Ironic Tempest Plugin Documentation

Release 2.12.1.dev27

OpenStack Foundation

Mar 12, 2025

CONTENTS

This repository contains a Tempest plugin for OpenStack Bare Metal and Bare Metal Introspection projects.

- Free software: Apache license

- Documentation: https://docs.openstack.org/ironic-tempest-plugin

- Source: https://opendev.org/openstack/ironic-tempest-plugin

- Bugs: https://storyboard.openstack.org/#!/project/951

Contents:

INSTALLATION

At the command line:

```
$ pip install ironic-tempest-plugin
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv ironic-tempest-plugin
$ pip install ironic-tempest-plugin
```

USAGE

2.1 Configuring

Update your Tempest configuration to enable support for ironic:

```
[service_available]
ironic = True
```

If introspection tests are needed, also enable support for ironic-inspector:

```
[service_available]
ironic_inspector = True
```

See the following example configurations for more details:

2.1.1 Full cloud with the Compute service and flat networking

This section documents running tempest on a full OpenStack cloud with the Compute, Image and Networking services enabled. The Bare Metal Introspection service (ironic-inspector) is not enabled. Flat networking is used.

Prerequisite

- Create a bare metal flavor in the Compute service in advance and record its ID (`<flavor uuid>` below).

- Create an image to use for instances and record its ID (`<image uuid>`). It can be either a whole disk or a partition image.

- Create and record the name or UUID of a flat network to use for bare metal instances (`<network name>`).

- Get the minimum and maximum API versions that you want to test against. Check the API version history to find the appropriate versions for your deployment.

> **Note**
>
> The minimum version can usually be set to `1.1`.

- Enroll at least one node and make it `available`.

Configuration

```ini
[service_available]
# Enable ironic tests.
ironic = True

# Disable ironic-inspector tests.
ironic-inspector = False

[baremetal]
# Minimum and maximum API versions to test against.
min_microversion = <min API version as X.Y>
max_microversion = <max API version as X.Y>
# Driver to use for API tests for Queens and newer:
driver = fake-hardware

[compute]
# Configure the bare metal flavor so that the Compute services provisions
# bare metal instances during the tests.
flavor_ref = <flavor uuid>
flavor_ref_alt = <flavor uuid>

# Configure the image to use.
image_ref = <image uuid>
image_ref_alt = <image uuid>

# Configure the network to use.
fixed_network_name = <network name>

[compute-feature-enabled]
# Ironic does not support this feature.
disk_config = False

# Not supported with flat networking.
interface_attach = False

[auth]
# Not supported with flat networking.
create_isolated_networks = False

[network]
# Required for flat networking.
shared_physical_network = True
```

2.2 Running

Run tests as described in the Tempest documentation. The following patterns can be used with `--regex` option to only run bare metal tests:

ironic

> all bare metal tests

`ironic_tempest_plugin.tests.api`

> only API tests using fake hardware, without other OpenStack services (these tests are run by jobs starting with `ironic-tempest-functional-python3`)

`ironic_tempest_plugin.tests.scenario`

> all integration tests, excluding the API tests with fake hardware (these tests are run by most of the jobs)

`ironic_standalone`

> standalone bare metal tests that do not use the Compute service (these tests are run by the jobs `ironic-standalone` and `ironic-standalone-redfish`)

InspectorBasicTest

> basic introspection tests (these tests are run by most of the jobs with `ironic-inspector` in their name)

InspectorDiscoveryTest

> introspection auto-discovery tests (these tests are run by the job `ironic-inspector-tempest-discovery` and require additional set up)

CONTRIBUTING

If you would like to contribute to the development of OpenStack, you must follow the steps in this page:

https://docs.openstack.org/infra/manual/developers.html

If you already have a good understanding of how the system works and your OpenStack accounts are set up, you can skip to the development workflow section of this documentation to learn how changes to OpenStack should be submitted for review via the Gerrit tool:

https://docs.openstack.org/infra/manual/developers.html#development-workflow

Pull requests submitted through GitHub will be ignored.

Bugs should be filed on Launchpad, not GitHub:

https://bugs.launchpad.net/ironic

INDICES AND TABLES

- genindex
- modindex
- search