

# Multi-Tenancy in vRealize Orchestrator

vRealize Orchestrator 7.6



vmware®

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

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

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

[docfeedback@vmware.com](mailto:docfeedback@vmware.com)

**VMware, Inc.**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

Copyright © 2008–2019 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

# Contents

<b>1</b>	<b>Multitenancy in VMware vRealize Orchestrator</b>	<b>4</b>
	Overview of Multitenancy in vRealize Orchestrator	4
<b>2</b>	<b>Enabling Multitenancy in vRealize Orchestrator</b>	<b>5</b>
	Enable vRealize Orchestrator Multitenancy	5
<b>3</b>	<b>Tenant Isolation in vRealize Orchestrator</b>	<b>6</b>
	Isolation of Access Rights in a Multitenant Orchestrator Environment	7
<b>4</b>	<b>Comparison of Single-Tenant and Multitenant Orchestrator Deployments</b>	<b>9</b>
<b>5</b>	<b>Managing Legacy Custom Content</b>	<b>10</b>
	Isolate Legacy Custom Content	10

# Multitenancy in VMware vRealize Orchestrator

1

*Multitenancy VMware vRealize Orchestrator* provides general information about the multitenant architecture introduced in VMware<sup>®</sup> vRealize Orchestrator 7.4.

## Intended Audience

This information is intended for the vRealize Automation system administrator, tenant administrators, and Orchestrator administrators.

## Overview of Multitenancy in vRealize Orchestrator

vRealize Orchestrator 7.4 introduces a multitenant architecture where multiple vRealize Automation tenants can share a single external or embedded vRealize Automation instance.

A tenant is an organizational unit in a vRealize Automation deployment. vRealize Orchestrator 7.4 introduces a multitenant architecture where multiple vRealize Automation tenants can share a single external or embedded vRealize Orchestrator instance. This is the vRealize Automation instance that Orchestrator uses as an authentication provider. For more information about multitenancy in vRealize Automation, see *Tenancy and User Roles in Preparing and Using Service Blueprints in vRealize Automation*.

The multitenancy feature in vRealize Automation is disabled by default to preserve backwards compatibility and because enabling it brings a major change to the user experience with the product. If you enable multitenancy in vRealize Orchestrator you cannot safely disable it after that.

---

**Note** Only vRealize Automation authentication is supported when Multitenancy is enabled.

---

# Enabling Multitenancy in vRealize Orchestrator

## 2

You can enable multitenancy only when Orchestrator is configured to use vRealize Automation as an authentication provider. Multitenancy is disabled by default.

### Enable vRealize Orchestrator Multitenancy

New vRealize Orchestrator installations are configured to run in a single-tenant mode. To run Orchestrator in a multitenant mode, you must enable it explicitly.

---

**Note** Turning multitenancy on is an irreversible change. Do not enable it if you are not aware of the purpose of the feature.

---

#### Procedure

- 1 Log in to the Orchestrator Appliance Linux console as root.
- 2 Stop the Orchestrator server service and the Control Center service.

```
service vco-server stop && service vco-configurator stop
```

- 3 Navigate to the `/var/lib/vco/tools/configuration-cli/bin` directory.

```
cd /var/lib/vco/tools/configuration-cli/bin
```

- 4 To enable the multitenancy feature, run the `vro-configure.sh` script.

```
./vro-configure.sh enable-multi-tenancy
```

- 5 Start the Orchestrator server service and the Control Center service.

```
service vco-server start && service vco-configurator start
```

You successfully enabled the multitenancy feature in vRealize Orchestrator.

# Tenant Isolation in vRealize Orchestrator

# 3

The Orchestrator multitenancy feature provides a certain level of isolation between tenants.

After enabling multitenancy, the objects that Orchestrator manages split into a system scope and a tenant-specific scope. These objects are workflows, actions, packages, configurations, categories, policies, policy templates, tasks, workflow runs, and others.

## System Scope

System scope is the semantic space that holds all the Orchestrator content that is shared between all tenants. The system content includes the following items:

- All objects included in the default Orchestrator plug-ins.
- Custom objects created before enabling the multitenancy feature.
- Objects created by the vRealize Automation system administrator.
- Predefined automation content (workflows, actions and other) that are managed by the system tenant and available for reading and invoking by all non-system tenants.

Tenants have a read-only access to this content and cannot create, modify, or delete any system-scope objects.

## Tenant-Specific Scope

Tenant-specific objects are associated with the tenant that created them. These objects can be workflows, actions, policies, policy templates, resources and others. Tenants can edit or delete content if they created it. They can run and view system content and their own tenant-specific content.

Tenants cannot view, edit, or delete system scope objects or objects created by other tenants.

## Orchestrator Plug-Ins in a Multitenant Environment

vRealize Orchestrator 7.4 does not support multitenancy of the Orchestrator plug-ins and plug-in inventory objects. Objects that belong to the plug-in inventory are a part of the system scope.

---

**Note** Objects, such as endpoints and inventory items, that you create by running workflows from the plug-in library are visible and accessible by all tenants.

---

## Resource Allocation

The Orchestrator server resources, such as CPU, memory, storage, network bandwidth, database space, maximum number of workflow runs, thread pools, and others are shared between all tenants. If one of the tenants reaches the limit of the allocated resources, all other tenants that use the same Orchestrator instance are affected.

## Security

The security isolation between tenants in vRealize Orchestrator 7.4 uses the system administrator and tenant administrator user roles as they are defined in vRealize Automation. For more information about user roles in vRealize Automation, see *User Roles Overview* in *Preparing and Using Service Blueprints in vRealize Automation*.

---

**Note** The vRealize Automation system administrator must be a member of the Orchestrator administrators group that you enter in the **Admin group** text box when you configure the authentication provider in Control Center.

---

User permissions that are configurable from the Orchestrator client do not correspond to any of the vRealize Automation user roles. You must configure them explicitly for a particular user or a group. For more information about setting user permissions, see *Using the VMware vRealize Orchestrator Client*.

## Isolation of Access Rights in a Multitenant Orchestrator Environment

When multitenancy is enabled, the system administrator and the tenant users have different privileges to manipulate objects in Orchestrator. These privileges depend on whether the objects belong to the system scope or to a tenant-specific scope.

**Table 3-1. Isolation Between Tenants**

Role	System Content	Tenant A Content	Tenant B Content
System Administrator	<ul style="list-style-type: none"> <li>Create, view, edit, delete, and restore system-scope objects</li> <li>Run system workflows</li> <li>Monitor the runs of the workflows that the system administrator started</li> </ul>	<p><b>Note</b> Unless the account designated as a system administrator is also an administrator of one of the existing tenants, the system administrator cannot access or manipulate any tenant-specific content.</p>	
Tenant A Administrator	<ul style="list-style-type: none"> <li>View system content</li> <li>Run system workflows</li> <li>Monitor system workflows started by any Tenant A user</li> </ul>	<ul style="list-style-type: none"> <li>Create, view, edit, delete, and restore objects that belong to Tenant A</li> <li>Run Tenant A workflows</li> <li>Monitor the runs of Tenant A workflows started by any Tenant A user</li> </ul>	Users from Tenant A cannot access any objects created by Tenant B users, except for the resources Tenant B users create by running workflows from the plug-in library.
Tenant B Administrator	<ul style="list-style-type: none"> <li>View system content</li> <li>Run system workflows</li> <li>Monitor system workflows started by any Tenant B user</li> </ul>	Users from Tenant B cannot access any objects created by Tenant A users, except for the resources that Tenant A users create by running workflows from the plug-in library.	<ul style="list-style-type: none"> <li>Create, view, edit, delete, and restore objects that belong to Tenant B</li> <li>Run Tenant B workflows</li> <li>Monitor the runs of Tenant B workflows started by any Tenant B user</li> </ul>
Solution User	<ul style="list-style-type: none"> <li>Create, view, edit, delete, and restore system-scope objects</li> <li>Run system workflows</li> </ul>	<ul style="list-style-type: none"> <li>Create, view, edit, delete, and restore objects that belong to Tenant A</li> <li>Run Tenant A workflows</li> </ul>	<ul style="list-style-type: none"> <li>Create, view, edit, delete, and restore objects that belong to Tenant B</li> <li>Run Tenant B workflows</li> </ul>



# Comparison of Single-Tenant and Multitenant Orchestrator Deployments

## 4

Since version 7.4, Orchestrator can work in a single-tenant mode or a multitenant mode according to the business needs and demands.

### Single-Tenant Deployment

Unless the multitenancy feature is enabled, Orchestrator works in a single-tenant mode. This means that all objects that form the Orchestrator content and runtime are shared between all users. You set different levels of permission on an object to limit the access that different users or user groups can have to the object. For more information about setting user permissions, see *Using the VMware vRealize Orchestrator Client*.

### Multitenant Deployment

When multitenancy is enabled, the tenant-specific objects in Orchestrator are isolated between the vRealize Automation tenants and from the system-scope objects. Tenant users can see the tenant-specific content by logging in to the Orchestrator client with their user name, password, and tenant ID.

---

**Note** The objects from the plug-in inventory are not multi-tenant. These objects are part of the system scope.

---

# Managing Legacy Custom Content

# 5

After you enable the multitenancy feature in vRealize Orchestrator, all the existing objects become system-scope objects.

Similarly to the objects and resources that are included in the Orchestrator platform out of the box, the custom objects that you created before enabling multitenancy are shared between all tenants in a read-only mode and only the system administrator can modify or delete them.

## Isolate Legacy Custom Content

If you want to prevent the custom content from becoming a system-scope content, you can export the custom objects and resources as a package and delete them from the Orchestrator server before enabling the multitenancy feature. After enabling multitenancy, you can import these objects to a particular tenant or tenants.

---

**Note** You can import the same package to multiple tenants independently. You cannot import to a tenant a package that exists in the system scope and you cannot import to the system scope a package that exists as a tenant-specific content.

---

### Prerequisites

Verify that multitenancy is not enabled on your vRealize Orchestrator 7.4 .

### Procedure

- 1 Log in to the Orchestrator client as an administrator.
- 2 Create a package for export.  
See [Create a Package](#).
- 3 Export the package.  
See [Export a Package](#).
- 4 Delete the exported package from the Orchestrator server.  
See [Remove a Package](#).
- 5 Enable multitenancy.  
See [Enable vRealize Orchestrator Multitenancy](#).

- 6 Log in to the Orchestrator client as a tenant administrator to import the package to the particular tenant.
- 7 Import the package.  
See [Import a Package](#).