openstack-tempest-skiplist Documentation

Release 0.0.1.dev521

Red Hat, Inc.

Jul 07, 2023

CONTENTS

openstack-tempest-skiplist is a tool for generate tempest skiplist based on different criteria such as release, installer type, and job name

CHAPTER

ONE

CONTENT:

1.1 Overview

openstack-tempest-skiplist will automatically generate the list of tests to be skipped by tempest based on the criterias passed to the tool

- Free software: Apache license
- Documentation: https://docs.openstack.org/openstack-tempest-skiplist/latest/
- Storyboard: https://storyboard.openstack.org/#!/project/1173
- Source: https://opendev.org/openstack/openstack-tempest-skiplist
- Bugs: https://storyboard.openstack.org/#!/project/1173
- Release notes: https://docs.openstack.org/releasenotes/openstack-tempest-skiplist

1.2 Installation

1.2.1 Git

1. Clone and change to the directory:

```
$ git clone https://opendev.org/openstack/openstack-tempest-skiplist
$ cd openstack-tempest-skiplist
```

2. Create a virtual environment using **virtualenv**:

```
$ virtualenv .venv
$ source .venv/bin/activate
```

3. Install requirements in the newly created virtual environment:

```
(.venv) $ pip install .
```

4. *(optional)* Instead of manual installation described in steps 2 and 3 above, tox can be used for installing the requirements as well. To create python 3.6 environment run following:

```
$ tox -epy36
$ source .tox/py36/bin/activate
```

1.2.2 Pip installation

Install `openstack-tempest-skiplist` via pip as follows:

```
$ pip install openstack-tempest-skiplist
```

1.3 Formatting

1.3.1 YAML File

The YAML file used by *openstack-tempest-skiplist* use the following pattern:

```yaml
known_failures:
  - test: 'full.tempest.test'
    bz: 'https://bugzilla.redhat.com/1'
    lp: 'https://launchpad.net/bugs/1'
    deployment:
      - 'undercloud'
      - 'overcloud'
    jobs:
      - job1
      - job2
    reason: 'default reason'
    releases:
      - name: master
        lp: 'https://launchpad.net/bugs/2'
        reason: 'Some reason'
      - name: train
        bz: 'https://bugzilla.redhat.com/train1'
      - name: ussuri
        bz: 'https://bugzilla.redhat.com/ussuri1'
```

1.3.2 YAML values

BZ and LP

One of them is required, not both, but its crucial that people are able to trackdown the reason the test is being added in the skiplist. Here there are two levels of lp and bz: the first one, is the global lp and bz and the second is based on the release. This is required because its possible to have the test failing in two different releases but with different reasons. If the lp or bz is set on test level, you dont need to add it per release, it will use the test level. The value of both lp and bz are their respective URL

Deployment

This is right now TripleO only configuration, since it deploys two different openstack instances, undercloud and overcloud. Default to overcloud.

Installers

This is a list of installers per release where the test will be skipped. The options for installers are tripleo and osp, where tripleo is the upstream installer and osp is the downstream installer. The default (if this option is not set) is tripleo and osp.

Jobs

This is a list of jobs where the test should be skipped. This is not a required field, however, once set, the tool will return the test only if the job matches.

Reason

This is a description about why this test is being skipped. It exists in two levels, the first in the test level, that will be the default reason, and in release level, where reason can be different.

Releases

Releases contain a list of releases that the test will be skipped. Its very common that in a release the test is passing, but in another dont, so we can manage it here. Here, its also required a reason, and a lp or a bz

1.3.3 Examples

Below are some examples of valid yaml files that can be used:

No releases

Since theres no releases, this will be valid for all releases:

```yaml
known_failures:
  - test: 'tempest_skip.tests.test_validate'
    bz: 'https://bugzilla.redhat.com/1'
    lp: 'https://launchpad.net/bugs/1'
    deployment:
      - 'undercloud'
    reason: 'This test will be skipped in any release'
```

With releases

As release is set, the test will be skipped only on the matching releases

```
known_failures:
  - test: 'tempest_skip.tests.test_validate'
    bz: 'https://bugzilla.redhat.com/1'
    lp: 'https://launchpad.net/bugs/1'
    deployment:
      - 'undercloud'
    reason: 'This test will be skipped in any release'
    releases:
      - name: rocky
        reason: 'Test failing in rock because of network'
        lp: 'https://launchpad.net/bugs/1'
      - name: ussuri
        reason: 'Test is failing in ussuri because of storage bug'
        bz: 'https://bugzilla.redhat.com/1'
```

With jobs

If a list of jobs is set, the test will be skipped only in the matching jobs

```
known_failures:
  - test: 'tempest_skip.tests.test_validate'
    bz: 'https://bugzilla.redhat.com/1'
    lp: 'https://launchpad.net/bugs/1'
    deployment:
      - 'undercloud'
    reason: 'This test will be skipped in any release'
    jobs:
      - tempest-test-job-skip1
      - tempest-test-job-skip2
```

With jobs and releases

This test will be skipped only when it matches both, job and release

```
known_failures:
  - test: 'tempest_skip.tests.test_validate'
    bz: 'https://bugzilla.redhat.com/1'
    lp: 'https://launchpad.net/bugs/1'
    deployment:
      - 'undercloud'
    reason: 'This test will be skipped in all releases'
    releases:
      - name: rocky
        reason: 'Test failing in rock because of network'
        lp: 'https://launchpad.net/bugs/1'
```

(continues on next page)

(continued from previous page)

```
      - name: ussuri
        reason: 'Test is failing in ussuri because of storage bug'
        bz: 'https://bugzilla.redhat.com/1'
    jobs:
      - tempest-test-job-skip1
      - tempest-test-job-skip2
```

With releases and installers

```
known_failures:
  - test: 'tempest_skip.tests.test_validate'
    bz: 'https://bugzilla.redhat.com/1'
    lp: 'https://launchpad.net/bugs/1'
    deployment:
      - 'overcloud'
    reason: 'This test will be skipped in any release'
    releases:
      - name: train
        reason: 'Test failing in train because of network'
        installers:
          - 'tripleo'
        lp: 'https://launchpad.net/bugs/1'
      - name: wallaby
        reason: 'Test is failing in /osp-17 because of storage bug'
        installers:
          - 'osp'
        bz: 'https://bugzilla.redhat.com/1'
```

1.4 Validate the yaml file

1.4.1 Validation

You can use **`tempest-skip validate`** command to validate if the yaml file is in the expected format for the allowed list:

```
$ tempest-skip validate --allowed --file good_file.yaml
```

or for the skipped list:

    $ tempest-skip validate skipped file good_file.yaml

This will return nothing if the file is valid, or an error otherwise:

```
$ tempest-skip validate --file bad_file.yaml
required key not provided @ data['known_failures'][0]['releases'][2]['reason']
```

1.5 List skipped tests

1.5.1 List skipped tests

You can use **tempest-skip list-skipped** command to list tests in the yaml file with one positional and two optional parameters which is in the expected format:

```
1. ``--file`` is the positional parameter - lists all the tests in the file::

    $ tempest-skip list-skipped yaml --file tempest_skip.yml

2. ``--release``, ``--deployment``, ``--installer`` and  ``--job`` are the
   optional parameters - list all the tests within a specific release,
   deployment or a specific job::

   $ tempest-skip list-skipped --file tempest_skip.yml --release train
   $ tempest-skip list-skipped --file tempest_skip.yml --job job1
   $ tempest-skip list-skipped --file tempest_skip.yml --release train --job␣
↪job1
   $ tempest-skip list-skipped --file tempest_skip.yml --deployment undercloud
   $ tempest-skip list-skipped --file tempest_skip.yml --installer tripleo
```

This will return any tests that match the job, as well as tests that doesnt have any job configured. This is required when you configure your zuul jobs to always parse the `--job` option. In this scenario, if a job2 is parsed, and there is no test with job2, it would return zero tests to be skipped, which is not the intent. The test with no job defined, means, skip everywhere, if you define the job in the test yaml file, it means, skip all the tests that doesnt have a job defined, plus this test.

1.6 List allowed tests

1.6.1 List allowed tests

You can use **tempest-skip list-allowed** command to list the tests to be executed with two positional parameters which are in the expected format:

```
1. ``--file`` is the positional parameter - list all the tests in the file
2. ``--group`` or ``--job`` - filter the tests for a specific job, or a
      specific group.
```

1.6.2 Job filter

The job filter, is as the name indicate, it checks the yaml file for a job that matches (must be full match, not partial), and list the tests related to that specific job:

```
$ tempets-skip list-allowed --file tempest_allow.yml --job job1
```

1.6.3 Group filter

The group filter, which have precedence on the `--job`, will list the tests for a particular group. This is good when you have several jobs, that run a specific set of tests. In this case, you dont need to repeat the same set of tests for several different jobs:

```
$ tempest-skip list-allowed --file tempest_allow.yml --group default_group
```

1.6.4 Release filter

The release filter, which is default to master, filter based on group or job for an specific release.

1.6.5 Multiple Groups with same name

Multiple groups can have same name with different tests and releases. This behaviour allows us to classify the tests on the basis of releases. Here, we cannot define a release more than once in such groups i.e groups having same name should mandatorily have different releases:

```
- name: featureset062  # standalone-jobs
  tests:
    - 'octavia_tempest_plugin.tests.scenario.v2.test_healthmonitor'
    - 'octavia_tempest_plugin.tests.scenario.v2.test_listener'
  releases:
    - master
    - wallaby
- name: featureset062  # standalone-jobs
  tests:
    - 'octavia_tempest_plugin.tests.scenario.v2.test_healthmonitor'
    - 'octavia_tempest_plugin.tests.scenario.v2.test_l7policy'
  releases:
    - train
```

1.6.6 Wildcard filter for releases

If in the releases list in the yaml file, the release `all` is set, that means, it will not matter which release is passed to `tempest-skip` command, it will be included in the final list.

1.7 Adding tests to skiplist

1.7.1 Adding tests

Of course it is possible to directly edit the yaml file and add the test itself, but that may lead to failures, as for example duplicated entries, identation issues, having a namespace test instead of full path test name.

In order to avoid that, you can use the *addtest* command that will identify if the test is already on the skiplist, avoiding duplication. It will also generate the yaml file properly as well as avoid the use of test namespace. For example, if a user tries to add the tempest.scenario, it will skip all the tests under

tempest.scenario, which is not the desirable behavior. However, it is a lot of work to add each entry under tempest.scenario, since we must repeat all the reasons, bugzilla, releases, etc.

The addtest command solves all these problems for you. First of all, when you try to add a test passing a test namespace, addtest will give you a list of tests under that particular namespace, where you can choose from the list which ones you would like to add. Select the ones you want and you are done!

1.7.2 Examples

The command below add the test tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_connectivity_betw

```
$ tempest-skip addtest \
               --file roles/validate-tempest/vars/tempest_skip.yml \
               --release master \
               --test tempest.scenario.test_network_basic_ops.
↪TestNetworkBasicOps.test_connectivity_between_vms_on_different_networks \
               --reason 'Failing on network' \
               --lp https://launchpad.net/bug/12345
```

In this example, we are adding the full path test, and so, the command will not prompt you a list of tests that you want to choose. If for example, only the namespace be parsed, you will be prompted with a list of tests under that namespace:

```
$ tempest-skip addtest \
               --file roles/validate-tempest/vars/tempest_skip.yml \
               --release master \
               --test tempest.scenario.test_network_basic_ops \
               --reason 'Failing on network' \
               --lp https://launchpad.net/bug/12345
```

And this is the output:

```
[?] These are the tests available on the namespace, choose which ones you␣
↪want to add. Press space to select:
 > o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_
↪connectivity_between_vms_on_different_networks
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_hotplug_
↪nic
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_mtu_
↪sized_frames
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_network_
↪basic_ops
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_port_
↪security_macspoofing_port
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_
↪preserve_preexisting_port
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_router_
↪rescheduling
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_subnet_
↪details
   o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_update_
↪instance_port_admin_state
```

(continues on next page)

(continued from previous page)

```
  o tempest.scenario.test_network_basic_ops.TestNetworkBasicOps.test_update_
↪router_admin_state
```

You can use the arrow keys on your keyboard to navigate through the list, and space to select. Once you are done, just press enter, the command will go through each test, check if the test exists or not. If it exists, it will also check the release exists, and properly add the test.

Once you are done, you can validate if the yaml file was generated properly with the command:

```
$ tempest-skip validate --file roles/validate-tempest/vars/tempest_skip.yml
```

There are some arguments you can pass to the addtest, the required ones are:

- file
- lp or bz
- reason

All the other arguments have default values, for example, *release* default value is master and *deployment* default value is overcloud

For more information, use:

```
$ tempest-skip addtest --help
```

- search