Watcher Dashboard Documentation

Release 14.1.0.dev18

OpenStack Foundation

Nov 11, 2025

CONTENTS

OpenStack Watcher provides a flexible and scalable resource optimization service for multi-tenant OpenStack-based clouds. Watcher provides a complete optimization loopincluding everything from a metrics receiver, complex event processor and profiler, optimization processor and an action plan applier. This provides a robust framework to realize a wide range of cloud optimization goals, including the reduction of data center operating costs, increased system performance via intelligent virtual machine migration, increased energy efficiency and more!

Watcher project consists of several source code repositories:

- watcher - is the main repository. It contains code for Watcher API server, Watcher Decision Engine and Watcher Applier.
- python-watcherclient - Client library and CLI client for Watcher.
- watcher-dashboard - Watcher Horizon plugin.

The documentation provided here is continually kept up-to-date based on the latest code, and may not represent the state of the project at any specific prior release.

CHAPTER

ONE

INSTALL GUIDE

1.1 Installation

First off, create a virtual environment and install the Horizon dependencies:

```
$ git clone https://github.com/openstack/horizon
$ cd horizon
$ python tools/install_venv.py
```

We will refer to the folder you are now in as <HORIZON_DIR>. If you want more details on how to install Horizon, you can have a look at the Horizon documentation, especially their quickstart tutorial.

Then, you need to install Watcher Dashboard on the server running Horizon. To do so, you can issue the following commands:

```
$ git clone https://opendev.org/openstack/watcher-dashboard
$ cd watcher-dashboard
$ pip install -e .
```

We will refer to the folder you are now in as <DASHBOARD_DIR>.

The next step is now to register the Watcher Dashboard plugins against your Horizon. To do so, you can execute the tools/register_plugin.sh:

```
$ cd <DASHBOARD_DIR>
$ ./tools/register_plugin.sh . <HORIZON_DIR>
```

This script will then create the needed symlinks within Horizon so that it can load the Watcher plugin when it starts.

If you wish to have Horizon running being an Apache server, do not forget to start the service via the following command:

```
$ sudo service apache2 restart
```

For more details on how to configure Horizon for a production environment, you can refer to their online installation guide.

1.2 DevStack setup

Add the following to your DevStack `local.conf` file

```
enable_plugin watcher-dashboard https://opendev.org/openstack/watcher-
↪dashboard
```

1.3 Unit testing

First of all, you have to create an environment to run your tests in. This step is actually part of the `run_tests.sh` script which creates and maintains a clean virtual environment.

Here below is the basic command to run Watcher Dashboard tests:

```
$ ./run_tests.sh
```

The first time you will issue the command above, you will be asked if you want to create a virtual environment. So unless you have installed everything manually (in which case you should use the `-N` flag), you need to accept

1.4 Integration testing

Before being able to run integration tests, you need to have a Horizon server running with Watcher Dashboard plugin configured. To do so, you can run a test server using the following command:

```
$ ./run_tests.sh --runserver 0.0.0.0:8000
```

By default, integration tests expect to find a running Horizon server at `http://localhost:8000/` but this can be customized by editing the `watcher_dashboard/test/integration_tests/horizon.conf` configuration file. Likewise, this Horizon will be looking, by default, for a Keystone backend at `http://localhost:5000/v2.0`. So in order to customize its location, you will have to edit `watcher_dashboard/test/settings.py` by updating the `OPENSTACK_KEYSTONE_URL` variable.

To run integration tests:

```
$ ./run_tests.sh --integration
```

You can use PhantomJS as a headless browser to execute your integration tests. On an Ubuntu distribution you can install it via the following command:

```
$ sudo apt-get install phantomjs
```

Then you can run your integration tests like this:

```
$ ./run_tests.sh --integration --selenium-headless
```

Please note that these commands are also available via `tox`.

Note

As of the Mitaka release, the dashboard for watcher is now maintained outside of the Horizon codebase, in this repository.

1.5 Policies

You can enable policies on Watcher `Optimization` panel, by updating in the `<HORIZON_DIR>/openstack_dashboard/settings.py` configuration file the following parameters

> POLICY_FILES = { infra-optim: watcher_policy.json, }

You can also update the file `<HORIZON_DIR>/openstack_dashboard/conf/watcher_policy.conf` to customize your policies.

1.6 Links

Watcher project: https://opendev.org/openstack/watcher/

Watcher at github: https://github.com/openstack/watcher

Watcher at wiki.openstack.org: https://wiki.openstack.org/wiki/Watcher

Launchpad project: https://launchpad.net/watcher

Join us on IRC (Internet Relay Chat):

```
Network: OFTC (https://www.oftc.net/)
Channel: #openstack-watcher
```

Or send an email to openstack-discuss@lists.openstack.org using [watcher] in object

CHAPTER

TWO

DEVELOPER GUIDE

2.1 Contributing to Watcher

If youre interested in contributing to the Watcher project, the following will help get you started.

2.1.1 Contributor License Agreement

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

See also

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

2.1.2 LaunchPad Project

Most of the tools used for OpenStack depend on a launchpad.net ID for authentication. After signing up for a launchpad account, join the openstack team to have access to the mailing list and receive notifications of important events.

See also

- https://launchpad.net/
- https://launchpad.net/watcher
- https://launchpad.net/watcher-dashboard
- https://launchpad.net/~openstack

2.1.3 Project Hosting Details

Bug tracker
https://launchpad.net/watcher-dashboard

Blueprints
https://blueprints.launchpad.net/watcher-dashboard

Mailing list (prefix subjects with `[watcher]` for faster responses)
http://lists.openstack.org/cgi-bin/mailman/listinfo/openstack-discuss

Wiki
https://wiki.openstack.org/wiki/Watcher

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

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

IRC Channel
`#openstack-watcher` (changelog)

Weekly Meetings
On Wednesdays at 14:00 UTC on even weeks in the `#openstack-meeting-4` IRC channel, 13:00 UTC on odd weeks in the `#openstack-meeting-alt` IRC channel (meetings logs)

CHAPTER

THREE

INDICES AND TABLES

- genindex
- search