An overview of the OpenStack dashboard graphical user interface
An overview of the OpenStack dashboard graphical user interface

OpenStack Team
rhos-docs@redhat.com
Abstract

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

PREFACE .................................................................................................................. 3  

CHAPTER 1. THE OPENSTACK DASHBOARD ................................................................. 4  

CHAPTER 2. THE ADMIN TAB .................................................................................. 5  
  2.1. VIEW ALLOCATED FLOATING IP ADDRESSES ............................................. 6  

CHAPTER 3. THE PROJECT TAB .............................................................................. 7  

CHAPTER 4. THE IDENTITY TAB ............................................................................ 9  

CHAPTER 5. CUSTOMIZING THE DASHBOARD ......................................................... 10  
  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 ............................................. 11  
  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
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

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>View basic reports.</td>
</tr>
<tr>
<td>Resource Usage</td>
<td>Use the following tabs to view the following usages:</td>
</tr>
<tr>
<td></td>
<td>• <strong>Usage Report</strong> - View the usage report.</td>
</tr>
<tr>
<td></td>
<td>• <strong>Stats</strong> - View the statistics of all resources.</td>
</tr>
<tr>
<td>Hypervisors</td>
<td>View the hypervisor summary.</td>
</tr>
<tr>
<td>Host Aggregates</td>
<td>View, create, and edit host aggregates. View the list of availability zones.</td>
</tr>
<tr>
<td>Instances</td>
<td>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.</td>
</tr>
<tr>
<td>Volumes</td>
<td>View, create, edit, and delete volumes, and volume types.</td>
</tr>
<tr>
<td>Flavors</td>
<td>View, create, edit, view extra specifications for, and delete flavors. Flavors are the virtual hardware templates in OpenStack.</td>
</tr>
<tr>
<td>Parameter Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Images</td>
<td>View, create, edit properties for, and delete custom images.</td>
</tr>
<tr>
<td>Networks</td>
<td>View, create, edit properties for, and delete networks.</td>
</tr>
<tr>
<td>Routers</td>
<td>View, create, edit properties for, and delete routers.</td>
</tr>
<tr>
<td>Floating IPs</td>
<td>View allocated floating IP addresses for all projects.</td>
</tr>
<tr>
<td>Defaults</td>
<td>View and edit the default quotas (maximum limits) for resources in the environment.</td>
</tr>
<tr>
<td>Metadata Definitions</td>
<td>Import, view, and edit metadata definition namespaces, and associate the metadata definitions with specific resource types.</td>
</tr>
</tbody>
</table>

**System Information**

Contains the following tabs:

- **Services** - View a list of the services.
- **Compute Services** - View a list of all Compute services.
- **Network Agents** - View the network agents.
- **Block Storage Services** - View a list of all Block Storage services.
- **Orchestration Services** - View a list of all Orchestration services.

### 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.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>View reports for the project.</td>
</tr>
<tr>
<td>Instances</td>
<td>View, launch, create a snapshot from, stop, pause, or reboot instances, or connect to them through the console.</td>
</tr>
</tbody>
</table>
| Volumes        | Use the following tabs to complete these tasks:  
  - **Volumes** - View, create, edit, and delete volumes.  
  - **Volume Snapshots** - View, create, edit, and delete volume snapshots. |
| Images         | 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. |
Use the following tabs to complete these tasks:

- **Security Groups** - View, create, edit, and delete security groups and security group rules.
- **Key Pairs** - View, create, edit, import, and delete key pairs.
- **Floating IPs** - Allocate an IP address to or release it from a project.
- **API Access** - View API endpoints, download the OpenStack RC file, download EC2 credentials, and view credentials for the logged-in project user.

### Table 3.2. The Network Tab

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Topology</td>
<td>View the interactive topology of the network.</td>
</tr>
<tr>
<td>Networks</td>
<td>Create and manage public and private networks and subnets.</td>
</tr>
<tr>
<td>Routers</td>
<td>Create and manage routers.</td>
</tr>
<tr>
<td>Trunks</td>
<td>Create and manage trunks. Requires the <code>trunk</code> extension enabled in OpenStack Networking (neutron).</td>
</tr>
</tbody>
</table>

### Table 3.3. The Object Store Tab

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Containers</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Table 3.4. The Orchestration Tab

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stacks</td>
<td>Orchestrate multiple composite cloud applications using templates, through both an OpenStack-native REST API and a CloudFormation-compatible Query API.</td>
</tr>
</tbody>
</table>
CHAPTER 4. THE IDENTITY TAB

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

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 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. This customization allows you to modify the following elements:

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

NOTE
To ensure continued support for modified 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. To pull the horizon container image, run the following command:

```
$ sudo podman pull registry.redhat.io/rhosp-beta/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.

Make a directory for your theme:

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

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-beta/openstack-horizon /usr/bin/sleep infinity
```

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 .
```
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. 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-beta/openstack-horizon
MAINTAINER Acme
```
LABEL name="rhosp16-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/rhosp16-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 `rhosp16-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:

```
```

**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 `DockerHorizonConfigImage` and `DockerHorizonImage` parameters to point to your modified container image. For example:

```
parameter_defaults:
...  
DockerHorizonConfigImage: 192.0.2.5:8787/rhosp16-rhel8/openstack-horizon-mytheme:0-1
DockerHorizonImage: 192.0.2.5:8787/rhosp16-rhel8/openstack-horizon-mytheme:0-1
...  
```
CHAPTER 5. CUSTOMIZING THE DASHBOARD

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:

```yaml
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:

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