Installing and Configuring VMware vRealize Orchestrator

February 2022

vRealize Orchestrator 8.7

vmware®

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

https://docs.vmware.com/

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

Contents

Installing and Configuring VMware vRealize Orchestrator 6

1 Introduction to VMware vRealize Orchestrator 7
- Key Features of the Orchestrator Platform 7
- vRealize Orchestrator User Roles 9
- vRealize Orchestrator Architecture 11
- vRealize Orchestrator Plug-Ins 11

2 vRealize Orchestrator System Requirements 13
- Default Appliance Components 13
- Hardware Requirements 14
- Scalability Maximums 14
- Network Requirements 14
- Ports and Endpoints 15
- Browser Support 15
- Internationalization Support 15

3 Setting Up vRealize Orchestrator Components 17
- vCenter Server Setup 17
- Authentication Methods 17

4 Installing vRealize Orchestrator 19
- Download and Deploy the vRealize Orchestrator Appliance 19
  - Power on the vRealize Orchestrator Appliance and Open the Home Page 21
  - Enable or Disable SSH Access to the vRealize Orchestrator Appliance 22

5 Initial Configuration 23
- Configuring a Standalone vRealize Orchestrator Server 23
  - Configure a Standalone vRealize Orchestrator Server with vRealize Automation Authentication 23
  - Configure a Standalone vRealize Orchestrator Server with vSphere Authentication 25
- vRealize Orchestrator Feature Enablement with Licenses 26
- vRealize Orchestrator Database Connection 27
- Manage Certificates 27
  - Manage vRealize Orchestrator Certificates 28
    - Generate a Custom TLS Certificate for vRealize Orchestrator 28
    - Set a Custom TLS Certificate for vRealize Orchestrator 29
    - Import a Trusted Certificate with the Control Center 32

Configuring the vRealize Orchestrator Plug-Ins 32
Manage vRealize Orchestrator Plug-Ins 32
Install or Update a vRealize Orchestrator Plug-In 33
Delete a Plug-In 33
vRealize Orchestrator High Availability 34
Scalability Maximums 34
Configure a vRealize Orchestrator Cluster 35
Removing an vRealize Orchestrator Cluster Node 37
Scale Out a Standalone vRealize Orchestrator Deployment 37
Monitoring an vRealize Orchestrator Cluster 38
Configuring the Customer Experience Improvement Program 39
Categories of Information That VMware Receives 39
Join or Leave the Customer Experience Improvement Program 39

6 Using the vRealize Orchestrator API Services 41
Managing SSL Certificates Through the REST API 41
Delete a TLS Certificate by Using the REST API 42
Import TLS Certificates by Using the REST API 42
Create a Keystore by Using the REST API 43
Delete a Keystore by Using the REST API 44
Add a Key by Using the REST API 44

7 Additional Configuration Options 46
Reconfiguring Authentication 46
Change the Authentication Provider 46
Change the Authentication Parameters 47
Configuring the Workflow Run Properties 47
vRealize Orchestrator Log Files 48
Logging Persistence 48
vRealize Orchestrator Logs Configuration 49
Configure Logging Integration with vRealize Log Insight 49
Create or Overwrite a Syslog Integration in vRealize Orchestrator 50
Delete a Syslog Integration in vRealize Orchestrator 51
Enable Kerberos Debug Logging 52
Enabling the Opentracing and Wavefront Extensions 52
Configure the Opentracing Extension 53
Configure the Wavefront Extension 54
Enable Time Synchronization for vRealize Orchestrator 55
Deactivate Time Synchronization for vRealize Orchestrator 57
Configure vRealize Orchestrator Kubernetes CIDR 57
Update the DNS Settings for vRealize Orchestrator 58

8 Configuration Use Cases and Troubleshooting 60
Verify the vRealize Orchestrator server build number 60
Configure vRealize Orchestrator Plugin for vSphere Web Client 61
Cancel Running Workflows 62
Enable vRealize Orchestrator Server Debugging 62
Resize the vRealize Orchestrator Appliance Disks 64
How to Scale the Heap Memory Size of the vRealize Orchestrator Server 65
Disaster Recovery of vRealize Orchestrator by Using Site Recovery Manager 66
Configure Virtual Machines for vSphere Replication 66
Create Protection Groups 67
Create a Recovery Plan 69
Organize Recovery Plans in Folders 70
Edit a Recovery Plan 70

9 Setting System Properties 72
Setting Server File System Access for Workflows and Actions 72
Rules in the js-io-rights.conf File Permitting Write Access to the vRealize Orchestrator System 72
Set Server File System Access for Workflows and Actions 73
Set Access to Operating System Commands for Workflows and Actions 74
Set JavaScript Access to Java Classes 75
Set Custom Timeout Property 76
Adding a JDBC Connector for the vRealize Orchestrator SQL Plug-In 77
Set Scheduled Task and Policy Authentication Token Renewal Property 78

10 Where to Go from Here 79

Installing and Configuring VMware vRealize Orchestrator

Installing and Configuring VMware vRealize Orchestrator provides information and instructions about installing and configuring VMware® vRealize Orchestrator.

Intended Audience

This information is intended for advanced vSphere administrators and experienced system administrators who are familiar with virtual machine technology and data center operations.

Introduction to VMware vRealize Orchestrator

1

VMware vRealize Orchestrator is a development- and process-automation platform that provides a library of extensible workflows to allow you to create and run automated, configurable processes to manage VMware products as well as other third-party technologies.

vRealize Orchestrator automates management and operational tasks of both VMware and third-party applications such as service desks, change management systems, and IT asset management systems.

This chapter includes the following topics:

- Key Features of the Orchestrator Platform
- vRealize Orchestrator User Roles
- vRealize Orchestrator Architecture
- vRealize Orchestrator Plug-Ins

Key Features of the Orchestrator Platform

vRealize Orchestrator is composed of three distinct layers: an orchestration platform that provides the common features required for an orchestration tool, a plug-in architecture to integrate control of subsystems, and a library of workflows. vRealize Orchestrator is an open platform that can be extended with new plug-ins and content, and can be integrated into larger architectures through a REST API.

vRealize Orchestrator includes several key features that help with running and managing workflows.

Persistence

A production-grade PostgreSQL database is used to store relevant information, such as processes, workflow states, and the vRealize Orchestrator configuration.

Central management

vRealize Orchestrator provides a central tool to manage your processes. The application server-based platform, with full version history, can store scripts and process-related primitives in the same storage location. This way, you can avoid scripts without versioning and proper change control on your servers.

Check-pointing

Every step of a workflow is saved in the database, which prevents data-loss if you must restart the server. This feature is especially useful for long-running processes.

Control Center

Control Center is a Web-based portal that increases the administrative efficiency of vRealize Orchestrator instances by providing a centralized administrative interface for runtime operations, workflow monitoring, and correlation between the workflow runs and system resources.

Versioning

All vRealize Orchestrator platform objects have an associated version history. Version history is useful for basic change management when distributing processes to project stages or locations.

Git integration

With the vRealize Orchestrator Client, you can integrate a Git repository to further improve version and source control of your vRealize Orchestrator content. With Git, you can manage workflow development across multiple vRealize Orchestrator instances. See *Using Git with the vRealize Orchestrator Client* in the *Using the VMware vRealize Orchestrator Client* guide.

Scripting engine

The Mozilla Rhino JavaScript engine provides a way to create building blocks for the vRealize Orchestrator Client platform. The scripting engine is enhanced with basic version control, variable type checking, name space management, and exception handling. The engine can be used in the following building blocks:

- Actions
- Workflows
- Policies

Workflow engine

The workflow engine allows you to automate business processes. It uses the following objects to create a step-by-step process automation in workflows:

- Workflows and actions that vRealize Orchestrator Client provides.
- Custom building blocks created by the customer.
- Objects that plug-ins add to vRealize Orchestrator Client.

Users, other workflows, schedules, or policies can start workflows.

Policy engine

You can use the policy engine to monitor and generate events to react to changing conditions in the vRealize Orchestrator Client server or a plugged-in technology. Policies can aggregate

events from the platform or the plug-ins, which helps you to handle changing conditions on any of the integrated technologies.

vRealize Orchestrator Client

Create, run, edit, and monitor workflows with the vRealize Orchestrator Client. You can also use the vRealize Orchestrator Client to manage action, configuration, policy, and resource elements. See *Using the vRealize Orchestrator Client*.

Development and resources

The vRealize Orchestrator landing page provides quick access to resources to help you develop your own plug-ins, for use in vRealize Orchestrator. You will also find information about using the vRealize Orchestrator REST API to send requests to the vRealize Orchestrator server.

Security

vRealize Orchestrator provides the following advanced security functions:

- Public Key Infrastructure (PKI) to sign and encrypt content imported and exported between servers.
- Digital Rights Management (DRM) to control how exported content can be viewed, edited, and redistributed.
- Transport Layer Security (TLS) to provide encrypted communications between the vRealize Orchestrator Client, vRealize Orchestrator server, and HTTPS access to the Web front end.
- Advanced access rights management to provide control over access to processes and the objects manipulated by these processes.

Encryption

vRealize Orchestrator uses a FIPS-compliant Advanced Encryption Standard (AES) with a 256-bit cipher key for encryption of strings. The cipher key is randomly generated and is unique across appliances that are not part of a cluster. All nodes in a cluster share a cipher key.

vRealize Orchestrator User Roles

vRealize Orchestrator provides different tools and interfaces based on the specific responsibilities of the global user roles. In vRealize Orchestrator you can have users with full rights, that are a part of the administrator group (**administrators**), developers (**workflow designers**), troubleshooting users (**viewers**), and users with limited access.

vRealize Orchestrator user roles are managed in the **Role Management** menu of the vRealize Orchestrator Client. For more information on configuring user roles in the vRealize Orchestrator Client, see *Assign Roles in the vRealize Orchestrator Client* in the *Using the VMware vRealize Orchestrator Client* guide.

Note For vRealize Orchestrator deployments authenticated with vRealize Automation, or using a vRealize Automation license, user roles are assigned with the Identity and Access Management service of the vRealize Automation platform. See *Configure vRealize Orchestrator Client Roles in vRealize Automation* in *Using the VMware vRealize Orchestrator Client*.

User Role	Description
Administrator	This user has full access to all vRealize Orchestrator platform capabilities and content, including content created by specific groups. Primary administrator user responsibilities include: ■ Installing and configuring vRealize Orchestrator. ■ Adding users to the vRealize Orchestrator Client, assigning roles, and creating and deleting groups. See *Create Groups in the vRealize Orchestrator Client* in *Using the VMware vRealize Orchestrator Client*. ■ Creating an integration with a Git repository for the developers in their vRealize Orchestrator environment. See *Configure a Connection to a Git Repository* in *Using the VMware vRealize Orchestrator Client*. ■ Troubleshooting their vRealize Orchestrator environment through features like workflow validation and debugging workflow scripts.
Viewer	This user has read-only access to all vRealize Orchestrator Client, including all groups and group content. This user can view but cannot create, edit, or run content, or export workflow runs, workflow run logs, or packages. Viewers are not limited by group permissions. **Note** The viewer role is supported only for vRealize Orchestrator instances authenticated with vRealize Automation. This role is not mapped to a vRealize Automation role by default so it must be explicitly assigned to users.
Workflow Designer	This user can extend the vRealize Orchestrator platform functionality by creating and editing objects. Workflow designers do not have access to the administrative and troubleshooting features of the vRealize Orchestrator Client. Primary workflow designer responsibilities include: ■ Creating, editing, running, and deleting vRealize Orchestrator objects like workflows, actions, policies, and configuration elements. ■ Scheduling workflow runs. See *Schedule Workflows in the vRealize Orchestrator Client* in *Using VMware vRealize Orchestrator Client*. ■ Adding content created by the workflow developer to groups they are assigned to. ■ Pushing local changes to the vRealize Orchestrator content inventory to the connect Git repository. See *Push Changes to a Git Repository* in *Using VMware vRealize Orchestrator Client*.
Users with limited rights	Users with no assigned role can still log in to the vRealize Orchestrator Client, but have limited access to client features and content. If they are assigned to a group, this user can view and run content included in that group.

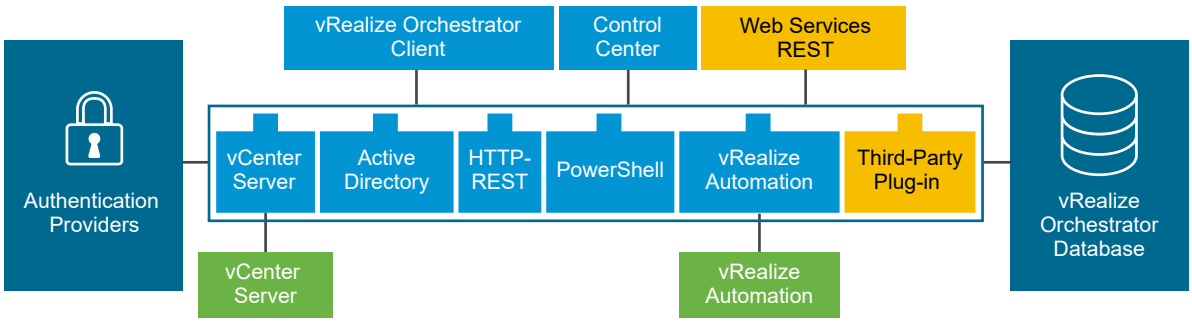
vRealize Orchestrator Architecture

vRealize Orchestrator contains a workflow library and a workflow engine to allow you to create and run workflows that automate orchestration processes. You run workflows on the objects of different technologies that vRealize Orchestrator accesses through a series of plug-ins.

vRealize Orchestrator provides a standard set of plug-ins, including plug-ins for vCenter Server and vRealize Automation, to allow you to orchestrate tasks in the different environments that the plug-ins expose.

vRealize Orchestrator also presents an open architecture for plugging in external third-party applications to the orchestration platform. You can run workflows on the objects of the plugged-in technologies that you define yourself. vRealize Orchestrator connects to an authentication provider to manage user accounts and to a preconfigured PostgreSQL database to store information from the workflows that it runs. You can access vRealize Orchestrator, the objects it exposes, and the vRealize Orchestrator workflows through the vRealize Orchestrator Client, or through Web services. Monitoring and configuration of vRealize Orchestrator workflows and services is done through the vRealize Orchestrator Client and Control Center.

Figure 1-1. VMware vRealize Orchestrator Architecture



vRealize Orchestrator Plug-Ins

Plug-ins allow you to use vRealize Orchestrator to access and control external technologies and applications. By exposing an external technology in an vRealize Orchestrator plug-in, you can incorporate objects and functions in workflows that access the objects and functions of that external technology.

The external technologies that you can access by using plug-ins include virtualization management tools, email systems, databases, directory services, and remote-control interfaces.

vRealize Orchestrator provides a set of standard plug-ins that you can use to incorporate into workflows such technologies as the VMware vCenter Server API and email capabilities. By using the plug-ins, you can automate the delivery of new IT services or adapt the capabilities of existing infrastructure and application services. In addition, you can use the vRealize Orchestrator open plug-in architecture to develop plug-ins for accessing other applications.

The vRealize Orchestrator plug-ins that VMware develops are distributed as `.vmoapp` files.

For more information about the vRealize Orchestrator plug-ins, see Using the VMware vRealize Orchestrator Plug-Ins.

For more information about third-party vRealize Orchestrator plug-ins, see VMware Marketplace.

vRealize Orchestrator System Requirements

2

Your system must meet the technical requirements that are necessary for vRealize Orchestrator to work properly.

For a list of the supported versions of vCenter Server, the vSphere Web Client, vRealize Automation, and other VMware solutions, see VMware Product Interoperability Matrix.

This chapter includes the following topics:

- vRealize Orchestrator Appliance Components
- Hardware Requirements for the vRealize Orchestrator Appliance
- vRealize Orchestrator Scalability Maximums
- Network Requirements for vRealize Orchestrator
- vRealize Orchestrator Ports and Endpoints
- Browsers Supported by vRealize Orchestrator
- Level of Internationalization and Localization Support

vRealize Orchestrator Appliance Components

The vRealize Orchestrator Appliance is a Photon-based virtual appliance running in containers.

The vRealize Orchestrator Appliance includes the following components:

- An infrastructure level Kubernetes layer.
- A preconfigured PostgreSQL database.
- The core vRealize Orchestrator services: the server service, Control Center service, and orchestration UI service.

The default vRealize Orchestrator Appliance database configuration is production ready.

Note To use the vRealize Orchestrator Appliance in a production environment, you must configure the vRealize Orchestrator server to authenticate through vRealize Automation or vSphere.See Configuring a Standalone vRealize Orchestrator Server.

Hardware Requirements for the vRealize Orchestrator Appliance

The vRealize Orchestrator Appliance is a preconfigured Photon-based virtual machine that runs in containers. Before you deploy the appliance, verify that your system meets the minimum hardware requirements.

The vRealize Orchestrator Appliance has the following hardware requirements:

- 4 CPUs
- 12 GB of memory
- 200 GB hard disk

Do not reduce the default memory size, because the vRealize Orchestrator server requires at least 8 GB of free memory.

vRealize Orchestrator Scalability Maximums

The scalability limit table outlines the recommended maximums on vRealize Orchestrator 8.x deployments.

Component	Scale targets	More information
Virtual machines	35,000	
vCenter Server connections	10	See vCenter Server Setup
Active nodes in a cluster	3	See Configure a vRealize Orchestrator Cluster
Concurrent running workflows	300 per node	See Configuring the Workflow Run Properties
Queued running workflows	10,000 per node	
Preserved workflow runs	100 per node	
Log event expiration days	15	

Network Requirements for vRealize Orchestrator

Each vRealize Orchestrator node requires a network setup.

The network requirements for vRealize Orchestrator are:

- Single, static IPv4 and Network Address
- Reachable DNS server set manually

- Valid fully-qualified domain name (FQDN) set manually that can be resolved both forward and in reverse through the DNS server

Note IP address change or hostname change after installation is not supported and results in a broken setup that is not recoverable.

vRealize Orchestrator Ports and Endpoints

The vRealize Orchestrator Kubernetes service includes two endpoints and several main network ports.

vRealize Orchestrator Network Ports

You can access vRealize Orchestrator over port 443. The 443 port is secured with a self-signed certificate that is generated during the installation. When using an external load balancer, it must be set up to balance on port 443.

To view all vRealize Orchestrator ports, refer to the Ports and Protocols tool.

vRealize Orchestrator Endpoints

You can access the vRealize Orchestrator client and Control Center services at the following endpoints.

Service	Endpoint
vRealize Orchestrator Client	`https://your_orchestrator_FQDN/orchestration-ui`
Control Center	`https://your_orchestrator_FQDN/vco-controlcenter`

Browsers Supported by vRealize Orchestrator

Confirm that your browsers support vRealize Orchestrator.

To access the vRealize Orchestrator Client and Control Center, you must use one of the following browsers:

- Microsoft Edge
- Mozilla Firefox
- Google Chrome

Level of Internationalization and Localization Support

The vRealize Orchestrator Control Center and vRealize Orchestrator Client include support for non-English operating systems, non-English data formatting, and multi-language support for the Control Center and client user interface.

The vRealize Orchestrator Control Center and vRealize Orchestrator Client support the use of non-English operating systems, non-English input and output, and support for non-English formatting of data such as dates, time, and numbers.

The user interfaces of the vRealize Orchestrator and vRealize Orchestrator Client are localized to the following languages:

- Spanish
- French
- German
- Traditional Chinese
- Simplified Chinese
- Korean
- Japanese
- Italian
- Dutch
- Brazilian Portuguese
- Russian

Setting Up vRealize Orchestrator Components

3

When you download and deploy the vRealize Orchestrator Appliance, the vRealize Orchestrator server is preconfigured. After deployment, the services start automatically.

To enhance the availability and scalability of your vRealize Orchestrator setup, follow these guidelines:

- Install and configure an authentication provider and configure vRealize Orchestrator to work with the provider. See Configuring a Standalone vRealize Orchestrator Server.
- For clustered vRealize Orchestrator environments, install and configure a load balancing server and configure it to distribute the workload between the vRealize Orchestrator servers.

This chapter includes the following topics:

- vCenter Server Setup
- Authentication Methods

vCenter Server Setup

Increasing the number of vCenter Server instances in your vRealize Orchestrator setup causes vRealize Orchestrator to manage more sessions. Too many active sessions can cause vRealize Orchestrator to experience timeouts when more than 10 vCenter Server connections occur.

For a list of the supported versions of vCenter Server, see the VMware Product Interoperability Matrix.

Note If your network has sufficient bandwidth and latency, you can run multiple vCenter Server instances on different virtual machines in your vRealize Orchestrator setup. If you are using LAN to improve the communication between vRealize Orchestrator and vCenter Server, a 100-Mb line is mandatory.

Authentication Methods

To authenticate and manage user permissions, vRealize Orchestrator requires a connection to either vRealize Automation or a vSphere server instance.

When you download, and deploy vRealize Orchestrator Appliance, you must configure the server with a vRealize Automation or vSphere authentication. See Configuring a Standalone vRealize Orchestrator Server.

Note vRealize Orchestrator 8.x authentication with vRealize Automation is only supported with vRealize Automation 8.x.

Installing vRealize Orchestrator 4

vRealize Orchestrator consists of a server component and a client component.

To use vRealize Orchestrator, you must deploy the vRealize Orchestrator Appliance and configure the vRealize Orchestrator server.

You can change the default vRealize Orchestrator configuration settings by using the vRealize Orchestrator Control Center.

This chapter includes the following topics:

- Download and Deploy the vRealize Orchestrator Appliance

Download and Deploy the vRealize Orchestrator Appliance

Before you can access the vRealize Orchestrator content and services, you must download and deploy the vRealize Orchestrator Appliance.

Prerequisites

- Verify that you have a running vCenter Server instance. The vCenter Server version must be 6.0 or later.
- Verify that the host on which you are deploying the vRealize Orchestrator Appliance meets the minimum hardware requirements. See Hardware Requirements for the vRealize Orchestrator Appliance.
- If your system is isolated and without Internet access, you must download the `.ova` file for the appliance from the VMware website.

Procedure

1. Log in to the vSphere Web Client as an **administrator**.
2. Select an inventory object that is a valid parent object of a virtual machine, such as a data center, folder, cluster, resource pool, or host.
3. Select **Actions > Deploy OVF Template**.
4. Enter the file path or the URL to the `.ova` file and click **Next**.
5. Enter a name and location for the vRealize Orchestrator Appliance, and click **Next**.

6 Select a host, cluster, resource pool, or vApp as a destination on which you want the appliance to run, and click **Next**.

7 Review the deployment details, and click **Next**.

8 Accept the terms in the license agreement and click **Next**.

9 Select the storage format you want to use for the vRealize Orchestrator Appliance.

Format	Description
Thick Provisioned Lazy Zeroed	Creates a virtual disk in a default thick format. The space required for the virtual disk is allocated when the virtual disk is created. If any data remains on the physical device, it is not erased during creation, but is zeroed out on demand later on first write from the virtual machine.
Thick Provisioned Eager Zeroed	Supports clustering features such as Fault Tolerance. The space required for the virtual disk is allocated when the virtual disk is created. If any data remains on the physical device, it is zeroed out when the virtual disk is created. It might take much longer to create disks in this format than to create disks in other formats.
Thin Provisioned Format	Saves hard disk space. For the thin disk, you provision as much datastore space as the disk requires based on the value that you select for the disk size. The thin disk starts small and, at first, uses only as much datastore space as the disk needs for its initial operations.

10 Click **Next**.

11 Configure the network settings and enter the **root** password.

When configuring the network settings of the vRealize Orchestrator Appliance, you must use the IPv4 protocol. For both DHCP and Static network configurations, you must add a fully qualified domain name (FQDN) for your vRealize Orchestrator Appliance.

If the host name displayed in the shell of the deployed vRealize Orchestrator Appliance is *photon-machine*, the preceding network configuration requirements are not met.

12 (Optional) Configure additional network settings for the vRealize Orchestrator Appliance, such as enabling SSH access.

Note When configuring a Kubernetes network, the values of the internal cluster CIDR and internal service CIDR must allow for at least 1024 hosts. Because of this requirement, the network mask value must be 22 or less. Network mask values higher than 22 are invalid. The Kubernetes network properties have to following default values:

Kubernetes network property	Default value	Property description
Kubernetes internal cluster CIDR	10.244.0.0/22	The CIDR used for pods running inside the Kubernetes cluster.
Kubernetes internal service CIDR	10.244.4.0/22	The CIDR used for Kubernetes services inside the Kubernetes cluster.

Note You can also change the Kubernetes CIDR network properties after deployment. See Configure vRealize Orchestrator Kubernetes CIDR.

13 (Optional) To enable FIPS mode for the vRealize Orchestrator Appliance, set **FIPS Mode** to **`strict`**.

Note FIPS 140-2 enablement is supported only for new vRealize Orchestrator environments. If you want to enable FIPS mode on your environment, you must do so during installation.

14 Click **Next**.

15 Review the **Ready to complete** page and click **Finish**.

Results

The vRealize Orchestrator Appliance is successfully deployed.

What to do next

Log in to the vRealize Orchestrator Appliance command line as **root** and confirm that you can perform a forward or reverse DNS lookup.

- To perform a forward DNS lookup, run the `nslookup` *your_orchestrator_FQDN* command. The command must return the vRealize Orchestrator Appliance IP address.
- To perform a reverse DNS lookup, run the `nslookup` *your_orchestrator_IP* command. The command must return the vRealize Orchestrator Appliance FQDN.

Note If you have not enabled SSH during deployment, you can also perform DNS lookups from the virtual machine console in the vSphere Web Client.

Power on the vRealize Orchestrator Appliance and Open the Home Page

To use the standalone vRealize Orchestrator Appliance, you must first power it on.

Procedure

1 Log in to the vSphere Web Client as an **administrator**.

2 Right-click the vRealize Orchestrator Appliance and select **Power > Power On**.

3 In a Web browser, navigate to the host address of your vRealize Orchestrator Appliance virtual machine that you configured during the OVA deployment.

   https://*your_orchestrator_FQDN*/vco.

Enable or Disable SSH Access to the vRealize Orchestrator Appliance

You can enable or disable SSH access to the vRealize Orchestrator Appliance.

Prerequisites

- Download and deploy the vRealize Orchestrator Appliance.
- Verify that the vRealize Orchestrator Appliance is up and running.

Procedure

1 Log in to the vRealize Orchestrator Appliance command line as **root**.

2 To enable SSH access, run the `/usr/bin/toggle-ssh enable` command.

3 To disable SSH access, run the `/usr/bin/toggle-ssh disable` command.

Initial Configuration

5

Before you begin automating tasks and managing systems and applications with vRealize Orchestrator, you must use the vRealize Orchestrator Control Center to configure an external authentication provider. You can also use the vRealize Orchestrator Control Center for additional configuration tasks such as managing license and certificate information, installing plug-ins, and monitoring the state of your vRealize Orchestrator cluster.

This chapter includes the following topics:

- Configuring a Standalone vRealize Orchestrator Server
- vRealize Orchestrator Feature Enablement with Licenses
- vRealize Orchestrator Database Connection
- Manage Certificates
- Configuring the vRealize Orchestrator Plug-Ins
- vRealize Orchestrator High Availability
- Configuring the Customer Experience Improvement Program

Configuring a Standalone vRealize Orchestrator Server

Although the vRealize Orchestrator Appliance is a preconfigured Photon-based virtual machine, you must configure an authentication provider before you access the full functionality of the vRealize Orchestrator Control Center and vRealize Orchestrator Client.

Configure a Standalone vRealize Orchestrator Server with vRealize Automation Authentication

To prepare the vRealize Orchestrator Appliance for use, you must configure the host settings and the authentication provider. You can configure vRealize Orchestrator to authenticate with vRealize Automation. Use vRealize Automation authentication with vRealize Automation 8.x.

Prerequisites

- Download and deploy the latest version of the vRealize Orchestrator Appliance. See Download and Deploy the vRealize Orchestrator Appliance.

- Install and configure vRealize Automation 8.x and verify that your vRealize Automation server is running. See the vRealize Automation documentation.

  Important The product version of the vRealize Automation authentication provider must match the product version your vRealize Orchestrator deployment. For example, to authenticate a vRealize Orchestrator 8.7 deployment, you must use a vRealize Automation 8.7 deployment.

If you plan to create a cluster:

- Set up a load balancer to distribute traffic among multiple instances of vRealize Orchestrator. See VMware vRealize Orchestrator 8.x Load Balancing Guide.

Procedure

1. Access the Control Center to start the configuration wizard.
   a. Navigate to https://*your_orchestrator_FQDN*/vco-controlcenter.
   b. Log in as **root** with the password you entered during OVA deployment.
2. Configure the authentication provider.
   a. On the **Configure Authentication Provider** page, select **vRealize Automation** from the **Authentication mode** drop-down menu.
   b. In the **Host address** text box, enter your vRealize Automation host address and click **CONNECT**.

      The format of the vRealize Automation host address must be https://*your_vra_hostname*.
   c. Click **Accept Certificate**.
   d. Enter the credentials of the vRealize Automation organization owner under which vRealize Orchestrator will be configured. Click **REGISTER**.
   e. Click **SAVE CHANGES**.

      A message indicates that your configuration is saved successfully.

Results

You have successfully finished the vRealize Orchestrator server configuration.

What to do next

- Verify that **CSP** is the configured license provider at the **Licensing** page.
- Verify that the node is configured properly at the **Validate Configuration** page.

  Note Following the configuration of the authentication provider, the vRealize Orchestrator server restarts automatically after 2 minutes. Verifying the configuration immediately after authentication can return an invalid configuration status.

Configure a Standalone vRealize Orchestrator Server with vSphere Authentication

You register the vRealize Orchestrator server with a vCenter Single Sign-On server by using the vSphere authentication mode. Use vCenter Single Sign-On authentication with vCenter Server 6.0 and later.

Prerequisites

- Download and deploy the latest version of the vRealize Orchestrator Appliance. See Download and Deploy the vRealize Orchestrator Appliance.
- Install and configure a vCenter Server with vCenter Single Sign-On running. See the vSphere documentation.

If you plan to create a cluster:

- Set up a load balancer to distribute traffic among multiple instances of vRealize Orchestrator. See VMware vRealize Orchestrator 8.x Load Balancing Guide.

Procedure

1 Access the Control Center to start the configuration wizard.

   a Navigate to https://*your_orchestrator_FQDN*/vco-controlcenter.

   b Log in as **root** with the password you entered during OVA deployment.

2 Configure the authentication provider.

   a On the **Configure Authentication Provider** page, select **vSphere** from the **Authentication mode** drop-down menu.

   b In the **Host address** text box, enter the fully qualified domain name or IP address of the Platform Services Controller instance that contains the vCenter Single Sign-On and click **Connect**.

   Note If you use an external Platform Services Controller or multiple Platform Services Controller instances behind a load balancer, you must manually import the certificates of all Platform Services Controllers that share a vCenter Single Sign-On domain.

   Note To integrate a different vSphere Client with your configured vRealize Orchestrator environment, you must configure vSphere to use the same Platform Services Controller registered to vRealize Orchestrator. For High Availability vRealize Orchestrator environments, you must replicate the PCS instances behind the vRealize Orchestrator load balancer server.

   c Review the certificate information of the authentication provider and click **Accept Certificate**.

d Enter the credentials of the local administrator account for the vCenter Single Sign-On domain. Click **REGISTER**.

By default, this account is **`administrator@vsphere.local`** and the name of the default tenant is **`vsphere.local`**.

e In the **Admin group** text box, enter the name of an administrators group and click **SEARCH**.

For example, **`vsphere.local\vcoadmins`**

f Select the administration group you want to use.

g Click **SAVE CHANGES**.

A message indicates that your configuration is saved successfully.

Results

You have successfully finished the vRealize Orchestrator server configuration.

What to do next

- Verify that **CIS** is the configured license provider at the **Licensing** page.
- Verify that the node is configured properly at the **Validate Configuration** page.

Note Following the configuration of the authentication provider, the vRealize Orchestrator server restarts automatically after 2 minutes. Verifying the configuration immediately after authentication can return an invalid configuration status.

vRealize Orchestrator Feature Enablement with Licenses

Access to certain vRealize Orchestrator features is based on the license applied to your vRealize Orchestrator deployment.

After authentication, your vRealize Orchestrator instance is assigned a license based on the authentication provider. Licenses control access to the following vRealize Orchestrator features:

- Git integration
- Role management
- Multi-language support (Python, Node.js, and PowerShell)

You can manually change the license of the vRealize Orchestrator server from the **Licenses** page of the Control Center.

Note There is no limit to the number of vRealize Orchestrator deployments to which you can apply the same license, regardless of the license type. For vRealize Automation licenses, having a deployed and configured vRealize Automation environment is not required.

Authentication	License	Git Integration	Role management	Multi-language support
vSphere	vSphere vCloud Suite Standard	No	No	No
vSphere	vRealize Automation vRealize Suite Advanced or Enterprise vCloud Suite Advanced or Enterprise	Yes	Yes	Yes
vRealize Automation	vRealize Automation vRealize Suite Advanced or Enterprise vCloud Suite Advanced or Enterprise	Yes	Roles are managed from the vRealize Automation instance used to authenticate vRealize Orchestrator.	Yes

Note vRealize Suite Standard licenses do not include vRealize Automation, so they do not support access to vRealize Orchestrator features.

vRealize Orchestrator Database Connection

The vRealize Orchestrator server requires a database for storing data.

The deployed vRealize Orchestrator Appliance includes a preconfigured PostgreSQL database used by the vRealize Orchestrator server to store data.

The postgreSQL database is not accessible for users.

Manage Certificates

Issued for a particular server and containing information about the server public key, the certificate allows you to sign all elements created in vRealize Orchestrator and guarantee authenticity. When the client receives an element from your server, typically a package, the client verifies your identity and decides whether to trust your signature.

- Manage vRealize Orchestrator Certificates

  You can manage the vRealize Orchestrator certificates from the **Certificates** page in the vRealize Orchestrator Control Center or with the vRealize Orchestrator Client, by using the *ssl_trust_manager* tagged workflows .

Manage vRealize Orchestrator Certificates

You can manage the vRealize Orchestrator certificates from the **Certificates** page in the vRealize Orchestrator Control Center or with the vRealize Orchestrator Client, by using the *ssl_trust_manager* tagged workflows .

Import a Certificate to the Orchestrator Trust Store

vRealize Orchestrator Control Center uses a secure connection to communicate with vCenter Server, relational database management system (RDBMS), LDAP, Single Sign-On, and other servers. You can import the required TLS certificate from a URL or a PEM-encoded file.
Each time you want to use a TLS connection to a server instance, you must import the corresponding certificate from the **Trusted Certificates** tab on the **Certificates** page and import the corresponding TLS certificate.

You can load the TLS certificate in vRealize Orchestrator from a URL address or a PEM-encoded file.

Option	Description
Import from URL or proxy URL	The URL of the remote server: `https://your_server_IP_address` or `your_server_IP_address:port`
Import from file	Path to the PEM-encoded certificate file. **Note** You can also import a trusted certificate by running the **Import a trusted certificate from a file** workflow in the vRealize Orchestrator Client. The file imported through this workflow must be DER-encoded.

For more information on importing a certificate, see Import a Trusted Certificate with the Control Center.

Package Signing Certificate

Packages exported from an vRealize Orchestrator server are digitally signed. Import, export, or generate a new certificate to be used for signing packages. Package signing certificates are a form of digital identification that is used to guarantee encrypted communication and a signature for your Orchestrator packages.

The vRealize Orchestrator Appliance includes a package signing certificate that is generated automatically, based on the network settings of the appliance. If the network settings of the appliance change, you must generate a new package signing certificate manually. After generating a new package signing certificate, all future exported packages are signed with the new certificate.

Generate a Custom TLS Certificate for vRealize Orchestrator

You can use the vRealize Orchestrator Appliance to generate a new TLS certificate for your environment or set an existing custom certificate.

The vRealize Orchestrator Appliance includes a Trusted Layer Security (TLS) certificate that is generated automatically, based on the network settings of the appliance. If the network settings of the appliance change, you must generate a new certificate manually. You can create a certificate chain to guarantee encrypted communication and provide a signature for your packages. However, the recipient cannot be sure that the self-signed package is in fact a package issued by your server and not a third party claiming to be you. To prove the identity of your server, use a certificate signed by a Certificate Authority (CA).

vRealize Orchestrator generates a server certificate that is unique to your environment. The private key is stored in the `vmo_keystore` table of the vRealize Orchestrator database.

Note To configure your vRealize Orchestrator Appliance to use an existing custom TLS certificate, see Set a Custom TLS Certificate for vRealize Orchestrator.

Prerequisites

Verify that SSH access for the vRealize Orchestrator Appliance is enabled. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.

Procedure

1. Log in to the vRealize Orchestrator Appliance command line over SSH as **root**.
2. Run the `vracli certificate ingress --generate auto --set stdin` command.
3. To apply the custom certificate to your vRealize Orchestrator Appliance, run the deployment script.
   a. Navigate to the `/opt/scripts/` directory.

   ```
   cd /opt/scripts/
   ```

   b. Run the `./deploy.sh` script.

   Important Do not interrupt the deployment script. You receive the following message when the script finishes running:

   ```
   Prelude has been deployed successfully.
   To access, go to your_orchestrator_address
   ```

What to do next

To confirm that the new certificate chain is applied, run the `vracli certificate ingress --list` command.

Set a Custom TLS Certificate for vRealize Orchestrator

Set a custom TLS Certificate for your vRealize Orchestrator Appliance.

The vRealize Orchestrator Appliance includes a Trusted Layer Security (TLS) certificate that is generated automatically, based on the network settings of the appliance.

You can configure your vRealize Orchestrator Appliance to use an existing custom TLS certificate. You can set the certificate by importing the relevant PEM file from your local machine into the vRealize Orchestrator Appliance. You can also set your custom TLS certificate by copying the certificate chain directly into the vRealize Orchestrator Appliance. Both procedures require you to run the `./deploy.sh` script before the new TLS certificate can be used in your vRealize Orchestrator deployment.

For information on generating a new custom TLS certificate, see Generate a Custom TLS Certificate for vRealize Orchestrator.

Prerequisites

- Verify that SSH access for the vRealize Orchestrator Appliance is enabled. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.
- Verify that the PEM file containing the TLS certificate contains the following components in the set order:
  a The private key for the certificate.
  b The primary certificate.
  c If applicable, the Certificate Authority (CA) intermediate certificate or certificates.
  d The root CA certificate.

  For example, the TLS certificate can have the following structure:

```
 -----BEGIN RSA PRIVATE KEY-----
<Private Key>
-----END RSA PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
<Primary TLS certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Intermediate certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA certificate>
-----END CERTIFICATE-----
```

Procedure

1 Set the certificate by importing the PEM file into the vRealize Orchestrator Appliance.

   a Import the certificate PEM from your local machine by running a secure copy (SCP) command from an SSH shell.

   For Linux, you can use a terminal SCP command:

   ```
   scp ~/PEM_local_filepath/your_cert_file.PEM root@orchestrator_FQDN_or_IP:/
   PEM_orchestrator_filepath/your_cert_file.PEM
   ```

   For Windows, you can use a PuTTY client PSCP command:

   ```
   pscp C:\PEM_local_filepath\your_cert_file.PEM root@<orchestrator_FQDN_or_IP>:/
   PEM_orchestrator_filepath/your_cert_file.PEM
   ```

   b Log in to the vRealize Orchestrator Appliance command line over SSH as **root**.

   c Run the `vracli certificate ingress --set` *your_cert_file.PEM* command.

2 (Optional) Set the certificate by copying the certificate chain directly into the appliance.

   a Log in to the vRealize Orchestrator Appliance command line over SSH as **root**.

   b Run the `vracli certificate ingress --set stdin` command.

   c Copy and paste the certificate chain, and press Ctrl+D.

3 To apply the new TLS certificate, run the deployment script.

   a Navigate to the `/opt/scripts/` directory.

   ```
   cd /opt/scripts/
   ```

   b Run the `./deploy.sh` script.

   Important Do not interrupt the deployment script. You receive the following message when the script finishes running:

   ```
   Prelude has been deployed successfully.
   To access, go to https://your_orchestrator_FQDN
   ```

Results

You have set custom TLS certificate for your vRealize Orchestrator Appliance.

What to do next

To confirm that the new certificate chain is applied, run the `vracli certificate ingress --list` command.

Import a Trusted Certificate with the Control Center

To communicate with other servers securely, the vRealize Orchestrator server must be able to verify their identity. For this purpose, you might need to import the TLS certificate of the remote entity to the vRealize Orchestrator trust store. To trust a certificate, you can import it to the trust store either by establishing a connection to a specific URL, or directly as a PEM-encoded file.

Procedure

1. Log in to Control Center as **`root`**.
2. Go to the **Certificates** page.
3. Select **Trusted Certificates** and click **Import**.
4. To import the certificate from a file, select **Import from a PEM-encoded file**.
5. Browse to the certificate file and click **Import**.
6. To import the certificate from a URL address, select **Import from URL**.
7. Enter the URL address where your certificate is stored and click **Import**.

Results

You have successfully imported a remote server certificate to the vRealize Orchestrator trust store.

Configuring the vRealize Orchestrator Plug-Ins

The vRealize Orchestrator Appliance provides access to a preinstalled library of default plug-ins. The default vRealize Orchestrator plug-ins are configured with plug-in specific workflows run in the vRealize Orchestrator Client.

The default vRealize Orchestrator plug-ins come with configuration workflows. You can run these workflows from the vRealize Orchestrator Client to register endpoints for management.

The configuration workflows have the *configuration* tag. For example, to access workflows that are used to manage AMQP brokers and subscriptions, enter the tags *AMQP* and *Configuration* in the search text box of the workflow library.

Manage vRealize Orchestrator Plug-Ins

On the **Manage Plug-Ins** page of vRealize Orchestrator Control Center, you can view a list of all plug-ins that are installed in vRealize Orchestrator and perform basic management actions.

Install or Upgrade a Plug-In

With the vRealize Orchestrator plug-ins, the vRealize Orchestrator server can integrate with other software products. vRealize Orchestrator comes with a set of preinstalled default plug-ins. You can further expand the capabilities of the vRealize Orchestrator platform by installing custom plug-ins.

You can install or upgrade plug-ins from the **Manage Plug-Ins** page of the vRealize Orchestrator. The file extension that can be used is `.vmoapp`.

For more information on installing or upgrading vRealize Orchestrator plug-ins, see Install or Update a vRealize Orchestrator Plug-In.

Change Plug-In Logging Level

Instead of changing the logging level for vRealize Orchestrator, you can change it only for specific plug-ins.

Disable a Plug-In

You can disable a plug-in by deselecting the **Enable plug-in** option next to the name of the plug-in.

This action does not remove the plug-in file. For more information on uninstalling a plug-in in vRealize Orchestrator, see Delete a Plug-In.

Install or Update a vRealize Orchestrator Plug-In

You can install or update third-party plug-ins in the vRealize Orchestrator Control Center.

Prerequisites

Download the *.dar* or *.vmoapp* file of the plug-in.

Note The preferred file format for vRealize Orchestrator plug-ins is *.vmoapp*.

Procedure

1. Log in the Control Center as **root**.
2. Select the **Manage Plug-ins** page.
3. Click **Browse** and select the *.dar* or *.vmoapp* file of the plug-in you want to install or update.
4. Click **Upload**.
5. Review the plug-in information, if applicable, accept the end-user license agreement, and click **Install**.

   The plug-in is installed or updated and the vRealize Orchestrator server service is restarted.

What to do next

Verify that the correct plug-in information is listed on the **Manage Plug-ins** page.

Delete a Plug-In

You can delete third-party plug-ins from the vRealize Orchestrator Appliance through Control Center.

Note Starting with vRealize Orchestrator 8.0, you no longer delete the plug-in package manually from the vRealize Orchestrator Client.

Procedure

1 Log in to the Control Center as **root**.

2 Select **Manage Plug-ins**.

3 Find the plug-in you want to delete and click the delete icon ( ).

4 Confirm that you want to delete the plug-in, and click **Delete**.

Results

You deleted the plug-in from the vRealize Orchestrator Appliance.

vRealize Orchestrator High Availability

To increase the availability of the vRealize Orchestrator services, start multiple vRealize Orchestrator server instances in a cluster with a shared database. vRealize Orchestrator works as a single instance until it is configured to work as part of a cluster.

Multiple vRealize Orchestrator server instances with identical server and plug-ins configurations work together in a cluster and share one database.

All vRealize Orchestrator server instances communicate with each other by exchanging heartbeats. Each heartbeat is a timestamp that the node writes to the shared database of the cluster at a certain time interval. Network problems, an unresponsive database server, or overload might cause an vRealize Orchestrator cluster node to stop responding. If an active vRealize Orchestrator server instance fails to send heartbeats within the failover timeout period, it is considered non-responsive. The failover timeout is equal to the value of the heartbeat interval multiplied by the number of the failover heartbeats. It serves as a definition for an unreliable node and can be customized according to the available resources and the production load.

An vRealize Orchestrator node enters standby mode when it loses connection to the database, and remains in this mode until the database connection is restored. The other nodes in the cluster take control of the active work, by resuming all interrupted workflows from their last unfinished items, such as scriptable tasks or workflow invocations.

You can monitor the state of your vRealize Orchestrator cluster from the **System** tab of the vRealize Orchestrator Client dashboard. To configure the cluster heartbeat, number of failover heartbeats, and the number of active nodes, navigate to the **Orchestrator Cluster Management** page of the vRealize Orchestrator Control Center.

vRealize Orchestrator Scalability Maximums

The scalability limit table outlines the recommended maximums on vRealize Orchestrator 8.x deployments.

Component	Scale targets	More information
Virtual machines	35,000	
vCenter Server connections	10	See vCenter Server Setup
Active nodes in a cluster	3	See Configure a vRealize Orchestrator Cluster
Concurrent running workflows	300 per node	See Configuring the Workflow Run Properties
Queued running workflows	10,000 per node	
Preserved workflow runs	100 per node	
Log event expiration days	15	

Configure a vRealize Orchestrator Cluster

You can configure your new vRealize Orchestrator deployment to run in high availability by deploying three nodes and connecting them as a cluster.

A vRealize Orchestrator cluster consists of three vRealize Orchestrator instances that share a common PostgreSQL database. The database of the configured vRealize Orchestrator cluster can only run in asynchronous mode.

To create a vRealize Orchestrator cluster, you must select one vRealize Orchestrator instance to be the primary node of the cluster. After configuring the primary node, you join the secondary nodes to it.

The created vRealize Orchestrator cluster is pre-configured with automatic failover.

Note Failure of the automatic failover can lead to loss of database data.

Prerequisites

- Download and deploy three standalone vRealize Orchestrator instances. See Download and Deploy the vRealize Orchestrator Appliance.

  Note The recommended number of nodes that can be used to create a clustered vRealize Orchestrator environment is three.

- Verify that SSH access is enabled for all vRealize Orchestrator nodes. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.
- Configure a load balancer server. See VMware vRealize Orchestrator 8.x Load Balancing Guide.

Procedure

1 Configure the primary node.

   a Log in to the vRealize Orchestrator Appliance command line of the primary node over SSH as **root**.

   b To configure the cluster load balancer server, run the `vracli load-balancer set` *load_balancer_FQDN* command.

   c Log in to the Control Center of the primary node and select **Host Settings**.

   d Click **Change** and set the host address of the connected load balancer server.

   e Configure the authentication provider. See Configuring a Standalone vRealize Orchestrator Server.

2 Join secondary nodes to primary node.

   a Log in to the vRealize Orchestrator Appliance command line of the secondary node over SSH as **root**.

   b To join the secondary node to the primary node, run the `vracli cluster join` *primary_node_hostname_or_IP* command.

   c Enter the root password of the primary node.

   d Repeat the procedure for other secondary node.

3 (Optional) If your primary node uses a custom certificate, you must either set the certificate in the appliance or generate a new certificate. See Generate a Custom TLS Certificate for vRealize Orchestrator.

   Note The file containing the certificate chain must be PEM-encoded.

4 Finish the cluster deployment.

   a Log in to the vRealize Orchestrator Appliance command line of the primary node over SSH as **root**.

   b To confirm that all nodes are in a ready state, run the `kubectl -n prelude get nodes` command.

   c Run the `/opt/scripts/deploy.sh` script and wait for the deployment to finish.

Results

You have created a vRealize Orchestrator cluster. After creating the cluster, you can access your vRealize Orchestrator environment only from the FQDN address of your load balancer server.

Note Because you can only access the Control Center of the cluster with the root password of the load balancer, you cannot edit the configuration of a cluster node if it has a different root password. To edit the configuration of this node, remove it from the load balancer, edit the configuration in the Control Center, and add the node back to the load balancer.

What to do next

To monitor the state of the vRealize Orchestrator cluster, log in to the vRealize Orchestrator Client and navigate to the **System** tab of the dashboard. See Monitoring an vRealize Orchestrator Cluster.

Removing an vRealize Orchestrator Cluster Node

You can delete an vRealize Orchestrator so you can reduce your cluster capacity.

After removing a node from your vRealize Orchestrator cluster, that node will no longer be functional. If you want to use this node again, you must delete its vRealize Orchestrator Appliance from your vCenter Server and deploy it again. See Download and Deploy the vRealize Orchestrator Appliance.

Prerequisites

Create a vRealize Orchestrator cluster. See Configure a vRealize Orchestrator Cluster.

Procedure

1. Log in to the vRealize Orchestrator Appliance command line of the node you want to remove as **root**.
2. To remove the node from your vRealize Orchestrator, run the `vracli cluster leave` command.
3. Log in to the vRealize Orchestrator Appliance command line of one of the remaining nodes as **root**.
4. Run the `kubectl -n prelude get nodes` command and confirm that the removed node is no longer part of the cluster.

Scale Out a Standalone vRealize Orchestrator Deployment

You can increase the availability and scalability of your configured vRealize Orchestrator deployment by scaling it out.

Prerequisites

- Download, deploy, and configure a vRealize Orchestrator instance. See Download and Deploy the vRealize Orchestrator Appliance and Configuring a Standalone vRealize Orchestrator Server.
- Download and deploy two additional vRealize Orchestrator instances. See Download and Deploy the vRealize Orchestrator Appliance.
- Configure a load balancer server. See VMware vRealize Orchestrator 8.x Load Balancing Guide.

Procedure

1 Configure the primary node.
   a Log in to the Control Center of your configured vRealize Orchestrator deployment as **root**.
   b Select **Configure Authentication Provider** and unregister your authentication provider.
   c Select **Host Settings** and enter the host name of the load balancer server.
   d Select **Configure Authentication Provider** and register your authentication provider again.
   e Log in to the vRealize Orchestrator Appliance command line of the configured instance as **root**.
   f To stop all the services of the vRealize Orchestrator instance, run the `/opt/scripts/deploy.sh --onlyClean` command.
   g To set the load balancer, run `vracli load-balancer set` *load_balancer_FQDN*.
   h (Optional) If your vRealize Orchestrator instance uses a custom certificate, run the `vracli certificate ingress --set` *your_cert_file.pem* command.

   Note The file containing the certificate chain must be PEM-encoded.

2 Join secondary nodes to the configured instance.
   a Log in to the vRealize Orchestrator Appliance command line of the secondary node as **root**.
   b To join the secondary node to the configured instance, run the `vracli cluster join` *primary_node_hostname_or_IP* command.
   c Repeat for the other secondary node.

3 Finish the scale-out process.
   a Log in to the vRealize Orchestrator Appliance command line of the configured instance as **root**.
   b Run `/opt/scripts/deploy.sh` and wait for the script to finish.

Results

You have scaled out your vRealize Orchestrator deployment.

Monitoring an vRealize Orchestrator Cluster

You can monitor your existing vRealize Orchestrator cluster through the **System** tab of the vRealize Orchestrator Client dashboard.

The recommended method for monitoring the configuration synchronization states of the vRealize Orchestrator instances is through the **System** tab of the vRealize Orchestrator Client dashboard.

Note If you are unable to access the vRealize Orchestrator Client dashboard, you can also monitor the states of your vRealize Orchestrator instances by running the `kubectl get pods -n prelude` command from the vRealize Orchestrator Appliance command line.

Configuration Synchronization State	Description
RUNNING	The vRealize Orchestrator service is available and can accept requests.
STANDBY	The vRealize Orchestrator service cannot process requests because: ■ The node is part of a High Availability (HA) cluster and remains in a standby mode until the primary node fails. ■ The service cannot verify the configuration prerequisites, like a valid connection to the database, authentication provider, and the vRealize Orchestrator instance license.
Failed to retrieve the service's health status	The vRealize Orchestrator server service cannot be contacted because it is either stopped or a network issue is present.
Pending restart	Control Center detects a configuration change and the vRealize Orchestrator server restarts automatically.

Configuring the Customer Experience Improvement Program

If you choose to participate in the Customer Experience Improvement Program (CEIP), VMware receives anonymous information that helps to improve the quality, reliability, and functionality of VMware products and services.

Categories of Information That VMware Receives

The Customer Experience Improvement Program (CEIP) provides VMware with information that enables VMware to improve our products and services and to fix problems.

Details regarding the data collected through CEIP and the purposes for which it is used by VMware are set in the Trust & Assurance Center at http://www.vmware.com/trustvmware/ceip.html. To join or leave the CEIP for this product, see Join or Leave the Customer Experience Improvement Program.

Join or Leave the Customer Experience Improvement Program

Join the Customer Experience Improvement Program from the vRealize Orchestrator Appliance command line.

Procedure

1. Log in to vRealize Orchestrator Appliance command line as **root**.
2. To join the Customer Experience Improvement Program, run the `vracli ceip on` command.
3. Review the Customer Experience Improvement Program information, and run the `vracli ceip on --acknowledge-ceip` command.
4. Restart the vRealize Orchestrator services.
   a. To restart the server service, run the `kubectl -n prelude exec -it` *your_vro_pod* `-c vco-server-app /bin/bash` command.
   b. To stop the service, run the `kill 1` command.
   c. To restart the Control Center service run the `kubectl -n prelude exec -it` *your_vro_pod* `-c vco-controlcenter-app /bin/bash` command.
   d. To stop the service, run the `kill 1` command.
5. To leave the Customer Experience Improvement Program, run the `vracli ceip off` command.
6. Repeat the steps for restarting the services.

Using the vRealize Orchestrator API Services

6

In addition to configuring vRealize Orchestrator by using Control Center, you can modify the vRealize Orchestrator server configuration settings by using the vRealize Orchestrator REST API, the Control Center REST API, or the command-line utility, stored in the appliance.

The Configuration plug-in is included in the vRealize Orchestrator package, by default. You can access the Configuration plug-in workflows from either the vRealize Orchestrator workflow library or the vRealize Orchestrator REST API. With these workflows, you can change the trusted certificate and keystore settings of the vRealize Orchestrator server. For information on all available vRealize Orchestrator REST API service calls, see the *vRealize Orchestrator Server API* documentation, located at https://*your_orchestrator_FQDN*/vco/api/docs.

- Managing TLS Certificates and Keystores by Using the REST API

  In addition to managing TLS certificates by using Control Center, you can also manage trusted certificates and keystores when you run workflows from the Configuration plug-in or by using the REST API.

Managing TLS Certificates and Keystores by Using the REST API

In addition to managing TLS certificates by using Control Center, you can also manage trusted certificates and keystores when you run workflows from the Configuration plug-in or by using the REST API.

The Configuration plug-in contains workflows for importing and deleting TLS certificates and keystores. You can access these workflows by navigating to **Library > Workflows > SSL Trust Manager** and **Library > Workflows > Keystores** in the vRealize Orchestrator Client. You can also run these workflows by using the vRealize Orchestrator REST API.

The Control Center REST API provides access to resources for configuring the vRealize Orchestrator server. You can use the Control Center REST API with third-party systems to automate the vRealize Orchestrator configuration. The root endpoint of the Control Center REST API is https://*your_orchestrator_FQDN*/vco/api. For information on all available service calls that you can make to the Control Center REST API, see the *vRealize Orchestrator Control Center API* documentation, at https://*your_orchestrator_FQDN*/vco-controlcenter/docs.

Delete a TLS Certificate by Using the REST API

You can delete a TLS certificate by running the Delete trusted certificate workflow of the Configuration plug-in or by using the REST API.

Procedure

1 Make a `GET` request at the URL of the Workflow service of the Delete trusted certificate workflow.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Delete trusted
certificate
```

2 Retrieve the definition of the Delete trusted certificate workflow by making a `GET` request at the URL of the definition.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/8a70a326-
ffd7-4fef-97e0-2002ac49f5bd
```

3 Make a `POST` request at the URL that holds the execution objects of the Delete trusted certificate workflow.

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/8a70a326-
ffd7-4fef-97e0-2002ac49f5bd/executions/
```

4 Provide the name of the certificate you want to delete as an input parameter of the Delete trusted certificate workflow in an execution-context element in the request body.

Import TLS Certificates by Using the REST API

You can import TLS certificates by running a workflow from the Configuration plug-in or by using the REST API.

You can import a trusted certificate from a file or a URL. See Import a Trusted Certificate with the Control Center

Procedure

1 Make a `GET` request at the URL of the Workflow service.

Option	Description
Import trusted certificate from a file	Imports a trusted certificate from a file.
Import trusted certificate from URL	Imports a trusted certificate from a URL address.
Import trusted certificate from URL using proxy server	Imports a trusted certificate from a URL address by using a proxy server.
Import trusted certificate from URL with certificate alias	Imports a trusted certificate with a certificate alias, from a URL address.

To import a trusted certificate from a file, make the following `GET` request:

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Import
                trusted certificate from a file
```

2 Retrieve the definition of the workflow by making a `GET` request at the URL of the definition.

To retrieve the definition of the Import trusted certificate from a file workflow, make the following `GET` request:

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5
```

3 Make a `POST` request at the URL that holds the execution objects of the workflow.

For the Import trusted certificate from a file workflow, make the following `POST` request:

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/
93a7bb21-0255-4750-9293-2437abe9d2e5/executions
```

4 Provide values for the input parameters of the workflow in an execution-context element of the request body.

Parameter	Description
cer	The CER file from which you want to import the TLS certificate. This parameter is applicable for the Import trusted certificate from a file workflow.
url	The URL from which you want to import the TLS certificate. For non-HTPS services, the supported format is *IP_address_or_DNS_name:port*. This parameter is applicable for the Import trusted certificate from URL workflow.

Create a Keystore by Using the REST API

You can create a keystore by running the Create a keystore workflow of the Configuration plug-in or by using the REST API.

Procedure

1 Make a `GET` request at the URL of the Workflow service of the Create a keystore workflow.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Create a keystore
```

2 Retrieve the definition of the Create a keystore workflow by making a `GET` request at the URL of the definition.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/
```

3 Make a `POST` request at the URL that holds the execution objects of the Create a keystore workflow.

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/executions/
```

4 Provide the name of the keystore you want to create as an input parameter of the Create a keystore workflow in an execution-context element in the request body.

Delete a Keystore by Using the REST API

You can delete a keystore by running the Delete a keystore workflow of the Configuration plug-in or by using the REST API.

Procedure

1 Make a `GET` request at the URL of the Workflow service of the Delete a keystore workflow.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Delete a keystore
```

2 Retrieve the definition of the Delete a keystore workflow by making a `GET` request at the URL of the definition.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/
7a3389eb-1fab-4d77-860b-81b66bb45b86/
```

3 Make a `POST` request at the URL that holds the execution objects of the Delete a keystore workflow.

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/
7a3389eb-1fab-4d77-860b-81b66bb45b86/executions/
```

4 Provide the keystore you want to delete as an input parameter of the Delete a keystore workflow in an execution-context element in the request body.

Add a Key by Using the REST API

You can add a key by running the Add key workflow of the Configuration plug-in or by using the REST API.

Procedure

1 Make a `GET` request at the URL of the Workflow service of the Add key workflow.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows?conditions=name=Add key
```

2 Retrieve the definition of the Add key workflow by making a `GET` request at the URL of the definition.

```
GET https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/
```

3 Make a `POST` request at the URL that holds the execution objects of the Add key workflow.

```
POST https://{orchestrator_host}:{port}/vco/api/workflows/6c301bff-e8fe-4ae0-
ad08-5318178594b3/executions/
```

4 Provide the keystore, key alias, PEM-encoded key, certificate chain and key password as input parameters of the Add key workflow in an execution-context element in the request body.

Additional Configuration Options 7

You can use the Control Center to change the default vRealize Orchestrator behavior.

This chapter includes the following topics:

- Reconfiguring Authentication
- Configuring the Workflow Run Properties
- vRealize Orchestrator Log Files
- Enabling the Opentracing and Wavefront Extensions
- Enable Time Synchronization for vRealize Orchestrator
- Deactivate Time Synchronization for vRealize Orchestrator
- Configure vRealize Orchestrator Kubernetes CIDR
- Update the DNS Settings for vRealize Orchestrator

Reconfiguring Authentication

After you set up the authentication method during the initial configuration of Control Center, you can change the authentication provider or the configured parameters at any time.

Change the Authentication Provider

To change the authentication mode or the authentication provider connection settings, you must first unregister the existing authentication provider.

Procedure

1. Log in to Control Center as **`root`**.
2. On the **Configure Authentication Provider** page, click the **UNREGISTER** button next to the host address text box to unregister the authentication provider that is in use.

Results

You have successfully unregistered the authentication provider.

What to do next

Reconfigure the authentication in Control Center. See Configuring a Standalone vRealize Orchestrator Server.

Change the Authentication Parameters

When you use vSphere as an authentication provider in Control Center, you can change the default tenant of the vRealize Orchestrator administrators group.

Prerequisites

Configure vSphere as the authentication provider for your vRealize Orchestrator deployment. See Configure a Standalone vRealize Orchestrator Server with vSphere Authentication.

Note The vRealize Automation authentication does not include these parameters.

Procedure

1. Log in to the Control Center as **root**.
2. Select **Configure Authentication Provider**.
3. Click the **CHANGE** button next to the **Default tenant** text box.
4. Replace the name of the tenant.
5. Click the **CHANGE** button next to the **Admin group** text box.

   Note If you do not reconfigure the administrators group, it remains empty and you are no longer able to access Control Center.
6. Enter the name of an administrator group and click **SEARCH**.
7. Select an administrator group.
8. Change the administrators group.
9. To finish editing the authentication parameters, click **SAVE CHANGES**.

Configuring the Workflow Run Properties

By default, you can run up to 300 workflows per node, and up to 10,000 workflows can be queued if the number of actively running workflows is reached.

When the vRealize Orchestrator node has to run more than 300 concurrent workflows, the pending workflow runs are queued. When an active workflow run completes, the next workflow in the queue starts to run. If the maximum number of queued workflows is reached, the next workflow runs fail until one of the pending workflows starts to run.

You can configure the workflow run properties on the **Advanced Options** page in Control Center.

Option	Description
Enable safe mode	If safe mode is enabled, all running workflows are canceled and are not resumed on the next vRealize Orchestrator node start.
Number of concurrent running workflows	The number of workflows that run simultaneously. The default is 300 workflows per node.
Maximum amount of running workflows in the queue	The number of workflow run requests that the vRealize Orchestrator server accepts before becoming unavailable. The default is 10,000 workflows per node.
Maximum number of preserved runs per workflow	The maximum number of finished workflow runs that are kept as history per workflow. If the number is exceeded, the oldest workflow runs are deleted. The default is 100 runs per workflow.
Log events expiration days	The number of days that log events are kept in the database before they are purged. The default is 15 days.

vRealize Orchestrator Log Files

VMware Technical Support routinely requests diagnostic information when you submit a support request. This diagnostic information contains product-specific logs and configuration files from the host on which the product runs.

vRealize Orchestrator Appliance logs are stored in the `/data/vco/usr/lib/vco/app-server/logs/` directory. You export the logs of your vRealize Orchestrator Appliance deployment by logging in to the appliance command line and running the `vracli log-bundle` command. The generated log bundle is saved on the root folder of your vRealize Orchestrator Appliance.

Logging Persistence

You can log information in any kind of vRealize Orchestrator script, for example workflow, policy, or action. This information has types and levels. The type can be either persistent or non-persistent. The level can be `DEBUG`, `INFO`, `WARN`, `ERROR`, `TRACE`, and `FATAL`.

Table 7-1. Creating Persistent and Non-Persistent Logs

Log Level	Persistent Type	Non-Persistent Type
`DEBUG`	Server.debug("short text", "long text");	System.debug("text")
`INFO`	Server.log("short text", "long text");	System.log("text");
`WARN`	Server.warn("short text", "long text");	System.warn("text");
`ERROR`	Server.error("short text", "long text");	System.error("text");

Persistent Logs

Persistent logs (server logs) track past workflow run logs and are stored in the vRealize Orchestrator database.

Non-Persistent Logs

When you use a non-persistent log (system log) to create scripts, the vRealize Orchestrator server notifies all running vRealize Orchestrator applications about this log, but this information is not stored in the database. When the application is restarted, the log information is lost. Non-persistent logs are used for debugging purposes and for live information. To view system logs, you must select a completed workflow run in the vRealize Orchestrator Client and select the **Logs** tab.

vRealize Orchestrator Logs Configuration

On the **Configure Logs** page in Control Center, you can set the level of server log and the scripting log that you require. If either of the logs is generated multiple times a day, it becomes difficult to determine what causes problems.

The default log level of the server log and the scripting log is `INFO`. Changing the log level affects all new messages that the server enters in the logs and the number of active connections to the database. The logging verbosity decreases in descending order.

Caution Only set the log level to `DEBUG` or `ALL` to debug a problem. Do not use these settings in a production environment because it can seriously impair performance.

Generate vRealize Orchestrator Logs

You can export the logs of your deployment by logging in to the vRealize Orchestrator Appliance command line as **root** and running the `vracli log-bundle` command. The generated log bundle is stored in the root folder of the appliance.

.

Note When you have more than one vRealize Orchestrator instance in a cluster, the log-bundle includes the logs from all vRealize Orchestrator instances in the cluster.

Configure Logging Integration with vRealize Log Insight

You can configure vRealize Orchestrator to send your logging information to a vRealize Log Insight server.

You can configure a logging integration to a vRealize Log Insight server through the vRealize Orchestrator Appliance command line.

Note For information on configuring a logging integration with a remote syslog server, see Create or Overwrite a Syslog Integration in vRealize Orchestrator.

Prerequisites

- Configure your vRealize Log Insight server. See *vRealize Log Insight Documentation*.
- Verify that your vRealize Log Insight version is 4.7.1 or later.

Procedure

1 Log in to the vRealize Orchestrator Appliance command line as **root**.

2 To configure the logging integration with vRealize Log Insight, run the `vracli vrli set` *vRLI_FQDN* command.

> **Note** If your vRealize Orchestrator instance uses a self-signed certificate, you can disable the SSL authentication by including the optional `-k` or `--insecure` argument.

What to do next

For more information on vRealize Log Insight configuration options, run the `vracli vrli -h` command.

Create or Overwrite a Syslog Integration in vRealize Orchestrator

You can configure vRealize Orchestrator to send your logging information to one or more remote syslog servers.

The `vracli remote-syslog set` command is used to create a syslog integration or overwrite existing integrations.

vRealize Orchestrator remote syslog integration supports three connection types:

- Over UDP.
- Over TCP without TLS.

> **Note** To create a syslog integration without using TLS, add the `--disable-ssl` flag to the `vracli remote-syslog set` command.

- Over TCP with TLS.

For information on configuring a logging integration with vRealize Log Insight, see Configure Logging Integration with vRealize Log Insight.

Prerequisites

Configure one or more remote syslog servers.

Procedure

1 Log in to the vRealize Orchestrator Appliance command line as **root**.

2 To create an integration to a syslog server, run the `vracli remote-syslog set` command.

```
vracli remote-syslog set -id name_of_integration protocol_type://
syslog_URL_or_FQDN:syslog_port
```

Note If you do not enter a port in the `vracli remote-syslog set` command, the port value defaults to 514.

Note You can add a certificate to the syslog configuration. To add a certificate file, use the `--ca-file` flag. To add a certificate as plaintext, use the `--ca-cert` flag.

3 (Optional) To overwrite an existing syslog integration, run the `vracli remote-syslog set` and set the `-id` flag value to the name of the integration you want to overwrite.

Note By default, the vRealize Orchestrator Appliance requests that you confirm that you want to overwrite the syslog integration. To skip the confirmation request, add the `-f` or `--force` flag to the `vracli remote-syslog set` command.

What to do next

To review the current syslog integrations in the appliance, run the `vracli remote-syslog` command.

Delete a Syslog Integration in vRealize Orchestrator

You can delete syslog integrations from your vRealize Orchestrator Appliance by running the `vracli remote-syslog unset` command.

Prerequisites

Create one or more syslog integrations in the vRealize Orchestrator Appliance. See Create or Overwrite a Syslog Integration in vRealize Orchestrator.

Procedure

1 Log in to the vRealize Orchestrator Appliance command line as **root**.

2 Delete syslog integrations from the vRealize Orchestrator Appliance.

a To delete a specific syslog integration, run the `vracli remote-syslog unset -id` *Integration_name* command.

b To delete all syslog integrations on the vRealize Orchestrator Appliance, run the `vracli remote-syslog unset` command without the `-id` flag.

Note By default, the vRealize Orchestrator Appliance requests that you confirm that you want to delete all syslog integrations. To skip the confirmation request, add the `-f` or `--force` flag to the `vracli remote-syslog unset` command.

Enable Kerberos Debug Logging

You can troubleshoot vRealize Orchestrator plug-in problems by modifying the Kerberos configuration file used by the plug-in.

The Kerberos configuration file is located in the `/data/vco/usr/lib/vco/app-server/conf/` directory of the vRealize Orchestrator Appliance.

Procedure

1. Log in to the vRealize Orchestrator Appliance command line as **root**.
2. Run the `kubectl -n prelude edit deployment vco-app` command.
3. In the deployment file, locate and edit the `-Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf'` string.

   ```
   -Djava.security.krb5.conf=/usr/lib/vco/app-server/conf/krb5.conf
   -Dsun.security.krb5.debug=true'
   ```

4. Save the changes and exit the file editor.
5. Run the `kubectl -n prelude get pods` command.

   Wait until all pods are running.
6. Verify that the Kerberos debug logging is enabled.

   ```
   kubectl -n prelude log {vco_app_name} -c vco-server-app | grep krb5
   ```

   Verify that the logs contain a similar message.

   ```
   kubectl -n prelude log vco-app-5c965f9b9d-v8srd -c vco-server-app | grep krb5
   12:23:05,417 INFO O11N:75 - Sysprop: java.security.krb5.conf = /usr/lib/vco/app-server/
   conf/krb5.conf
   12:23:05,421 INFO O11N:75 - Sysprop: sun.security.krb5.debug = true
   2019-10-22 12:23:38.521+0000 [Thread-19] INFO {} [O11N] Sysprop: java.security.krb5.conf
   = /usr/lib/vco/app-server/conf/krb5.conf
   2019-10-22 12:23:38.525+0000 [Thread-19] INFO {} [O11N] Sysprop: sun.security.krb5.debug =
   true
   Java config name: /usr/lib/vco/app-server/conf/krb5.conf
   EType: sun.security.krb5.internal.crypto.Aes256CtsHmacSha1EType
   ```

Enabling the Opentracing and Wavefront Extensions

The Opentracing and Wavefront extensions for vRealize Orchestrator provide tools for gathering data about your vRealize Orchestrator environment. You can use this data for troubleshooting the vRealize Orchestrator system and workflows.

Before you can configure vRealize Orchestrator to use the Opentracing and Wavefront extensions, you must enable them in the vRealize Orchestrator Appliance.

Prerequisites

- Verify that the vRealize Orchestrator Appliance SSH service is enabled. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.
- If you have enabled previous versions of the Opentracing or Wavefront extensions, you must remove them before enabling the current version. For example if you have previously enabled version 8.1.0 of the Wavefront extension, you must run the `rm /data/vco/usr/lib/vco/app-server/extensions/wavefront-8.1.0.jar` command.

Procedure

1. Log in to the vRealize Orchestrator Appliance over SSH as **root**.
2. To list all available extensions, run the `ls /data/vco/usr/lib/vco/app-server/extensions/` command.
3. Run the following command to enable the Opentracing extension:

```
mv /data/vco/usr/lib/vco/app-server/extensions/opentracing-8.7.0.jar.inactive /
data/vco/usr/lib/vco/app-server/extensions/opentracing-8.7.0.jar
```

4. Run the following command to enable the Wavefront extension:

```
mv /data/vco/usr/lib/vco/app-server/extensions/wavefront-8.7.0.jar.inactive /
data/vco/usr/lib/vco/app-server/extensions/wavefront-8.7.0.jar
```

5. Log in to the Control Center and confirm that the extensions appear in the **Extension Properties** page.

What to do next

Configure Opentracing and Wavefront integration with vRealize Orchestrator in the **Extension Properties** page. See Configure the Opentracing Extension and Configure the Wavefront Extension.

Configure the Opentracing Extension

The Opentracing extension sends data about workflow runs to a Jaeger server. Data includes the workflow status, input and output parameters, the user that initiated the workflow run, and the workflow ID data.

Prerequisites

- Verify sure that Opentracing is enabled in the vRealize Orchestrator Appliance. See Enabling the Opentracing and Wavefront Extensions.
- Deploy a Jaeger server for use in the Opentracing extension. For more information, see the Getting Started with Jaeger documentation.

Procedure

1. Log in to the Control Center as **root**.

2 Select the **Extension Properties** page.

3 Select the Opentracing extension.

4 Enter the Jaeger server host address and port.

> **Note** Insert two forward slashes ("//") before entering the server address.

5 Click **Save**.

Results

You have configured the Opentracing extension for vRealize Orchestrator.

What to do next

- To access the Jaeger UI containing the data collected by the Opentracing extension, visit the host address entered during configuration.
- Under the **Service** option, select **Workflows**.
- To specify what data to view, use the **Tags** option. For example, to view data about failed workflows, enter `status=failed`.

Configure the Wavefront Extension

Use the Wavefront extension to gather metric data about your vRealize Orchestrator system and workflows.

Prerequisites

1 Verify that Wavefront is enabled in the vRealize Orchestrator Appliance. See Enabling the Opentracing and Wavefront Extensions.

2 Import the Wavefront Certificate:

   a Log in to the vRealize Orchestrator Control Center as **root**.

   b Select the **Certificates** page.

   c Click the **Import** drop-down menu and select **Import from URL**.

   d Enter the Wavefront URL and click **Import**.

3 Configure a Wavefront proxy. For more information, see Installing and Managing Wavefront Proxies.

Procedure

1 Log in to the vRealize Orchestrator Control Center as **root**.

2 Select the **Extension Properties** page.

3 Select the Wavefront extension.

4 Configure the Wavefront properties.

Option	Description
Proxy	The Wavefront proxy address.
Host	Optional. The Wavefront host address.
Token	Optional. The Wavefront API token. For more information on generating a Wavefront API token, see Generating an API Token .
Prefix	Add prefix labels for each metric sent to Wavefront. Prefix labels are separated by a dot symbol.

5 (Optional) Select **Send default dashboard on next start**.

6 Click **Save**.

Results

You have configured the Wavefront extension for vRealize Orchestrator.

What to do next

- To access the metrics collected by Wavefront, access the dashboard on the address entered during configuration.
- To get notifications about specific events in your vRealize Orchestrator environment, you can use Wavefront Alerts. For more information, see the Wavefront Alerts documentation.

Enable Time Synchronization for vRealize Orchestrator

You can enable time synchronization on your vRealize Orchestrator deployment with the vRealize Orchestrator Appliance command line.

You can configure time synchronization for your standalone or clustered vRealize Orchestrator deployment by using the Network Time Protocol (NTP) communication protocol. vRealize Orchestrator supports two, mutually exclusive, NTP configurations:

NTP configuration	Description
ESXi	This configuration can be used when the ESXi server hosting the vRealize Orchestrator Appliance is synchronized with an NTP server. If you are using a clustered deployment, all ESXi hosts must be synchronized with an NTP server. For more information on configuring NTP for ESXi, see Configuring Network Time Protocol (NTP) on an ESXi host using the vSphere Web Client. **Note** If your vRealize Orchestrator deployment is migrated to a ESXi host that is not synchronized to an NTP server, you can experience clock drift.
systemd	This configuration uses the systemd-timesyncd daemon to synchronize the clocks of your vRealize Orchestrator deployment. **Note** By default, the systemd-timesyncd daemon is enabled, but configured with no NTP servers. If the vRealize Orchestrator Appliance uses a dynamic IP configuration, the appliance can use any NTP servers received by the DHCP protocol.

Procedure

1. Log in to the vRealize Orchestrator Appliance command line as **root**.
2. Enable NTP with ESXi.
   a. Run the `vracli ntp esxi` command.
   b. (Optional) To confirm the status of the NTP configuration, run the `vracli ntp status` command.
3. Enable NTP with systemd.
   a. Run the `vracli ntp systemd --set` *FQDN_or_IP_of_systemd_server* command.

      Note You can add multiple systemd NTP servers by separating their network addresses with a comma. Each network address must be placed inside single quotation marks. For example, `vracli ntp systemd --set` *'ntp_address_1'*,*'ntp_address_2'*

   b. (Optional) To confirm the status of the NTP configuration, run the `vracli ntp status` command.

Results

You have enabled time synchronization for your vRealize Orchestrator deployment.

What to do next

The NTP configuration can fail if there is a time difference of above 10 minutes between the NTP server and the vRealize Orchestrator deployment. To resolve this problem, reboot the vRealize Orchestrator Appliance.

Deactivate Time Synchronization for vRealize Orchestrator

You can deactivate the Network Time Protocol (NTP) time synchronization on your vRealize Orchestrator deployment with the vRealize Orchestrator Appliance command line.

You can also reset the NTP configuration of your vRealize Orchestrator Appliance to the default state by running the `vracli ntp reset` command.

Prerequisites

Verify that you have configured time synchronization with ESXi or systemd. See Enable Time Synchronization for vRealize Orchestrator.

Procedure

1. Log in to the vRealize Orchestrator Appliance command line as **root**.
2. To deactivate time synchronization with ESXi or systemd, run the `vracli ntp disable` command.
3. (Optional) To confirm the status of the NTP configuration, run the `vracli ntp status` command.

Configure vRealize Orchestrator Kubernetes CIDR

You can change the Kubernetes Classless Inter-domain Routing (CIDR) subnet masks after deployment.

The vRealize Orchestrator Appliance configures and runs a Kubernetes cluster. The pods and services in this cluster are deployed in separate IPv4 subnets, represented by the internal cluster CIDR and internal service CIDR, respectively. The default values of the subnet masks set during OVF deployment are the following:

Kubernetes network property	Default value	Property description
`cluster-cidr`	10.244.0.0/22	The CIDR used for pods running inside the Kubernetes cluster.
`service-cidr`	10.244.4.0/22	The CIDR used for Kubernetes services inside the Kubernetes cluster.

The default CIDR network addresses can create a conflict with outside private networks that you might be using. In such scenarios, you can change the configuration of these CIDR values either during or after deploying your vRealize Orchestrator Appliance.

Note For information on changing the CIDR configuration during appliance deployment, see Download and Deploy the vRealize Orchestrator Appliance.

Prerequisites

- Verify that the CIDR address values support at least 1024 hosts.

- The internal cluster CIDR and internal service CIDR must not share the same subnet value.
- The CIDR value for one of the subnets cannot include the value you want to add to the other subnet.

Note For example, the `cluster-cidr` value cannot be **`10.244.4.0/22 10.244.4.0/24`**, because this would also include the subnet value for the `service-cidr` property. Each subnet value must be added separately.

Procedure

1. Log in to the vRealize Orchestrator Appliance as **root**.
2. Run the `vracli upgrade exec -y --prepare --profile k8s-subnets` command.
3. Back up your vRealize Orchestrator deployment by taking a virtual machine (VM) snapshot. See Take a Snapshot of a Virtual Machine.

   Caution vRealize Orchestrator 8.x does not currently support memory snapshots. Before taking the snapshot of your vRealize Orchestrator deployment, verify that the **Snapshot the virtual machine's memory** option is deactivated.
4. Change the values of the cluster CIDR and service CIDR subnets by running the `vracli network k8s-subnets` command.

   ```
   vracli network k8s-subnets --cluster-cidr <CIDR_value> --service-cidr <CIDR_value>
   ```
5. To finish the CIDR configuration process, run the `vracli upgrade exec` command.

Update the DNS Settings for vRealize Orchestrator

An administrator can update the DNS settings of the vRealize Orchestrator deployment by using the `vracli network dns` command.

Prerequisites

Verify that the vRealize Orchestrator Appliance SSH service is enabled. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.

Procedure

1. Log in to the vRealize Orchestrator Appliance command-line over SSH as **root**.

   Note For clustered deployments, log in to appliance of any node in the cluster.
2. To set new DNS servers to your vRealize Orchestrator deployment, run the `vracli network dns set` command.

   ```
   vracli network dns set --servers DNS1,DNS2
   ```

3 Verify that the new DNS servers are properly applied to all vRealize Orchestrator nodes by running the `vracli network dns status` command.

4 To stop the vRealize Orchestrator services in your deployment, run the following set of commands:

```
/opt/scripts/svc-stop.sh
sleep 120
/opt/scripts/deploy.sh --onlyClean
```

5 Restart the vRealize Orchestrator nodes and wait for them to start completely.

6 Log in to the command-line for each vRealize Orchestrator node over SSH and verify that the new DNS servers are listed in the `/etc/resolve.conf` file.

7 To start the vRealize Orchestrator services, run the `/opt/scripts/deploy.sh` script on one of the nodes in your deployment.

Results

The vRealize Orchestrator DNS settings are changed as specified.

Configuration Use Cases and Troubleshooting

8

The configuration use cases provide task flows that you can perform to meet specific configuration requirements of your vRealize Orchestrator server and troubleshooting topics to understand and solve a problem.

This chapter includes the following topics:

- Verify the vRealize Orchestrator server build number
- Configure vRealize Orchestrator Plugin for vSphere Web Client
- Cancel Running Workflows
- Enable vRealize Orchestrator Server Debugging
- Resize the vRealize Orchestrator Appliance Disks
- How to Scale the Heap Memory Size of the vRealize Orchestrator Server
- Disaster Recovery of vRealize Orchestrator by Using Site Recovery Manager

Verify the vRealize Orchestrator server build number

In certain scenarios, you might be required to verify the server build number of your vRealize Orchestrator deployment.

You can verify your vRealize Orchestrator server build number by navigating to https://*your_orchestrator_FQDN*/vco/api/about. Your server build number is displayed in the `<ns2:build-number>` tags.

Verifying your server build number can be useful in use cases such as providing additional information to a support request (SR) that you have logged with VMware Support.

Note The vRealize Orchestrator server build number is different from the build number of your vRealize Orchestrator Appliance. To verify the build number of your appliance, log in to the vRealize Orchestrator Appliance command line and run the `vracli version` command. Verifying the appliance build number can help you confirm if your upgrade to the latest version of vRealize Orchestrator is successful.

Configure vRealize Orchestrator Plugin for vSphere Web Client

To use the vRealize Orchestrator plug-in for the vSphere Web Client, you must register vRealize Orchestrator as an extension of vCenter Server.

After you register your vRealize Orchestrator server with vCenter Single Sign-On and configure it to work with vCenter Server, you must register vRealize Orchestrator as an extension of vCenter Server.

Prerequisites

- Verify that SSH access is enabled for the vRealize Orchestrator Appliance. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.
- You must register vRealize Orchestrator with vSphere authentication to the same Platform Services Controller that your managed vCenter Server authenticates with.
- Copy the `vco-plugin.zip` to the vRealize Orchestrator Appliance:
  a Download the `vco-plugin.zip` file from the VMware Technology Network.
  b Open an SSH client.

  Note For Linux or MacOS environments, you can use the Terminal command-line interface. For Windows environments, you can use the PuTTY client.

  c To copy the `vco-plugin.zip` file, run the secure copy command.

```
For Linux/MacOS: scp ~/<zip_download_dir>/vco-plugin.zip
root@<orchestrator_FQDN_or_IP>:/data/vco/usr/lib/vco/downloads/vco-plugin.zip
```

```
For Windows: pscp C:\<zip_download_dir>\vco-plugin.zip root@<orchestrator_FQDN_or_IP>:/
data/vco/usr/lib/vco/downloads/vco-plugin.zip
```

Procedure

1. Log in to the vRealize Orchestrator Client.
2. Navigate to **Library > Workflows**.
3. Search for the **Register vCenter Orchestrator as a vCenter Server extension** workflow, and click **Run**.
4. Select the vCenter Server instance to register vRealize Orchestrator with.
5. Enter https://*your_orchestrator_FQDN* or the service URL of the load balancer that redirects the requests to the vRealize Orchestrator server nodes.
6. Click **Run**.

Cancel Running Workflows

You can use the vRealize Orchestrator Control Center to cancel workflows that do not finish properly.

Procedure

1 Log in to Control Center as **root**.

2 Click **Troubleshooting**.

3 Cancel running workflows.

Option	Description
Cancel all workflow runs	Enter a workflow ID, to cancel all tokens for that workflow.
Cancel workflow runs by ID	Enter all token IDs, you want to cancel. Separate IDs with a comma.
Cancel all running workflows	Cancel all running workflows on the server.

Note Operations where you cancel workflows by ID might not be successful, as there is no reliable way to cancel the run thread immediately.

Results

On the next server start, the workflows are set in a canceled state.

Enable vRealize Orchestrator Server Debugging

You can start the vRealize Orchestrator server in debug mode to debug issues when developing a plug-in.

Prerequisites

Install and configure the Kubernetes command-line tool on your local machine. See Install and Set Up kubectl.

Procedure

1 Log in to the vRealize Orchestrator Appliance command line as **root**.

2 Run the `kubectl -n prelude edit deployment vco-app` command.

3 Edit the deployment `YAML` file, by adding a debug environment variable to the `vco-server-app` container. The variable must be added under the `env` section of the `vco-server-app` container.

```
containers:
      - command:
      ...
        env:
        - name: DEBUG_PORT
```

```
          value: "your_desired_debug_port"
      ...
    name: vco-server-app
      ...
```

Note When adding the debug environment variable to the `env` section, you must follow the YAML indentation formatting as presented in the preceding example.

4 Save the changes to the deployment file.

   If the edit to the deployment file is successful, you receive the `deployment.extensions/vco-app edited` message.

5 Generate the Kubernetes configuration file, by running the `vracli dev kubeconfig` command.

   As kubeconfig is a developer environment, you are prompted to confirm that you want to continue. Enter **`yes`** to continue or **`no`** to stop.

6 Copy the content of the generated configuration file from `apiVersion: v1` up to and including the `client-key-data` content.

7 Save the generated Kubernetes configuration file on your local machine.

8 Log out of the vRealize Orchestrator Appliance.

9 Finish configuring the debug mode on your local machine.

   a Open a command-line shell.

   b Bind the *KUBECONFIG* environment variable to the saved configuration file.

   Note This example is based on a Linux environment.

   ```
   export KUBECONFIG=/file/path/fileName
   ```

   c To validate that the services are running, run the `kubectl cluster-info` command.

   d To finish configuring the debug mode, perform the following Kubernetes API request.

   Note The value of the *localhost_ debug_ port* variable is the port set in your remote debugging configuration of your Integrated Development Environment (IDE). The value of the *vro_ debug_ port* variable is generated during step 3 of this procedure.

   ```
   kubectl port-forward pod/vco_app_pod_ID localhost_debug_port:vro_debug_port
   ```

Important When configuring your debugging tool, provide the DNS and IP settings of the local machine where you performed the port forward command.

Results

You have configured server debugging for your vRealize Orchestrator Appliance.

Resize the vRealize Orchestrator Appliance Disks

You can modify the disk size of the vRealize Orchestrator Appliance by editing the disk size settings of the vRealize Orchestrator Appliance virtual machine in vSphere.

Prerequisites

Verify that the vRealize Orchestrator Appliance SSH service is enabled. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.

Procedure

1 Verify the currently available disk space in the vRealize Orchestrator Appliance.

Note The vRealize Orchestrator Appliance disks need at least 20 percent free disk space.

a Log in to the vRealize Orchestrator Appliance command line over SSH as **root**.

b Run the `vracli disk-mgr` command.

2 Resize the disk of the vRealize Orchestrator Appliance virtual machine in vSphere.

a Log in to the vSphere Client as an **administrator**.

b Right-click on the virtual machine and select **Edit Settings**.

c On the **Virtual Hardware** tab, expand **Hard disk** to view and change the disk settings, and click **OK**.

For more information on changing the disk size of vSphere virtual machines, see *Change the Virtual Disk Configuration* in *vSphere Virtual Machine Administration*.

3 Trigger the automatic resize in the Photon OS.

a Log in to the vRealize Orchestrator Appliance command line over SSH as **root**.

b Run the `vracli disk-mgr resize` command.

Note You can track the progress of the disk resize procedure at `/var/log/vmware/prelude/disk_resize.log`.

You have resized the vRealize Orchestrator Appliance disks.

4 Verify that the success of the disk resize procedure by running the `disk-mgr` command.

```
vracli disk-mgr
```

What to do next

To troubleshoot problems with the disk resize procedure, see KB 79925.

How to Scale the Heap Memory Size of the vRealize Orchestrator Server

You can scale the heap memory size of the vRealize Orchestrator server by creating a custom profile and modifying the resource metrics file.

You can adjust the heap memory size of the vRealize Orchestrator server, so your orchestration environment can manage changing workloads. For example, you can increase the heap memory of your vRealize Orchestrator deployment if you are planning to manage multiple vCenter servers.

Prerequisites

- Enable SSH access to the vRealize Orchestrator Appliance. See Enable or Disable SSH Access to the vRealize Orchestrator Appliance.
- Increase the RAM of the virtual machine on which vRealize Orchestrator is deployed up to the next suitable increment. Because it is important that enough memory is left available for the rest of the services, the vRealize Orchestrator Appliance resources must be scaled up first. For example, If the desired heap memory is `7G` then the vRealize Orchestrator Appliance RAM should be increased with `4G` respectively because the subtraction between the default heap value of `3G` and the desired heap memory is `4G`.For information on increasing the RAM of a virtual machine in vSphere, see *Change the Memory Configuration* in *vSphere Virtual Machine Administration*.

Procedure

1. Log in the vRealize Orchestrator Appliance command line over SSH as **root**.
2. To create the custom profile directory and the required directory tree that is used when the profile is active, run the following script:

```
vracli cluster exec -- bash -c 'base64 -d <<<
IyBDcmVhdGUgY3VzdG9tIHByb2ZpbGUgZGlyZWN0b3J5Cm1rZGlyIC1wIC9ldGMvdm13YXJlLXByZWx1ZGUvcHJvZml
sZXMvY3VzdG9tLXByb2ZpbGUvCgojIENyZWF0ZSB0aGUgcmVxdWlyZWQgZGlyZWN0b3J5IHRyZWUgdGhhdCB3aWxsIG
JlIHVzZWQgd2hlbiB0aGUgcHJvZmlsZSBpcyBhY3RpdmUKbWtkaXIgLXAgL2V0Yy92bXdhcmUtcHJlbHVkZS9wcm9ma
Wxlcy9jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLwoKIyBDcmVhdGUgImNoZWNrIiBmaWxlIHRoYXQgaXMg
YW4gZXhlY3V0YWJsZSBmaWxlIHJ1biBieSBkZXBsb3kgc2NyaXB0LgpjYXQgPDxFT0YgPiAvZXRjL3Ztd2FyZS1wcmV
sdWRlL3Byb2ZpbGVzL2N1c3RvbS1wcm9maWxlL2NoZWNrCiMhL2Jpbi9iYXNoCmV4aXQgMApFT0YK2htb2QgNzU1IC
9ldGMvdm13YXJlLXByZWx1ZGUvcHJvZmlsZXMvY3VzdG9tLXByb2ZpbGUvY2hlY2sKCiMgQ29weSB2Uk8gcmVzb3VyY
2UgbWV0cmljcyBmaWxlIHRvIHlvdXIgY3VzdG9tIHByb2ZpbGUKY2F0IDw8RU9GID4gL2V0Yy92bXdhcmUtcHJlbHVk
ZS9wcm9maWxlcy9jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLzkwLXJlc291cmNlcy55YW1sCnBvbHlnbG9
0UnVubmVyTWVtb3J5TGltaXQ6IDYwMDBNCnBvbHlnbG90UnVubmVyTWVtb3J5UmVxdWVzdDogMTAwME0KcG9seWdsb3
RSdW5uZXJNZW1vcnlMaW1pdFZjbzogNTYwME0KCnNlcnZlck1lbW9yeUxpbWl0OiA2RwpzZXJ2ZXJNZW1vcnlSZXF1Z
XN0OiA1RwpzZXJ2ZXJKdm1IZWFwTWF4OiA0RwoKY29udHJvbENlbnRlck1lbW9yeUxpbWl0OiAxLjVHCmNvbnRyb2xD
ZW50ZXJNZW1vcnlSZXF1ZXN0OiA3MDBtCkVPRgpjaG1vZCA2NDQgL2V0Yy92bXdhcmUtcHJlbHVkZS9wcm9maWxlcy9
jdXN0b20tcHJvZmlsZS9oZWxtL3ByZWx1ZGVfdmNvLzkwLXJlc291cmNlcy55YW1sCg== | bash'
```

3. Edit the resource metrics file in your custom profile with the desired memory values.

```
vi /etc/vmware-prelude/profiles/custom-profile/helm/prelude_vco/90-resources.yaml
```

4 Save the changes to the resource metrics file and run the `deploy.sh` script.

```
/opt/scripts/deploy.sh
```

Results

You have changed the heap memory size of your vRealize Orchestrator server.

Disaster Recovery of vRealize Orchestrator by Using Site Recovery Manager

You must configure Site Recovery Manager to protect your vRealize Orchestrator. Secure this protection by completing the common configuration tasks for Site Recovery Manager.

Prepare the Environment

You must ensure that you meet the following prerequisites before you start configuring Site Recovery Manager.

- Verify that vSphere 6.0 or later is installed on the protected and recovery sites.
- Verify that you are using Site Recovery Manager 8.1 or later.
- Verify that vRealize Orchestrator is configured.

Configure Virtual Machines for vSphere Replication

You must configure the virtual machines for vSphere Replication or array based replication in order to use Site Recovery Manager.

To enable vSphere Replication on the required virtual machines, perform the following steps.

Procedure

1 In the vSphere Web Client, select a virtual machine on which vSphere Replication should be enabled and click **Actions > All vSphere Replication Actions > Configure Replication**.

2 In the **Replication type** window, select **Replicate to a vCenter Server** and click **Next**.

3 In the **Target site** window, select the vCenter for the recovery site and click **Next**.

4 In the **Replication server** window, select a vSphere Replication server and click **Next**.

5 In the **Target location** window, click **Edit** and select the target datastore, where the replicated files will be stored and click **Next**.

6 In the **Replication options** window, keep the default setting and click **Next**.

7 In the **Recovery settings** window, enter time for **Recovery Point Objective (RPO)** and **Point in time instances**, and click **Next**.

8 In the **Ready to complete** window, verify the settings and click **Finish**.

9 Repeat these steps for all virtual machines on which vSphere Replication must be enabled.

Create Protection Groups

You create protection groups to enable Site Recovery Manager to protect your virtual machines.

You can organize protection groups in folders. The **Protection Groups** tab displays the names of the protection groups, but does not display in which folder they are placed. If you have two protection groups with the same name in different folders, it might be difficult to tell them apart. Therefore, ensure that protection group names are unique across all folders. In environments in which not all users have view privileges for all folders, to be sure of the uniqueness of protection group names, do not place protection groups in folders.

When you create protection groups, wait to ensure that the operations finish as expected. Make sure that Site Recovery Manager creates the protection group and that the protection of the virtual machines in the group is successful.

Prerequisites

Verify that you performed one of the following tasks:

- Included virtual machines in datastores for which you configured array-based replication.
- Satisfied the requirements in *Prerequisites for Storage Policy Protection Groups* and reviewed the *Limitations of Storage Policy Protection Groups* in the *Site Recovery Manager Administration* guide.
- Configured vSphere Replication on your virtual machines.
- Performed a combination of some or all the above.

Procedure

1. In the vSphere Client or vSphere Web Client, click **Site Recovery > Open Site Recovery**.
2. On the Site Recovery home tab, select a site pair and click **View Details**.
3. Select the **Protection Groups** tab, and click **New** to create a protection group.
4. On the Name and direction page, enter a name and description for the protection group, select a direction, and click **Next**.
5. On the Protection group type page, select the protection group type, and click **Next**.

Option	Action
Create an array-based replication protection group	Select **Datastore groups (array-based replication)** and select an array pair.
Create a vSphere Replication protection group	Select **Individual VMs (vSphere Replication)**.
Create a storage policy protection group	Select **Storage Policies (array-based replication)**.

6 Select datastore groups, virtual machines, or storage policies to add to the protection group.

Option	Action
Array-based replication protection groups	Select datastore groups and click **Next**. When you select a datastore group, the virtual machines that the group contains appear in the Virtual machines table.
vSphere Replication protection groups	Select virtual machines from the list, and click **Next**. Only virtual machines that you configured for vSphere Replication and that are not already in a protection group appear in the list.
Storage policy protection groups	Select storage policies from the list, and click **Next**.

7 On the Recovery plan page, you can optionally add the protection group to a recovery plan.

Option	Action
Add to existing recovery plan	Adds the protection group to an existing recovery plan.
Add to new recovery plan	Adds the protection group to a new recovery plan. If you select this option, you must enter a recovery plan name.
Do not add to recovery plan now.	.Select this option if you do not want to add the protection group to a recovery plan.

8 Review your settings and click **Finish**.

You can monitor the progress of the creation of the protection group on the **Protection Group** tab.

- For array-based replication and vSphere Replication protection groups, if Site Recovery Manager successfully applied inventory mappings to the protected virtual machines, the protection status of the protection group is *OK*.
- For storage policy protection groups, if Site Recovery Manager successfully protected all the virtual machines associated with the storage policy, the protection status of the protection group is *OK*.
- For array-based replication and vSphere Replication protection groups, if you did not configure inventory mappings, or if the Site Recovery Manager was unable to apply them, the protection status of the protection group is *Not Configured*.
- For storage policy protection groups, if Site Recovery Manager cannot protect all the virtual machines associated with the storage policy, the protection status of the protection group is *Not Configured*.

What to do next

For array-based replication and vSphere Replication protection groups, if the protection status of the protection groups is *Not Configured*, apply inventory mappings to the virtual machines:

- To apply site-wide inventory mappings, or to check that inventory mappings that you have already set are valid, see *Configure Inventory Mappings* in the *Site Recovery Manager Administration* guide. To apply these mappings to all the virtual machines, see *Apply Inventory Mappings to All Members of a Protection Group* in the *Site Recovery Manager Administration* guide.
- To apply inventory mappings to each virtual machine in the protection group individually, see *Configure Inventory Mappings for an Individual Virtual Machine in a Protection Group* in the *Site Recovery Manager Administration* guide.

For storage policy protection groups, if the protection status of the protection group is *Not Configured*, verify that you have satisfied the requirements in *Prerequisites for Storage Policy Protection Groups* and reviewed the *Limitations of Storage Policy Protection Groups* in the *Site Recovery Manager Administration* guide.

Create a Recovery Plan

You create a recovery plan to establish how Site Recovery Manager recovers virtual machines.

Procedure

1. In the vSphere Client or the vSphere Web Client, click **Site Recovery > Open Site Recovery**.
2. On the Site Recovery home tab, select a site pair, and click **View Details**.
3. Select the **Recovery Plans** tab, and click **New** to create a recovery plan.
4. Enter a name, description, and direction for the plan, select a folder, and click **Next**.
5. Select the group type from the menu.

Option	Description
Protection groups for individual VMs or datastore groups	Select this option to create a recovery plan that contains array-based replication and vSphere Replication protection groups.
Storage policy protection groups	Select this option to create a recovery plan that contains storage policy protection groups. If you are using stretched storage, select this option.

6. Select one or more protection groups for the plan to recover, and click **Next**.
7. From the **Test Network** drop-down menu, select a network to use during test recovery, and click **Next**.

   If there are no site-level mappings, the default option **Use site-level mapping** creates an isolated test network.
8. Review the summary information and click **Finish** to create the recovery plan.

Organize Recovery Plans in Folders

To control the access of different users or groups to recovery plans, you can organize your recovery plans in folders.

Organizing recovery plans into folders is useful if you have many recovery plans. You can limit the access to recovery plans by placing them in folders and assigning different permissions to the folders for different users or groups. For information about how to assign permissions to folders, see *Assign Site Recovery Manager Roles and Permissions* in the *Site Recovery Manager Administration* guide.

Procedure

1 On the **Site Recovery** home tab, select a site pair, and click **View Details**.

2 Click the **Recovery Plans** tab, and in the left pane right-click **Recovery Plans** and click **New Folder**.

3 Enter a name for the folder to create, and click **Add**.

4 Add new or existing recovery plans to the folder.

Option	Description
Create a new recovery plan	Right-click the folder and select **New Recovery Plan**.
Add an existing recovery plan	Right-click a recovery plan from the inventory tree and click **Move**. Select a target folder and click **Move**.

Edit a Recovery Plan

You can edit a recovery plan to change the properties that you specified when you created it. You can edit recovery plans from the protected site or from the recovery site.

Procedure

1 In the vSphere Client or the vSphere Web Client, click **Site Recovery > Open Site Recovery**.

2 On the **Site Recovery** home tab, select a site pair, and click **View Details**.

3 Click the **Recovery Plans** tab, right-click a recovery plan, and click **Edit**.

4 (Optional) Change the name or description of the plan, and click **Next**.

You cannot change the direction and the location of the recovery plan.

5 (Optional) Select or deselect one or more protection groups to add them to or remove them from the plan, and click **Next**.

6 (Optional) From the drop-down menu select a different test network on the recovery site, and click **Next**.

7 Review the summary information and click **Finish** to make the specified changes to the recovery plan.

You can monitor the update of the plan in the **Recent Tasks** view.

Setting System Properties

9

You can set system properties to change the default Orchestrator behavior.

This chapter includes the following topics:

- Setting Server File System Access for Workflows and Actions
- Set Access to Operating System Commands for Workflows and Actions
- Set JavaScript Access to Java Classes
- Set Custom Timeout Property
- Adding a JDBC Connector for the vRealize Orchestrator SQL Plug-In
- Set Scheduled Task and Policy Authentication Token Renewal Property

Setting Server File System Access for Workflows and Actions

In vRealize Orchestrator, the workflows and actions have limited access to specific file system directories. You can extend access to other parts of the server file system by modifying the `js-io-rights.conf` configuration file.

Rules in the js-io-rights.conf File Permitting Write Access to the vRealize Orchestrator System

The `js-io-rights.conf` file contains rules that permit write access to defined directories in the server file system.

Mandatory Content of the js-io-rights.conf File

Each line of the `js-io-rights.conf` file must contain the following information.

- A plus (`+`) or minus (`-`) sign to indicate whether rights are permitted or denied
- The read (`r`), write (`w`), and run (`x`) levels of rights

- The path on which to apply the rights.

  Note The root folder for the `js-io-rights.conf` file is always `/var/run/vco`. In the vRealize Orchestrator Appliance file system, this folder is located under `/data/vco/var/run/vco`. All content with access to the vRealize Orchestrator file system must be mapped under this root folder.

Default Content of the js-io-rights.conf File

The default content of the `js-io-rights.conf` configuration file in the Orchestrator Appliance is as follows:

```
-rwx /
+rwx /var/run/vco
+rx /etc/vco
-rwx /etc/vco/app-server/security/
+rx /var/log/vco/
```

The first two lines in the default `js-io-rights.conf` configuration file allow the following access rights:

`-rwx /`

All access to the file system is denied.

`+rwx /var/run/vco`

Read, write, and run access is permitted in the `/var/run/vco` directory.

Rules in the js-io-rights.conf File

vRealize Orchestrator resolves access rights in the order they appear in the `js-io-rights.conf` file. Each line can override the previous lines.

Important You can permit access to all parts of the file system by setting `+rwx /` in the `js-io-rights.conf` file. However, doing so represents a high security risk.

Set Server File System Access for Workflows and Actions

To change which parts of the server file system that workflows and the vRealize Orchestrator API can access, modify the `js-io-rights.conf` configuration file. The `js-io-rights.conf` file is created when a workflow attempts to access the vRealize Orchestrator server file system.

Procedure

1. Log in to the vRealize Orchestrator Appliance command line as **root**.
2. Navigate to the `/data/vco/var/run/vco/` directory.
3. Open the `js-io-rights.conf` configuration file in a text editor.

4 Add the necessary lines to the `js-io-rights.conf` file to allow or deny access to areas of the file system.

   For example, the following line denies the execution rights in the `/data/vco/var/run/vco/noexec` directory:

   ```
   -x /data/vco/var/run/vco/noexec
   ```

   `/data/vco/var/run/vco/noexec` retains execution rights, but `/data/vco/var/run/vco/noexec/bar` does not. Both directories remain readable and writable.

Results

You modified the access rights to the file system for workflows and for the vRealize Orchestrator API.

Set Access to Operating System Commands for Workflows and Actions

The vRealize Orchestrator API provides a scripting class, `Command`, that runs commands in the vRealize Orchestrator server host operating system. To prevent unauthorized access to the server host, by default, vRealize Orchestrator applications do not have permission to run the `Command` class. If vRealize Orchestrator applications require permission to run commands on the host operating system, you can activate the `Command` scripting class.

You grant permission to use the `Command` class by setting an vRealize Orchestrator configuration system property.

Procedure

1 Log in to Control Center as **`root`**.

2 Click **System Properties**.

3 Click **New**.

4 In the **Key** text box, enter **`com.vmware.js.allow-local-process`**.

5 In the **Value** text box, enter **`true`**.

6 In the **Description** text box, enter a description for the system property.

7 Click **Add**.

8 Click **Save changes** from the pop-up menu.

   A message indicates that you have saved successfully.

9 Wait for the vRealize Orchestrator server to restart.

Results

You granted permissions to vRealize Orchestrator applications to run local commands in the vRealize Orchestrator server host operating system.

Note By setting the `com.vmware.js.allow-local-process` system property to `true`, you allow the `Command` scripting class to write anywhere in the file system. This property overrides any file system access permissions that you set in the `js-io-rights.conf` file for the `Command` scripting class only. The file system access permissions that you set in the `js-io-rights.conf` file still apply to all scripting classes other than `Command`.

Set JavaScript Access to Java Classes

By default, vRealize Orchestrator restricts JavaScript access to a limited set of Java classes. If you require JavaScript access to a wider range of Java classes, you must set an vRealize Orchestrator system property.

Allowing the JavaScript engine full access to the Java virtual machine (JVM) presents potential security issues. Malformed or malicious scripts might have access to all the system components to which the user who runs the vRealize Orchestrator server has access. Therefore, by default the vRealize Orchestrator JavaScript engine can access only the classes in the `java.util.*` package.

If you require JavaScript access to classes outside of the `java.util.*` package, you can list in a configuration file the Java packages to which to allow JavaScript access. You then set the `com.vmware.scripting.rhino-class-shutter-file` system property to point to this file.

Procedure

1. Create a text configuration file to store the list of Java packages to which to allow JavaScript access.

   For example, to allow JavaScript access to all the classes in the `java.net` package and to the `java.lang.Object` class, you add the following content to the file.

   ```
   java.net.*
   java.lang.Object
   ```

2. Enter a name for the configuration file.
3. Save the configuration file in a subdirectory of `/data/vco/usr/lib/vco`.

   Note The configuration file cannot be saved under another directory.

4. Log in to Control Center as **`root`**.
5. Click **System Properties**.
6. Click **New**.
7. In the **Key** text box, enter **`com.vmware.scripting.rhino-class-shutter-file`**.

8 In the **Value** text box, enter `vco/usr/lib/vco/`*your_configuration_file_subdirectory*.

9 In the **Description** text box, enter a description for the system property.

10 Click **Add**.

11 Click **Save changes** from the pop-up menu.

A message indicates that you have saved successfully.

12 Wait for the vRealize Orchestrator server to restart.

Results

The JavaScript engine has access to the Java classes that you specified.

Set Custom Timeout Property

When vCenter Server is overloaded, it takes more time to return the response to the vRealize Orchestrator server than the 20000 milliseconds set by default. To prevent this situation, you must modify the vRealize Orchestrator configuration file to increase the default timeout period.

If the default timeout period expires before the completion of certain operations, the vRealize Orchestrator server log contains errors.

```
Operation 'getPropertyContent' total time : '5742228' for 1823 calls, mean time : '3149.0', min time : '0', max time : '32313' Timeout, unable to get property 'info' com.vmware.vmo.plugin.vi4.model.TimeoutException
```

Procedure

1 Log in to Control Center as **`root`**.

2 Click **System Properties**.

3 Click **New**.

4 In the **Key** text box enter **`com.vmware.vmo.plugin.vi4.waitUpdatesTimeout`**.

5 In the **Value** text box enter the new timeout period in milliseconds.

6 (Optional) In the **Description** text box enter a description for the system property.

7 Click **Add**.

8 Click **Save changes** from the pop-up menu.

A message indicates that you have saved successfully.

9 Restart the Orchestrator server.

Results

The value you set overrides the default timeout setting of 20000 milliseconds.

Adding a JDBC Connector for the vRealize Orchestrator SQL Plug-In

This example demonstrates how you can add a MySQL connector for the vRealize Orchestrator SQL plug-in.

Procedure

1 Add the MySQL connector.jar file to the vRealize Orchestrator Appliance.

   a Log in to the vRealize Orchestrator Appliance command line over SSH as **root**.

   b Navigate to the `/data/vco/var/run/vco` directory.

   ```
   cd /data/vco/var/run/vco
   ```

   c Create a `plugins/SQL/lib/` directory.

   ```
   mkdir -p plugins/SQL/lib/
   ```

   d Copy your MySQL connector.jar file from your local machine to the `/data/vco/var/run/vco/plugins/SQL/lib/` directory by running a secure copy (SCP) command.

   ```
   scp ~/local_machine_dir/your_mysql_connector.jar root@orchestrator_FQDN_or_IP:/
   data/vco/var/run/vco/plugins/SQL/lib/
   ```

   Note You can also use alternative methods for copying your connector.jar file to the vRealize Orchestrator Appliance, such as PSCP.

2 Add the new MySQL property to the Control Center.

   a Log in to the Control Center as **root**.

   b Select **System Properties**.

   c Click **New**.

   d Under **Key**, enter **`o11n.plugin.SQL.classpath`**.

   e Under **Value**, enter **`/var/run/vco/plugins/SQL/lib/your_mysql_connector.jar`**.

   Note The value text box can include multiple JDBC connectors. Each JDBC connector is separated by a semicolon (";"). For example:

   ```
   /var/run/vco/plugins/SQL/lib/your_mysql_connector.jar;/var/run/vco/plugins/SQL/lib/
   your_mssql_connector.jar;/var/run/vco/plugins/SQL/lib/your_other_connector.jar
   ```

f (Optional) Enter a description for the MySQL system property.

g Click **Add**, and wait for the vRealize Orchestrator server to restart.

Note Do not save your JDBC connector.jar file in another directory and do not set a different value to the `o11n.plugin.SQL.classpath` property. Doing so makes the JDBC connector unavailable to your vRealize Orchestrator deployment.

Set Scheduled Task and Policy Authentication Token Renewal Property

Manage how you can enable the renewal of the authentication tokens used in scheduled tasks or policies by setting a system property.

When a scheduled task is configured by non-administrator users in the vRealize Orchestrator Client without an end time, the authentication token for that scheduled workflow expires eight hours after the specified start time. Aside from scheduled tasks, this authentication token is also used for vRealize Orchestrator policies. To make sure that the scheduled workflows or policies in the vRealize Orchestrator deployment continue running, you can set a system property in the Control Center.

Note Authentication tokens cannot be renewed after 90 days of their initial start date.

Prerequisites

Verify that your vRealize Orchestrator deployment uses a vRealize Automation authentication provider or is integrated in vRealize Automation. The `com.vmware.o11n.auth.csp.renewTokens` system property is unavailable for vRealize Orchestrator deployments authenticated with vSphere.

Procedure

1 Log in to the Control Center as **root**.

2 Select **System Properties**.

3 Click **New**.

4 Under **Key**, enter **`com.vmware.o11n.auth.csp.renewTokens`**.

5 Under **Value**, enter **`true`**.

Note For vRealize Orchestrator deployments in vRealize Automation and vRealize Automation Cloud, long-running workflows started from vRealize Automation corrupt the authentication token after its expiration. The token is set to expire eight hours after the specified start time.

6 (Optional) Enter a description for the new system property.

7 Click **Add**, and wait for the vRealize Orchestrator server to restart.

Where to Go from Here

10

When you have installed and configured vRealize Orchestrator, you can use vRealize Orchestrator to automate frequently repeated processes related to the management of the virtual environment.

- Log in to the vRealize Orchestrator Client, run, and schedule workflows on the vCenter Server inventory objects or other objects that vRealize Orchestrator accesses through its plug-ins. See *Using the VMware vRealize Orchestrator Client*.
- Duplicate and modify the standard vRealize Orchestrator workflows and write your own actions and workflows to automate operations in vCenter Server.
- To extend the functionality of the vRealize Orchestrator platform, develop plug-ins.
- Manage your vRealize Orchestrator inventory across multiple vRealize Orchestrator instances with the integration of a remote Git repository. See *Using the VMware vRealize Orchestrator Client*.
- Run workflows on your vSphere inventory objects by using the vSphere Web Client.