Red Hat OpenStack Platform 16.2-Beta

Introduction to the OpenStack Dashboard

An overview of the Red Hat OpenStack Platform Dashboard graphical user interface
Red Hat OpenStack Platform 16.2-Beta Introduction to the OpenStack Dashboard

An overview of the Red Hat OpenStack Platform Dashboard graphical user interface

OpenStack Team
rhos-docs@redhat.com
Abstract

This guide provides an outline of the options available in the Red Hat OpenStack Platform Dashboard user interface.
# Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE .................................................. 3

DIRECT DOCUMENTATION FEEDBACK (DDF) FUNCTION NOT AVAILABLE IN THIS BETA RELEASE ...... 4

CHAPTER 1. THE RED HAT OPENSTACK PLATFORM DASHBOARD SERVICE (HORIZON) ............ 5
  1.1. THE ADMIN TAB ........................................ 5
      1.1.1. Viewing allocated floating IP addresses 6
  1.2. THE PROJECT TAB ..................................... 6
  1.3. THE IDENTITY TAB .................................... 8

CHAPTER 2. CUSTOMIZING THE DASHBOARD ........................................ 9
  2.1. OBTAINING THE HORIZON CONTAINER IMAGE ............... 9
  2.2. OBTAINING THE RCUE THEME ................................ 9
  2.3. CREATING YOUR OWN THEME BASED ON RCUE .......... 10
  2.4. CREATING A FILE TO ENABLE YOUR THEME AND CUSTOMIZE THE DASHBOARD ........ 10
  2.5. GENERATING A MODIFIED HORIZON IMAGE ................. 11
  2.6. USING THE MODIFIED CONTAINER IMAGE IN THE OVERCLOUD .... 12
  2.7. EDITING PUPPET PARAMETERS .................................. 12
  2.8. DEPLOYING AN OVERCLOUD WITH A CUSTOMIZED DASHBOARD 13
Making Open Source More Inclusive

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see our CTO Chris Wright’s message.
DIRECT DOCUMENTATION FEEDBACK (DDF) FUNCTION NOT AVAILABLE IN THIS BETA RELEASE

The Direct Documentation Feedback (DDF) function allows users to enter feedback directly on documentation pages on fully supported Red Hat documentation products. The DDF function is not available in this Red Hat OpenStack platform 16.2 beta documentation set.
CHAPTER 1. THE RED HAT OPENSTACK PLATFORM DASHBOARD SERVICE (HORIZON)

The Red Hat OpenStack Platform (RHOSP) Dashboard (horizon) is a web-based graphical user interface that you can use to manage RHOSP services.

To access the browser dashboard, you must install the Dashboard service, and you must know the dashboard host name, or IP, and login password. The dashboard URL is:

http://HOSTNAME/dashboard/

1.1. THE ADMIN TAB

In the Admin tab you can view usage and manage instances, volumes, flavors, images, projects, users, services, and quotas.

**NOTE**

The Admin tab displays in the main window when you log in as an admin user.

The following options are available in the Admin tab:

### Table 1.1. 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>● Usage Report - View the usage report.</td>
</tr>
<tr>
<td></td>
<td>● Stats - 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 with 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 Red Hat OpenStack Platform (RHOSP).</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>

<table>
<thead>
<tr>
<th>System Information</th>
<th>Contains the following tabs:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- Services - View a list of the services.</td>
</tr>
<tr>
<td></td>
<td>- Compute Services - View a list of all Compute services.</td>
</tr>
<tr>
<td></td>
<td>- Network Agents - View the network agents.</td>
</tr>
<tr>
<td></td>
<td>- Block Storage Services - View a list of all Block Storage services.</td>
</tr>
<tr>
<td></td>
<td>- Orchestration Services - View a list of all Orchestration services.</td>
</tr>
</tbody>
</table>

1.1.1. Viewing allocated floating IP addresses

You can use the Floating IPs panel to view a list of allocated floating IP addresses. You can access the same information from the command line with the `nova list --all-projects` command.

1.2. THE PROJECT TAB

In the Project tab you can view and manage project resources. Set a project as active in Identity > Projects to view and manage resources in that project.

The following options are available in the Project tab:

Table 1.2. The Compute tab

<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>
### Table 1.3. 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 1.4. 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 1.5. 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 with templates, through both an OpenStack-native REST API and a CloudFormation-compatible Query API.</td>
</tr>
</tbody>
</table>

1.3. THE IDENTITY TAB

In the **Identity** tab you can view and manage projects and users.

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

- **Projects** - View, create, edit, and delete projects, 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 is available when you log in as an admin user.

For more information about managing your cloud with the Red Hat OpenStack Platform dashboard, see the following guides:

- Creating and Managing Instances
- Creating and Managing Images
- Networking guide
- Users and Identity Management guide
CHAPTER 2. CUSTOMIZING THE DASHBOARD

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

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

NOTE

To ensure continued support for modified RHOSP container images, the resulting images must comply with the Red Hat Container Support Policy.

2.1. OBTAINING THE HORIZON CONTAINER IMAGE

To obtain a copy of the horizon container image, pull the image either into the undercloud or a separate client system that is running podman.

Procedure

- Pull the horizon container image:

  $ sudo podman pull registry.redhat.io/rhosp-beta/openstack-horizon:16.2-beta

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

2.2. OBTAINING THE RCUE THEME

The horizon container image uses 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. Create 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-beta/openstack-horizon /usr/bin/sleep infinity
3. Copy the RCUE theme from the container to your local directory:

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

4. Kill the container:

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

   **Result:** You now have a local copy of the RCUE theme.

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

To use RCUE as a basis, copy the entire RCUE theme directory rcue to a new location. This procedure uses `mytheme` as an example name.

**Procedure**

- Copy the theme:

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

To change the colors, graphics, fonts, and other elements of a theme, edit the files in mytheme. When you edit this theme, check for all instances of rcue including paths, files, and directories to ensure that you change them to the new mytheme name.

### 2.4. 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:

   ```python
   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.

2. Optional: You can also set custom parameters in the `_12_mytheme_theme.py` file. Use the following examples as a guide:

   **SITE_BRANDING**
   
   Set the HTML title that appears at the top of the browser window.

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

   **SITE_BRANDING_LINK**
   
   Changes the hyperlink of the theme logo, which normally redirects to `horizon:user_home`
Changes the hyperlink of the theme logo, which normally redirects to `horizon:user_home` by default.

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

### 2.5. GENERATING A MODIFIED HORIZON IMAGE

When your custom theme is ready, you can create a new container image that uses your theme.

**Procedure**

1. Use a **dockerfile** to generate a new container image using the original **horizon** image as a basis, as shown in the following example:

   ```
   FROM registry.redhat.io/rhosp-rhel8/openstack-horizon
   MAINTAINER Acme
   LABEL name="rhosp-beta/openstack-horizon-mytheme" vendor="Acme" version="0"
   release="1"
   COPY mytheme /usr/share/openstack-dashboard/openstack_dashboard/themes/mytheme
   COPY _12_mytheme_theme.py /var/lib/config-data/horizon/etc/openstack-dashboard/local_settings.d/_12_mytheme_theme.py
   RUN sudo chown horizon:horizon /var/lib/config-data/horizon/etc/openstack-dashboard/local_settings.d/_12_mytheme_theme.py
   
   $ sudo podman build . -t "172.24.10.10:8787/rhosp-rhel8/openstack-horizon:0-5" --log-level debug
   ```

   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 uses to pull images. In this instance, you push this image to the container registry of the undercloud, 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 instance, 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. If you generate a new version of this image, increment the `release`, for example, `0-2`.

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

3. Use the dockerfile to generate the new image:

   ```
   $ sudo openstack tripleo container image push --local 172.24.10.10:8787/rhosp-rhel8/openstack-horizon:0-5
   ```
5. Verify that the image has uploaded to the local registry:

```
[stack@director horizon-themes]$ curl http://172.24.10.10:8787/v2/_catalog | jq .repositories[] | grep -i horizon "rhosp-rhel8/openstack-horizon"
[stack@director horizon-themes]$ sudo openstack tripleo container image list | grep hori
<<<<<<<<<<<Uploaded
[stack@director ~]$]
```

**IMPORTANT**

If you update or upgrade 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.

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

To use the container image that you modified 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`.

**Procedure**

1. Edit the `DockerHorizonConfigImage` and `DockerHorizonImage` parameters to point to your modified container image:

   ```yaml
   parameter_defaults:
   ...
   ...
   ```

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

### 2.7. EDITING PUPPET PARAMETERS

Director provides a set of dashboard parameters that you can modify with environment files.

**Procedure**

- Use the `ExtraConfig` parameter to set Puppet hieradata. For example, the default help URL points to `https://access.redhat.com/documentation/en/red-hat-openstack-platform`. To modify this URL, use the following environment file content and replace the URL:

  ```yaml
  parameter_defaults:
  ExtraConfig:
  horizon::help_url: "http://openstack.example.com"
  ```
2.8. DEPLOYING AN OVERCLOUD WITH A CUSTOMIZED DASHBOARD

Procedure

- To deploy the overcloud with your dashboard customizations, include the following environment files in the `openstack overcloud deploy` command:
  - The environment file with your modified container image locations.
  - The environment file with additional dashboard modifications.
  - Any other environment files that are relevant to your overcloud configuration.

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