



# Red Hat OpenStack Platform 16.1

## Introduction to the OpenStack Dashboard

An overview of the OpenStack dashboard graphical user interface



# Red Hat OpenStack Platform 16.1 Introduction to the OpenStack Dashboard

---

An overview of the OpenStack dashboard graphical user interface

OpenStack Team  
rhos-docs@redhat.com

## Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This guide provides an outline of the options available in the OpenStack dashboard user interface.

---

## Table of Contents

|   |           |
|---|-----------|
| <b>PREFACE</b> .....  | <b>3</b>  |
| <b>CHAPTER 1. THE OPENSTACK DASHBOARD</b> .....                             | <b>4</b>  |
| <b>CHAPTER 2. THE ADMIN TAB</b> .....                                       | <b>5</b>  |
| 2.1. VIEW ALLOCATED FLOATING IP ADDRESSES .....                             | 6         |
| <b>CHAPTER 3. THE PROJECT TAB</b> .....                                     | <b>7</b>  |
| <b>CHAPTER 4. THE IDENTITY TAB</b> .....                                    | <b>9</b>  |
| <b>CHAPTER 5. CUSTOMIZING THE DASHBOARD</b> .....                           | <b>10</b> |
| 5.1. OBTAINING THE HORIZON CONTAINER IMAGE .....                            | 10        |
| 5.2. OBTAINING THE RCUE THEME .....   | 10        |
| 5.3. CREATING YOUR OWN THEME BASED ON RCUE .....                            | 11        |
| 5.4. CREATING A FILE TO ENABLE YOUR THEME AND CUSTOMIZE THE DASHBOARD ..... | 11        |
| 5.5. GENERATING A MODIFIED HORIZON IMAGE .....                              | 12        |
| 5.6. USING THE MODIFIED CONTAINER IMAGE IN THE OVERCLOUD .....              | 12        |
| 5.7. EDITING PUPPET PARAMETERS .....  | 13        |
| 5.8. DEPLOYING AN OVERCLOUD WITH A CUSTOMIZED DASHBOARD .....               | 13        |



# PREFACE

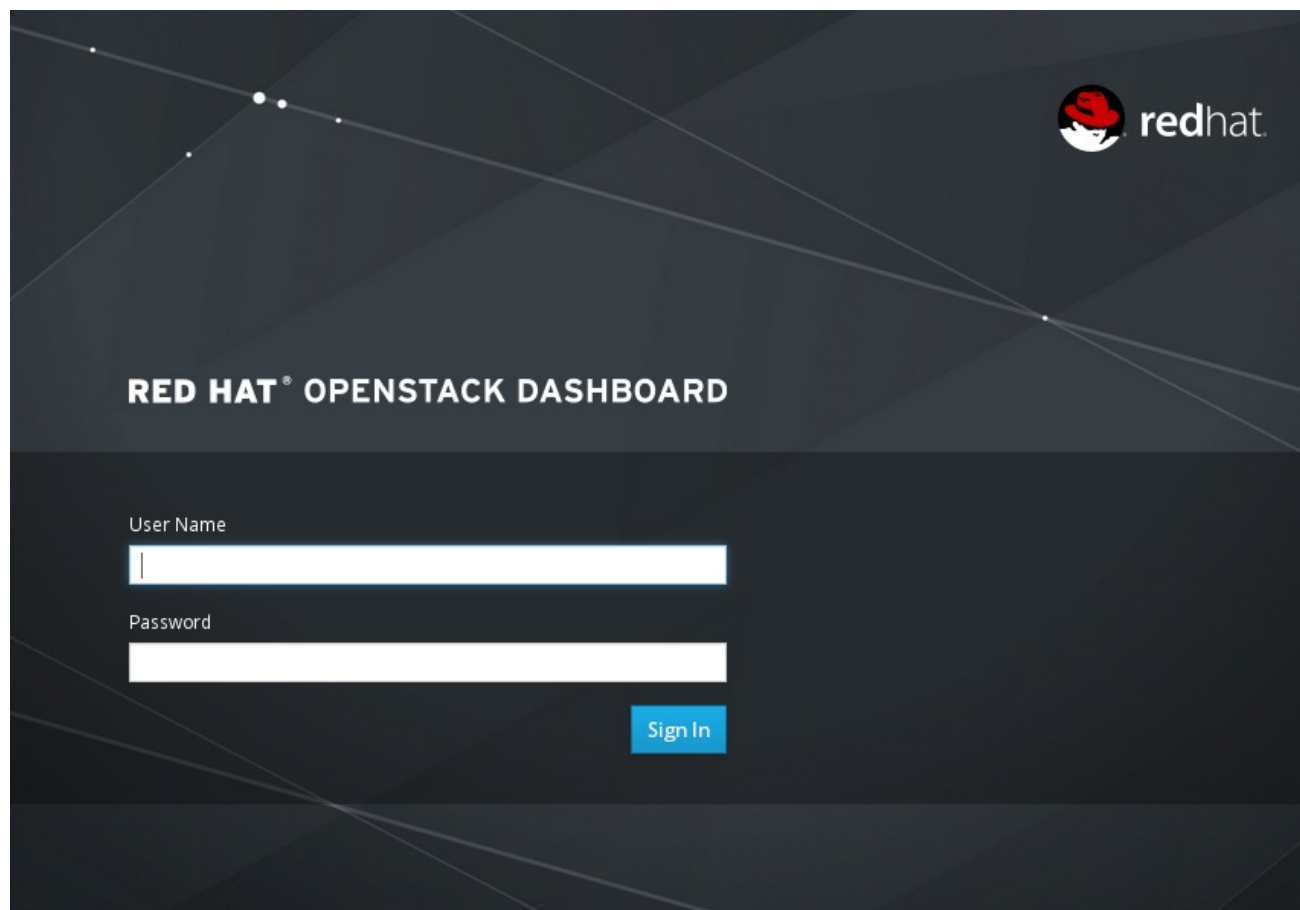
This document provides an outline of the options available in the OpenStack dashboard graphical user interface.

## CHAPTER 1. THE OPENSTACK DASHBOARD

The OpenStack dashboard is a web-based graphical user interface for managing OpenStack services.

To access the browser dashboard, the dashboard service must be installed, and you must know the dashboard host name (or IP) and login password. The dashboard URL is:

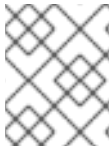
`http://HOSTNAME/dashboard/`





## CHAPTER 2. THE ADMIN TAB

The **Admin** tab provides an interface where administrative users can view usage and manage instances, volumes, flavors, images, projects, users, services, and quotas.



### NOTE

The **Admin** tab displays in the main window only if you have logged in as a user with administrative privileges.

The following options are available in the **Admin** tab:

**Table 2.1. System Panel**

| Parameter Name         | Description   |
|------------------------|---|
| <b>Overview</b>        | View basic reports.   |
| <b>Resource Usage</b>  | Use the following tabs to view the following usages: <ul style="list-style-type: none"> <li>● <b>Usage Report</b> - View the usage report.</li> <li>● <b>Stats</b> - View the statistics of all resources.</li> </ul>     |
| <b>Hypervisors</b>     | View the hypervisor summary.  |
| <b>Host Aggregates</b> | View, create, and edit host aggregates. View the list of availability zones.  |
| <b>Instances</b>       | View, pause, resume, suspend, migrate, soft or hard reboot, and delete running instances that belong to users of some, but not all, projects. Also, view the log for an instance or access an instance using the console. |
| <b>Volumes</b>         | View, create, edit, and delete volumes, and volume types.   |
| <b>Flavors</b>         | View, create, edit, view extra specifications for, and delete flavors. Flavors are the virtual hardware templates in OpenStack.   |

| Parameter Name       | Description  |
|----------------------|--|
| Images               | View, create, edit properties for, and delete custom images.   |
| Networks             | View, create, edit properties for, and delete networks.  |
| Routers              | View, create, edit properties for, and delete routers.   |
| Floating IPs         | View allocated floating IP addresses for all projects.   |
| Defaults             | View and edit the default quotas (maximum limits) for resources in the environment.  |
| Metadata Definitions | Import, view, and edit metadata definition namespaces, and associate the metadata definitions with specific resource types.  |
| System Information   | <p>Contains the following tabs:</p> <ul style="list-style-type: none"><li>● <b>Services</b> - View a list of the services.</li><li>● <b>Compute Services</b> - View a list of all Compute services.</li><li>● <b>Network Agents</b> - View the network agents.</li><li>● <b>Block Storage Services</b> - View a list of all Block Storage services.</li><li>● <b>Orchestration Services</b> - View a list of all Orchestration services.</li></ul> |

## 2.1. VIEW ALLOCATED FLOATING IP ADDRESSES

You can use the **Floating IPs** panel to view a list of allocated floating IP addresses. This information can also be accessed using the command line:

```
$ nova list --all-tenants
```

## CHAPTER 3. THE PROJECT TAB

The **Project** tab provides an interface for viewing and managing the resources of a project. Set a project as active in **Identity > Projects** to view and manage resources in that project.

The screenshot shows the 'Project' tab in the Red Hat Enterprise Linux OpenStack Platform. The top navigation bar includes 'Project', 'Admin', and 'Identity'. Below it, the 'Compute' section is active, with sub-tabs for 'Overview', 'Instances', 'Volumes', 'Images', and 'Access & Security'. The 'Overview' sub-tab is selected, displaying a 'Limit Summary' with seven circular progress indicators for various resources: Instances (Used 2 of 10), VCPUs (Used 2 of 20), RAM (Used 4GB of 50GB), Floating IPs (Allocated 0 of 50), Security Groups (Used 1 of 10), Volumes (Used 0 of 10), and Volume Storage (Used 0Bytes of 1000GB). Below this is a 'Usage Summary' section with a date range selector (From: 2015-07-01, To: 2015-07-09) and a 'Submit' button. It also displays usage statistics: Active Instances: 0, Active RAM: 0Bytes, This Period's VCPU-Hours: 0.00, This Period's GB-Hours: 0.00, and This Period's RAM-Hours: 0.00. A 'Download CSV Summary' button is available. At the bottom, a table with columns 'Instance Name', 'VCPUs', 'Disk', 'RAM', and 'Time since created' is shown, but it contains no data, displaying 'No items to display.' and 'Displaying 0 items'.

The following options are available in the **Project** tab:

**Table 3.1. The Compute Tab**

| Parameter Name   | Description  |
|------------------|--|
| <b>Overview</b>  | View reports for the project.  |
| <b>Instances</b> | View, launch, create a snapshot from, stop, pause, or reboot instances, or connect to them through the console.  |
| <b>Volumes</b>   | Use the following tabs to complete these tasks: <ul style="list-style-type: none"> <li>● <b>Volumes</b> - View, create, edit, and delete volumes.</li> <li>● <b>Volume Snapshots</b> - View, create, edit, and delete volume snapshots.</li> </ul> |
| <b>Images</b>    | View images, instance snapshots, and volume snapshots created by project users, and any images that are publicly available. Create, edit, and delete images, and launch instances from images and snapshots.                                       |

| Parameter Name    | Description   |
|-------------------|---|
| Access & Security | <p>Use the following tabs to complete these tasks:</p> <ul style="list-style-type: none"> <li>● <b>Security Groups</b> - View, create, edit, and delete security groups and security group rules.</li> <li>● <b>Key Pairs</b> - View, create, edit, import, and delete key pairs.</li> <li>● <b>Floating IPs</b> - Allocate an IP address to or release it from a project.</li> <li>● <b>API Access</b> - View API endpoints, download the OpenStack RC file, download EC2 credentials, and view credentials for the logged-in project user.</li> </ul> |

Table 3.2. The Network Tab

| Parameter Name   | Description  |
|------------------|--|
| Network Topology | View the interactive topology of the network.  |
| Networks         | Create and manage public and private networks and subnets.   |
| Routers          | Create and manage routers.   |
| Trunks           | Create and manage trunks. Requires the <b>trunk</b> extension enabled in OpenStack Networking (neutron). |

Table 3.3. The Object Store Tab

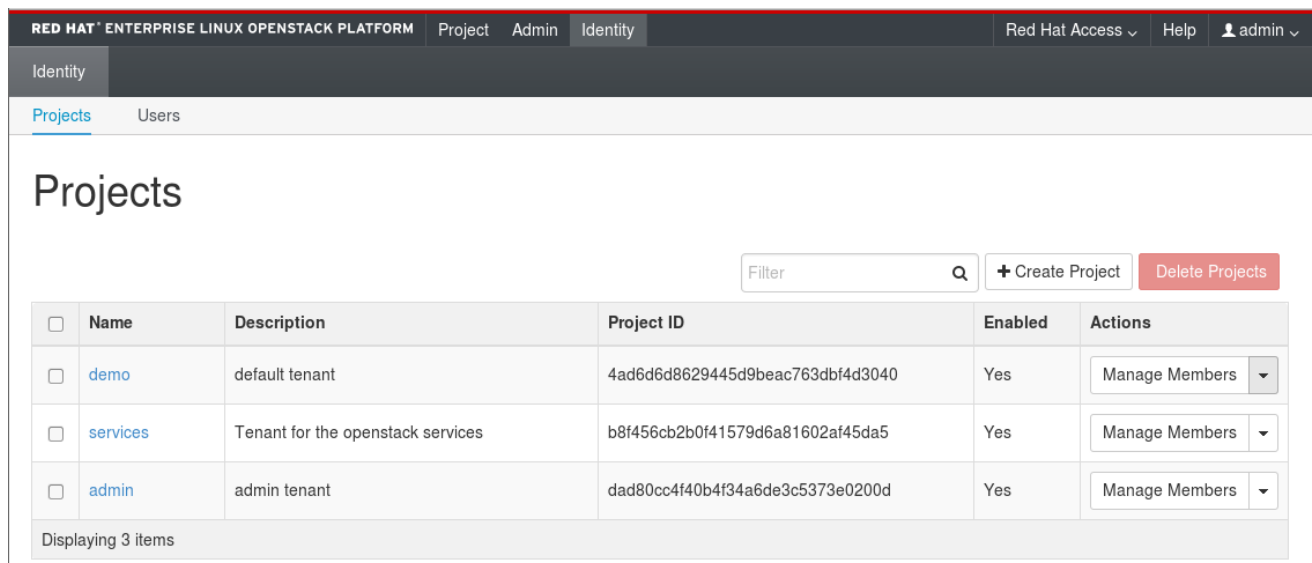
| Parameter Name | Description   |
|----------------|---|
| Containers     | Create and manage storage containers. A container is a storage compartment for data, and provides a way for you to organize your data. It is similar to the concept of a Linux file directory, but it cannot be nested. |

Table 3.4. The Orchestration Tab

| Parameter Name | Description   |
|----------------|---|
| Stacks         | Orchestrate multiple composite cloud applications using templates, through both an OpenStack-native REST API and a CloudFormation-compatible Query API. |

## CHAPTER 4. THE IDENTITY TAB

The **Identity** tab provides an interface for viewing and managing projects and users.



The screenshot shows the OpenStack Identity tab interface. At the top, there is a navigation bar with tabs for 'Project', 'Admin', and 'Identity'. Below this, there is a sub-navigation bar with 'Projects' and 'Users'. The 'Projects' section is active, displaying a table of projects. The table has columns for 'Name', 'Description', 'Project ID', 'Enabled', and 'Actions'. There are three projects listed: 'demo', 'services', and 'admin'. Each project has a 'Manage Members' button in the 'Actions' column. Above the table, there is a search filter and two buttons: '+ Create Project' and 'Delete Projects'.

| <input type="checkbox"/> | Name     | Description                       | Project ID                       | Enabled | Actions        |
|--------------------------|----------|-----------------------------------|----------------------------------|---------|----------------|
| <input type="checkbox"/> | demo     | default tenant                    | 4ad6d6d8629445d9beac763dbf4d3040 | Yes     | Manage Members |
| <input type="checkbox"/> | services | Tenant for the openstack services | b8f456cb2b0f41579d6a81602af45da5 | Yes     | Manage Members |
| <input type="checkbox"/> | admin    | admin tenant                      | dad80cc4f40b4f34a6de3c5373e0200d | Yes     | Manage Members |

Displaying 3 items

The following options are available in the **Identity** tab:

- **Projects** – View, create, edit, and delete projects/tenants, view project usage, add or remove users as project members, modify quotas, and set an active project.
- **Users** – View, create, edit, disable, and delete users, and change user passwords. The **Users** tab displays only if you are logged in as a user with administrative privileges.

For more information on procedures to manage your cloud using the OpenStack Dashboard, see the following guides:

- [Instances and Images guide](#) (also has procedures for volumes and containers)
- [Networking guide](#)
- [Users and Identity Management guide](#)

## CHAPTER 5. CUSTOMIZING THE DASHBOARD

The OpenStack dashboard for Red Hat OpenStack Platform (RHOSP) uses a default theme (**RCUE**), which is stored inside the **horizon** container. You can customize the look and feel of the OpenStack dashboard by adding your own theme to the container image and customizing certain dashboard parameters. You can modify the following elements:

- Logo
- Site colors
- Stylesheets
- HTML title
- Site branding link
- Help URL



### NOTE

To ensure continued support for modified Red Hat OpenStack Platform container images, the resulting images must comply with the [Red Hat Container Support Policy](#).

### 5.1. OBTAINING THE HORIZON CONTAINER IMAGE

You must obtain a copy of the horizon container image. You can pull this image either into the undercloud or a separate client system running **podman**.

#### Procedure

- To pull the horizon container image, run the following command:

```
$ sudo podman pull registry.redhat.io/rhosp-rhel8/openstack-horizon
```

You can now use this image as a basis for a modified image.

### 5.2. OBTAINING THE RCUE THEME

The **horizon** container image is configured to use the Red Hat branded RCUE theme by default. You can use this theme as a basis for your own theme and extract a copy from the container image.

#### Procedure

1. Make a directory for your theme:

```
$ mkdir ~/horizon-themes  
$ cd ~/horizon-themes
```

2. Start a container that executes a null loop. For example, run the following command:

```
$ sudo podman run --rm -d --name horizon-temp registry.redhat.io/rhosp-rhel8/openstack-horizon /usr/bin/sleep infinity
```

3. Copy the RCUE theme from the container to your local directory:

```
$ sudo podman cp -a horizon-temp:/usr/share/openstack-  
dashboard/openstack_dashboard/themes/rcue .
```

4. Kill the container:

```
$ sudo podman kill horizon-temp
```

You should now have a local copy of the RCUE theme.

### 5.3. CREATING YOUR OWN THEME BASED ON RCUE

To use RCUE as a basis, copy the entire RCUE theme directory **rcue** to a new location, for example **mytheme**:

```
$ cp -r rcue mytheme
```

To change a theme's colors, graphics, fonts, among others, edit the files in **mytheme**. When editing this theme, check for all instances of **rcue** and ensure you change them to the new **mytheme** name. This includes paths, files, and directories.

### 5.4. CREATING A FILE TO ENABLE YOUR THEME AND CUSTOMIZE THE DASHBOARD

To enable your theme in the dashboard container, you must create a file to override the **AVAILABLE\_THEMES** parameter.

#### Procedure

1. Create a new file called **\_12\_mytheme\_theme.py** in the **horizon-themes** directory and add the following content:

```
AVAILABLE_THEMES = [('mytheme', 'My Custom Theme', 'themes/mytheme')]
```

The **12** in the file name ensures this file is loaded after the RCUE file, which uses **11**, and overrides the **AVAILABLE\_THEMES** parameter.

You can also set custom parameters in the **\_12\_mytheme\_theme.py** file. For example:

#### SITE\_BRANDING

Set the HTML title that appears at the top of the browser window. For example:

```
SITE_BRANDING = "Example, Inc. Cloud"
```

#### SITE\_BRANDING\_LINK

Changes the hyperlink of the theme's logo, which normally redirects to **horizon:user\_home** by default. For example:

```
SITE_BRANDING_LINK = "http://example.com"
```

## 5.5. GENERATING A MODIFIED HORIZON IMAGE

Once your custom theme is ready, you can create a new container image that enables and uses your theme. Use a **dockerfile** to generate a new container image using the original **horizon** image as a basis. The following is an example of a **dockerfile**:

```
FROM registry.redhat.io/rhosp-rhel8/openstack-horizon
MAINTAINER Acme
LABEL name="rhosp-rhel8/openstack-horizon-mytheme" vendor="Acme" version="0" release="1"
COPY mytheme /usr/share/openstack-dashboard/openstack_dashboard/themes/mytheme
COPY _12_mytheme_theme.py /etc/openstack-dashboard/local_settings.d/_12_mytheme_theme.py
RUN sudo chown horizon:horizon /etc/openstack-
dashboard/local_settings.d/_12_mytheme_theme.py
```

Save this file in your **horizon-themes** directory as **dockerfile**.

To use the **dockerfile** to generate the new image, run the following command:

```
sudo podman build . -t "192.168.24.1:8787/rhosp-rhel8/openstack-horizon-mytheme:0-1"
```

The **-t** option names and tags the resulting image. It uses the following syntax:

```
[LOCATION]/[NAME]:[TAG]
```

### LOCATION

This is usually the location of the container registry that the overcloud eventually pulls uses to pull images. In this instance, you will push this image to the undercloud's container registry, so set this to the undercloud IP and port.

### NAME

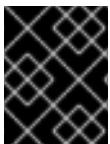
For consistency, this is usually the same name as the original container image followed by the name of your theme. In this case, it is **rhosp-rhel8/openstack-horizon-mytheme**.

### TAG

The tag for the image. Red Hat uses the **version** and **release** labels as a basis for this tag and it is usually a good idea to follow this convention. If you generate a new version of this image, increment the **release** (e.g. **0-2**).

Push the resulting image to the undercloud's container registry:

```
$ podman push 192.168.24.1:8787/rhosp-rhel8/openstack-horizon-mytheme:0-1
```



### IMPORTANT

If updating or upgrading Red Hat OpenStack Platform, you must reapply the theme to the new **horizon** image and push a new version of the modified image to the undercloud.

## 5.6. USING THE MODIFIED CONTAINER IMAGE IN THE OVERCLOUD

To use the resulting container image with your overcloud deployment, edit the environment file that contains the list of container image locations. This environment file is usually named **overcloud-images.yaml**.



Edit the **ContainerHorizonConfigImage** and **ContainerHorizonImage** parameters to point to your modified container image. For example:

```
parameter_defaults:
...
ContainerHorizonConfigImage: 192.168.24.1:8787/rhosp-rhel8/openstack-horizon-mytheme:0-1
ContainerHorizonImage: 192.168.24.1:8787/rhosp-rhel8/openstack-horizon-mytheme:0-1
...
```

Save this new version of the **overcloud-images.yaml** file.

## 5.7. EDITING PUPPET PARAMETERS

The director provides a set of horizon parameters you can modify using environment files. You can also use the **ExtraConfig** hook to set Puppet hieradata. For example, the default help URL points to <https://access.redhat.com/documentation/en/red-hat-openstack-platform>. You can modify this URL with the following environment file content:

```
parameter_defaults:
  ExtraConfig:
    horizon::help_url: "http://openstack.example.com"
```

## 5.8. DEPLOYING AN OVERCLOUD WITH A CUSTOMIZED DASHBOARD

To deploy the overcloud with your dashboard customizations, include the following environment files:

- The environment file with your modified container image locations.
- The environment file with additional dashboard modifications.
- Any other environment files relevant to your overcloud's configuration.

For example:

```
$ openstack overcloud deploy --templates \
  -e /home/stack/templates/overcloud-images.yaml \
  -e /home/stack/templates/help_url.yaml \
  [OTHER OPTIONS]
```