Heat Dashboard Documentation

Release 4.0.2.dev3

OpenStack Developers

Feb 01, 2024

CONTENTS

CHAPTER

ONE

HEAT DASHBOARD INSTALLATION GUIDE

This page describes the manual installation of heat-dashboard, while distribution packages may provide more automated process.

Note: This page assumes horizon has been installed. Horizon setup is beyond the scope of this page.

Install Heat Dashboard with all relevant packages to your Horizon environment.

```
pip install heat-dashboard
```

In most cases, heat-dashboard is installed into your python site-packages directory like `/usr/local/lib/python2.7/site-packages`. We refer to the directory of heat-dashboard as `<heat-dashboard-dir>` below and it would be `<site-packages>/heat_dashboard` if installed via pip. The path varies depending on Linux distribution you use.

To enable heat-dashboard plugin, you need to put horizon plugin setup files into horizon enabled directory.

The plugin setup files are found in `<heat-dashboard-dir>/enabled`.

```
$ cp <heat-dashboard-dir>/enabled/_[1-9]*.py \
      /usr/share/openstack-dashboard/openstack_dashboard/local/enabled
```

Note: The directory `local/enabled` may be different depending on your environment or distribution used. The path above is one used in Ubuntu horizon package.

Configure the policy file for heat-dashboard in OpenStack Dashboard `local_settings.py`.

```
POLICY_FILES['orchestration'] = '<heat-dashboard-dir>/conf/heat_policy.json
↪'
```

Note: If your `local_settings.py` has no `POLICY_FILES` yet, you need to define the default `POLICY_FILES` in `local_settings.py`. If you use the example `local_settings.py` file from horizon, what you need is to uncomment `POLICY_FILES` (which contains the default values).

You can also add additional configurations to `local_settings.py`. For more detail, see *Configuration*. You can also find an example file at `<heat-dashboard-dir>/heat_dashboard/local_settings.d`.

Compile the translation message catalogs of heat-dashboard.

```
$ cd <heat-dashboard-dir>
$ python ./manage.py compilemessages
```

Run the Django update commands. Note that `compress` is required when you enable compression.

```
$ cd <horizon-dir>
$ DJANGO_SETTINGS_MODULE=openstack_dashboard.settings python manage.py␣
↪collectstatic --noinput
$ DJANGO_SETTINGS_MODULE=openstack_dashboard.settings python manage.py␣
↪compress --force
```

Finally, restart your web server. For example, in case of apache:

```
$ sudo service apache2 restart
```

CHAPTER

TWO

HEAT DASHBOARD CONFIGURATION GUIDE

2.1 Configuration

Heat Dashboard has configuration option as below.

For more configurations, see Configuration Guide in the Horizon documentation.

2.1.1 OPENSTACK_HEAT_STACK

New in version 9.0.0(Mitaka).

Default:

```
{
    'enable_user_pass': True
}
```

A dictionary of settings to use with heat stacks. Currently, the only setting available is enable_user_pass, which can be used to disable the password field while launching the stack. Currently HEAT API needs user password to perform all the heat operations because in HEAT API trusts is not enabled by default. So, this setting can be set as False in-case HEAT uses trusts by default otherwise it needs to be set as True.

CHAPTER

THREE

HEAT DASHBOARD USER DOCUMENTATION

3.1 Launch and manage stacks

OpenStack Orchestration is a service that you can use to orchestrate multiple composite cloud applications. This service supports the use of both the Amazon Web Services (AWS) CloudFormation template format through both a Query API that is compatible with CloudFormation and the native OpenStack Heat Orchestration Template (HOT) format through a REST API.

These flexible template languages enable application developers to describe and automate the deployment of infrastructure, services, and applications. The templates enable creation of most OpenStack resource types, such as instances, floating IP addresses, volumes, security groups, and users. Once created, the resources are referred to as stacks.

The template languages are described in the Template Guide.

3.1.1 Launch a stack

1. Log in to the dashboard.
2. Select the appropriate project from the drop down menu at the top left.
3. On the *Project* tab, open the *Orchestration* tab and click *Stacks* category.
4. Click *Launch Stack*.
5. In the *Select Template* dialog box, specify the following values:

Template Source	Choose the source of the template from the list.
Template URL/File/Data	Depending on the source that you select, enter the URL, browse to the file location, or directly include the template.
Environment Source	Choose the source of the environment from the list. The environment files contain additional settings for the stack.
Environment File/Data	Depending on the source that you select, browse to the file location, directly include the environment

6. Click *Next*.
7. In the *Launch Stack* dialog box, specify the following values:

Stack Name	Enter a name to identify the stack.
Creation Timeout (minutes)	Specify the number of minutes that can elapse before the launch of the stack times out.
Rollback On Failure	Select this check box if you want the service to roll back changes if the stack fails to launch.
Password for user demo	Specify the password that the default user uses when the stack is created.
DBUsername	Specify the name of the database user.
LinuxDistribution	Specify the Linux distribution that is used in the stack.
DBRootPassword	Specify the root password for the database.
KeyName	Specify the name of the key pair to use to log in to the stack.
DBName	Specify the name of the database.
DBPassword	Specify the password of the database.
InstanceType	Specify the flavor for the instance.

8. Click *Launch* to create a stack. The *Stacks* tab shows the stack.

After the stack is created, click on the stack name to see the following details:

Topology The topology of the stack.

Overview The parameters and details of the stack.

Resources The resources used by the stack.

Events The events related to the stack.

Template The template for the stack.

3.1.2 Manage a stack

1. Log in to the dashboard.
2. Select the appropriate project from the drop down menu at the top left.
3. On the *Project* tab, open the *Orchestration* tab and click *Stacks* category.
4. Select the stack that you want to update.
5. Click *Change Stack Template*.
6. In the *Select Template* dialog box, select the new template source or environment source.
7. Click *Next*.

   The *Update Stack Parameters* window appears.
8. Enter new values for any parameters that you want to update.
9. Click *Update*.

3.1.3 Delete a stack

When you delete a stack, you cannot undo this action.

1. Log in to the dashboard.
2. Select the appropriate project from the drop down menu at the top left.
3. On the *Project* tab, open the *Orchestration* tab and click *Stacks* category.
4. Select the stack that you want to delete.
5. Click *Delete Stack*.
6. In the confirmation dialog box, click *Delete Stack* to confirm the deletion.

3.2 Generate a Heat Orchestration Template

Heat Dashboard provides a user-friendly interface to generate Heat Orchestration templates in a Drag and Drop way.

3.2.1 Generate a template

1. Log in to the dashboard.
2. On the *Project* tab, open the *Orchestration* tab and click *Template Generator* category.
3. Wait until the page is completely loaded. It may take several seconds.
4. Click the dropdown menu of Template Version, and choose an appropriate version.
5. Drag icons of resource types at the top of the page to the central canvas.
6. Click icons on the canvas to specify properties of resources.
7. Click EDIT button at the top of the canvas, to enable manipulate mode.
8. When in manipulate mode, click on CONNECT button to add an edge between icons.
9. Click edges to show details of connections.
10. Click the Generate Template button at the top-right of the page and generated template will be shown in a text box. You can also add modification to the template here.
11. Click CREATE STACK to jump to continue to *Launch Stack*.
12. Click DOWNLOAD STACK to download the generated template.
13. You can also click the Manage Drafts button at the top-right of the page, to temporarily save the editing canvas or to load a saved one.

3.2.2 Currently Supported resource types

13 types of resources are supported in the first release of Heat Dashboard.

1. OS::Cinder::Volume
2. OS::Cinder::VolumeAttachment
3. OS::Heat::ResourceGroup
4. OS::Neutron::FloatingIP
5. OS::Neutron::FloatingIPAssociation
6. OS::Neutron::Net
7. OS::Neutron::Port
8. OS::Neutron::Router
9. OS::Neutron::RouterInterface
10. OS::Neutron::SecurityGroup
11. OS::Neutron::Subnet
12. OS::Nova::KeyPair
13. OS::Nova::Server

CHAPTER

FOUR

CONTRIBUTOR DOCUMENTATION

4.1 How to Contribute

4.1.1 Contributor License Agreement

In order to contribute to the Heat Dashboard project, you need to have signed OpenStacks contributors agreement.

See also:

- https://docs.openstack.org/infra/manual/developers.html
- https://wiki.openstack.org/CLA

4.1.2 Project Hosting Details

Bug tracker https://storyboard.openstack.org/#!/project/992

Code Hosting https://opendev.org/openstack/heat-dashboard/

Code Review https://review.opendev.org/#/q/status:open+project:openstack/heat-dashboard,n,z

4.2 Use Heat Dashboard in DevStack

Set up your `local.conf` to enable heat-dashboard:

```
[[local|localrc]]
enable_plugin heat-dashboard https://opendev.org/openstack/heat-dashboard
```

Note: You also need to install Heat itself into DevStack to use Heat Dashboard.