blueprint **vCAC_LinkedClone**.
- 3 Complete the information on the **Blueprint Information** tab.
- 4 On the **Build Information** tab, select **Linked Clone** from the **Action** drop-down menu.
- 5 Select a snapshot of the vCAC_Clone-VFF17Y0 machine.
 - a Click the **Browse** icon next to the **Clone from** text box.
The Select Snapshot dialog box appears.
 - b Select the machine **vCAC_Clone-VFF17Y0** and click **Next**.
 - c Select the snapshot **vCAC_Clone-VFF17Y0_snapshot** and click **Next**.
 - d Click **OK** to save your changes.
vCAC_Clone-VFF17Y0_snapshot appears in the **Clone from** text box.
- 6 For a Windows-based vRealize Automation blueprint, enter 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 Configure the **Properties** and **Actions** tabs as required and click **OK** to save your changes.
The new blueprint, for example **vCAC_LinkedClone**, appears on the **Blueprints** page.

What to do next

Register the vRealize Automation blueprint configured for linked clone deployment for use in Application Services. See [Register the vRealize Automation Cloud Provider and Template](#).

Updating Existing Virtual Machine Templates in vRealize Automation

Before you can update the content of an existing Linux or Windows template, or 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 Application Services 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 the existing Application Services agent bootstrap and vRealize Automation 6.0 or later guest agent. In a PowerShell command window, run the following command.

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

Register the vRealize Automation Cloud Provider and Template

Application Services registers a cloud provider by connecting to a vRealize Automation business group.

A vRealize Automation business group associates users with a set of services and resources, often corresponding to a business, department, or other organizational unit. Each business group can have access to one or more vRealize Automation blueprints used to request machines. These blueprints can belong to one or more business groups. To request machines, a user must belong to at least one business group. The machine resource information in the vRealize Automation blueprint is added manually or obtained from the cloned or linked cloned vSphere template.

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Verify that vRealize Automation 6.1 or later is installed and configured.
- Verify that the vRealize Automation business group that you plan to use with Application Services is created and configured.
- Verify that you registered Application Services to vRealize Automation by using the CLI. See [Register Application Services to vRealize Automation](#).
- Verify that at least one vRealize Automation blueprint is available in the business group for the Application Services cloud provider to connect to. This blueprint must point to a virtual machine template that meets the Application Services requirements. See [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Clouds > Cloud Providers**.
- 2 Click the **Add** icon ()

3 Complete the cloud provider information.

For vRealize Automation 6.0 or later, when you register Application Services to vRealize Automation, the host name and business group name get populated for the logged-in user.

Option	Description
Name and Description	Include the vRealize Automation business group name in either of these fields. The text from these fields appears in the Deployment Profile wizard under the Cloud Provider column.
Cloud Provider Type	vCAC, vCloud, and EC2 are the supported types.
vCAC Infrastructure IP/Host Name/URL	Display the IP address or host URL for the registered vRealize Automation server. The field is read-only. Examples of a secure vCAC IP are https://192.0.2.115 and https://192.10.1.113:443. An example of a secure Host Name URL is https://vcac.it.vmware.com.
User Name and Password	Add business group manager credentials for the selected business group in vRealize Automation. The domain name is required when you enter the user name. An example of a user name is myname@mycompany.com.
Business Group	Select a business group from the Business Group drop-down menu. The menu contains the business groups of the logged-in user.
Sharing	Displays the Private status of the cloud provider. This value is always Private and is read-only.

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.

6 To register vRealize Automation blueprints, click **Edit**.

7 In the Templates section, click **+**.

8 Select the check boxes next to one or more cloud templates or vRealize Automation blueprints in the list, and click **OK**.

Only global blueprints and blueprints that belong to the selected business group of the cloud provider appear.

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.

9 To change existing information for a vRealize Automation cloud provider, click **Edit** in the toolbar, make your changes, and click **Save**.

10 Click **Save**.

As a best practice, refresh the cloud template information, vRealize Automation Infrastructure IP, and the host name when you edit a vRealize Automation cloud provider.

Note If the vRealize Automation blueprint name changes in the vRealize Automation environment, the new blueprint name does not appear in the Application Services 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 Application Services.

What to do next

Map the vRealize Automation cloud template to the Application Services logical templates. See [Add a Logical Template to the Library](#).

Create a deployment environment to deploy an application to the vRealize Automation environment. See [Create a vRealize Automation Deployment Environment](#).

Create a vRealize Automation Deployment Environment

You can map a deployment environment to a vRealize Automation reservation policy before you deploy applications to that cloud environment. If you do not select a reservation policy, vRealize Automation picks any suitable reservation policy at deployment time.

A vRealize Automation business group can have several reservations mapped to reservation policies. A reservation has a pool of resources such as memory, storage, and networking 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 vRealize Automation blueprint. When you request a machine from this blueprint, the virtual machine is saved in a designated storage space and assigned to a network that is already defined in the reservation. If you do not map a reservation policy to a blueprint, vRealize Automation assigns a reservation depending on the resource requirements of the virtual machine in the vRealize Automation blueprint.

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Verify that vRealize Automation 6.1 or later is installed and configured.
- Verify that the vRealize Automation business group that you plan to use with Application Services is created and configured.
- Verify that you have tenant administrator privileges to access the business groups.

Users with tenant administrator privileges create and maintain business groups. For more information about business groups, see *IaaS Configuration for Virtual Platforms*.

- Verify that you are a business group manager for the business group.

- Verify that a reservation policy is created and configured in the vRealize Automation. The Application Services deployment environment is equivalent to the reservation policy in vRealize Automation.
- Verify that one vRealize Automation cloud provider is registered in Application Services. See [Register the vRealize Automation Cloud Provider and Template](#).

Procedure

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

Option	Description
Deployment Environment name and Description	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.
Cloud Provider	If the cloud provider does not appear in the list, cancel the dialog box and select Clouds > Cloud Providers to add the vRealize Automation cloud provider.
Reservation Policy	<p>Click Select to select from a list of reservation policies that the cloud provider you selected provides.</p> <p>Reservation policies in vRealize Automation 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.</p> <p>If you select No reservation policy, vRealize Automation assigns a reservation based on availability of resources.</p>

- 4 Select a reservation policy and click **OK**.

A vRealize Automation reservation policy is mapped to the Application Services deployment environment name.

- 5 Click **Save**.
- 6 (Optional) To change the existing information for a vRealize Automation deployment environment, click **Edit** in the toolbar and 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](#), [Create a Policy Instance](#) or [Create a Puppet Solution Instance](#).

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

- Log in to Application Services as an **application cloud administrator**.
- Depending on your cloud provider, you must have at least one vCloud Director, vRealize Automation, or Amazon EC2 deployment environment mapped in Application Services. See [Create a vCloud Director Deployment Environment](#), [Create a vCloud Director Deployment Environment](#), [Create a vRealize Automation Deployment Environment](#), or [Create an Amazon EC2 Deployment Environment](#).
- If you are using a custom external service, verify that a preconfigured or custom external service is available in the library. See [Add an External Service to the Library](#).

Procedure

- 1 On the Application Services 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.

Option	Description
Name and Description	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.
External Service Version	If the external service does not appear in the drop-down list, cancel the dialog box and select Library > External Services to add a custom external service to the library.
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**.
- 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.

Note The composite deployment feature does not support policy compliance scans.

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Depending on your cloud provider, you must have at least one vCloud Director, vRealize Automation, or Amazon EC2 deployment environment mapped in Application Services. See [Create a vCloud Director Deployment Environment](#), [Create a vCloud Director Deployment Environment](#), [Create a vRealize Automation Deployment Environment](#), or [Create an Amazon EC2 Deployment Environment](#).
- If you are using a custom policy, verify that it is available in the library. See [Add a Policy to the Library](#).

Procedure

- 1 On the Application Services 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.

Option	Description
Name and Description	Include the policy name in either of these fields. The text from these fields appears in the policy instances section for the deployment environment.
Policy	If the policy does not appear in the drop-down list, cancel the dialog box and select Library > Policies to add a custom policy to the library.
Version	Click the down arrow to select from a list of versions that are associated with the policy.
Critical	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.

Note If you specify a new property value in the policy instance, the new property value overrides the existing library 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 library 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**.
- 6 Click the arrow next to the policy instance name to return to the previous page.

Register a Puppet Solution Instance

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

For the registration process, you can implement either the manual or the automated option. For the automated option instructions, see [Register a Puppet Master Solution Instance](#).

Procedure

1 [Prepare the Puppet Master Environment for Registration](#)

To register a Puppet Master within a deployment environment you must prepare the Puppet Master to work with the Application Services server.

2 [Create a Puppet Solution Instance](#)

A solution instance such as Puppet Master, must be created in an Application Services deployment environment 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 you must prepare the Puppet Master to work with the Application Services server.

You can also automate the registration process, see [Register a Puppet Master Solution Instance](#).

Prerequisites

- Verify that Puppet Enterprise version 3.0.1 to 3.2.3 or Puppet Open Source version 3.2.4 to 3.6.2 is installed.
- Verify that you have the latest supported version of the Puppet agents installed. See the Puppet Labs documentation Web site for the agent installation instructions.
- Make sure that there is a client machine available with Puppet installed besides the Puppet Master.
- Make sure the system times for the Application Services and the Puppet Master servers are synchronized. You can use an NTP server for this purpose. The time difference between the two servers cannot be more than 60 seconds.

Procedure

- 1 Navigate to the directory on your Puppet Master and create a directory called `appd_nodes`.
 - On Puppet Enterprise, navigate to `/etc/puppetlabs/puppet/manifests/`.
 - On Puppet Open Source, 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 'appd_nodes/*.pp'` to the first line of the file.
- 3 In the command line, create the `empty_manifest_file.pp` file.
 - On Puppet Enterprise, enter
`touch /etc/puppetlabs/puppet/manifests/appd_nodes/empty_manifest_file.pp`
 - On Puppet Open Source, enter
`touch /etc/puppet/manifests/appd_nodes/empty_manifest_file.pp`
- 4 Open the mcollective server configuration file and add the line `plugin.ssl_serializer = yaml`, if missing.
 - On Puppet Enterprise, open the file `/etc/puppetlabs/mcollective/server.cfg`.
 - On Puppet Open Source, open the file `/etc/mcollective/server.cfg`.
- 5 Download the `appdintegn.rb` and `appdintegn.ddl` agent files.
`http://Application_Services_IP/artifacts/solutions/puppet/mcollective/agent/appdintegn.ddl`
`http://Application_Services_IP/artifacts/solutions/puppet/mcollective/agent/appdintegn.rb`
- 6 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/`.
- 7 Restart MCollective to view the newly copied agent files.
 - On Puppet Enterprise, enter `/etc/init.d/pe-mcollective restart`.
 - On Puppet Open Source, enter `/etc/init.d/mcollective restart`.
- 8 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 Application Services 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.
- 9 (Optional) On the Puppet master, if the auto-sign for the certificate is not turned on enter the command `sudo puppet cert sign Name`.

- 10 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/
```

- 11 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
```

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

- On Puppet Enterprise, enter


```
puppet certificate find pe-internal-mcollective-servers --ssldir TempCredsDir  
--ca-location remote --ca_server CAPuppetMaster.
```
- On Puppet Open Source, enter


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

Application Services creates the node definition file `*.pp` for each virtual machine it provisions in the `/etc/puppetlabs/puppet/manifests/appd_nodes` or `/etc/puppet/manifests/appd_nodes` directory.

What to do next

Register the Puppet master within a deployment environment in Application Services. See [Create a Puppet Solution Instance](#).

Create a Puppet Solution Instance

A solution instance such as Puppet Master, must be created in an Application Services deployment environment 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 solution instance per a deployment environment. The solution instances of multiple deployment environments can point to the same Puppet Master. For information on using a Ruby script that runs locally on the Puppet Master server, see [Register a Puppet Master Solution Instance](#).

Prerequisites

- Log in to Application Services as an **application cloud administrator**.

- Depending on your cloud provider, you must have at least one vCloud Director, vRealize Automation, or Amazon EC2 deployment environment mapped in Application Services. See [Create a vCloud Director Deployment Environment](#), [Create a vCloud Director Deployment Environment](#), [Create a vRealize Automation Deployment Environment](#), or [Create an Amazon EC2 Deployment Environment](#).
- Make sure that the Puppet Master is configured to work with the Application Services server. See [Prepare the Puppet Master Environment for Registration](#) or [Register a Puppet Master Solution Instance](#).
- If you are using a custom policy, verify that it is available in the library. See [Add a Policy to the Library](#).

Procedure

- 1 Log in to the Application Services Web interface.
- 2 On the Application Services 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.

Option	Description
Name and Description	Include the solution name. You can add property or agent configuration information for the solution in the description section.
Solution	Select the available Puppet Master v1.0.0 solution from the drop-down list.

The solution properties and corresponding agents are populated.

- 6 In the Properties section, enter the required property values.

Property	Description
puppet_server	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 Application Services server must be able to contact Puppet Master at this DNS name, unless the puppet_server_ip property is also supplied.
puppet_server_ip	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 Application Services server or agent nodes.
puppet_environment	Name of the Puppet Master environment assigned to a puppet agent. This name corresponds to the environment configuration property of a puppet agent .

Property	Description
agent_run_interval	<p>How frequently the puppet agent applies the library. The default run interval is every 30 minutes (30m). You can specify the run interval in the following units.</p> <ul style="list-style-type: none"> ■ Seconds. For example, 30 or 30s. ■ Minutes. For example, 30m. ■ Hours. For example, 6h. ■ Days. For example, 5d. ■ Years. For example, 2y. <p>A run interval of 0 tells the puppet agent to run continuously.</p>
identity_private_key	<p>Private key of the certificate issued to the Application Services server by the Puppet Master.</p> <p>Application Services 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--'</p> <p>Copy contents of the file <i>TempCredsDir/private_keys/Name.pem</i> you created to prepare the Puppet Master, as the value for this property.</p>
identity_public_cert	<p>The X509 certificate corresponding to the <code>identity_private_key</code> property.</p> <p>The format of the certificate is PEM encoded including the header and trailer. The header for example is formatted as, '-----BEGIN CERTIFICATE--'</p> <p>Copy contents of the file <i>TempCredsDir/certs/Name.pem</i> you created to prepare the Puppet Master, as the value for this property.</p>
ca_cert	<p>The X509 certificate of the certificate authority that issues digital certificates used to authenticate with Puppet Master and MCollective.</p> <p>The format of the certificate is PEM encoded including the header and trailer. The header for example is formatted as, '-----BEGIN CERTIFICATE--'</p> <p>Copy contents of the file <i>TempCredsDir/certs/ca.pem</i> you created to prepare the Puppet Master, as the value for this property.</p>
mc_servers_shared_cert	<p>Shared server certificate or public key for the MCollective cluster.</p> <p>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--'</p> <p>Copy contents of the file <i>TempCredsDir/certs/pe-internal-mcollective-servers.pem</i> for Puppet Enterprise or <i>TempCredsDir/certs/mcollective-servers.pem</i> for Puppet Open Source you created to prepare the Puppet Master, as the value for this property.</p> <p>The trailer for example is formatted as, '-----END CERTIFICATE-----' or '-----END RSA PUBLIC KEY--'</p>
mc_messaging_server_password	<p>Password corresponding to the config setting <code>plugin.activemq.pool.1.password</code> in the MCollective server config file.</p> <p>For example, the MCollective server config file in the Puppet Enterprise is located at <code>/etc/puppetlabs/mcollective/server.cfg</code> .</p>
identity_cert_name	<p>Name of the identity certificate.</p> <p>This name is the string which you set when you generated the Application Services certificate to correspond to the Puppet Master during the preparation of the Puppet Master.</p>

Property	Description
puppet_version	Puppet Enterprise suite version for a Puppet Enterprise and Puppet package version for a Puppet open source. For example, the Puppet Enterprise version is 3.2.3 and the Puppet open source version is 3.6.2.
mc_messaging_server_username	User name corresponding to the config setting <code>plugin.activemq.pool.1.user</code> in the MCollective server config file. For example, the MCollective server config file in the Puppet Enterprise is located at <code>/etc/puppetlabs/mcollective/server.cfg</code> .
mc_messaging_server_port	Port corresponding to the config setting <code>plugin.activemq.pool.1.port</code> in the MCollective server config file. For example, the MCollective server config file in the Puppet Enterprise is located at <code>/etc/puppetlabs/mcollective/server.cfg</code> .
node_manifest_dir	Directory where the node definition manifests are located for virtual machines deployed using Application Services. The Puppet Master site.pp must be able to import *.pp files from this directory.
mc_appd_agent_ddl	Location of the MCollective Application Services agent DDL file. You do not need to override this value.
mc_client_setup_script	Location of the MCollective Client setup Beanshell script. You do not need to override this value.
global_conf	URL to download the Darwin global configuration for each node.
agent_post_install_wait	Time in seconds to wait after installing the agent.

7 In the Agents section, enter the required values.

Option	Description
pe_installer_payload_base_path	Base URI where the operating system and architecture specific Puppet Enterprise tar files are hosted. Not for Windows. The script detects the URI based on the <code>pe_installer_payload_base_path</code> and the current version of operating system and architecture on the machine. You can skip this auto detection by specifying a value for the <code>pe_installer_payload</code> property. Automatic detection assumes that the specific directory structure mirrors the structure of the Puppet Labs hosted tar files that are located at https://s3.amazonaws.com/pe-builds/ .
installer_payload_base_path	Windows version of <code>pe_installer_payload_base_path</code> .
is_enterprise	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.

- 8 In the Agents section, enter the optional value.

Option	Description
<code>pe_installer_payload</code>	Link to download tar file for Puppet Enterprise. Not for Windows. 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 <code>pe_installer_payload_base_path</code> and current version of operating system and architecture on the machine. An example URI is, https://s3.amazonaws.com/pe-builds/released/3.0.1/puppet-enterprise-3.0.1-all.tar.gz
<code>installer_payload</code>	Windows version of <code>pe_installer_payload</code> .

- 9 Click **Save**.

Application Services checks the property and agent values you entered and displays an error message if the required values are missing.

- 10 Click **Validate Connection** to verify 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, enter the command to remove the temporary credential directory you created when you downloaded the public certificate from the Puppet Master.

```
rm -rf TempCredsDir
```

Register a Puppet Master Solution Instance

You can register a Puppet Master solution instance in an Application Services deployment environment using a Ruby script that runs locally on the Puppet Master server.

For information on the equivalent alternative to using a Ruby script, see [Create a Puppet Solution Instance](#).

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- You must have a Puppet Master server available with Ruby version 1.8.7, 2.0.x for Puppet 3.2 and higher, or 2.1.x for Puppet 3.5 and higher installed on it.

Procedure

- 1 Log in to the Puppet Master server.
- 2 Download the Ruby script from the Application Services server URL.

```
http://ApplicationDirectorIP/artifacts/solutions/puppet/RegisterWithAppD.rb
```

- 3 (Optional) Run the Ruby script with the `-h` option to view the registration options.

```
ruby RegisterWithAppD.rb -h
```

Options:

<code>-i, --appd_ip APPSERVICESIP</code>	Specify the IP address or host name of the Application Services instance.
<code>-u, --appd_user USERNAME</code>	Specify the user name of the administrator account for the Application Services instance. The default is admin.
<code>-p, --appd_password PASSWORD</code>	Specify the password for the administrator account.
<code>-t, --appd_tenant TENANT</code>	Specify the name of the tenant assigned to the user.
<code>-g, --appd_group GROUP</code>	Specify the business group assigned to the user.
<code>-d, --appd_de_name DENAME</code>	Specify the name of the deployment environment to register this puppet master under.
<code>-n, --node_manifest_dir DIR</code>	Specify the directory where Application Services places node definition files for servers under its management. The default is the 'appd_nodes' sub-directory under the site manifest directory.
<code>-c, --cert_name CN</code>	Specify the 'subject' for the certificate that Application Services uses to authenticate to Puppet Master. The default is 'vmware-appd'.
<code>-s, --[no-]skip_mc_agent</code>	Install the Application Services MCollective agent and restart the MCollective daemon. Installs the MC agent by default. Use <code>-s</code> to skip.

- 4 Run the Ruby script to register the Puppet Master solution instance with the Application Services instance.

```
ruby RegisterWithAppD.rb -i APPSERVICESIP -u USERNAME -p PASSWORD
-d DENAME -t TENANT -g GROUP
```

- 5 Verify that you receive a notification when the script successfully registers the solution instance.
- 6 Log in to the Application Services Web interface.
- 7 On the Application Services title bar, click the drop-down menu and select **Clouds > Deployment Environments**.
- 8 Select the existing deployment environment with the registered solution instance.
- 9 Select the Solution Instances tab and click the solution instance.
- 10 If you are running the Ruby script on Puppet Master Open Source, perform the following steps.
 - a Click Edit in the toolbar.
 - b In the Agents section, set the `is_enterprise` property value to false.
 - c Click Save to finish.
- 11 Click **Validate Connection** to check the connection to the server.

A notification informs you that the connection is successful and a green check mark appears next to the Validate Connection button.

What to do next

Import Puppet content into the Application Services library as services. See [Import a Puppet Service to the Library](#).

Setting Up Application Provisioning for the vCloud Director Environment

10

To use Application Services 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 Application Services to create cloud providers and register them to the organization vDC. See [Set Up vCloud Director for Application Services](#). You must also create custom virtual machine templates in vCloud Director.

In Application Services, 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 Application Services library. 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](#).

This chapter includes the following topics:

- [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#)
- [Creating Windows Virtual Machine Templates in vCloud Director](#)
- [Create Linux Virtual Machine Templates in vCloud Director](#)
- [Verify Cloud Template Configuration from the vCloud Director Catalog](#)
- [Updating Existing Virtual Machine Templates in vCloud Director](#)
- [Exporting Virtual Machine Templates with OVF Format](#)
- [Register the vCloud Director Cloud Provider and Template](#)
- [Create a vCloud Director Deployment Environment](#)

Virtual Machine Requirements for Creating vCloud Director Custom Templates

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

Table 10-1. Virtual Machine Requirements for Custom Templates

Requirement	Description
Operating system	<p>The following operating systems are supported:</p> <ul style="list-style-type: none"> ■ Red Hat Enterprise Linux 6.1, 6.3, and 6.5 for 32- and 64-bit ■ Windows Server 2008 R2 Enterprise with Service Pack 1 for 64-bit, and Windows Server 2012 R2 for 32- and 64-bit ■ CentOS 5.10, 6.1, 6.3, 6.5 for 32- and 64-bit ■ SUSE Linux Enterprise 11 SP2 for 32- and 64-bit ■ Ubuntu 12.04.2 for 32- and 64-bit
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	<p>Install the supported JRE from one of the following packages, which are available in the Application Services virtual appliance:</p> <ul style="list-style-type: none"> ■ JRE for Windows <ul style="list-style-type: none"> http://Application_Services_IP/agent/jre-1.7.0_72-win64.zip <p>See Configure Windows Virtual Machine Template in vCloud Director.</p> ■ JRE for Linux <ul style="list-style-type: none"> ■ http://Application_Services_IP/agent/jre-1.7.0_72-lin32.zip ■ http://Application_Services_IP/agent/jre-1.7.0_72-lin64.zip <p>To install JRE on a virtual machine template, see Create Linux Virtual Machine Templates in vCloud Director.</p>
Linux agent bootstrap service	<p>Download the Linux agent bootstrap package <code>vmware-appdirector-agent-service</code> from the VMware product download site http://vmware.com/web/vmware/downloads. Install the Linux agent bootstrap script from one of the following packages:</p> <ul style="list-style-type: none"> ■ http://Application_Services_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_x86_64.rpm ■ http://Application_Services_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.rpm ■ http://Application_Services_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_amd64.deb ■ http://Application_Services_IP/agent/vmware-appdirector-agent-service_6.0.0.0-0_i386.deb <p>To install the agent bootstrap service on a virtual machine template, see Create Linux Virtual Machine Templates in vCloud Director.</p> <p>These files are downloaded if you use the <code>preparevCACTemplate.sh</code> script.</p>
Windows agent bootstrap service on vCloud Director	<p>Download the http://Application_Services_IP/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip file from the Application Services virtual appliance.</p> <p>To install the agent bootstrap service on a Windows-based virtual machine template, see Configure Windows Virtual Machine Template in vCloud Director.</p>
Supported Windows scripting	Application Services supports scripting with Windows CMD or PowerShell 2.0.
Supported Linux scripting	Application Services supports scripting with Bash.

Table 10-1. Virtual Machine Requirements for Custom Templates (Continued)

Requirement	Description
Linux commands	The following Linux commands must be available on the virtual machine: <ul style="list-style-type: none"> ■ wget ■ md5sum ■ grep ■ sed ■ setsid ■ awk ■ ifconfig ■ apt-get ■ yum
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 Application Services library.

Creating Windows Virtual Machine Templates in vCloud Director

With Windows virtual machine templates, you can use Application Services 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](#)

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](#)

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

- [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.

- [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 Application Services.

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 Application Services is created and configured with a user account that has privileges to create and share vCloud Director library 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 or Windows Server 2012 R2 operating system and your application.
- 3 Install Windows Server 2008 R2 Enterprise with Service Pack 1 or Windows Server 2012 R2 in the virtual machine, using the **Full Installation** option.
- 4 Install VMware Tools in the virtual machine.

What to do next

Install the supported Java SE 7 Runtime Environment package and Application Services agent bootstrap file in the Windows virtual machine. See [Configure Windows Virtual Machine Template in vCloud Director](#).

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 Application Services.

The agent bootstrap is a daemon that runs as an NT service on Windows after a virtual machine starts for the first time. The Application Services 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](#).

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_51-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\bin directory and enter **java -version** to verify the installation.

The installed version of Java appears.

- 5 Download and install the Application Services agent bootstrap file from http://Application_Director_IP/agent/vmware-appdirector-agent-bootstrap-windows_6.0.0.0.zip.
- 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 Application Services 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 (rkttools.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 enter **install.bat password=Password cloudProvider=vcd**.
- 11 From the command-line, enter **services.msc** and open the Application Services agent bootstrap service.
- 12 On the **Log On** tab, set the **Log on as** option to **This account** and enter 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 Enter **net start AppDAgentBootstrap** to verify that the service starts successfully.
 - b Enter **net stop AppDAgentBootstrap** to stop the service.
- 14 Delete the runtime log files generated during the template creation.
`agent_reset.bat`
- 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](#).

Add the vApp template to the vCloud Director catalog so that you can register it for use in Application Services. See [Add Windows Virtual Machine Templates to the vCloud Director Catalog](#).

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](#).

Prerequisites

- Verify that you have correctly created a Windows vApp and configured the virtual machine template. See [Create a vCloud Director vApp](#) and [Configure Windows Virtual Machine Template in vCloud Director](#).
- 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 enter the Windows domain name, domain account, and account password.
- 8 Select **Add to Library** 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 library. See [Add Windows Virtual Machine Templates to the vCloud Director Catalog](#).

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 Application Services.

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Verify that the Windows template meets the Application Services virtual machine requirements. See [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#).

- Complete the preparation tasks required to create a Windows vApp and configure the virtual machine template. See [Create a vCloud Director vApp](#) and [Configure Windows Virtual Machine Template in vCloud Director](#).

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 Library** to create a vApp template in the library.

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 Application Services. See [Register the vCloud Director Cloud Provider and Template](#).

Create Linux Virtual Machine Templates in vCloud Director

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

Prerequisites

vCloud Director

- 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 Application Services 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 Application Services virtual machine requirements. See [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#).
- 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_51-lin-ArchitectureName.zip`, where the *ArchitectureName* is 32 or 64.

b Create an `/opt/vmware-jre` directory.

c Unzip the JRE package.

```
unzip -d /opt/vmware-jre
```

d Enter `/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 Application Services 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 enter `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 enter `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 Library** to create a vApp template in the library.

What to do next

Register the Linux template for use in Application Services. See [Register the vCloud Director Cloud Provider and Template](#).

Verify Cloud Template Configuration from the vCloud Director Catalog

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

If your cloud template has a customization script, Application Services 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 Application Services 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-vmc/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 Application Services agent boot service ran and attempted to download the Application Services agent JAR file from the Application Services server.

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

What to do next

Verify that the items listed in [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#) 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 Application Services 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 Application Services 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, Application Services cannot access virtual machine templates.

Register the vCloud Director Cloud Provider and Template

Application Services 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 Application Services, see [Preparing to Install Application Services](#).

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

- Log in to Application Services as an **application cloud administrator**.
- Verify that vCloud Director 5.1.2 or 5.5 is installed and configured.
- Create virtual machine templates that meet Application Services requirements. See [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#).
- Verify that the vCloud organization that you plan to use with Application Services is created and configured with organization administrator privileges.

- When you map cloud templates for a cloud provider in Application Services, 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 Application Services cloud provider.

Procedure

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

Option	Description
Name and Description	Include the vCloud Director organization name in either of these fields. The text from these fields appears in the Deployment Profile wizard under the Cloud Provider column.
Cloud Provider Type	vCloud Director 5.1.2 or 5.5 is the supported type.
Cloud IP/Hostname	The cloud IP or host name must match the IP address or host name of the vCloud Director instance.
Organization Name	The organization name must match the name of an organization in vCloud Director.
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 Application Services 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 Application Services.

What to do next

Map these vCloud Director cloud templates to the Application Services logical templates. See [Add a Logical Template to the Library](#).

Create a deployment environment to deploy an application to the vCloud Director environment. See [Create a vCloud Director Deployment Environment](#).

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

- Log in to Application Services as an **application cloud administrator**.
- Verify that at least one vCloud Director cloud provider is registered in Application Services. See [Register the vCloud Director Cloud Provider and Template](#).
- Verify that an organization vDC is created and configured in the vCloud Director organization. The Application Services deployment environment is equivalent to the organization vDC in vCloud Director.

For information about setting up vCloud Director for use with Application Services, see [Preparing to Install Application Services](#).

- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Clouds > Deployment Environments**.
- 2 Click **New**.

3 Complete the deployment environment information.

Option	Description
Deployment Environment name and Description	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.
Cloud Provider	If the cloud provider does not appear in the list, cancel the dialog box and select Clouds > Cloud Providers to add the vCloud Director cloud provider.
Organization vDC URI	Click Select to select from a list of vDCs that the cloud provider you selected provides.

4 Select an organization vDC and click **OK**.

A vCloud Director organization vDC is mapped to the Application Services 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](#), [Create a Policy Instance](#) or [Create a Puppet Solution Instance](#).

Setting Up Application Provisioning for the Amazon EC2 Environment

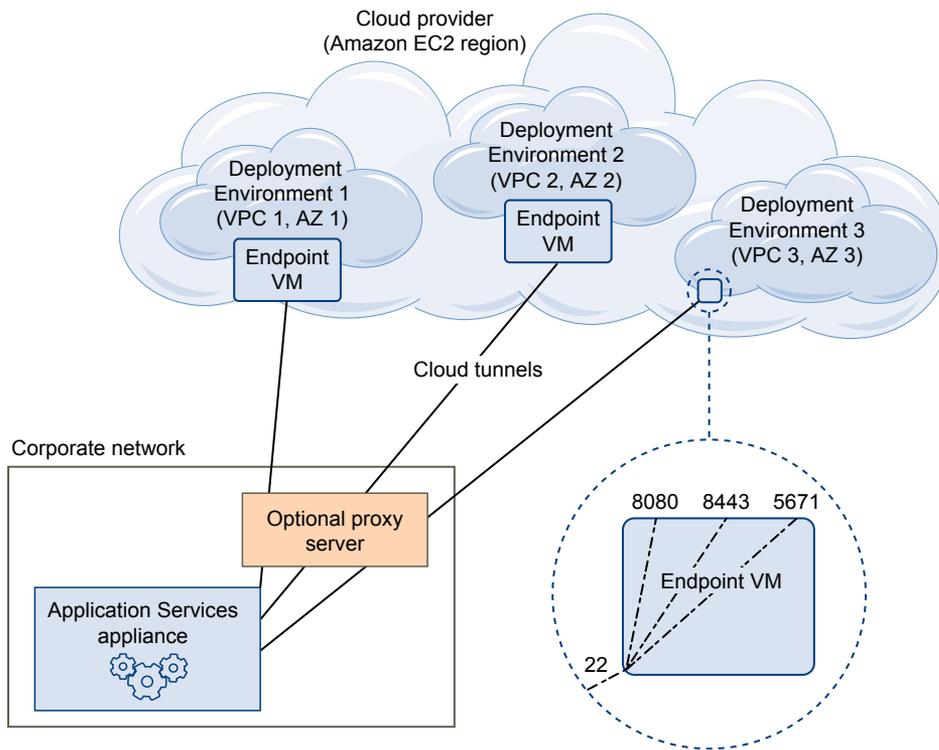
11

You must set up one or more cloud environments to use Application Services to deploy applications in an Amazon EC2 environment.

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 Application Services, 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 Application Services, 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 Application Services library. 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 Application Services 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 Application Services appliance in your corporate network to your Amazon VPC environment. Application deployments in the VPC communicate with the Application Services appliance through the Endpoint VM. You can create and manage the cloud tunnels using the CLI. See [Managing Cloud Tunnels](#).

Figure 11-1. Application Services and Amazon EC2 Topology

Familiarize yourself with the key concepts that relate to setting up and configuring a cloud environment for application provisioning. See [Key Concepts](#).

This chapter includes the following topics:

- [Configure Amazon EC2 Environment for Application Services](#)
- [Virtual Machine Requirements for Creating Amazon EC2 Custom Templates](#)
- [Create Amazon EC2 Virtual Machine Templates or AMIs](#)
- [Register the Amazon EC2 Cloud Provider and Template](#)
- [Create an Amazon EC2 Deployment Environment](#)

Configure Amazon EC2 Environment for Application Services

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 Application Services 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 Application Services.
- 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.

Port Access	Source or Destination
Inbound 22	0.0.0.0/0
Inbound ALL	SubnetofVPC
Outbound ALL	0.0.0.0/0
(Optional) Inbound 2222	0.0.0.0/0

What to do next

Create a Linux virtual machine in your VPC. See [Create an Endpoint VM](#).

Create an Endpoint VM

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

An Application Services 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 Application Services 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 Application Services](#).

Procedure

- 1 Install one Endpoint VM in each VPC to manage using Application Services.
- 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-eb`s AMI or the `amazon/ami-vpc-nat-1.0.0-beta.x86_64-eb`s 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 Application Services server. See [Create a Cloud Tunnel to Connect to Amazon EC2](#).

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 Application Services server.

Your network from Application Services 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 Application Services appliance over the cloud tunnel.

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Verify that the Endpoint VM is properly set up and configured. See [Configure Amazon EC2 Environment for Application Services](#).
- 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 Application Services is running.
The proxy server or the network must permit access to the standard SSH port 22 outside the network.
- Start the Application Services CLI. See [Start the CLI Remotely](#).

Procedure

- 1 Use the SSH client to copy the downloaded private key file for the Endpoint VM to the Application Services appliance and copy the file to the `/tmp` directory.
- 2 In the roo shell, create a secure cloud tunnel instance.

```
create-cloud-tunnel --name TunnelName --description "TunnelDescription" --enabled false --
externalAddress EndpointVMElasticIP
--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.

Application Services 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](#) and [Create Amazon EC2 Virtual Machine Templates or AMIs](#).

You can also manage the existing cloud tunnel connection in CLI. See [Managing Cloud Tunnels](#).

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 Application Services.

Table 11-1. Virtual Machine Requirements for Custom AMIs

Requirement	Description
Operating system	CentOS 6.3
JRE	<p>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 Application Services virtual appliance:</p> <p>JRE for Linux</p> <ul style="list-style-type: none"> ■ http://Application_Director_IP/agent/jre-7u45-linux-amd64.rpm ■ http://Application_Director_IP/agent/jre-7u45-linux-i586.rpm <p>To install JRE on an RPM-based AMI, see Create Amazon EC2 Virtual Machine Templates or AMIs.</p>
Linux agent bootstrap service	<p>Download the Linux agent bootstrap package <code>vmware-appdirector-agent-service-ec2</code> from the VMware product download site http://vmware.com/web/vmware/downloads. Install the Linux agent bootstrap script from one of the following packages:</p> <ul style="list-style-type: none"> ■ http://Application_Director_IP/agent/vmware-appdirector-agent-service-ec2_6.0.0.0-0_x86_64.rpm ■ http://Application_Director_IP/agent/vmware-appdirector-agent-service-ec2_6.0.0.0-0_i386.rpm <p>To install the agent bootstrap service on an RPM-based AMI, see Create Amazon EC2 Virtual Machine Templates or AMIs.</p> <p>These files are downloaded if you use the <code>preparevCACTemplate.sh</code> script.</p>
Supported Linux scripting	Application Services supports scripting with Bash.

Table 11-1. Virtual Machine Requirements for Custom AMIs (Continued)

Requirement	Description
Linux commands	The following Linux commands must be available on the virtual machine: <ul style="list-style-type: none"> ■ 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 Application Services library.

Create Amazon EC2 Virtual Machine Templates or AMIs

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

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 Application Services applications will run and confirm that the AMIs are accessible to the Region.
- Verify that the Amazon EC2 AMI meets the Application Services virtual machine requirements. See [Virtual Machine Requirements for Creating Amazon EC2 Custom Templates](#).

For more information about using AMIs, see the Amazon 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 Application Services, 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, enter `rpm -i jre-7u45-linux-ArchitectureName.rpm` to install the package.
- 4 Download and install the Application Services agent bootstrap service for an RPM-based AMI.
 - a On a machine in the same network as Application Services, 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, enter `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.

- 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 Application Services cloud provider and provide access to the AMIs or virtual machine templates. See [Register the Amazon EC2 Cloud Provider and Template](#).

Register the Amazon EC2 Cloud Provider and Template

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

The following Amazon Region code values are supported when specifying cloud provider information. For the most current list of supported Amazon Region code values, see *Amazon AWS Regions and Endpoints* documentation. For example, see

http://docs.aws.amazon.com/general/latest/gr/rande.html#ec2_region.

Table 11-2. Common Supported Amazon Region Code Values

Amazon Region Code Values	Region Names
ap-northeast-1	Asia Pacific (Tokyo) region
ap-southeast-1	Asia Pacific (Singapore) region
ap-southeast-2	Asia Pacific (Sydney) region
eu-west-1	EU (Ireland) region
sa-east-1	South America (Sao Paulo) region
us-east-1	US East (N. Virginia) region
us-west-1	US West (N. California) region
us-west-2	US West (Oregon) region

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Verify that an AWS user account is set up with access keys.
- Create virtual machine templates that meet virtual private cloud (VPC) requirements. See [Virtual Machine Requirements for Creating Amazon EC2 Custom Templates](#).

Procedure

- On the Application Services title bar, click the drop-down menu and select **Clouds > Cloud Providers**.
- Click **New**.

3 Complete the cloud provider information.

Option	Description
Name and Description	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.
Cloud Provider Type	Amazon EC2 is the supported type.
Amazon Region	Specify the Amazon Region code value. For example, if your region is US East (N. Virginia), enter the Amazon Region code us-east-1 . Entering an unsupported value prevents the Availability Tunnel from displaying content when you attempt to create a new deployment environment.
Proxy URL	If your corporate network requires a proxy server for external connections, add the Web proxy URL to enable the virtual private cloud (VPC) server to connect to Amazon EC2.
User Name and Password	Access keys set up on the AWS account.

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

5 Click **Save**.

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**.

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** and make your changes.

11 Click **Save**.

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

What to do next

Create a deployment environment to deploy an application to the cloud environment. See [Create an Amazon EC2 Deployment Environment](#).

Map these AMIs to the Application Services logical templates. See [Add a Logical Template to the Library](#).

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 Application Services.

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

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Verify that the Amazon EC2 environment is configured to establish a secure connection with the Application Services appliance. See [Configure Amazon EC2 Environment for Application Services](#).
- Verify that at least one Amazon cloud provider is registered in Application Services. See [Register the Amazon EC2 Cloud Provider and Template](#).
- Verify that a secure cloud tunnel connection is available.

Procedure

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

Option	Description
Deployment Environment and Description	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.
Cloud Provider	If the cloud provider does not appear in the list, cancel the dialog box and select Clouds > Cloud Providers to add the Amazon EC2 cloud provider.
Virtual Private Cloud	Click Select to view a list of available Amazon VPCs in the Region.
Availability Zone	Click Select to view a list of Availability Zones in the Amazon Region.
Cloud Tunnel	Select the cloud tunnel instance that connects the Application Services 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 Validate to check the secure SSH connection status of the cloud tunnel.

- 4 Click **Save**.
- 5 To change existing information for an Amazon EC2 deployment environment, click **Edit** and make your changes.
- 6 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](#), [Create a Policy Instance](#) or [Create a Puppet Solution Instance](#).

Developing Application Services Components

12

Application Services provides an open framework to create and develop components that can be installed in a virtual machine.

The Application Services components in the library 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 Application Services library in a test environment.

Familiarize yourself with the key concepts that appear frequently in topics about developing deployable components. See [Key Concepts](#).

This chapter includes the following topics:

- [Defining Component Actions](#)
- [Configuring Component Properties](#)
- [Application Services Predefined Properties](#)
- [Secured Component Properties](#)
- [Required Component Properties](#)
- [Best Practices for Developing Components](#)

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 application 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 *Using Application Services Library 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 Application Services 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](#).

You can add a service or custom task in an application deployment and define the component action. See [Add a Service to the Library](#) and [Add a Custom Task to the Library](#).

Supported Action and Custom Task Scripts

Application Services 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](#).

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](#).

Table 12-1. Action Script Exit and Return Codes

Script Type	Description
Bash	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> .
Windows CMD	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.
Windows PowerShell	You can use <code>exit 0</code> to indicate success status and <code>exit non-zero</code> for error status.
BeanShell	You can use <code>System.exit(0);</code> to indicate success status and <code>System.exit(1);/non-zero</code> for error status.

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 Application Services 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 Application Services agent in the virtual machine communicates with the Application Services 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](#)

Application Services supports string, array, content, computed, boolean, single select, integer, and double properties.

- [Defining Property Values](#)

An application catalog administrator can define properties in the library for services and custom tasks.

- [Binding to Other Properties](#)

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

- [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.

- [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.

Types of Properties

Application Services supports string, array, content, computed, boolean, single select, integer, and double 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	Script Syntax	Sample Usage
admin_email = "admin@email987.com"	Bash - \$admin_email	echo \$admin_email
	Windows CMD - %admin_email%	echo %admin_email%
	Windows PowerShell - \$admin_email	write-output \$admin_email
	BeanShell - admin_email	print(admin_email);

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.

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

Content Property

The content property value is a URL to a file to download content. Application Services 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, suppose Hyperic artifacts are hosted in the Application Services appliance and the URL points to that location in the appliance. The Application Services agent downloads the artifacts from the specified location into the deployed virtual machine.

Sample String Property	Script Syntax	Sample Usage
HQ_PACKAGE = "http://DarwinServerIP/artifacts/services/hyperic/hyperic-hq-agent-linux.tar.gz"	Bash - <code>\$HQ_PACKAGE</code>	<pre>tar -zxvf \$HQ_PACKAGE</pre>
	Windows CMD - <code>%HQ_PACKAGE%</code> %	<pre>start /wait c:\unzip.exe %HQ_PACKAGE%</pre>

Sample String Property	Script Syntax	Sample Usage
	Windows PowerShell - \$HQ_PACKAGE	& c:\unzip.exe \$HQ_PACKAGE
	BeanShell - HQ_PACKAGE	<pre> import java.io.BufferedOutputStream; 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 BufferedOutputStream(new FileOutputStream(filePath)); 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(); </pre>

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	Script Syntax	Sample Usage
my_unique_id = ""	Bash - \$my_unique_id	export my_unique_id="0123456789"
	Windows CMD - %my_unique_id%	set my_unique_id=0123456789
	Windows PowerShell - \$my_unique_id	\$my_unique_id = "0123456789"
	BeanShell - my_unique_id	my_unique_id = "0123456789";

Boolean Property

The boolean property type provides Yes and No choices in the Value drop-down menu, but the value you select changes to True or False when you click on another part of the page. Application Services displays the Yes and No values to avoid confusion and provide consistency with the vRealize Automation console interface, but changes Yes to True and No to False for use in action scripts.

Single Select Property

The single select property lets you specify multiple string values from which the user can select only one value.

Integer Property

The integer property type accepts zero, a positive integer, or a negative integer as a value.

Double Property

The double property type stores floating point values with precision to four places.

Property Type Reference

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

Property Type	Property Type to Bind	Auto-Bind Expose	Auto-Bind Consume	Accept Initial Property Value
String	String, Array, Content, and Computed	Yes	Yes	Yes
Content	N/A	N/A	N/A	Yes
Array	String, Content, and Computed	N/A	Yes	Yes
Computed	N/A	Yes	N/A	N/A

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](#). To understand the Auto-Bind expose and Auto-Bind consume concepts, see [Auto-Binding to Other Properties](#).

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

Sample Property Type	Property Type to Bind	Binding Outcome (A binds to B)
String (property A)	String (property B="Hi")	A="Hi"
String (property A)	Content (property B="http://my.com/content")	A="http://my.com/content"
String (property A)	Array (property B=["1","2"])	A=["1","2"]
String (property A)	Computed (property B="Hello")	A="Hello"

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

Sample Property Type	Property Type to Bind	Binding Outcome (A binds to B)
Array (property A)	String (property B="Hi")	A="Hi"
Array (property A)	Content (property B="http://my.com/content")	A="http://my.com/content"
Array (property A)	Computed (property B="Hello")	A="Hello"

Defining Property Values

An application catalog administrator can define properties in the library for services and custom tasks.

To allow for customization of a component script, the default value for a property in the library 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 application 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-overridable by default, but the application catalog administrator can choose to enable the overridable feature. For properties that can be overridden, the application architect might force a value for the property in some regulated environments, and disable the overridable 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 Library](#) and [Add a Custom Task to the Library](#).

Binding to Other Properties

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

The application 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, Application Services 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](#).

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, Application Services provides either a *NodeName:NICx_ip* or *self:NICx_ip* property. Where *x* reflects the NIC number. See [Predefined IP Address Property](#).

Figure 12-1. Binding to Another Property in a Clustered Node

Edit Property for Apache_LB

Name: use_ajp

Description: Enable this when configuring proxy to Tomcat instance that is using AJP. Set to YES or NO.

Type: String

Library Value: NO

Blueprint Value: Use library value
 Type your own value
 bind to another property in this blueprint
 bind to an artifact

Auto-Bind Type: None

Auto-Bind Tags:

Required Secured Overridable at Deployment

Save Cancel

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 application catalog administrator, the Auto-Bind type appears 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 application 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 Library](#).

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 12-2. Preconfigured WAR Application Component

Application Component	Property	Auto-Bind Type	Auto-Bind Tag
Linux WAR file	Context	Expose	Context
Linux WAR file	service_start	Consume	Servlet Container, Start

Table 12-2. Preconfigured WAR Application Component (Continued)

Application Component	Property	Auto-Bind Type	Auto-Bind Tag
Linux WAR file	service_stop	Consume	Servlet Container, Stop
Linux WAR file	webapps_dir	Consume	Servlet Container, DIR

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 12-3. Predefined Node and Node Array Properties

Node Type	Property	Auto-Bind Type	Auto-Bind Tag
Node, Node Array	hostname	Expose	Hostname
Node, Node Array	ip	Expose	IP Address
Node Array	node-array-index	Expose	Node Array Index

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 12-4. Property Type and Node Type Compatibility

Source Property Type	Target Property Type	Node Type	Compatible
Array	String	Node	Yes
Array	Computed	Node	Yes
Array	String	Node Array	Yes
Array	Computed	Node Array	Yes
String	String	Node	Yes
String	Computed	Node	Yes
String	String	Node Array	No
String	Computed	Node Array	No

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 `progress_status="completed"`
- Windows CMD `set progress_status=completed`
- Windows PowerShell `$progress_status="completed"`
- BeanShell `progress_status="completed"`

Note Array and content property do not support passing modified property values between action scripts of life cycle stages and components.

Application Services Predefined Properties

Application Services 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.

The following IP addresses are available in the `nodeName:NICx_ip` properties, where `-x-` is the NIC number.

In the sample Clustered Dukes 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 machine 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](#).

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 Application Services.

If a property is changed from secured to unsecured, for security purposes, Application Services 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 library, blueprint, or deployment profile. For example, an application catalog administrator can define a required property, mark it as overridable, and not set any value for the property in the library. 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 library, as they include a number of examples of how to define properties and action scripts.

Note Use the sample components in the library only in a test environment.

Follow these best practices when developing components in Application Services.

- 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 Dukes 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 Application Services server.

Managing the Application Services Library

13

With the prepopulated, extensible library 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 Application Services includes many predefined logical templates, services, and external services. An application 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 library are available to all user groups in Application Services. You must add logical templates for each group outside the Default group.

You can use the Application Services library 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 library. See [Key Concepts](#).

This chapter includes the following topics:

- [Add Operating Systems to the Library](#)
- [Add Tags to the Library](#)
- [Add a Service to the Library](#)
- [Import a Puppet Service to the Library](#)
- [Add an External Service to the Library](#)
- [Add a Logical Template to the Library](#)
- [Add a Policy to the Library](#)
- [Add a Custom Task to the Library](#)

Add Operating Systems to the Library

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

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

Note You cannot delete an operating system from the library.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Familiarize yourself with the preinstalled operating systems on the Operating Systems page.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Operating Systems**.
- 2 Click **New**.
- 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.

Option	Description
x86_32	Use for 32-bit operating systems.
i386	Use for 32-bit operating systems.
x86_64	Use for 64-bit operating systems.

- 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 library or to an application blueprint.

Add Tags to the Library

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 library.

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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Familiarize yourself with the preinstalled tags listed on the Tags page.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Tags**.
- 2 Click **New**.
- 3 Enter 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**.

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 library or to an application blueprint.

Add a Service to the Library

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

Procedure

- 1 [Create a Service Version in the Library](#)

Use the Application Services library to create software service versions. The Application Services library also contains predefined services.

2 Define Service Version Properties

Application Services 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

The application 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

You can copy or edit a service from an existing service version.

5 Delete a Service from the Library

You can delete a service from the **Services** page.

Create a Service Version in the Library

Use the Application Services library to create software service versions. The Application Services library 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](#).

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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify whether you must add items to the operating systems and descriptive tag lists. See [Add Operating Systems to the Library](#) and [Add Tags to the Library](#).
- 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 **Library > Services** to view the predefined services.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Services**.

2 Click **New**.

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 **Library > 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, Application Services 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 **Library > 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

Configure the service version properties. See [Define Service Version Properties](#).

Define Service Version Properties

Application Services 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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- To view examples about how to define a service, select **Library > Services** to view the predefined services.
- Verify that a service version is available in Application Services. See [Create a Service Version in the Library](#).

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 Enter 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 vRealize Automation, 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.

Option	Description
None	The property does not Auto-Bind.
Consume	The property value is set with autobinding.
Expose	The property can be used to customize another property value with autobinding.

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.

Property Type	Cannot Set Auto-Bind Type to
Array	Expose
Computed	Consume
Content	Expose or Consume
Single Select	Expose or Consume

- 7 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.

- 8 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**.

- 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, Application Services 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.

- 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 application 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 service property definitions.

What to do next

Create action scripts for the service version. See [Add Action Scripts to the Service Version](#).

Add Action Scripts to the Service Version

The application 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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- 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 12 Developing Application Services Components](#).
- To view examples about how to add scripts to a service, select **Library > Services** to view the predefined services.
- Verify that properties are defined for the service version. See [Define Service Version Properties](#).

Procedure

1 Open an existing service version and click **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.

An application 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 **Reboot** check box to restart the virtual machine after the script runs successfully, during an application deployment.

7 Click **OK**.

- 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 **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 **Save**.

- 11 Click **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](#).

Maintain Service Versions

You can copy or edit a service from an existing service version.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- 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 Library](#).
- 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 12 Developing Application Services Components](#).
- Create a copy of the service to preserve the original for future reference and edit the copied version.
- You must be a member of the owning business group for the service version.

If you are not a member of the business group that owns the service version, the **Copy** and **Edit** buttons are disabled.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > 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 Library](#).

Delete a Service from the Library

You can delete a service from the **Services** page.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.

- 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.

- You must be a member of the owning business group for the service version.

If you are not a member of the business group that owns the service version, the **Delete** button is disabled.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > 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 Library

Puppet services can be used to build application blueprints. You must import Puppet modules and defined resources into the Application Services library as services.

You can import multiple Puppet modules at the same time.

After you import the Puppet modules, these services have a special tag called Puppet Services. The newly imported services do not have action scripts. Application Services 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 library. The default version is 1.0.0-unknown for Puppet modules with missing version information.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Start the Application Services CLI. See [Start the CLI Remotely](#).
- 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

- ◆ After you log in to the CLI program and in the `roo` shell, enter the command to import the Puppet module such as Apache.

```
import-puppet-manifest --typeFilter "^apache$" --targetGroup pubs
```

The parameters in the command are case sensitive. If you use the incorrect case, the command is ignored.

Parameter	Description
<code>typeFilter</code>	Limits the import to types matching the specified search expression. For details about expression syntax refer to the Puppet documentation or type <code>puppet man resource_type</code> to view the man pages. For example, to import the <code>mysql</code> module, type <code>import-puppet-manifests --typeFilter "^mysql\$"</code> .
<code>osfamily</code>	Specifies the operating system for the imported type. The options are <code>WINDOWS</code> and <code>LINUX</code> . Linux is the default operating system.
<code>puppetPath</code>	Specifies the location of the Puppet binary file.
<code>shared</code>	Indicates whether the artifact is shared or private. You can define <code>true</code> or <code>false</code> in the command. The default setting <code>false</code> indicates that the artifact is private to the user's group performing the import operation.
<code>targetGroup</code>	Specifies the business group for the imported manifests. This is the owning business group of the service.

The command imports the `apache` Puppet module to the library.

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](#) and [Setting Up and Configuring a Deployment Profile](#).

Add an External Service to the Library

In the Application Services library, 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 Library](#)

With Application Services you can create basic or advanced external services.

2 [Define External Service Properties](#)

Application Services 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](#)

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](#)

You can copy a basic or advanced external service from an existing version or modify the service.

5 [Delete an External Service](#)

To manage your external services you can delete duplicate or outdated services.

Create an External Service in the Library

With Application Services 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 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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify whether you must add items to the operating systems and descriptive tag lists. See [Add Operating Systems to the Library](#) and [Add Tags to the Library](#).
- To view examples about how to name or describe an external service, select **Library > External Services** to view the predefined external services.

Procedure

1 On the Application Services title bar, click the drop-down menu and select **Library > External Services**.

2 Click **New**.

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 **Library > 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, Application Services 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 **Library > Operating Systems** to create an operating system name.

What to do next

Configure the external service version properties. See [Define External Service Properties](#).

Define External Service Properties

Application Services 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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.

- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- To view examples about how to define an external service, select **Library > External Services** to view the predefined services.
- Verify that an external service version is available in Application Services. See [Create an External Service in the Library](#).

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 Enter 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 vRealize Automation, 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.

Option	Description
None	The property does not Auto-Bind.
Consume	The property value is set with autobinding.
Expose	The property can be used to customize another property value with autobinding.

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, Application Services resets the property value, for security purposes. You must set a new value for the property.

- 9 From the Override drop-down menu, select a property override option.

Option	Description
Blueprint	Property is overridable in the application blueprint. The cloud admin cannot view this property when registering the external service Instance to a deployment environment.
Deployment Environment	Cloud admin can override the property in the deployment environment. The property appears in the application blueprint canvas and user can Auto-Bind to a property. The property value is not editable in the application blueprint.
Not Overridable	Property is not overridable.

- 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 application 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](#).

Create provider specification for the custom external service version. See [Create Provider Specification for an Advanced External Service Version](#).

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 view examples about how to create a provider specification for an external service from the **Library > External Services** page.

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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).

- 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.
- Verify that properties are defined for the external service version. See Application Services. See [Create an External Service in the Library](#).

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.
An application 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**.
- 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](#).

Add the external service to your application blueprint. See [Add an External Service to an Advanced Blueprint](#).

Update external services or copy an existing external service. See [Maintain External Service Versions](#).

Maintain External Service Versions

You can copy a basic or advanced external service from an existing version or modify the service.

You can view examples about how to define an external service from the **Library > External Services** page.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that an external service version is available in Application Services. See [Create an External Service in the Library](#).
- Create a copy of the external service to preserve the original for future reference and edit the copied version.
- You must be a member of the owning business group for the external service version.

If you are not a member of the group that owns the external service version, the **Copy** and **Edit** buttons are disabled.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > 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 Library](#).

Delete an External Service

To manage your external services you can delete duplicate or outdated services.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that the external service is not used in a blueprint. See [Add an External Service to an Advanced Blueprint](#).
- Verify that you are a member of the owning business group for 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 Application Services title bar, click the drop-down menu and select **Library > External Services**.
- 2 Open the existing external service.
- 3 Click **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 Library

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, vRealize Automation, 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 Application Services user interface session expires after 30 minutes of inactivity. If the session expires, any changes that were not saved are lost.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Create cloud templates in vCloud Director, vRealize Automation, or Amazon EC2 that meet the requirements for working properly in Application Services. See [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#), [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#), or [Virtual Machine Requirements for Creating Amazon EC2 Custom Templates](#).
- Verify that at least one cloud provider is registered in Application Services. See [Register the vCloud Director Cloud Provider and Template](#), [Register the vRealize Automation Cloud Provider and Template](#), or [Register the Amazon EC2 Cloud Provider and Template](#).
- 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 Application Services to Use a Proxy for External URLs](#).
- Verify whether you must add items to the operating systems and descriptive tag lists. See [Add Operating Systems to the Library](#) and [Add Tags to the Library](#).
- 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 Library](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Logical Templates**.
- 2 Click **New**.

- 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.

Option	Description
Name	Logical template version name stays the same.
Version	Add a qualifier to denote major and minor versions such as 1.2.1 or 1.0.1-CentOS56-32bit .
Description	Add detailed information about the logical template. For example, describe a script you are using or the amount of disk size required.
Tags	<p>Categorize logical templates based on the functions that they provide. Application Services organizes the templates based on tags you see when you create a deployment blueprint for an application.</p> <p>Note Only tags designated as Server Type appear in the drop-down menu.</p> <p>You can add multiple tags.</p> <p>To use a tag that is not in the list, click Cancel, and select Library > Tags to create a tag.</p>
Supported OS	<p>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, Application Services prevents you from adding that service.</p> <p>To use an operating system name that is not in the list, click Cancel, and select Library > Operating Systems to create an operating system name.</p>

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 **Library > 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](#).

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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Create cloud templates in vCloud Director, vRealize Automation, or Amazon EC2 that meet the requirements for working properly in Application Services. See [Virtual Machine Requirements for Creating vCloud Director Custom Templates](#), [Virtual Machine Requirements for Creating vRealize Automation Custom Templates](#), [Virtual Machine Requirements for Creating Amazon EC2 Custom Templates](#).
- 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 Application Services to Use a Proxy for External URLs](#).
- Verify whether you must add items to the operating systems and descriptive tag lists. See [Add Operating Systems to the Library](#) and [Add Tags to the Library](#).
- If you plan to have application or services preinstalled in a template, create and configure the services. See [Add a Service to the Library](#).
- Verify that you are a member of the business group that owns the logical template version.

If you are not part of the business group that owns the logical template version, the **Copy** and **Edit** buttons are disabled.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > 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 library. See [Delete a Logical Template from the Library](#).

Delete a Logical Template from the Library

You can delete a logical template from the **Logical Templates** page.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that the logical templates in application blueprints and all of the other objects referencing the template are deleted.
- Verify that you are a member of the owning business group for the logical template version.

If you are not part of the business group that owns the logical template version, the **Delete** button is disabled.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Logical Templates**.
- 2 Open the logical template.
- 3 Click **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 Library

You can create policy definitions to capture a custom set of rules that assess the compliance state of an application deployment.

Application Services 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.

Policies are visible to all business groups in their tenant, but are not visible in other tenants.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- To view examples about how to name or define a policy, select **Library > Policies** to view the predefined policy.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Policies**.

- 2 Click **New**.

- 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, Application Services 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](#).

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. Application Services 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 Application Services 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.

Script Input	
Variable	Description
var min_cpu_count	Corresponds to min_cpu_count property and the value for the script is set to consume.
eventPayload	<p>Includes the details of the deployment assessed for policy compliance.</p> <p>For regular properties, var eventPayload must be added to access the eventPayload object.</p> <p>The policy script might regard the eventPayload variable as a java object with the following properties:</p> <p>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.</p> <p>blueprint of type Blueprint as defined in V2 API Represents the actual blueprint object that's referenced from deploymentProfile.</p> <p>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.</p> <p>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.</p>

Policy scripts are expected to output the following properties to communicate the result of the compliance assessment. Scripts must declare them as variables.

Script Output Variable	Description
complianceResult	<p>Type of string that is mandatory. If script fails to set it then the compliance result is assumed as an Error.</p> <p>The valid values for the variable are:</p> <p>Compliant Indicates that a deployment is compliant against the policy being assessed.</p> <p>Non_Compliant Indicates that a deployment violates the policy being assessed.</p> <p>Error Indicates failure to produce an assessment result.</p>
complianceMessage	<p>Type of string.</p> <p>This optional value provides a high-level summary of the reason behind policy violation. Value can be any string with less than 2048 characters.</p>

The scripts can create log messages with the standard println function available in Java Script. The log is captured by Application Services, which is useful to diagnose errors in policy scripts or provide details for a policy assessment result.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that at least one policy is created in the library. See [Add a Policy to the Library](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > 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**.

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](#).

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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- To view examples about policy definition and action, select **Library > Policies** to view the predefined policies.
- Verify that a policy version is available in the Application Services library. See [Add a Policy to the Library](#).
- Create a copy of the policy to preserve the original for future reference and edit the copied version.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Policies**.
- 2 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.

- 3 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](#).

Delete a Policy

If the policy definitions are not applicable to your application deployment, you can delete the policy from the Application Services library.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that the policy does not have existing policy instances in the deployment environment. See [Create a vCloud Director Deployment Environment](#), [Create a vRealize Automation Deployment Environment](#), or [Create an Amazon EC2 Deployment Environment](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Policies**.
- 2 Open the existing policy.
- 3 Remove the existing policy version and confirm your selection.
- 4 Click the arrow next to the policy name to return to the Policies page.
- 5 Delete the policy and confirm your selection.

Add a Custom Task to the Library

With Application Services, you can create a custom task to perform customized tasks in the application deployment such as run security patches.

The Application Services 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 Application Services user interface session expires after 30 minutes of inactivity. If the session expires, any changes that were not saved are lost.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.

- Verify whether you must add items to the operating systems and descriptive tag lists. See [Add Operating Systems to the Library](#) and [Add Tags to the Library](#).
- 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 Application Services to Use a Proxy for External URLs](#).
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Tasks**.
- 2 Click **New**.
- 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.

Option	Description
Name	Task version name stays the same.
Description	Add detailed information about the custom task. For example, describe what the script does when added to a node in the execution plan.
Version	You can specify major, minor, or micro releases, with or without qualifiers. For example, you might use unique version numbers such as 1.0 or 1.5.
Supported OSes	If the scripts used in this task can run only on particular operating systems, select those operating systems here. In the execution plan, Application Services 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 Library > Operating Systems .

- 6 In the Properties section, define the variables used for a custom task.

Option	Description
Property Name	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.
Description	Include details about the custom task.
Type	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.

Option	Description
Value	Type the value to substitute for this property when the custom task runs. For example, for a property called <code>remove_all</code> , you might type either <code>true</code> or <code>false</code> in this text box.
Secured	Select the check box for passwords you define or other properties whose values you want to obscure. For example, the <code>JBOSS_JMX_PWD</code> property in the JBoss service is secured. If a property is changed from Secured to Unsecured, Application Services resets the property value, for security purposes. You must set a new value for the property.
Delete button	Removes the selected row from the Properties section.

- 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](#). Modify the custom task to support your current needs. See [Maintain Custom Task Versions](#).

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

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify whether you must add items to the operating systems and descriptive tag lists. See [Add Operating Systems to the Library](#) and [Add Tags to the Library](#).

- 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 Application Services to Use a Proxy for External URLs](#).
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > 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**.

Creating Applications

Application Services 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 library
You can use the Application Services sample templates or create custom templates. The cloud templates that contain operating system images and might have services preinstalled and configured are mapped to Application Services logical templates.
- Application infrastructure components and scripts from the library
The included sample services are reusable software components. Application Services 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.

Sharing

Application Services relies on vRealize Automation for

A logged-in Application Services user belongs to one or more business groups. One or more business groups belong to a single tenant. One or more tenants

Note For information about deleting an application deployment from Application Services, see [Delete an Application Deployment from Application Services](#).

Familiarize yourself with the key concepts that appear frequently in topics about creating applications. See [Key Concepts](#).

This chapter includes the following topics:

- [Create an Application](#)
- [Create an Application Version](#)
- [Model an Application Blueprint](#)
- [Configure an Application Blueprint](#)

- [Creating an Advanced Blueprint](#)
- [Copy an Application Version](#)
- [Copy an Application](#)
- [Delete an Application Version](#)

Create an Application

With Application Services, you can model your application deployment, create dependencies, and edit the application configurations.

You can specify applications as shared throughout the tenant or private to the owning business 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 the owning business group.

Prerequisites

- Log in to Application Services as an **application architect**.
- 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 business group.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Applications**.
- 2 Click **+ Create New Application**.
- 3 Enter the name of the application in the **Name** text box.
- 4 (Optional) Enter a description of the application in the **Description** text box.
- 5 Select the owning business group for the application in the **Business Group** drop-down menu.
- 6 Select the sharing status of the application in the **Sharing** drop-down menu.

Only members of the owning business group can access a private application. Members of all business groups in the tenant can access a shared application.

- 7 Click **Save**.

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.

Prerequisites

- Log in to Application Services as an **application architect**.

- 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 Application Services title bar, click the drop-down menu and select **Applications**.
- 2 Open an application.
- 3 Click **+ Create Application Version**.
- 4 Enter the major and minor version of the application in the **Version** text box.
For example, you might also add a qualifier to the major and minor versions such as **1.0.0-SNAPSHOT**.
- 5 (Optional) Enter a description of the application version in the **Description** text box.
- 6 Click **Save**.

The 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](#).

Model an Application Blueprint

The application blueprint provides detailed control over installation dependencies, configuration changes, and editable Application Servicesscripts. 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 Application Services library. On the right is a list of the services available from the library 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 Application Services user interface session expires after 30 minutes of inactivity. If the session expires, any changes that were not saved are lost.

Prerequisites

- Log in to Application Services as an **application architect**.
- Verify that at least one application version is created in Application Services. See [Create an Application Version](#).

Procedure

- 1 On the Application Services 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.

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, Application Services 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 an 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 application catalog administrator sets the supported operating systems and components for a library service. The application catalog administrator can also add to the list of operating systems and tags that are already available in the library. For example, the MySQL service in the library 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**.

What to do next

Configure the services and application components you added to the application blueprint. See [Configure an Application Blueprint](#).

Configure an Application Blueprint

You can configure the properties and action scripts of the services and application components to customize your application deployment.

Prerequisites

- Log in to Application Services as an **application architect**.
- 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 Application Services to Use a Proxy for External URLs](#).
- Familiarize yourself with the basic concepts of binding to another property if you plan to customize your application. See [Binding to Other Properties](#).
- Make sure that there are services and application components in the application blueprint. See [Model an Application Blueprint](#).

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 application 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. An application catalog administrator can edit a service script from **Library > 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 or artifact, select the property value or artifact name 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. Binding to an artifact lets you use a component without knowing its physical location.

- 4 When you are finished creating the blueprint for the application, click **Save**.

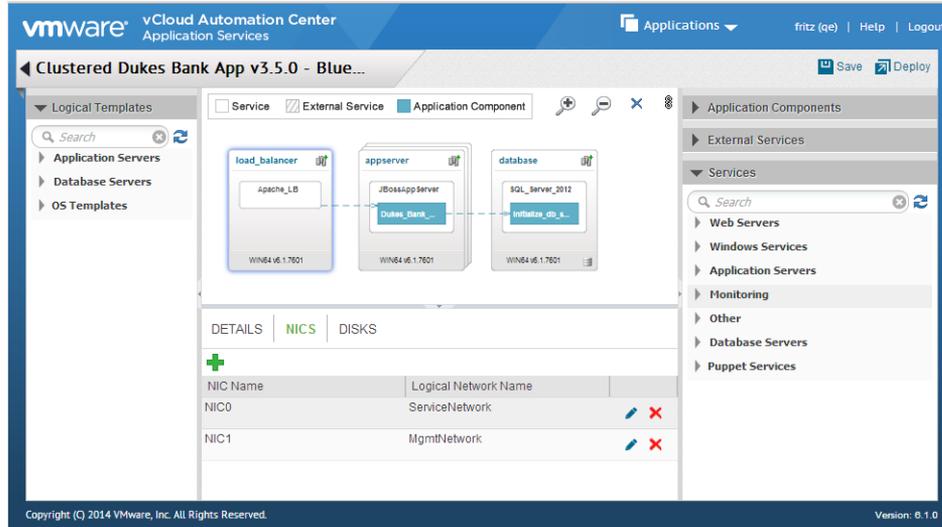
Application Services checks the application topology you created and displays a message box listing any errors. For example, you see a message if a property type that you selected is not compatible with a script type, or if a service or component is missing a required script. Some errors require correction before you can save the application.

What to do next

Create a deployment profile. See [Create a Deployment Profile](#).

Creating an Advanced Blueprint

With Application Services, you can create an advanced application blueprint with dependencies between components, clustered nodes, and multiple networks.

Figure 14-1. Clustered Windows Application Example

- [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.

- [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.

- [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.

- [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.

- [Add an External Service to an Advanced Blueprint](#)

You can add basic or advanced external services to a blueprint if your application needs the external services 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 **Add a Dependency** () in the toolbar at the top of 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 16 Deploying Applications](#).

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 12 Developing Application Services Components](#).

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 Dukes 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 16 Deploying Applications](#).

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 Application Services, 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 Dukes 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](#).

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 (Optional) To be part of the same network as another node, pick the network name from the drop-down menu.
- 4 Click a different part of the canvas or press Enter to finish adding the NIC.
- 5 Click **Save** in the toolbar above the canvas to save your changes.

What to do next

Deploy the application. See [Chapter 16 Deploying Applications](#).

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 Application Services 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](#)

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 14-1. Disk Options

Option	Description
Name	The unique disk name within the node.
Mount Path	The path for the disk to be mounted. Application Services does not mount the disk if the mount path is not specified or if the disk is not formatted.
File System	File systems such as ext3 or NTFS can be specified.
Size (GB)	The size of the disk to be provisioned.
Tags	A string specified that you can leverage to determine the mapping between disk and storage.
Description	Specify more information for the disk.

- (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 14-2. Disk Tags

Tag	Description
Archive	Stores database archive files.
Binary	Stores binary files, such as Oracle binary.
Data	Stores data files, such as Oracle table space files.
Log	Stores database online logs.

- When you are finished, click **Save** in the toolbar above the canvas.

What to do next

Deploy the application. See [Chapter 16 Deploying Applications](#)

Add an External Service to an Advanced Blueprint

You can add basic or advanced external services to a blueprint if your application needs the external services in order to work.

Prerequisites

- Familiarize yourself with the basic concepts of basic or advanced external services. See [Add an External Service to the Library](#).
- Understand the basic concepts of defining a service. See [Define Service Version Properties](#).
- Familiarize yourself with how to create dependencies among nodes. See [Create a Dependency Between Components](#).

Procedure

- Open an existing application blueprint.
- Select and drag an external service to the blueprint canvas.
For example, you can add the vPostgres external services in the blueprint.
- (Optional) Create dependencies among applicable components in the blueprint.
A blue dotted line appears and points to the dependent component.
- Click the property you want to edit.
You can edit or define the external service properties just like any other service in the library.
- Click **Save**.

What to do next

[Chapter 16 Deploying Applications](#).

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

- Log in to Application Services as an **application architect**.
- 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](#).
- Verify that you are a member of the business group that owns the application version.

If you are not a member of the business group that owns the application version, the **Copy** button is disabled.

Procedure

- 1 On the Application Services 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](#).

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

- Log in to Application Services as an **application architect**.
- 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](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Applications**.
- 2 From the Applications page, point to the application card view.
- 3 Click **Copy** () 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](#).

Delete an Application Version

If your application version becomes obsolete, you can remove it from the appliance.

When you delete an application version, the blueprints are removed. You cannot delete an application version with deployment profiles.

Prerequisites

- Log in to Application Services as an **application architect**.
- 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](#).
- Verify that you are a member of the business group that owns the application version.

If you are not a member of the business group that owns the application version, the **Delete** button is disabled.

Procedure

- 1 On the Application Services 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 Artifacts

Artifact management lets your organization deploy an application with build files and other types of software artifacts without regard to their physical location.

An artifact is a script or the output of a build process. With artifact management, an application blueprint specifies an artifact by type and name, but not by location or unique identifier. Artifact management monitors the physical location and identity of artifacts and supplies the required artifact during deployment.

In artifact management, an application catalog administrator creates artifact repository specifications and maps them to artifact repository instances. An artifact repository specification identifies a store that can contain an unlimited number of artifacts. It has properties that identify an artifact repository instance and the artifacts it contains. Application Services provides out-of-the-box artifact repository specifications for Jenkins and YUM, and the capacity to define additional artifact repository specifications.

An artifact repository specification has agent- and server-side scripts that can fetch artifacts from an artifact repository instance. A script can produce metadata and store it with an artifact. It can also track and search for an artifact. The script lifecycles are resolve artifact, download artifact, and find tracking IDs. You can write your own Bash, Windows Cmd, Windows Powershell, BeanShell, and JavaScript scripts.

An application architect and other application administrators create artifact specifications, map them to artifact repository instances such as actual Jenkins build projects, and then bind the artifact specifications to the properties of services, external service, or application components such as WAR or JAR files in the application blueprints. An artifact specification identifies an artifact only by name, description, and business group. The deployment environment you select determines the actual artifact repository instance, and the deployment profile you use specifies the actual artifact. You can map an artifact to one repository instance per deployment environment.

This chapter includes the following topics:

- [Create an Artifact Repository Specification](#)
- [Create an Artifact Repository Instance](#)
- [Create an Artifact Specification](#)
- [Map an Artifact Specification to an Artifact Repository Instance](#)
- [Bind an Artifact to an Application Blueprint](#)

- [Deploy an Application with Artifacts](#)

Create an Artifact Repository Specification

An artifact repository specification is a template for an artifact repository instance.

An artifact repository specification defines properties to access an artifact repository instance and identify the artifacts it contains. The specification contains an action script that uses the property values to resolve the artifact repository instance and the artifact during deployment.

Prerequisites

- Log in to Application Services as an **application architect** or an **application catalog administrator**.
- Verify that you are a member of the appropriate business group.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Artifact Repositories**.

- 2 Click the Add icon ()

- 3 Enter a name and description for the artifact repository specification.

- 4 Enter a tracking identifier to identify the physical artifact in the repository.

Use a unique string to serve as a build ID, job ID, commit ID, version number, or other identifier. The system uses this identifier to locate different versions of the same artifact in the same repository. You can use this identifier in action scripts.

- 5 Select a business group from the **Business Group** drop-down menu.

Only business group users can access the artifact repository instances created from this specification.

- 6 Add a repository access property.

A repository access property specifies the server URL and any other information required to locate an artifact repository instance.

- a Click the Add icon () for Repository Access Properties.

You can use an empty row in the Repository Access Properties list instead, if available.

- b Enter a property name in the **Property Name** text box.

- c (Optional) Enter a description in the **Description** text box.

- d Select a property type from the **Type** drop-down menu.

Type	Description
String	Specifies unique identifier for repositories created from this specification. Accepts a maximum of 1024 alphanumeric characters.
Integer	Specifies single integer with positive or negative value.
Content	Specifies URL of object to be downloaded as an artifact by scripts.
Array	Specifies an array in the form "value1","value2",
Computed	Specifies computed value to be provided by scripts
Boolean	Specifies YES or NO.
Double	Specifies a double integer.
Single Select	Defines a single select statement.

- e (Optional) Enter a property value in the **Value** text box.

Enter a value here only if you want to use the same value for every artifact repository instance created from the specification.

- f (Optional) Select the **Secured** check box.

When selected, hides the value of the property by masking the characters.

7 Add an artifact identifier property.

- a Click the Add icon () for Artifact Identifier Properties.

You can use an empty row in the Artifact Identifier Properties list instead, if available.

- b Enter a property name in the **Property Name** text box.
- c (Optional) Enter a description in the **Description** text box.
- d Select a property type from the **Type** drop-down menu.

Type	Description
String	Specifies unique identifier for repositories created from this specification. Accepts a maximum of 1024 alphanumeric characters.
Integer	Specifies single integer with positive or negative value.
Content	Specifies URL of object to be downloaded as an artifact by scripts.
Array	Specifies an array in the form "value1","value2",
Computed	Specifies computed value to be provided by scripts
Boolean	Specifies YES or NO.
Double	Specifies a double integer.
Single Select	Defines a single select statement.

- e (Optional) Enter a property value in the **Value** text box.

Enter a value here only if you want to use the same value for every artifact in an artifact repository instance.

- f (Optional) Select the **Secured** check box.

When selected, hides the value of the property by masking the characters.

- 8 Create a script to resolve the location of an artifact.

- a Select the script type from the **Script Type** drop-down menu for the RESOLVE_ARTIFACT script.

- b Click in the **Script** text box to edit the script.

- c Enter the script text in the edit window.

You can change the script type or select a property to insert with the drop-down menus located at the top of the edit window.

- d Click OK to close the edit window and save the script.

The script text appears in the Script text box.

- 9 Click **Save** to save your changes when finished.

What to do next

[Create an Artifact Repository Instance.](#)

Create an Artifact Repository Instance

You create an artifact repository instance by associating a physical repository with an artifact repository specification.

Prerequisites

- Log in to Application Services as an **application cloud administrator**.
- Verify that you are a member of the appropriate business group.
- Verify that a suitable artifact repository specification exists.
- Verify that a suitable deployment environment exists.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Clouds > Deployment Environments**.
- 2 Click the deployment environment you want to use.
The deployment details window appears.
- 3 Click the **Artifact Repository Instances** tab.

- 4 Click the Add icon (+).

The New Artifact Repository Instances page appears.

- 5 Select an artifact repository specification from the **Artifact** drop-down menu.
- 6 Enter a name in the **Name** text box.
- 7 (Optional) Enter a description in the **Description** text box.
- 8 Enter all of the required property values in the Properties section.
- 9 Click **Save**.

What to do next

[Create an Artifact Specification.](#)

Create an Artifact Specification

An artifact specification describes a logical artifact located in one or more artifact repository instances. An artifact specification does not provide any physical details.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that you are a member of the appropriate business group.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Artifacts**.
- 2 Click the Add icon (+) for Artifacts.
- 3 Enter the name of the artifact in the **Name** text box.
- 4 (Optional) Enter a description in the **Description** text box.
- 5 Select a business group from the **Business Group** drop-down menu.
Only users of the business group can access the artifact.
- 6 Click **Save** when finished.

What to do next

[Map an Artifact Specification to an Artifact Repository Instance.](#)

Map an Artifact Specification to an Artifact Repository Instance

When you map an artifact specification to an artifact repository instance, you specify where to find the artifact during deployment.

You can also change a current mapping of an artifact repository instance to an artifact specification.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that you are a member of the appropriate business group.
- Verify that a suitable artifact specification exists. See [Create an Artifact Specification](#).
- Verify that a suitable artifact repository instance exists. See [Create an Artifact Repository Instance](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Library > Artifacts**.
- 2 Click the Add icon () for Artifact Repositories.
The Repositories for an artifact window appear.
- 3 Select an artifact name from the **Artifact Name** drop-down menu.
Any artifact repository instances already mapped to the artifact specification appear in the Artifact Repository Instances list.
- 4 Click the Add icon () for Artifact Repository Instances.
A new row with a drop-down menu appears in the list.
- 5 Select an artifact repository instance from the drop-down menu.
The property values for the selected instance appear in the Artifact Identifier table.
- 6 (Optional) Enter a new value for a property to replace the current value, if required.
- 7 (Optional) Change the current mapping of an artifact repository instance to an artifact specification.
 - a Select the artifact specification in the **Artifact Name** drop-down menu.
 - b Select the artifact repository instance you want to change.
 - c Select a different artifact repository instance.
- 8 Click OK.

What to do next

[Bind an Artifact to an Application Blueprint](#)

Bind an Artifact to an Application Blueprint

When you bind an artifact to an application blueprint, you actually bind an artifact specification to a property of an application component, service, or external service in the blueprint.

Prerequisites

- Log in to Application Services as an **application architect**.
- Verify that you are a member of the appropriate business group.
- Verify that a suitable mapping of an artifact specification with an artifact repository instance exists. See [Map an Artifact Specification to an Artifact Repository Instance](#).
- Verify that an application and application blueprint exist. See [Create an Application](#) and [Configure an Application Blueprint](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Applications**.
- 2 Open an application, an application version, and then a blueprint.
- 3 Select an existing application component, service, or external service, or add a new one.
- 4 Click the **Properties** tab.
- 5 Locate the property to bind to the artifact.
 For example, a JAR file has the `jar_file` property where you can specify a downloadable file URL. Alternatively, you can generate and bind a unique text string as a package name for installing packages from a yum repository.
- 6 Click the Edit icon for the property.
 The Edit Property dialog box appears.
- 7 Select an artifact name from the **Blueprint Value** drop-down menu labeled bind to an artifact.
- 8 Click **Save** when finished.
- 9 Click **OK** when prompted to save or discard blueprint changes.

What to do next

[Deploy an Application with Artifacts](#).

Deploy an Application with Artifacts

You can deploy an application with one or more artifacts as application components.

The deployment environment you select for a deployment profile determines which artifact repository instance to use.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that you are a member of the appropriate business group.
- Verify that a deployment environment with an artifact repository instance exists. See [Create an Artifact Repository Instance](#).
- Verify that an application and application version exist.
- Verify that an application blueprint with an artifact binding exists. See [Bind an Artifact to an Application Blueprint](#).
- Verify that a deployment profile exists.
- Verify access to the application properties of the deployment profile. See [Configure Application Properties](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Applications**.
- 2 Open the application, application version, and deployment profile in order.
- 3 Open the Application Properties page in the Deployment Profile wizard.
- 4 Click the tab for the type of object mapped. For example, click **Application Component** or **Service** to display the corresponding properties.
- 5 Locate the artifact repository and artifact names in a **New Value** text box.
The values have Repository: and Artifact: prefixes.
- 6 Click the names in the **New Value** text box.
The **Value for Property** dialog box appears.
- 7 (Optional) You can enter the value for the tracking identifier defined in the artifact repository specification.
The tracking identifiers for the out-of-the-box artifact repository specifications are JOB_ID for a Jenkins build and VERSION for a Yum build. Other specifications can have other tracking identifiers. You can leave the text box blank and enter the tracking identifier later when you deploy, either returning here to enter the tracking identifier, or using Quick Deploy on the application.
- 8 (Optional) You can also override artifact management for the deployment. Specify the URL for the application component in the type your own value text box. The URL then takes precedence.
- 9 Click OK.

Deploying Applications

Application Services simplifies and automates deployments of multi-tier enterprise applications in hybrid cloud environments.

With Application Services, you can create different deployment profiles for a life cycle, such as development, testing, and production. You can add deployment profiles to a composite deployment profile, bind their properties, and deploy them together and in order. The deployment profile settings are saved in Application Services. You can reuse a saved deployment profile to deploy an application version to a supported cloud environment.

If you update a deployed application, Application Services creates an update profile based on the modified values. This update profile includes an update script generated from the defined update life cycle stage.

To familiarize yourself with the key concepts relating to deploying applications, see [Key Concepts](#).

This chapter includes the following topics:

- [Setting Up and Configuring a Deployment Profile](#)
- [Deploying with Deployment Profiles](#)
- [Publishing Deployment Profiles to the vCloud Automation Center Service Catalog](#)
- [Request a vRealize Automation Service Catalog Item](#)
- [Using the Deployment Summary Page](#)
- [Using the Composite Deployment Summary Page](#)
- [Understanding the Deployment and Update Process](#)
- [Understanding Deployment Failures](#)

Setting Up and Configuring a Deployment Profile

Deployment profiles let you configure settings such as cloud templates, networks, and application property values that are allowed for use in specific deployment environments.

In the deployment profile, you can review the execution plan, add custom tasks to the execution plan, review the deployment profile settings, and make changes before you deploy the application.

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

- Log in to Application Services as an **application publisher and deployer**.
- Verify that you are a member of the appropriate business group.
- Verify that at least one application is created in Application Services. See [Chapter 14 Creating Applications](#).

Procedure

- 1 On the Application Services 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 Add icon (+).
A selection menu appears.
- 4 Select Create Deployment Profile from the menu.
The menu also contains Create Composite Deployment Profile which lets you create a cluster of existing deployment profiles.
- 5 Enter a name for the deployment profile.
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.
- 6 If creating a deployment profile for a shared application, select a business group from the drop-down menu.
You cannot select a business group for a private application.
- 7 Click **Deploy**.

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](#).

Configure the Deployment Environment

You can 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 Application Services to override the vRealize Automation blueprint custom properties, mapping predefined 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, Application Services 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, Application Services 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 Application Services 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.

You can connect to external cloud networks by mapping a cloud template. To connect to external cloud network:

- Ensure that the network profiles should be of external type.
- Ensure that you assign it to the network path or create reservations.

Select the cloud network name from the drop-down list and choose a network map to the NICs defined in the nodes or clusters. Click the icon next to the cloud network name to view the network details.

You can optionally define a vRealize Automation blueprint to contain individual custom properties or collections of custom properties grouped as build profiles. These custom properties are applied to a virtual machine when it is created. Application Services lets you override the vRealize Automation blueprint custom properties or add to the existing properties. For example, to override the existing vRealize Automation 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.

For related information about defining and using custom properties on blueprints, see *Custom Properties Reference*.

Note Do not use the Application Services reserved and internal properties as your custom properties.

See [Application Services Reserved and Internal Properties](#).

Note While the **Extra Configuration** option remains available, it is recommended that you add custom properties to the machine blueprint in vRealize Automation, rather than add them using the **Extra Configuration** option. Custom properties that you add to the machine blueprint in vRealize Automation are visible as node properties when you request application provisioning from the vRealize Automation catalog. However, you can use the **Extra Configuration** option to add additional custom properties that are not present as part of the machine blueprint, including custom properties that are not part of the blueprint's associated build profiles or business group.

Prerequisites

- Log in to Application Services as an **application cloud administrator** and an **application publisher and deployer**.
- Verify that at least one cloud template is mapped to each logical template used in the blueprint. See [Add a Logical Template to the Library](#).
- Verify that at least one application is created in Application Services. See [Chapter 14 Creating Applications](#).
- 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](#) or [Create an Amazon EC2 Deployment Environment](#).
- Register your cloud template to a Application Services cloud provider. See [Register the vCloud Director Cloud Provider and Template](#), [Register the vRealize Automation Cloud Provider and Template](#) and [Register the Amazon EC2 Cloud Provider and Template](#).
- Familiarize yourself with custom properties for Application Services deployments and reserved custom properties. See vRealize Automation *Custom Properties Reference*.
- Verify that a deployment profile is created. See [Create a Deployment Profile](#).

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 Application Services. You can also ask your cloud administrator to register a cloud template for you.

For vRealize Automation, 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 vRealize Automation deployments, click the **Extra Configuration** icon (🔗) to add custom properties to each node in the application blueprint.

Do not use Application Services reserved and internal properties.

- 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 vRealize Automation 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 Application Properties](#).

Application Services 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.

Reserved and Internal Properties

Reserved Properties	Internal Properties
agent.download.url	Hostname
agent.jar.md5sum	VirtualMachine.Cpu.Count
agent.max.logsize	VirtualMachine.Memory.Size
agent.root.folder	VirtualMachine.DiskN.Size
amqp.heartbeat	VirtualMachine.Admin.CustomizeGuestOSDelay
amqp.host	VirtualMachine.Admin.UseGuestAgent
amqp.port	VirtualMachine.SoftwareN.Name
queue.name	VirtualMachine.SoftwareN.ScriptPath
server.url	
temp.key	
APPD_REQUEST_ID	
DEPLOYMENT_NAME	
DEPLOYMENT_URI	
DEPLOYMENT_LOCATION_URI	
VM_URI	
VM_NAME	

Reserved Properties	Internal Properties
os.M_NAMEfamily	
APPD_REQUEST_ID	

Configure Application Properties

You can define new values for node properties, such as host name or vCPU. You can also define new values 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 library, blueprint, or deployment profile. The system defines node properties, such as memory allocation and number of CPUs, but you can override them. For example, a needed service might have a JVM heap size of 512 MB. But for a large deployment, you can override that setting and change the size to 1024 MB.

You can set a host name so that the virtual machine can be easily identified in the vCloud Director or vRealize Automation 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

- Log in to Application Services as an **application publisher and deployer**.
- Verify that at least one application is created in Application Services. See [Chapter 14 Creating Applications](#).
- Verify that the **Deployment Environment** tab is configured. See [Configure the Deployment Environment](#).

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](#).

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 vRealize Automation 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. Application Services 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 Application Services fails the deployment in an idle state after three days.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that at least one application is created in Application Services. See [Chapter 14 Creating Applications](#).
- Verify that the **Application Properties** tab is configured. See [Configure Application Properties](#).
- Verify that at least one custom task is created in the Application Services catalog. See [Add a Custom Task to the Library](#).

- 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](#) or [Create an Amazon EC2 Deployment Environment](#).
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).

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 **Library 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.
- 9 Click **Save**.
- 10 Click **OK**.

The deployment profile is listed for the application version.

What to do next

Use the deployment profile to deploy the application. See [Deploy with a Single Deployment Profile](#).

Use an Existing Deployment Profile

You can reuse an existing deployment profile for an application version.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Register the uploaded CentOS 6.3 32-bit template to a cloud provider. See [Register the vCloud Director Cloud Provider and Template](#), [Register the vRealize Automation Cloud Provider and Template](#), or [Register the Amazon EC2 Cloud Provider and Template](#).
- Verify that at least one cloud template is mapped to each logical template used in the blueprint. See [Add a Logical Template to the Library](#).
- Verify that a deployment profile is available in Application Services. See [Setting Up and Configuring a Deployment Profile](#).
- 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](#) or [Create an Amazon EC2 Deployment Environment](#).

Procedure

- 1 On the Application Services 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](#) and [Deploy with a Single Deployment Profile](#).

Create a Composite Deployment Profile

A composite deployment profile joins multiple deployment profiles into a single deployable unit with shared property bindings.

Prerequisites

Log in to Application Services as an **application publisher and deployer**.

Procedure

1 [Set Up a Composite Deployment Profile](#)

A composite deployment profile lets you join together multiple deployment profiles. The result is a composite deployment of applications.

2 [Select Deployment Profiles](#)

You can select one or more deployment profiles and determine the order of their deployment.

3 [Bind Properties](#)

You can bind the properties of nodes, services, and application components used in the associated applications.

Set Up a Composite Deployment Profile

A composite deployment profile lets you join together multiple deployment profiles. The result is a composite deployment of applications.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that you are a member of the appropriate business group.
- Verify that an application exists. See [Copy an Application](#).
- Verify that the application has an application version. See [Create an Application Version](#).
- Verify that one or more deployment profiles exist for the application version. See [Create a Deployment Profile](#).

Procedure

1 On the Application Services 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 Add icon (+).

A menu appears.

- 4 Select Create Composite Deployment Profile from the menu.

- 5 Enter a name for the composite deployment profile.

- 6 Select the business group and click **Deploy**.

The business group must contain the deployment profiles to include in the composite deployment profile.

The Add Deployment Profiles page appears.

What to do next

Your new composite deployment profile is set up, but empty. You need to select the deployment profiles.

Select Deployment Profiles

You can select one or more deployment profiles and determine the order of their deployment.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that multiple deployment profiles exist. See [Create a Deployment Profile](#).
- [Set Up a Composite Deployment Profile](#).

Procedure

- 1 Select an application from the **Application** drop-down menu.

The application must contain a deployment profile to include in the composite deployment profile.

- 2 Select an application version from the **Application Version** drop-down menu.

- 3 Select the deployment profile from the **Deployment Profile** drop-down menu.

The menu displays only deployment profiles in the same business group as the composite deployment profile.

- 4 Click **Add**.

The deployment profile appears in the selection table.

- 5 Repeat [Step 1](#) through [Step 4](#) to add another deployment profile.

If you add the same deployment profile multiple times, the system appends an underscore and an incremented sequence number to the alias of each duplicate.

- 6 (Optional) Change the deployment order of a deployment profile.

By default, the deployment profiles deploy in top-down order. You can change the order by changing the order numbers. Reordering a deployment profile does not alter its physical position.

- a Click the order number of the deployment profile to change.
- b Type over the number, or click the up or down arrow.

- 7 (Optional) Click the View Details link of a deployment profile to display information such as its application and deployment environment.

You must close the View Details pane to proceed.

- 8 Click **Visualize** to display the deployment profiles in graphical form.

The View Details link also appears on the graphical representation of a deployment profile.

- 9 Click **Next**.

The Bind Properties page appears.

What to do next

You have selected and ordered your deployment profiles. Now you can bind properties connected with those deployment profiles.

Bind Properties

You can bind the properties of nodes, services, and application components used in the associated applications.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- [Select Deployment Profiles](#).

Procedure

- 1 Click the Add icon (+).
- 2 Click **Choose** in the Property column.
The **Select a Property** dialog box appears.
- 3 Select one of the deployment profiles you have added to the composite deployment profile from the **Deployment Profile** drop-down menu.
- 4 Select a radio button for **Service** or **Application Component**.
The selection can vary with different deployment profiles.
- 5 Select an item from the corresponding drop-down menu.
Your choice determines the properties available for selection.
- 6 Select a property from the **Property** drop-down menu.

- 7 Click **OK**.
- 8 Click **Choose** in the Bind To Property column.
- 9 Select one of the deployment profiles you have added to the composite deployment profile from the **Deployment Profile** drop-down menu.

The menu displays the deployment profiles to be deployed before the current deployment profile. These deployment profiles have smaller order numbers than the current deployment profile.

- 10 Select a radio button for **Node**, **Service**, or **Application Component**.

The selection can vary with different deployment profiles.

- 11 Select a property from the **Bind To Property** drop-down menu.

- 12 Click **OK**.

- 13 Repeat the procedure to bind additional properties, as needed.

- 14 Click **Save**.

Deploying with Deployment Profiles

After saving your deployment profile or composite deployment profile, you can deploy the associated application or applications.

Deploy with a Single Deployment Profile

You can deploy the application from the Application Services user interface.

You can also deploy an application from the command-line interface. See [Deploying and Updating an Application Using CLI](#).

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that at least one deployment profile is available in Application Services. See [Create a Deployment Profile](#).

Procedure

- 1 On the Application Services 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.
- 4 Follow the prompts in the Deployment Profile wizard and make any changes.
- 5 Review the deployment profile settings.

You can make necessary changes to the deployment profile before you deploy the application.

6 Click **Deploy**.

A deployment summary page appears and refreshes approximately every 30 seconds to display the deployment status.

7 (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](#).

Check the status of an in-progress deployment. See [Using the Deployment Summary Page](#).

Deploy with a Composite Deployment Profile

You can deploy multiple applications in a chosen order and with bound properties with a composite deployment profile from the Application Services user interface.

You can also deploy an application from the command-line interface. See [Deploying and Updating an Application Using CLI](#).

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that a composite deployment profile is available in Application Services. See [Create a Composite Deployment Profile](#).

Procedure

1 On the Application Services 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 composite deployment profile, or use an existing profile.

You must have one or more deployment profiles before you can create a composite deployment profile.

4 Review the selected deployment profiles and property bindings.

You can make changes to the composite deployment profile before you deploy the applications it represents.

5 Click **Deploy**.

A composite deployment summary page appears and refreshes in approximately 30 seconds to display the deployment status of each deployment profile.

6 (Optional) Click **Refresh** in the toolbar to update the real-time status of the deployments.

- 7 Click a name in the Deployment Name column to display the deployment summary page for the deployment profile.

Clicking the Back arrow returns you to the composite deployment summary page.

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](#).

Check the status of an in-progress deployment. See [Using the Deployment Summary Page](#).

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 vRealize Automation 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

- Log in to Application Services as an **application publisher and deployer**.
- Verify that at least one deployment profile is available in Application Services. See [Create a Deployment Profile](#).
- You must be a member of the owning business group for the associated deployment profile.
If you are not a member of the business group that owns the associated deployment profile, the **Quick Deploy** button () is disabled.

Procedure

- 1 On the Application Services 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 **Quick Deploy** () 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](#).

Publishing Deployment Profiles to the vCloud Automation Center Service Catalog

With Application Services, you can publish a deployment profile or composite deployment profile to the vRealize Automation service catalog. vRealize Automation users can request the catalog item to start using the application or applications in that environment.

Publish a Single Deployment Profile

You can publish an Application Services deployment profile to the vRealize Automation service catalog. vRealize Automation users can request the catalog item to start using the application in that environment.

You cannot delete a published deployment profile from the vRealize Automation service catalog.

If you publish a deployment that has mapped artifacts in its blueprint, you can override the tracking identifier and content URL often used in such blueprints from the vRealize Automation service catalog.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that a deployment profile is available in Application Services. See [Setting Up and Configuring a Deployment Profile](#).
- vRealize Automation IaaS environment and entitlements must be properly set up and configured. See *vRealize Automation Installation and Configuration* and *IaaS Configuration for Virtual Platforms* documentation.
- Verify 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 vRealize Automation service catalog or quick deploy the latest application version and associated deployment profile.

Procedure

- 1 On the Application Services 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 changes to the deployment profile before publishing it.

- 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 appear in the vRealize Automation service catalog. This name and description are different from the deployment profile name and description.

- 8 Log in to your tenant in vRealize Automation.

The vRealize Automation console opens.

- 9 Select **Administration > Catalog Management > Catalog Items** 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 vRealize Automation catalog item for development. See [Request a vRealize Automation Service Catalog Item](#).

Publish a Composite Deployment Profile

You can publish an Application Services composite deployment profile to the vRealize Automation service catalog. vRealize Automation users can request the catalog item to start using the applications in that environment.

You cannot delete a published deployment profile from the vRealize Automation service catalog.

If you publish a deployment that has mapped artifacts in its blueprint, you can override the tracking identifier and content URL often used in such blueprints from the vRealize Automation service catalog.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that a composite deployment profile is available in Application Services. See [Setting Up and Configuring a Deployment Profile](#).
- vRealize Automation IaaS environment and entitlements must be properly set up and configured. See *vRealize Automation Installation and Configuration* and *IaaS Configuration for Virtual Platforms* documentation.
- Verify that if there is a custom task in any of the deployment profiles, all of the required property values in that deployment profile are defined so that the SSO user can request the item from the vRealize Automation service catalog or quick deploy the latest application version and associated deployment profile.

Procedure

- 1 On the Application Services 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 composite deployment profile or create one.
You must have one or more deployment profiles available to create a composite deployment profile.
- 4 Review the deployment profiles and the property bindings.
You can make changes to the composite deployment profile before publishing it.
- 5 Click **Publish**.
- 6 In the Publish dialog box, append the name of the composite deployment profile with more information, add a description, and click **OK**.

For example, add an optional description to the Description section about the applications to be deployed.

The published name and description appear in the vRealize Automation service catalog. This name and description are different from the deployment profile name and description.

- 7 Log in to your tenant in vRealize Automation.
The vRealize Automation console opens.
- 8 Select **Administration > Catalog Management > Catalog Items** and open the published item in the catalog.
- 9 Assign an entitled service to the catalog item from the Service drop-down menu and click **Update**.
The catalog item list appears.
- 10 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.
- 11 Click **Update**.

What to do next

Request a vRealize Automation catalog item for development. See [Request a vRealize Automation Service Catalog Item](#).

Request a vRealize Automation Service Catalog Item

When you request a service catalog item you provision the item to the designated cloud environment.

A typical user in vRealize Automation does not have direct access to the Application Services environment. Therefore, the user can access the vRealize Automation service catalog, which is a self-service portal and request Application Services application deployments.

Prerequisites

- Verify that you can log in to a vRealize Automation IaaS environment as an application publisher and deployer. For information about how to install and configure vRealize Automation and set entitlements on items in the catalog, see the vRealize Automation documentation.
- Verify that at least one vRealize Automation catalog item has an entitlement and a service assigned to it. See [Publish a Single Deployment Profile](#).
- Log in to the vRealize Automation console as an **application publisher and deployer**.

Procedure

- 1 Select **Administration > Catalog Management > Catalog Items**.
- 2 Locate a catalog item that has an entitled service assigned to it.
- 3 Click the **Catalog** tab and navigate to the service.

- 4 Click **Request** on the service icon and complete the description.

For example, you can request an item for development and testing purposes.

- 5 Click **Next** to review the properties of the item.

If a deployment uses artifact mapping in its blueprint, and that blueprint includes a tracking identifier or content URL as a property value, you can override that value in the service catalog.

You can edit node properties such as CPU, memory, cluster size and hostname, including custom properties that were defined on the node's machine blueprint.

- 6 Click **Submit** to request the catalog item.

When the deployment is complete, Application Services creates resources in vRealize Automation that appear in the **Items** tab.

What to do next

Select the vRealize Automation resource to track the progress. You can also use the Application Services deployment summary page to check the status in real time. See [Using the Deployment Summary Page](#).

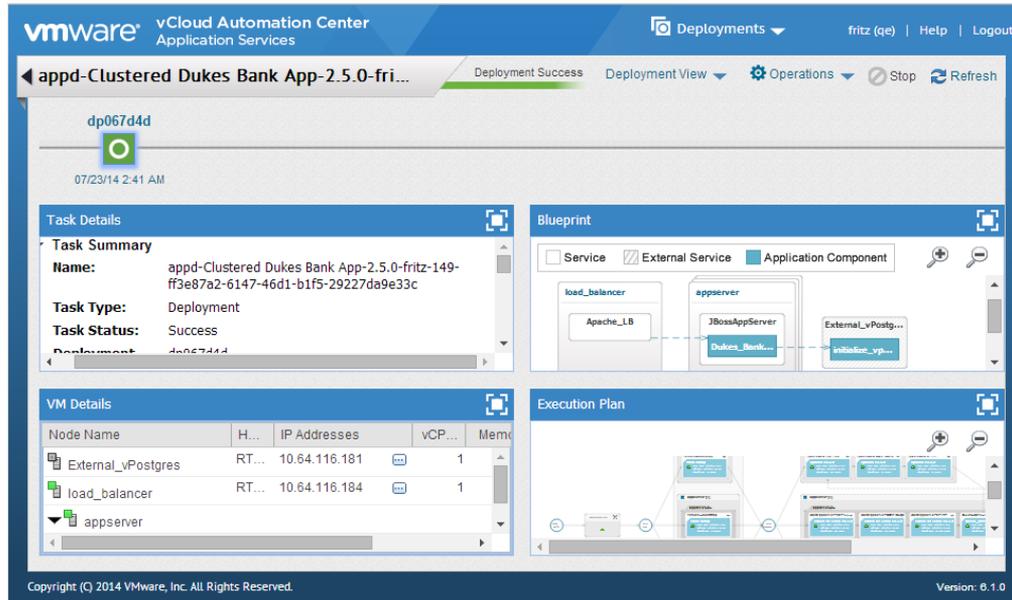
Using the Deployment Summary Page

Application Services 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 vRealize Automation user interfaces to check the status of the deployed virtual machines. To view the status of a Application Services 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.

Figure 16-1. Deployment Summary Page



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](#) and [View Deployed VM Details and Execution Plan of an Application](#).

Using the Composite Deployment Summary Page

Application Services provides a graphical user interface for checking the status of application deployments in real time on the composite deployment summary page.

You can also use the vCloud Director and vRealize Automation user interfaces to check the status of the deployed virtual machines. To view the status of a Application Services deployment in an Amazon VPC and associated Availability Zone, see [Amazon AWS Documentation](#).

On the composite deployment summary page, when deployments are running, the overall deployment status of the deployments appear in the toolbar. The page also displays the order, status, deployment names, deployment profile names, deployment environments, applications, application versions, the logged-in user who started the process, the start times, and the end times of the deployments.

The names in the Deployment Name column are links to the deployment summary pages of the respective deployment profiles. Clicking a name opens the deployment summary page for the deployment profile. For information about the deployment summary page, see [Using the Deployment Summary Page](#).

After the deployments finish, the status bar turns red or green depending on the success or failure of the tasks in the deployments.

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 Application Services, 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 vRealize Automation 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. With composite deployment profiles, these steps are repeated for each application deployed.

- 1 Application Services 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 Application Services 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, Application Services provisions the scaled clustered virtual machines by instantiating the cloud templates from the vCloud Director catalog or vRealize Automation 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 Application Services 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 Application Services 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 Application Services server.
- 7 The agent in each virtual machine downloads the execution plan from Application Services 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 Application Services 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.

Understanding Deployment Failures

When you deploy an application to the cloud or update a deployed application in the cloud, virtual machines are created and software is provisioned in the virtual machines. During these processes, deployment failures can occur that can affect your deployment.

Failures can occur in several deployment phases. In Application Services, you can identify where failures occurred by viewing the execution plan. You can use the **Resume Deployment** option, which is available in the **Execution Plan** panel in the Deployment View, to resume a failed deployment.

Table 16-1. Deployment Phases

Deployment Phases	Kinds of Failure
Pre-Provisioning	<ul style="list-style-type: none"> ■ Policy validation ■ Template validation ■ DE reachability validation
Provisioning	<p>Boot Strap failure occurs for the following reasons:</p> <ul style="list-style-type: none"> ■ Maximum virtual machines exhausted ■ OS install ■ IP setup ■ Storage issue <p>Agent bootstrap failure occurs for the following reasons:</p> <ul style="list-style-type: none"> ■ Firewall issue in the template ■ Gateway issue in the template ■ Timeout occurs during communication from agent to Application Services due to network and firewall issues

Table 16-1. Deployment Phases (Continued)

Deployment Phases	Kinds of Failure
Pre-Execution	vRealize Automation network bootstrap issues.
Execution	<ul style="list-style-type: none"> ■ Content download failure during install or configure phase ■ Script failures during install or configure start phase
Post-Execution	Deprovisioning failures can occur when external services are part of deployment.

Custom tasks cannot be added or deleted in the resume operation. Deployment failures that occur due to custom task failures cannot be resumed.

Resolve Deployment Failure

Deployment failures in any phase of application provisioning flow always require an application deployment tear down. Because deployment failures occur frequently and result in critical issues, they must be fixed without requiring a new deployment. The Resume Deployment operation lets you restart the failed deployments by fixing the error conditions.

Pre-Provisioning Failures

Pre-provisioning failures occur at an early stage in a deployment. Whenever a deployment fails in this phase, an error message appears. Correct the error before you trigger the resume deployment operation in the same deployment. You can then restart the pre-provisioning process.

In a pre-provisioning failure, errors often occur because of policy and template validation.

Provisioning Failure

Provisioning failure can occur when the host provisioning or agent bootstrap fails. A red icon appears in the host or agent bootstrap node. When the bootstrap fails in one of the virtual machines due to an Application Services agent installation issue or any issues related to the agent, the system tears down the newly provisioned virtual machine and reprovisions it. When the deployment resumes, the virtual machines that are not working correctly are torn down and provisioned again.

Note Network Bootstrap failure is not supported.

Execution

Execution failures can occur due to content download failure or script failure. When the resume option is enabled, you can edit the properties and script for the failed component. When you resume the deployment, execution starts with the first script of the component such as install, configure, or start scripts. When you resume the virtual machine or script, it might not reinstall or start the installation as it is already installed. You must modify the script to get the virtual machine to a clean state to start installing.

Updating Application Deployments

17

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.

Important Applications deployed with composite deployment profiles do not support resume, any type of update, or assisted teardown. However, such applications do support quick teardown.

You can deploy a saved update profile multiple times to update existing deployments. With vCloud Application Director 6.0 and 6.0.1 and Application Services 6.1, you can initiate an update process for existing deployments in vCloud Director and vRealize Automation 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 and 6.0.1 and Application Services 6.1 do 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, Application Services 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, Application Services 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, Application Services 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](#)
- [Initiate an Update Process to Scale In Deployments](#)
- [Initiate an Update Process to Modify Configurations](#)
- [Use an Existing Update Profile](#)
- [Promote an Update Profile](#)
- [Rollback an Update Process](#)
- [Understanding Run Custom Task Update](#)
- [Troubleshoot Failed Update Process to Scale Deployments](#)
- [Troubleshoot Failed Update Process to Modify Configuration](#)

Initiate an Update Process to Scale Out Deployments

With Application Services, 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](#). With REST APIs in Application Services, you can automate the scale out of a deployed application. See *Using Application Services REST APIs* document document.

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

- Log in to Application Services as an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- The deployed application must include at least one clustered node. See [Specify a Node as a Cluster](#).
- If you plan to add a custom task, verify that at least one custom task is created in the Application Services library. See [Add a Custom Task to the Library](#).
- Verify that the initial deployment is successfully deployed to a cloud environment.

You cannot scale clustered nodes from a failed deployment or after a failed scale operation.

- Contact your cloud administrator to get information about the deployment environment storage space limit.

Procedure

- 1 On the Application Services 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 Update Profiles 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](#).

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](#).

Initiate an Update Process to Scale In Deployments

With Application Services, 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.

During scale-in, you can attach a custom task with a scaled-in node and perform the update. However, when this scale-in update profile is reused, the node does not have any custom task associated with it. This is because, during the first time scale-in, the custom task information is deleted along with the scaled-in node. Hence it cannot be used for scale-in reuse.

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 Application Services, you can automate the scale in of a deployed application. See [Using Application Services REST APIs](#) document.

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

- Log in to Application Services as an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- The deployed application must include at least one clustered node. See [Specify a Node as a Cluster](#).
- If you plan to add a custom task, verify that at least one custom task is created in the Application Services library. See [Add a Custom Task to the Library](#).
- Verify that the initial deployment is successfully deployed to a cloud environment.

You cannot scale clustered nodes from a failed deployment or after a failed scale operation.

- Contact your cloud administrator to get information about the deployment environment storage space limit.

Procedure

- 1 On the Application Services 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 Update Profiles 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.

Application Services 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](#).

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](#).

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.

Note 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](#). With REST APIs in Application Services, you can automate the modification of configurations of a deployed application. See *Using Application Services REST APIs* document.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- If you plan to add a custom task, verify that at least one custom task is created in the Application Services library. See [Add a Custom Task to the Library](#).

- 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 Application Services 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 Update Profiles 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 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 **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.

- 9 (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.

- 10 (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.

- 11 (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.

- 12 (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](#).

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](#).

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

- Log in to Application Services as an **application publisher and deployer**.
- Verify that at least one deployment is successfully updated in Application Services.

Procedure

1 On the Application Services 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 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

Complete the tasks to update the deployment. See [Initiate an Update Process to Scale Out Deployments](#), [Initiate an Update Process to Scale In Deployments](#), and [Initiate an Update Process to Modify Configurations](#).

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

- Log in to Application Services as an **application publisher and deployer**.

Procedure

- 1 On the Application Services 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](#).

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, Application Services performs a compatibility check between the promoted update profile and the deployment before applying the changes.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that at least one update profile is promoted to one or multiple deployment environments. See [Promote an Update Profile](#).

Procedure

1 On the Application Services 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**.

Application Services 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 Application Services 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

- Log in to Application Services as an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- Verify that the initial deployment is successfully deployed to a cloud environment.

Procedure

- 1 On the Application Services 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**.
- 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 changes 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**.

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](#).

Understanding Run Custom Task Update

Run Custom Task update allows you to run a series of arbitrary custom tasks on the virtual machines of an existing deployment.

Run Custom Task update allows you to perform the following tasks.

- **Operate at virtual machine level.** Lets you operate on several components simultaneously in components and services.
- **Execute scripts without property change.** Lets you execute the scripts without changing any service or component properties in an existing deployment.

To trigger a Run Custom Task update through the REST API, you must specify the following information.

- All the tasks that need to be executed.
- Input parameters for every task.
- The node on which each task is executed.
- The execution dependency for each task in a node. For example, one or more tasks might wait for the other tasks to execute and complete.

When one Run Custom Task update starts, it displays in the deployment overview page. The deployment timeline has a node that represents the current execution of the Run Custom Task update.

Run Custom Task supports rollback operation. When the custom task fails to execute or you want to revert the changes made earlier, you must execute another custom task to revert the change. This is because Application Services cannot generate a rollback script for arbitrary custom task.

There are two new REST APIs introduced to manage Run Custom Task.

- Create and Schedule a Run Custom Task Update Profile

- Update and Schedule a Run Custom Task Update Profile

For more information on REST APIs, see *Using Application Services REST APIs*.

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 the scale in update process.
- 3 Initiate another scale out update process on the previously failed deployment.

Application Services 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 and 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.

Application Services 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.

Deploying Predefined Library Components

18

Application Services library 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 Application Services Default group can access and deploy the sample applications.

All of the icons rendered for the sample templates and services in the library are based on the name of the template or service. For example, the official **MySQL** icon appears next to the MySQL service in the library. 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 library components. See [Key Concepts](#).

This chapter includes the following topics:

- [Using the Sample Templates](#)
- [Available Sample Applications for Deployment](#)
- [Update Deployed Sample Applications](#)
- [Sample Application Services Library Tasks](#)
- [Sample Library Services](#)

Using the Sample Templates

Application Services 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 Application Services Readme file.

Upload the CentOS 6.3 32-bit OVF package to the vCloud Director catalog or vRealize Automation. For more information on the vCloud Director catalog or vRealize Automation, 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.

```
wget http://UbuntuMirrorSite/libpython2.7_2.7.3-0ubuntu3.4_amd64.deb
```

- 2 Install the python package to the Ubuntu template.

```
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 [Application Services System Requirements](#).

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.

```
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.

```
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, external services, and application components that are fully functional when deployed to the cloud.

An application architect can use the Application Services 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](#), [Register the vRealize Automation Cloud Provider and Template](#), or [Register the Amazon EC2 Cloud Provider and Template](#).
- Map the cloud template to the CentOS63 32bit logical template. See [Add a Logical Template to the Library](#).
- Log in to Application Services as an **application architect** and an **application publisher and deployer**.
- Understand the basic concepts of creating a deployment profile and deploying an application. See [Chapter 16 Deploying Applications](#).
- 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 Application Services title bar, click the drop-down menu and select **Applications**.
- 2 Select **Nanotrader**.
- 3 Click the Blueprint screenshot at the top of the screen.

- In the blueprint editor, select the service and SCRIPT component and add URL values to download the applicable files.

Service and Component	Description
vFabric_Web_Server	spring-nanotrader-web.tgz file for the deployment_archive property
vFabric_tc_Server	spring-nanotrader-web.war file for the external_template property
spring-nanotrader-services	spring-nanotrader-services.war file for the war_file property

- Click **Save**.
- On the Application Services title bar, click the drop-down menu and select **Applications**.
- Open the Nanotrader application that you configured.
- Create a deployment profile and deploy the application.
- 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.

- 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](#) and [Initiate an Update Process to Modify Configurations in Sample Applications](#).

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](#).
- Map the cloud template to the Windows Server 2008 R2 SP1 logical template. See [Add a Logical Template to the Library](#).

- Log in to Application Services as an **application architect** and an **application publisher and deployer**.
- 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 16 Deploying Applications](#).

Procedure

- 1 On the Application Services 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 Application Services 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://Load_Balancer_IP:8081`** URL.

The `Load_Balancer_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](#) and [Initiate an Update Process to Modify Configurations in Sample Applications](#).

Deploy Clustered Dukes Bank Application

The Clustered Dukes Bank application is a sample three-tier Application Services 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](#) and [Register the vCloud Director Cloud Provider and Template](#).
- Map the cloud template to the CentOS63 32bit or Windows Server 2008 R2 SP1 logical template. See [Add a Logical Template to the Library](#).
- Log in to Application Services as an **application architect** and an **application publisher and deployer**.
- 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 Application Services to Use a Proxy for External URLs](#).
- 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 16 Deploying Applications](#).

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Applications**.
- 2 Click **Clustered Dukes Bank App**.
- 3 Select the application version.

Option	Action
Linux-based Clustered Dukes Bank	Select version 2.1.0.
Windows-based Clustered Dukes Bank	Select version 3.0.0.

- 4 Click the Blueprint screenshot at the top of the screen.
- 5 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 Application Services agent in the JBossAppServer service should be able to access the installed JRE.
- 6 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.
- 7 Create a deployment profile and deploy the application.

- 8 To access the deployed application, open a supported Web browser and type the `http://Load_Balancer_IP:8081/bank/main.faces` URL.

The `Load_Balancer_IP` is the IP address of the deployed load balancer.

- 9 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](#) and [Initiate an Update Process to Modify Configurations in Sample Applications](#).

Deploy jPetStore Application

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.

To deploy the blueprint for the jPetStore application, you must download the VMware vFabric SQLFire 1.0.0 installer (`vFabric_SQLFire_10_Installer.jar`) file and point to that file using the SQLFire service property `SQLFire10_Installer`.

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](#) and [Register the vCloud Director Cloud Provider and Template](#).
- Map the cloud template to the CentOS63 32bit disk logical template jPetStore 1.0.0. See [Add a Logical Template to the Library](#).
- Log in to Application Services as an **application publisher and deployer**.
- If your system requires a proxy to access the Internet, verify that your proxy connections are properly configured. See [Configure Application Services to Use a Proxy for External URLs](#).
- Understand the basic concepts of creating a deployment profile and deploying an application. See [Chapter 16 Deploying Applications](#).

Procedure

- 1 Download the vFabric SQLFire 1.0.0 installer package (`vFabric_SQLFire_10_Installer.jar`) from the VMware Downloads page (<https://my.vmware.com/web/vmware/downloads>) to your own httpd server or the httpd server on the Application Services server.

If you use the httpd server on the Application Services server, download the `vFabric_SQLFire_10_Installer.jar` file to the `/opt/vmware/darwin/htdocs/artifacts/services/sqlfire` folder.

You must enter your customer account credentials to download the installer package.

- 2 On the Application Services title bar, click the drop-down menu and select **Applications**.

- 3 Click the **jPetStore** application to open jPetStore version 1.0.0.
- 4 Click the **View Blueprint** image to open the jPetStore v1.0.0 - Blueprint page.
 - a Click **SQLFire** in the jPetStore blueprint image.
 - b Click the **Properties** tab in the row that contains the **Details**, **Properties**, and **Actions** tabs.
 - c Click the Edit (✎) icon in the row that contains the SQLFire10_Installer property name.
 - d Specify a blueprint value on the **Edit Properties for SQLFire** page by using one of the following options.
 - Click **Use library value** to accept the default library value of `http://${darwin.content.server.ip}/artifacts/services/sqlfire/vFabric_SQLFire-1.0.jar`. The library value points to the httpd server at the Application Services server.

Note To use this library value, the `vFabric_SQLFire_10_Installer.jar` that you downloaded from the VMware Downloads page must be located at `/opt/vmware/darwin/htdocs/artifacts/services/sqlfire` and be named `vFabric_SQLFire-1.0.jar`.

 - Click in the **Type your own value** text box and enter the filepath to the `vFabric_SQLFire_10_Installer.jar` file at a different httpd location.
 - e Click **Save** to close the **Edit Properties for SQLFire** page.
- 5 Click **Deploy** to deploy the blueprint.
- 6 Create a deployment profile.
- 7 Deploy the application.
- 8 To access the deployed application, open a supported Web browser and enter `http://jPetStore_VM_IP:8080/jpetstore-1.0.0` in the address bar.

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](#), [Register the vRealize Automation Cloud Provider and Template](#), or [Register the Amazon EC2 Cloud Provider and Template](#).
- Map the cloud template to the CentOS63 32bit logical template. See [Add a Logical Template to the Library](#).
- Log in to Application Services as an **application publisher and deployer**.

- Verify that Application Services is configured to use a proxy. See [Configure Application Services to Use a Proxy for External URLs](#).
- Understand the basic concepts of creating a deployment profile and deploying an application. See [Chapter 16 Deploying Applications](#).

Procedure

- 1 On the Application Services 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.

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

- Log in to Application Services as an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- The deployed application must include at least one clustered node. See [Specify a Node as a Cluster](#).

- Verify that the initial deployment is successfully deployed to a cloud environment.
You cannot scale clustered nodes from a failed deployment or after a failed scale operation.
- Contact your cloud administrator to get information about the deployment environment storage space limit.

Procedure

- 1 On the Application Services 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 Update Profiles 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](#).

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

- Log in to Application Services as an **application publisher and deployer**.
- Familiarize yourself with the basic concepts of defining and configuring component properties and actions. See [Chapter 12 Developing Application Services Components](#).
- 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 Application Services 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 Update Profiles 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.

Sample Application	Service Version	Properties
Nanotrader	vFabric_Web_Server 5.1.1	http_port, deployment_archive, and webserver_conf_file
	vFabric_tc_Server 2.7.1	port, db_ip, db_port, and jdbc_url
	vFabric_SQLFire_Server 1.0.3	schema_file and dataload_file
Clustered DotShoppingCart	Apache_LB 2.2.22	http_port
	SQL_Server_2008 1.0.0	SA_PWD
Clustered Dukes Bank 3.0.0	Apache_LB 2.2.22	http_port and http_proxy_port
	JBossAppServer 5.1.0	JBOSS_JMX_USER and JBOSS_JMX_PWD
	SQL_Server_2008 1.0.0	SA_PWD
	initialize_db_script	db_username and db_password
	Dukes_Bank_App	JAR_FILE, EAR_FILE, db_password, and db_user
Clustered Dukes Bank 2.1.0	Apache_LB 2.2.22	http_port and http_proxy_port
	JBossAppServer 5.1.0	JBOSS_JMX_USER and JBOSS_JMX_PWD
	MySQL 5.0.0	db_port and db_root_password

Sample Application	Service Version	Properties
	initialize_db_script	db_username, db_password, init_db_username, and init_db_password
	Dukes_Bank_App	JAR_FILE, EAR_FILE, db_port, db_password, and db_user

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](#).

Sample Application Services Library 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 library. See [Add a Custom Task to the Library](#).

- [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.

- [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.

- [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.

- [Add an 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.

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 Application Services as an **application publisher and deployer**.
- Verify that Application Services is configured to use a proxy. See [Configure Application Services to Use a Proxy for External URLs](#).
- 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](#).

Procedure

1 On the Application Services 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 **Expand Cluster** (🔍).

If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.

7 Click **Add Script Task** (➤) 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 appears.

8 Select the predefined task from the **Library Task Name** drop-down menu.

The supported operating systems, predefined task details, script, and property details appear.

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**.

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 Application Services library. See [Add a Custom Task to the Library](#).

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 Application Services as an **application publisher and deployer**.
- 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 Application Services is configured to use a proxy. See [Configure Application Services to Use a Proxy for External URLs](#).

Procedure

- 1 On the Application Services 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 **Expand Cluster** (🔍).

If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.

- 7 Click **Add Script Task** (➤) 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 appears.

- 8 Select the predefined task from the **Library Task Name** drop-down menu.

The supported operating systems, predefined task details, script, and property details appear.

- 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**.

- 11 Click **OK**.

The APT Repository Config predefined task is added to the execution plan.

- 12 Review the deployment profile settings and deploy the application.

What to do next

Determine whether to add a customized task to the Application Services library. See [Add a Custom Task to the Library](#).

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 Application Services with a user account that has an Application Services role assigned to it.
- Log in to Application Services as an **application publisher and deployer**.
- 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 Application Services is configured to use a proxy. See [Configure Application Services to Use a Proxy for External URLs](#).

Procedure

- 1 On the Application Services 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 **Expand Cluster** .
If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.
- 7 Click **Add Script Task**  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 appears.

- 8 Select the predefined task from the **Library Task Name** drop-down menu.
The supported operating systems, predefined task details, script, and property details appear.
- 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**.

11 Click **OK**.

The YUM Repository Config predefined task is added to the execution plan.

12 Review the deployment profile settings and deploy the application.

What to do next

Consider adding a customized task to the Application Services library. See [Add a Custom Task to the Library](#).

Add an 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 Application Services as an **application publisher and deployer**.
- 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 Application Services is configured to use a proxy. See [Configure Application Services to Use a Proxy for External URLs](#).

Procedure

1 On the Application Services 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 **Expand Cluster** (🔍).

If the clustered node is not expanded, the predefined task is added only to the first virtual machine in the cluster.

- 7 Click **Add Script Task** () 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 appears.

- 8 Select the predefined task from the **Library Task Name** drop-down menu.
- The supported operating systems, predefined task details, script, and property details appear.
- 9 On the **Properties** tab, configure the properties.
- Select the `rh_n_username` property, enter the username value used to register the virtual machine with the Red Hat Network, and click **Save**.
 - Select the `rh_n_password` property, enter the password value used to register the virtual machine with the Red Hat Network, and click **Save**.
- 10 Click **OK**.

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 Application Services library. See [Add a Custom Task to the Library](#).

Sample Library Services

Application Services includes predefined components, such as services, in its library that are reusable components in several applications. These services are available to all business groups in Application Services.

Library Services

On the Application Services title bar, you can click the drop-down menu and select **Library > Services** to view the available sample services. The Library 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, Puppet Services, and Other.

Note Use the predefined sample library services only in a test environment.

Managing Deployments

When you deploy an application, an item is added to the Deployments page in Application Services.

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, start an update process, tear down a deployed application from the cloud, and remove an application deployment record from Application Services.

Note Policy scans do not function with composite deployments.

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](#)
- [View Deployed VM Details and Execution Plan of an Application](#)
- [Start a Policy Scan](#)
- [Tear Down an Application from the Cloud](#)
- [Scale In Deployments from vCloud Automation Center](#)
- [Scale Out Deployments from vCloud Automation Center](#)
- [Tear Down an Application from vCloud Automation Center](#)
- [Delete an Application Deployment from Application Services](#)
- [Cancel a Deployment or an Update Process](#)
- [View Policy Compliance Summary](#)

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](#). The composite deployment summary page shows details for all of its application deployments, and provides ready access to their deployment summary pages, as described in [Using the Composite Deployment Summary Page](#).

All Application Services roles can view deployments in their business groups.

Prerequisites

Verify that an application is deployed or an update process is initiated. See [Deploy with a Single Deployment Profile](#) or [Chapter 17 Updating Application Deployments](#).

Procedure

- 1 On the Application Services 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 dependent task, or virtual machines acquiring IP addresses, click the **Expand** icon () to open the applicable window in the deployment summary page.
- 4 Click the **Minimize** icon () 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 value 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, business 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 library, 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](#).

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 node name, logical template, and cloud template in vCloud Director, vRealize Automation, 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 vRealize Automation deployments.

Prerequisites

Verify that an application is deployed or that an update process is initiated. See [Deploy with a Single Deployment Profile](#) or [Chapter 17 Updating Application Deployments](#).

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, vRealize Automation, 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.
- 5 To view the custom properties defined for each node to override properties in the vRealize Automation blueprint, click the Extra Configuration icon (.
- 6 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.

- 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.

- Review the status of each task.

Depending on the status of the task, an icon appears next to it.

Icon	Description
	Task has not yet begun or did not run.
	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.
	Expand and view the details of each provisioning task.
	Task requires reboot .
	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.
	Expand and view the logs of one or more failed provisioning tasks.
	Task is waiting for one of the dependencies to finish running

- For details about a component or action script and its properties, expand the **Execution Plan** status window, click the arrow 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.

- To view an action script, expand the **Execution Plan** status window, click the arrow icon (▼) next to the task, and select **View Action Script**.
- To access the virtual machine log files, click the arrow icon (▼) next to the task and select **View Virtual Machine Logs**.
- To view the property values for the script, click the arrow icon (▼) next to the task and select **View Component Properties**.
- 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](#).

To delete a deployment record from Application Services, see [Delete an Application Deployment from Application Services](#).

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. Application Services 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

- Log in to Application Services as an **application publisher and deployer**.
- Verify that a policy or policy instance has changed.
- Verify that deployments with the obsolete policy definitions are available in Application Services.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Deployments**.
- 2 Enter 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](#).

Tear Down an Application from the Cloud

In Application Services, you can start the teardown process to remove the vCloud Director vApp and associated virtual machines, vRealize Automation 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 Application Services user interface or the command-line interface. See [Using CLI to Tear Down a Deployment](#). 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 Application Services 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 Application Services. To remove the deployment record from the Deployments page, see [Delete an Application Deployment from Application Services](#).

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- 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 Application Services title bar, click the drop-down menu and select **Deployments**.
- 2 Enter 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.

A composite application deployment supports only the quick tear down process, and all of its application deployments must be torn down at the same time.

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 Application Services](#).

Scale In Deployments from vCloud Automation Center

You can request to scale in one or more clustered nodes in an existing deployment from the vRealize Automation console. You can configure only the cluster size of nodes modeled as clusters in the application blueprint.

You cannot scale in a deployment that uses a blueprint that has a mapped artifact. For more information about artifact management, see [Chapter 15 Working with Artifacts](#).

Prerequisites

- Log in to the vRealize Automation console as a **tenant administrator**.
- Verify that you have access to at least one deployed application with a clustered node from vRealize Automation.

Procedure

- 1 Select **Items > Application Deployment**.
- 2 Enter the name of the deployment in the search text box and click the magnifying glass icon () or press Enter.
- 3 Select the deployment from the search result list.
The Item Details page appears.
- 4 Select **Scale In** from the **Actions** menu.
The **Request Information** tab on the Scale In dialog box appears.
- 5 Enter a description in the **Description** text box.
This required description helps you monitor the progress of the request after submission.
- 6 (Optional) Enter the reasons for the request in the **Reasons** text box.
- 7 Click **Next**.
- 8 Enter the number of clusters to scale in by clicking the up and down arrows in the **Scale in by** text box.
- 9 Click **Submit**.
The Request confirmation box appears.
- 10 Click **OK**.

What to do next

You can monitor the status of your scale in request on the **Requests** tab on the vRealize Automation console.

Scale Out Deployments from vCloud Automation Center

You can request to scale out one or more clustered nodes in an existing deployment from the vRealize Automation console. You can configure only the cluster size of nodes modeled as clusters in the application blueprint.

You cannot scale out a deployment that uses a blueprint that has a mapped artifact. For more information about artifact management, see [Chapter 15 Working with Artifacts](#).

Prerequisites

- Log in to the vRealize Automation console as a **tenant administrator**.
- Verify that you have access to at least one deployed application with clustered nodes from vRealize Automation.

Procedure

- 1 Select **Items > Application Deployment**.
- 2 Enter the name of the deployment in the search text box and click the magnifying glass icon () or press Enter.
- 3 Select the deployment from the search result list.
The Item Details page appears.
- 4 Select **Scale Out** from the **Actions** menu.
The **Request Information** tab on the Scale Out dialog box appears.
- 5 Enter a description in the **Description** text box.
This required description helps you monitor the progress of the request after submission.
- 6 (Optional) Enter the reasons for the request in the **Reasons** text box.
- 7 Click **Next**.
- 8 Enter the number of clusters to scale out by clicking the up and down arrows in the **Scale out by** text box.
- 9 Click **Submit**.
The Request confirmation box appears.
- 10 Click **OK**.

What to do next

You can monitor the status of your scale out request on the **Requests** tab on the vRealize Automation console.

Tear Down an Application from vCloud Automation Center

You can request the teardown of standard and composite applications and their virtual machines from the vRealize Automation console.

You can also tear down a deployed application from the Application Services user interface or the command-line interface. See [Tear Down an Application from the Cloud](#) and [Using CLI to Tear Down a Deployment](#). In the Application Services 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 in vCenter Server and vRealize Automation that are part of an application, or if an application is not used, you can use vRealize Automation to tear down the application. All virtual machines in the application are removed from their hosts.

Deployment tear down with vRealize Automation does not remove the deployment record from the Deployments page in Application Services. To remove the deployment record from the **Deployments** page, see [Delete an Application Deployment from Application Services](#).

Prerequisites

- Log in to the vRealize Automation console as a **tenant administrator**.
- Verify that you have access to at least one deployed application in vRealize Automation.
- 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 Select **Items > Application Deployment**.
- 2 Enter the name of the deployment to tear down in the search text box and click the magnifying glass icon () or press Enter.
- 3 Select the deployment to tear down from the search result list.
The Item Details page appears.
- 4 Select **Destroy** from the **Actions** menu.
The **Request Information** tab on the Destroy dialog box appears.

- 5 Enter a description in the **Description** text box.

This required description helps you monitor the progress of the request after submission.

- 6 (Optional) Enter the reasons for the request in the **Reasons** text box.

- 7 Click **Next**.

- 8 Select a **Choose a teardown option** radio button on the **Teardown Deployment** tab.

Option	Description
Assisted Teardown	Destroy all virtual machines deployed with the application and execute the TEARDOWN life cycle stage script.
Quick Teardown	Destroy all virtual machines deployed with the application, but do not execute the TEARDOWN life cycle stage script.

- 9 Click **Submit**.

The Request confirmation box appears.

- 10 Click **OK**.

What to do next

You can monitor the status of your teardown request on the **Requests** tab on the vRealize Automation console.

To delete the record of a deployment from the Deployments page, see [Delete an Application Deployment from Application Services](#).

Delete an Application Deployment from Application Services

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 Application Services.

Deleting a deployment from the Deployments page in Application Services does not delete the deployed application, vApp, and its virtual machines for vCloud Director, vRealize Automation 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](#).

Prerequisites

Log in to Application Services as an **application publisher and deployer**.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Deployments**.
- 2 Enter 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 Click **Delete Deployment**.
- 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, vRealize Automation, or Amazon EC2 to remove the residual components in the cloud.

Option	Description
vCloud Director	Delete the vApp and associated virtual machines.
vRealize Automation	Delete the vRealize Automation virtual machine and the associated virtual machine in vCenter Server.
Amazon EC2	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.

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 stop the deployment or update process and the provisioning in the cloud environment.

When you stop a deployment or an update process, the status changes to STOPPING and remains in that state until Application Services successfully stops the deployment or update process. The status then changes to STOPPED. Application Services does not delete the deployment or update process.

You cannot stop or resume composite deployments.

Note If the deployment status does not change to STOPPED in a reasonable amount of time, the process may be hung. You can force the deployment to stop by clicking **Stop** again. Only force the process to stop if the deployment status has not changed to STOPPED after an unreasonably long time. For related information see *Application Services Troubleshooting*.

Prerequisites

- Log in to Application Services as an **application publisher and deployer**.
- Verify that you have a deployment or update process in progress.

Procedure

- 1 Enter the name of the deployment or update process to cancel in the search text box.
- 2 From the search result list, select the deployment or update process.
The deployment summary page opens.
- 3 Click **Stop** in the toolbar above the deployment summary.

4 Confirm your selection.

Note If the deployment status does not change to STOPPED in a reasonable amount of time, the process may be hung, in which case you can force a hard stop by clicking **Stop** again. Clicking **Stop** a second time forces the deployment to stop immediately without completing the normal sequence of events employed by the stop process. You are prompted to confirm this action by clicking **OK**. Only force the process to stop if the deployment status has not changed to STOPPED after an unreasonably long time.

When you stop a deployment normally, currently running processes are stopped gracefully. When you force a stop, the deployment is stopped abruptly and currently running processes remain running. In either case, the associated VMs are not cleaned up.

What to do next

When you stop a deployment normally, you can resume it, tear it down from the cloud, or delete the application deployment record from vRealize Automation. See [Chapter 19 Managing Deployments](#).

For an update process, you can continue to interact with the deployment by starting another update process. See [Chapter 17 Updating Application Deployments](#).

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

- Log in to Application Services as an **application publisher and deployer**.
- Verify that a policy instance is applied to the deployment.

Procedure

- 1 On the Application Services title bar, click the drop-down menu and select **Deployments**.
- 2 Enter 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.

- 5 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](#).

Using the Application Services CLI

20

The Application Services CLI is a Spring Roo-based client that communicates with the Application Services server over HTTPS using REST APIs.

The **admin** user uses the Application Services CLI to register the Application Services appliance with the vRealize Automation server. User accounts with the application publisher and deployer role can use the CLI to deploy or tear down applications from the cloud. For information about user roles, see [Chapter 5 Setting Up Users and Groups](#).

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

Application Services User Names and Passwords Must Be ISO-8859-1 Compliant

Although the Application Services user interface supports the unicode character set for user name and password values, the Application Services CLI only supports the ISO-8859-1 character set. If you intend to use the Application Services CLI, use Application Services user name and password values that are compliant with the ISO-8859-1 character set.

This chapter includes the following topics:

- [Start the CLI Remotely](#)
- [General CLI Options](#)
- [Managing Cloud Tunnels](#)
- [Deploying and Updating an Application Using CLI](#)
- [Using CLI to Tear Down a Deployment](#)

Start the CLI Remotely

You can start the Application Services CLI from a remote machine.

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

Some CLI commands use names of business groups or deployments that can contain non-English characters. To display these characters, you must run the CLI in a fully internationalized shell client or terminal interface such as Putty or iTerm2. Also set the environment variable `LC_CTYPE` to `en_US.UTF-8` to enable non-English character input.

If you log on to the CLI with `su - root`, the variable is set automatically. If you omit the hyphen and log on with `su root`, you must set the variable with the following command.

```
export LC_CTYPE=en_US.UTF-8;
```

Prerequisites

- Verify that you know the password for the Application Services appliance.
- Verify that you installed Java JRE 1.7 on your remote machine.
- Make sure that the remote machine can connect to the Application Services appliance using HTTPS.

Procedure

- 1 Download the `darwin-cli.jar` file from the Application Services server `http://DarwinServerIP/tools/darwin-cli.jar` to a folder with write permissions on the remote machine.

Replace *DarwinServerIP* with the Application Services server IP address.

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 Application Services server.

```
login --serverUrl https://DarwinServerIP:8443/darwin --username UserName --password password
--tenantId tenantid
```

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 omit the `--password` parameter, the system prompts you to enter a password. The system does not save your password as plain text.

The `--tenantId` parameter logs you in to the tenant identified by the tenant name *tenantid*. If you omit the `--tenantId` parameter, the tenant is the default `vsphere.local`.

The command line prompt displays your user name, domain, and tenant.

General CLI Options

With the Application Services command-line interface options, you can 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. Your access to the command options depends on your assigned user role. For information about user roles and their allowed functions, see [Chapter 5 Setting Up Users and Groups](#).

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.

Table 20-1. General CLI Commands

CLI Command	Description
help	Lists use information.
cliversion	Displays the CLI version information.
login	Logs in the current user to the <code>roo</code> shell.
logout	Logs out the current user without closing the <code>roo</code> shell prompt. You can log out and log in as a different user.
status	Indicates whether a user is logged in. If a user is logged in, user name is displayed.
exit	Exits the CLI program.

Managing Cloud Tunnels

You can perform operations such as creating, updating, enabling, or deleting a secure cloud tunnel connection between the Application Services appliance and an Endpoint VM in an Amazon EC2 VPC.

Log in to Application Services as an **application cloud administrator**.

See [Create a Cloud Tunnel to Connect to Amazon EC2](#).

Table 20-2. Using the CLI to Manage Cloud Tunnels

CLI Command	Description
<pre>create-cloud-tunnel --name TunnelName --description "TunnelDescription" --enabled false --externalAddress EndpointVMElasticIP --sshPort 22 --internalAddress EndpointVMPrivateIP --proxyUrl ProxyURL --username UserName --privateKeyPath PrivateKeyFilePath</pre>	<p>Creates a secure cloud tunnel.</p> <p>For example, you can use the following command to create a cloud tunnel called EC2 Tunnel.</p> <pre>create-cloud-tunnel --name EC2 Tunnel --description "Test EC2 tunnel connection" --enabled false -- externalAddress 50.18.100.100 --sshPort 22 --internalAddress 192.0.2.255 --proxyUrl http://proxy.vmware.com:3128 -- username root --privateKeyPath /tmp/private_key_file</pre>
<pre>update-cloud-tunnel --name PreviousTunnelName --new-name NewTunnelName --description "TunnelDescription" --externalAddress EndpointVMElasticIP --sshPort 22 --internalAddress EndpointVMPrivateIP --proxyUrl ProxyURL --username UserName --privateKeyPath PrivateKeyFilePath</pre>	<p>Changes the values of various parameters of an existing cloud tunnel.</p> <p>For most of the parameters, this command works only if the cloud tunnel is disabled.</p>
<pre>list-cloud-tunnels --name TunnelName</pre>	<p>Retrieves a list of all the available cloud tunnels or a particular cloud tunnel, when specified.</p> <p>To view the details of a particular cloud tunnel, use the command <code>list-cloud-tunnels --name EC2 Tunnel</code>.</p>
<pre>enable-cloud-tunnel --name TunnelName</pre>	<p>Enables an existing cloud tunnel.</p> <p>This command initiates a background operation that establishes the secure cloud tunnel connection.</p>
<pre>disable-cloud-tunnel --name TunnelName</pre>	<p>Disables an existing cloud tunnel.</p> <p>This command initiates a background operation that disconnects the secure cloud tunnel connection.</p>
<pre>delete-cloud-tunnel --name TunnelName</pre>	<p>Deletes an existing cloud tunnel.</p> <p>The cloud tunnel must be disabled for it to be deleted.</p> <p>Removes the cloud tunnel connection between the Application Services appliance and the Amazon EC2 environment.</p>
<pre>test-cloud-tunnel --name TunnelName</pre>	<p>Verifies a cloud tunnel secure SSH connection and reports the current status.</p> <p>Enabling or disabling a cloud tunnel initiates the connection or disconnection operation in the background. This command reports the status of that background operation. In addition, the command checks whether the connection will function properly with Application Services during a deployment.</p> <p>Use this command after you enable or disable a cloud tunnel to confirm that your request succeeded.</p>

Deploying and Updating an Application Using CLI

To deploy an application, you must log in as an application publisher and deployer.

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 Application Services user interface to perform all of these tasks. See [Quick Deploy an Application](#), [Using the Deployment Summary Page](#), [Initiate an Update Process to Scale Out Deployments](#), or [Initiate an Update Process to Modify Configurations](#).

You can also view the deployed virtual machine status from the vCloud Director, vRealize Automation, and Amazon EC2 user interfaces.

Table 20-3. Deploy or Update an Application in the CLI

CLI Command	Description
<code>deploy-application --destination <i>ApplicationName-ApplicationVersion-DeploymentProfileName</i></code>	<p>Deploys an application blueprint.</p> <p>For example, to deploy the Clustered Dukes Bank application, version 2.1.0, with the prod-dep deployment profile, use the command <code>deploy-application --destination "Clustered Dukes Bank App-2.1.0-prod-dep"</code>.</p> <p>Enter dep 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.</p>
<code>deployment-status --deploymentName <i>DeploymentName</i></code>	<p>Shows the status of the latest deployment.</p> <p>If an application deployment fails, you can collect the virtual machine-specific log files to troubleshoot the problem. See the <i>VMware vCloud Automation Center Application Services Troubleshooting</i> guide .</p>

Table 20-3. Deploy or Update an Application in the CLI (Continued)

CLI Command	Description
<pre>deploy-application --destination ApplicationNameApplicationVersion- DeploymentProfileName --propertiesFile FileName.xml</pre>	<p>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 <code>global_conf</code>, <code>http_node_port</code>, and <code>http_server_port</code> properties in the <code>Apache_LB</code> and <code>AppServer</code> nodes.</p>
<pre><?xml version="1.0" encoding="UTF-8"?> <ns1:config-update-properties xmlns:ns1="http://www.test.com/darwin/schema/beans/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/dukesbank-2b-Update1-IndexPage.ear]]></value> </property> <property> <key>JAR_FILE</key> <value><![CDATA[http://192.10.1.113/Share/UPRConfig/CDB/mysql-connector-java-5.1.8.jar]]></value> </property> </node-component> </node> </ns1:config-update-properties></pre>	<pre><?xml version="1.0" encoding="UTF-8"?> <ns1:config-update-properties xmlns:ns1="http://www.test.com/darwin/schema/beans/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/dukesbank-2b-Update1-IndexPage.ear]]></value> </property> <property> <key>JAR_FILE</key> <value><![CDATA[http://192.10.1.113/Share/UPRConfig/CDB/mysql-connector-java-5.1.8.jar]]></value> </property> </node-component> </node> </ns1:config-update-properties></pre>
	<p>For example, to deploy the Clustered Dukes Bank application version 2.1.0, with the <code>staging-dep</code> deployment profile, and <code>new.props</code> properties file, use the command <code>deploy-application --destination "Clustered Dukes Bank App-2.1.0-DP_MAN_VCD" --propertiesFile ~/new.props.xml</code></p> <p>Enter <code>deploy</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.</p>

Table 20-3. Deploy or Update an Application in the CLI (Continued)

CLI Command	Description
<pre>update-scaleout --deploymentName <i>DeploymentName</i> -- propertiesFile <i>FileName.xml</i></pre>	<p>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.</p> <pre data-bbox="820 426 1410 659"><?xml version="1.0" encoding="UTF-8"?> <ns1:scaleout-properties xmlns:ns1="http://www.test.com/darwin/schema/beans/api"> <cluster-node> <name>appserver</name> <scale-out-by>1</scale-out-by> </cluster-node> </ns1:scaleout-properties></pre> <p>For example, to scale out the Clustered Dukes Bank application, version 2.1.0 with scaleout properties file, use the command <code>update-scaleout --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-6-cc0a3b20-43a0-4a22-be3e-49d4f31ab8e8" --propertiesFile ~/scaleout.xml</code>. Enter 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.</p>
<pre>update-config --deploymentName <i>DeploymentName</i> -- configUpdatePropertiesFile <i>ConfigUpdatePropertiesFileName.xml</i></pre>	<p>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 <code>global_conf</code> and <code>db_port</code> properties in the vFabric tc Server service of the AppServer node.</p> <pre data-bbox="820 1224 1410 1640"><?xml version="1.0" encoding="UTF-8"?> <ns1:config-update-properties xmlns:ns1="http://www.test.com/darwin/schema/beans/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></pre> <p>For example, to modify a configuration in the Clustered Dukes Bank application, version 2.1.0 with configupdate properties file, use the command <code>update-config --deploymentName "appd-Clustered Dukes Bank App-2.1.0-admin-2-19d63535-673e-4766-b380-de4e6ec3676a" --configUpdatePropertiesFile ~/configupdate.xml</code>.</p>

Table 20-3. Deploy or Update an Application in the CLI (Continued)

CLI Command	Description
	Enter 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, you must log in as an application publisher and deployer.

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 Application Services user interface. See [Tear Down an Application from the Cloud](#).

Table 20-4. Remove Deployment in the CLI

CLI Command	Description
<code>teardown --deploymentName <i>DeploymentName</i></code>	<p>Tear down a deployed application from the vCloud Director.</p> <p>For example, the <code>teardown --name appd-TestApp-1.0.0-admin-3-a99309aa-aa5b-45c2-b6a0-bac4a421178e</code> command removes the application from the cloud.</p> <p>Enter teardown 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.</p>

Using the CLI Import and Export Functions

21

You can import and export preconfigured application blueprints and their associated deployment profiles, services, external services, policies, artifact repositories, and available custom tasks across Application Services instances to help you further customize your applications.

You can import and export packages between different instances of Application Services 6.1, and import packages from Application Services 5.0, 5.2, and 6.0 to 6.1.

You can start the CLI from an Application Services remote machine. If you use a fully internationalized shell client or terminal interface, you can also run CLI commands that contain non-English characters. See [Start the CLI Remotely](#).

You must log in as an application architect and application catalog administrator to import and export Application Services packages.

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](#)

The CLI export command creates a package that consists of applications and their associated blueprints and deployment profiles, external services, policies, artifact repositories, services, logical templates, and available custom tasks to export between different Application Services instances.

- [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, artifact repositories, services, logical templates, and available custom tasks between different Application Services 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, artifact repositories, services, logical templates, and available custom tasks to export between different Application Services instances.

The export package has two formats, a compressed archive file format and an uncompressed file format. The compressed format has a `.zip` file type and is the default. The uncompressed format is an XML file and can have any file type. The `import-package` command accepts both formats.

You cannot export objects with names that contain a tab, underscore, new line, or carriage return character.

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 Application Services instance.

When you run the import command, the entire package is imported to the designated server. You cannot selectively import elements from the export package.

The export-package command

The CLI export-package command has the following format.

```
export-package --exportFilePath pathname --fromGroup obgname --CommandOption [--CommandOption] ...]
[--uncompressed]
```

The following table describes the required components of the export-package command.

CLI Command Component	Description
export-package	The export-package command name.
--exportFilePath <i>pathname</i>	Specifies the path name of the export package to be created. For example, --exportFilePath /home/dev/joomla.zip specifies the location and name of the joomla.zip export package. You must use the --uncompressed option when the export package has a .xml file extension.
--fromGroup <i>busgroupname</i>	Specifies the owning business group of the object to export. Policies are excluded. Anyone in the tenant can access a policy, so the owning business group is ignored.
--CommandOption	Specifies the type, name, and version of the object to export. For example, the command option --applicationVersion distmaps:1.0.0 specifies an application with the name distmaps and the version 1.0.0. An export-package command can have multiple command options, but must have at least one command option.

The following table describes the command options you can use when exporting one or more objects.

CLI Command Option	Description
--applicationVersion <i>Name:VersionID</i>	Specifies the object and version to export.
--serviceVersion <i>Name:VersionID</i>	<ul style="list-style-type: none"> ■ <i>Name</i> is the name of the application, service, script task, external service, policy, or artifact repository to export. A name is case-insensitive. If a name contains a space, enclose it in double quotes. ■ <i>VersionID</i> is the version identifier for the object in the form <i>major . minor . micro {-qualifier}</i>, for example 1.0.0-beta.
--scriptTaskVersion <i>Name:VersionID</i>	
--externalServiceVersion <i>Name:VersionID</i>	
--policyVersion <i>Name:VersionID</i>	You can export multiple objects of the same type with a single export command by separating the version names with commas. If a version name in a list of version names contains a space, enclose the entire list in double quotes.

```
export-package --exportFilePath pkgname.zip --fromGroup Dev
--applicationVersion "aname1:1.0.0,aname two:1.0.0,aname3:1.0.0"
```

CLI Command Option	Description
<code>--repoVersion Name:1.0.0</code>	<p>You can also export multiple types of objects with the same export command.</p> <pre>export-package --exportFilePath pkgname.zip --fromGroup Dev --applicationVersion aname:1.0.0 --serviceVersion sname:1.0.0 -- scriptTaskVersion stname:1.0.0</pre> <p>The ALL keyword exports all versions of the object type in the specified business group.</p> <pre>export-package --exportFilePath pkgname.zip --fromGroup Dev --applicationVersion ALL --serviceVersion ALL -- scriptTaskVersion ALL</pre>
<code>--uncompressed</code>	Export the package in uncompressed format. Required for export packages with a .xml file extension.

Exporting dependencies

When you export an object, you also export its dependencies. For a high-level object like an application, dependencies can include services, VM templates, script tasks, and more. The export command validates the owning business group of the top-level object, but not of the dependent objects which might belong to different business groups. The command simply exports the dependent objects as part of the top-level object. For example, suppose an application belongs to the Development business group but a service in the application belongs to the Production business group. A member of the Development group can export the application which includes the service. However, a member of the Production business group can export only the service but not the application. Of course, a member of both business groups can export both objects together or independently.

Export requirements

The following table shows when you must belong to the owning business group to export or import an object.

Exported Object	Business Group Requirements
Application	The following tasks require membership in the owning business group.
Service	<ul style="list-style-type: none"> Export the object.
External Service	<ul style="list-style-type: none"> Export the private object as a dependency.
VmTemplate	<p>The following tasks allow membership in any business group in the tenant.</p> <ul style="list-style-type: none"> View a shared application or service. Export the shared object as a dependency.
Script Task (always shared)	<p>The following tasks require membership in the owning business group.</p> <ul style="list-style-type: none"> Export the object. <p>The following task allows membership in any business group in the tenant.</p> <ul style="list-style-type: none"> View the object.

Exported Object	Business Group Requirements
Policy (always shared)	The following tasks allow membership in any business group in the tenant. <ul style="list-style-type: none"> Export the object.
Artifact repository (always shared)	The following tasks require membership in the owning business group. <ul style="list-style-type: none"> Export the object.

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, artifact repositories, services, logical templates, and available custom tasks between different Application Services instances.

You can import packages created with the export command only. You cannot import a package that has been altered or created by any means other than the export command.

You must belong to the owning business group of every object in the import package to import the objects.

When you run the import-package command, you import the entire package to the targeted server. You cannot selectively import objects from the package.

You cannot import external services with the same name into different business groups, and you cannot import an external service as a dependency if it is owned by another business group.

- Make sure the application that contained the deployment profile in the source appliance, is either a shared application or privately owned by the target group in the target appliance. Importing a deployment profile into a private application owned by any business group other than the target group is not supported.
- Make sure the blueprint that was used when creating the deployment profile is not changed when importing, any slight change in the blueprint will not allow nodes and components to map and this use case is not supported.

You can import a deployment profile as a dependency of a shared application, or as a dependency of a private application owned by the target business group. Also ensure that the blueprint used when creating the deployment profile does not change during the import because the slightest change prevents nodes and components from mapping. 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](#).

For the import process to finish successfully, the objects in the package cannot include non-ASCII characters.

The import-package command

The CLI import-package command has the following format.

```
import-package --importFilePath pathname --targetGroup obgname --ConflictResolutionAction action
```

The following table describes the required components of the import-package command.

CLI Command Component	Description
<code>import-package</code>	The import-package command name.
<code>--importFilePath <i>pathname</i></code>	Specifies the path name of the import package. For example, <code>--importFilePath /home/dev/joomla.zip</code> specifies the location and name of the <code>joomla.zip</code> import package.
<code>--targetGroup <i>busgroupname</i></code>	Specifies the owning business group of the object to import. Policies are excluded. Anyone in the tenant can access a policy, so the owning business group is ignored.
<code>--ConflictResolutionAction <i>action</i></code>	Specifies the CHECK, SKIP, OVERWRITE, or IMPORTASNEW conflict resolution action. An import-package command must specify one conflict resolution action.

The following table describes the conflict resolution actions to use when importing objects with an import package.

Conflict Resolution Action	Description
CHECK	Lists all objects in the import package and indicates whether an object exists on the target. Determines a match by comparing the object names and version identifiers. A version identifier has the form <i>major.minor.micro.qualifier</i> , for example, <code>2.0.1.beta</code> .
SKIP	Copies the object from the package if the object does not exist on the target. Does not copy the object if the object exists on the target. A copied object belongs to the owning business group specified by <code>--TargetGroup</code> .
OVERWRITE	Overwrites the contents of an object with contents from the package if the object exists on the target. Copies an object that does not exist on the target to the owning business group specified by <code>--TargetGroup</code> . The logged-in user must belong to the owning business group of the objects in the package to import them.
IMPORTASNEW	Copies an object from the package to the target with a new name if the object exists on the target. The new object belongs to the owning business group specified by <code>--TargetGroup</code> . You must specify a suffix for the renamed object with the <code>--suffix</code> option.

The following table describes the command options available.

CLI Command Option	Description
<code>--shared</code>	Shares the new items in the package with the members of all business groups in the tenant. If omitted, makes the new items in the package private to the members of the owning business group.
<code>--suffix <i>sfxtext</i></code>	Adds an underscore and the specified text represented by <i>sfxtext</i> to the end of every object copied from the package as a match. For example, if you specify <code>--suffix NEW</code> and copy <code>apache:1.0.0</code> from the package, the new name of the object is <code>apache _NEW</code> with version <code>1.0.0</code> . The owning business group for the new object is the target group specified by <code>--targetGroup</code> .

Example: SKIP and OVERWRITE

```
import-package --importFilePath /home/dev/dukes.zip --conflictResolutionAction SKIP --targetGroup  
Development
```

```
import-package --importFilePath /home/dev/dukes.zip --conflictResolutionAction OVERWRITE --targetGroup  
Development
```