OpenStack-Ansible Documentation: os_tempest role

Release 18.1.0.dev464

OpenStack-Ansible Contributors

May 12, 2025

CONTENTS

This is the Ansible role to deploy OpenStack Tempest.

tags openstack, cloud, ansible, os_tempest

category *nix

CHAPTER

ONE

CONTENT:

1.1 Overview

1.2 Team and repository tags

1.3 OpenStack-Ansible os_tempest role

Ansible role to install OpenStack Tempest.

Documentation for the project can be found at: https://docs.openstack.org/openstack-ansible-os_tempest/latest

Release notes for the project can be found at: https://docs.openstack.org/releasenotes/openstack-ansible-os_tempest

The project source code repository is located at: https://opendev.org/openstack/openstack-ansible-os_tempest

The project home is at: https://launchpad.net/openstack-ansible

The project bug tracker is located at: https://bugs.launchpad.net/openstack-ansible

Tempest is a testing framework consisting of a set of integration tests to test any deployed OpenStack cloud.

1.3.1 os_tempest mission

To provide a re-usable ansible role which installs, configures and runs Tempest.

1.3.2 Why?

The reason we have come up with this idea is because every OpenStack project uses playbooks and shell scripts to install, run and configure Tempest which are only slightly different but their purpose is the same.

When every project uses its own way to use Tempest, its really harder to cooperate (cross projects) together to solve any issues which may occur.

Thats where the re-usability steps in. By using the same role we can faster react to any issues which occurred in one project and may have an effect on another one.

1.3.3 Advantages

- maintenance of only one set of playbooks and scripts
- heads-up for issues related to particular tests
- bigger focus on development and maintenance of the one set of playbooks and scripts
- decreasing of time consumption needed to install, configure and run Tempest for new OpenStack projects - no need to write their CI Tempest procedures from scratch

1.4 User Guide

1.4.1 Installation

This page describes how to install `os_tempest` role.

To clone or view the source code of `os_tempest`, visit the role repository for os_tempest.

Install dependencies via `ansible-galaxy`:

```
$ mkdir ~/.ansible/roles -p
$ git clone https://opendev.org/openstack/openstack-ansible-os_tempest ~/.
↪ansible/roles/os_tempest
$ ansible-galaxy install -r ~/.ansible/roles/os_tempest/requirements.yml --
↪roles-path=~/.ansible/roles/
```

Then you need to export a couple of variables, *ANSIBLE_ROLES_PATH* which points to the directory where `os_tempest` was cloned and *ANSIBLE_ACTION_PLUGINS* which points to the location of `config_template` plugin. In this case its:

```
$ export ANSIBLE_ROLES_PATH=$HOME/.ansible/roles
$ export ANSIBLE_ACTION_PLUGINS=~/.ansible/roles/config_template/action
```

Then create a `playbook.yaml`, you can find an example one here. Then dont forget to set the name of the cloud youre going to run the role against, see this page.

1.4.2 Usage

Execute by ansible-playbook

First you need to install `os_tempest` role. For more information about the installation process refer to the Installation page.

After the role is installed enter the *openstack-ansible-os_tempest* directory.

First thing which needs to be done in order to execute `os_tempest` role is setting a cloud name. For information on how to do that, please, have a look at Set the name of the cloud page.

An example `playbook.yml` can be seen below in Example playbook section.

After the required variables in the `playbook.yml` file are set you can execute the role as follows:

```
$ ansible-playbook playbook.yaml
```

Example playbook

```
---
- hosts: localhost
  become: true
  vars:
    tempest_cloud_name: mycloud
    tempest_run: true
  roles:
    - os_tempest
```

Dependencies

This role requires the following packages to be installed on the target host:

- pip >= 7.1
- python-virtualenv

1.4.3 Default variables

```
## Verbosity Options
debug: False

stestr_executable: "{{ _stestr_executable | default('stestr') }}"

# Install openstack tempest
# set the tempest_install_method to source or distro
# on choosing source it will install from git or venv
# on choosing distro it will install based on distribution
tempest_install_method: "source"
tempest_venv_python_executable: "{{ openstack_venv_python_executable |␣
↪default(_tempest_venv_python_executable) | default('python3') }}"
```

(continues on next page)

(continued from previous page)

```
# Set the package install state for distribution and pip packages
# # Options are 'present' and 'latest'
tempest_package_state: "latest"

# Set the host which will execute the shade modules
# for the service setup. The host must already have
# clouds.yaml properly configured.
tempest_service_setup_host: "{{ openstack_service_setup_host | default(
↪'localhost') }}"
tempest_service_setup_host_python_interpreter: "{{ openstack_service_setup_
↪host_python_interpreter | default((tempest_service_setup_host == 'localhost
↪') | ternary(ansible_playbook_python, ansible_facts['python']['executable
↪'])) }}"

# Toggle whether tempest actually executes
tempest_run: no
# Toggle whether tempest cleanup executes prior and after regular tempest run
tempest_cleanup: no
# if tempest_cleanup_dry_run is set to true, tempest cleanup will log all␣
↪found
# leftover resources to a dry_run.json file, none resources will be deleted
#tempest_cleanup_dry_run: no

# Credential Provider Mechanisms
# If tempest_test_accounts is not an empty string, then use_dynamic_
↪credentials will be turned off
tempest_test_accounts: ""
tempest_test_accounts_file_path: "{{ tempest_workspace }}/etc/tempest_
↪accounts.yaml"
# tempest_create_isolated_networks has an effect only when use_dynamic_
↪credentials is enabled
tempest_create_isolated_networks: true
# tempest_fixed_network_name does not have an effect when dynamic credentials␣
↪are used together with tempest_create_isolated_networks
tempest_fixed_network_name: public

# Tempest Resources
# Toggle whether default resources are implemented
tempest_default_role_resources: yes
tempest_public_net_create: true
tempest_private_net_create: false
tempest_router_create: false
tempest_images_create: true
tempest_flavors_create: true
tempest_projects_create: "{{ tempest_public_net_create or tempest_private_net_
↪create or tempest_router_create }}"

# Define 0 (serial) or more to use a non default concurrency
```

(continues on next page)

(continued from previous page)

```
#tempest_run_concurrency:

# Define the worker file to be used by tempest
# This worker file is parsed to stestr to manually schedule tempest tests
# tempest_test_worker_file_path:

# We comment `tempest_git_repo` so that we do not attempt to build the wheel␣
↪from this repo/branch.
# Instead, we want tempest to get built from the stable release defined in␣
↪global requirements.
#tempest_git_repo: https://opendev.org/openstack/tempest
tempest_git_install_branch: master
tempest_upper_constraints_url: "{{ requirements_git_url | default('https://
↪releases.openstack.org/constraints/upper/' ~ requirements_git_install_
↪branch | default('master')) }}"
tempest_git_constraints:
  - "--constraint {{ tempest_upper_constraints_url }}"

tempest_pip_install_args: "{{ pip_install_options | default('') }}"

# Name of the virtual env to deploy into
tempest_venv_tag: "{{ venv_tag | default('untagged') }}"
tempest_venv_bin: "/openstack/venvs/tempest-{{ tempest_venv_tag }}/bin"

# The location where the tempest logs will be placed
tempest_log_dir: "/var/log/tempest"

## Tempest Plugins
# Extra plugins can be defined in ``tempest_extra_plugins``.
# tempest_extra_plugins:
#   - name: custom-tempest-plugin
#     repo: https://opendev.org/openstack/custom-tempest-plugin
#     branch: master
#     install: true
tempest_extra_plugins: []
tempest_plugins: "{{ (_tempest_plugins + tempest_extra_plugins) | selectattr(
↪'install', 'equalto', true) }}"

tempest_plugin_barbican_git_repo: https://opendev.org/openstack/barbican-
↪tempest-plugin
tempest_plugin_barbican_git_install_branch: master

tempest_plugin_cinder_git_repo: https://opendev.org/openstack/cinder-tempest-
↪plugin
tempest_plugin_cinder_git_install_branch: master

tempest_plugin_cloudkitty_git_repo: https://opendev.org/openstack/cloudkitty-
↪tempest-plugin
tempest_plugin_cloudkitty_git_install_branch: master
```

(continues on next page)

(continued from previous page)

```
tempest_plugin_designate_git_repo: https://opendev.org/openstack/designate-
↪tempest-plugin
tempest_plugin_designate_git_install_branch: master

tempest_plugin_glance_git_repo: https://opendev.org/openstack/glance-tempest-
↪plugin
tempest_plugin_glance_git_install_branch: master

tempest_plugin_heat_git_repo: https://opendev.org/openstack/heat-tempest-
↪plugin
tempest_plugin_heat_git_install_branch: master

tempest_plugin_ironic_git_repo: https://opendev.org/openstack/ironic-tempest-
↪plugin
tempest_plugin_ironic_git_install_branch: master

tempest_plugin_keystone_git_repo: https://opendev.org/openstack/keystone-
↪tempest-plugin
tempest_plugin_keystone_git_install_branch: master

tempest_plugin_magnum_git_repo: https://opendev.org/openstack/magnum-tempest-
↪plugin
tempest_plugin_magnum_git_install_branch: master

tempest_plugin_manila_git_repo: https://opendev.org/openstack/manila-tempest-
↪plugin
tempest_plugin_manila_git_install_branch: master

tempest_plugin_murano_git_repo: https://opendev.org/openstack/murano-tempest-
↪plugin
tempest_plugin_murano_git_install_branch: master

tempest_plugin_neutron_git_repo: https://opendev.org/openstack/neutron-
↪tempest-plugin
tempest_plugin_neutron_git_install_branch: master

tempest_plugin_novajoin_git_repo: https://opendev.org/x/novajoin-tempest-
↪plugin
tempest_plugin_novajoin_git_install_branch: master

tempest_plugin_octavia_git_repo: https://opendev.org/openstack/octavia-
↪tempest-plugin
tempest_plugin_octavia_git_install_branch: master

tempest_plugin_senlin_git_repo: https://opendev.org/openstack/senlin-tempest-
↪plugin
tempest_plugin_senlin_git_install_branch: master
```

(continues on next page)

(continued from previous page)

```
tempest_plugin_sahara_git_repo: https://opendev.org/openstack/sahara-tests
tempest_plugin_sahara_git_install_branch: master

tempest_plugin_telemetry_git_repo: https://opendev.org/openstack/telemetry-
↪tempest-plugin
tempest_plugin_telemetry_git_install_branch: master

tempest_plugin_trove_git_repo: https://opendev.org/openstack/trove-tempest-
↪plugin
tempest_plugin_trove_git_install_branch: master

tempest_plugin_zaqar_git_repo: https://opendev.org/openstack/zaqar-tempest-
↪plugin
tempest_plugin_zaqar_git_install_branch: master

tempest_plugin_zun_git_repo: https://opendev.org/openstack/zun-tempest-plugin
tempest_plugin_zun_git_install_branch: master

# tempest_workspace where tempest can be runned
tempest_workspace: "{{ ansible_facts['env']['HOME'] }}/workspace"

# The location where the test include/exclude lists will be placed
tempest_test_includelist_file_path: "{{ tempest_workspace }}/etc/tempest_
↪includelist.txt"
tempest_test_excludelist_file_path: "{{ tempest_workspace }}/etc/tempest_
↪excludelist.txt"

# Tests to execute:
# This sets up a list of tests to execute based on what's deployed in the␣
↪environment.
# The list gets added to the includelist which tempest executes.
# Defaults to tempest_test_whitelist for backwards compatibility and␣
↪migration purposes.
tempest_test_includelist: "{{ tempest_test_whitelist | default(tempest_test_
↪default_includelist) }}"
tempest_test_default_includelist:
  - "smoke"
  - "{{ (tempest_service_available_ceilometer | bool) | ternary('tempest.api.
↪telemetry', '') }}"
  - "{{ (tempest_service_available_heat | bool) | ternary('tempest.api.
↪orchestration.stacks.test_non_empty_stack', '') }}"

# TODO(arxcruz) Right now, we have the tempest_test_whitelist parsed as -e
# argument in ansible in several of our jobs, so it is right now hard to get
# rid of all of it in all the places. Once we implement the include list in␣
↪the
# openstack-tempest-skiplist side, we can remove it
# Extra test to be executed
tempest_test_extra_test: []
```

(continues on next page)

(continued from previous page)

```
# Tests being skipped by os_tempest
# Example:
# tempest_test_excludelist:
#   - test: tempest.scenario.test.minimum_basic
#     reason: This test is failing
#     lp: 'https://bugs.launchpad.net/openstack-ansible/+bug/123456'
#     bz: 'https://bugzilla.redhat.com/show_bug.cgi?id=123456'
# OR
# tempest_test_excludelist:
#   - 'tempest.scenario.test.minimum_basic'
# Defaults to tempest_test_blacklist for backwards compatibility and␣
↪migration purposes.
tempest_test_excludelist: "{{ tempest_test_blacklist | default([]) }}"

# Private network configuration
# Currently supports 2 types
# vlan - will need to make sure your seg id and subnet cidr are correct
# vxlan - default, can change subnet cidr and seg id

tempest_private_net_name: "private"
tempest_private_subnet_name: "private-subnet"
tempest_private_subnet_cidr: "192.168.74.0/28"
tempest_private_net_provider_type: "vxlan"
tempest_private_net_seg_id:
# If you choose vlan as private network provider type, you must set a physical
# name for it
# tempest_private_net_physical_name: "private"

# Public network configuration
# Currently supports 2 types
# Flat - default
# Vlan - make sure you override seg id, cidr, provider and physical
tempest_public_net_name: "public"
tempest_public_subnet_name: "public-subnet"
tempest_public_subnet_cidr: "10.1.13.0/24"
# Neutron default gateway to first ip of subnet, usually .1
# tempest_public_subnet_gateway_ip:
tempest_public_net_provider_type: "flat"
# TODO(chkumar246):
# The use of _type is to provide backwards compatibility for
# overrides in S and can be removed in T.
tempest_public_net_physical_name: "{{ tempest_public_net_physical_type |␣
↪default('flat') }}"
tempest_public_net_seg_id: ""
tempest_public_router_external: "True"
# Example allocation range:
# tempest_public_subnet_allocation_pools: "10.1.13.150-10.1.13.200"
tempest_public_subnet_allocation_pools: ""
```

(continues on next page)

(continued from previous page)

```
tempest_compute_image_ssh_user: cirros
tempest_compute_run_ssh: True
tempest_network_ping_gateway: False

tempest_dashboard_url: "https://{{ external_lb_vip_address | default('127.0.0.
↪1') }}/"

# var for setting tempest_service_available_{sevice_name} vars
# Example:
# tempest_services:
#   - cinder
#   - aodh
# It will set tempest_service_available_aodh and
# tempest_service_available_cinder to true.
tempest_services: []

tempest_service_available_aodh: "{{ groups['aodh_all'] is defined and groups[
↪'aodh_all'] | length > 0 }}"
tempest_service_available_barbican: "{{ groups['barbican_all'] is defined and␣
↪groups['barbican_all'] | length > 0 }}"
tempest_service_available_ceilometer: "{{ groups['ceilometer_all'] is defined␣
↪and groups['ceilometer_all'] | length > 0 }}"
tempest_service_available_cinder: "{{ groups['cinder_all'] is defined and␣
↪groups['cinder_all'] | length > 0 }}"
tempest_service_available_cloudkitty: "{{ groups['cloudkitty_all'] is defined␣
↪and groups['cloudkitty_all'] | length > 0 }}"
tempest_service_available_designate: "{{ groups['designate_all'] is defined␣
↪and groups['designate_all'] | length > 0 }}"
tempest_service_available_glance: "{{ groups['glance_all'] is defined and␣
↪groups['glance_all'] | length > 0 }}"
tempest_service_available_gnocchi: "{{ groups['gnocchi_all'] is defined and␣
↪groups['gnocchi_all'] | length > 0 }}"
tempest_service_available_heat: "{{ groups['heat_all'] is defined and groups[
↪'heat_all'] | length > 0 }}"
tempest_service_available_horizon: "{{ groups['horizon_all'] is defined and␣
↪groups['horizon_all'] | length > 0 }}"
tempest_service_available_ironic: "{{ groups['ironic_all'] is defined and␣
↪groups['ironic_all'] | length > 0 }}"
tempest_service_available_magnum: "{{ groups['magnum_all'] is defined and␣
↪groups['magnum_all'] | length > 0 }}"
tempest_service_available_manila: "{{ groups['manila_all'] is defined and␣
↪groups['manila_all'] | length > 0 }}"
tempest_service_available_mistral: "{{ groups['mistral_all'] is defined and␣
↪groups['mistral_all'] | length > 0 }}"
tempest_service_available_murano: "{{ groups['murano_all'] is defined and␣
↪groups['murano_all'] | length > 0 }}"
tempest_service_available_neutron: "{{ groups['neutron_all'] is defined and␣
↪groups['neutron_all'] | length > 0 }}"
```

(continues on next page)

(continued from previous page)

```
tempest_service_available_neutron_bgpvpn: "{{ (groups['neutron_all'] is␣
↪defined) and (groups['neutron_all'] | length > 0) and ('bgpvpn' in neutron_
↪plugin_base | default([])) }}"
tempest_service_available_neutron_vpnaas: "{{ (groups['neutron_all'] is␣
↪defined) and (groups['neutron_all'] | length > 0) and ('vpnaas' in neutron_
↪plugin_base | default([])) }}"
tempest_service_available_nova: "{{ groups['nova_all'] is defined and groups[
↪'nova_all'] | length > 0 }}"
tempest_service_available_novajoin: False
tempest_service_available_octavia: "{{ groups['octavia_all'] is defined and␣
↪groups['octavia_all'] | length > 0 }}"
tempest_service_available_panko: "{{ groups['panko_all'] is defined and␣
↪groups['panko_all'] | length > 0 }}"
tempest_service_available_sahara: "{{ groups['sahara_all'] is defined and␣
↪groups['sahara_all'] | length > 0 }}"
tempest_service_available_senlin: "{{ groups['senlin_all'] is defined and␣
↪groups['senlin_all'] | length > 0 }}"
tempest_service_available_swift: "{{ (groups['swift_all'] is defined and␣
↪groups['swift_all'] | length > 0) or (groups['ceph-rgw'] is defined and␣
↪groups['ceph-rgw'] | length > 0) or (ceph_rgws is defined and ceph_rgws |␣
↪length > 0) }}"
tempest_service_available_trove: "{{ groups['trove_all'] is defined and␣
↪groups['trove_all'] | length > 0 }}"
tempest_service_available_whitebox: False
tempest_service_available_zaqar: "{{ groups['zaqar_all'] is defined and␣
↪groups['zaqar_all'] | length > 0 }}"
tempest_service_available_zun: "{{ groups['zun_all'] is defined and groups[
↪'zun_all'] | length > 0 }}"

# Var for setting ssl verification
tempest_keystone_interface_insecure: "{{ (keystone_service_internaluri_
↪insecure | default(false)) | bool }}"

tempest_pip_packages:
  - "{{ (tempest_git_repo is defined) | ternary('git+' ~ (tempest_git_repo |␣
↪default('https://opendev.org/openstack/tempest.git')) ~ '@' ~ tempest_git_
↪install_branch ~ '#egg=tempest', 'tempest') }}"
  - cmd2
  - ddt
  - junitxml
  - lxml
  - nose
  - python-cinderclient
  - python-glanceclient
  - python-heatclient
  - python-keystoneclient
  - python-manilaclient
  - python-memcached
  - python-neutronclient
```

(continues on next page)

(continued from previous page)

```
  - python-novaclient
  - python-openstackclient
  - python-senlinclient
  - python-saharaclient
  - python-subunit
  - python-swiftclient
  - python-troveclient
  - testscenarios
  - os-testr

# The list of images for tempest to download for the current architecture, as␣
↪defined
# in this role vars/main.yml file
# To override this list, use a list of the form
# tempest_images:
#   - url: ...          where to download from (required)
#     checksum: ...     checksum to validate downloaded file, format:
↪<algorithm>:<checksum> (optional)
#     format: ...       format to use when uploading to glance (required)
#     name: ...         name to use when uploading to glance (optional)
#     properties:       a dict of custom properties to attach to the image in␣
↪glance (optional)
#       <property>: <value>
tempest_images: "{{ tempest_images_map[ansible_facts['architecture']] }}"

# The location where images are downloaded to
tempest_image_dir: "{{ lookup('env', 'HOME') }}/tempest-images"

tempest_flavors:
  - name: tempest1
    id: 201
    ram: 256
    disk: 1
    vcpus: 1
  - name: tempest2
    id: 202
    ram: 512
    disk: 1
    vcpus: 1

# The projects for tempest to use
tempest_projects:
  - "tempest"

## Tunable overrides
tempest_tempest_conf_overrides: {}

## The name of cloud from clouds.yaml
tempest_cloud_name: "default"
```

(continues on next page)

(continued from previous page)

```
## The name of domain from clouds.yaml
tempest_domain_name: "Default"

## The name of interface from clouds.yaml
tempest_interface_name: "internal"

# The default endpoint type to use by tempest
tempest_endpoint_type: "internal"

# python-tempestconf variables
# The tempest_use_tempestconf by default is set to false, set to true if you
# want to generate the tempest.conf file with this tool, instead of
# tempest.conf from the template
tempest_use_tempestconf: false
tempest_tempestconf_venv_tag: "{{ venv_tag | default('untagged') }}"
tempest_tempestconf_venv_bin: "/openstack/venvs/tempestconf-{{ tempest_
↪tempestconf_venv_tag }}/bin"

# We comment out `tempest_tempestconf_git_repo` so that the repo_build role␣
↪does not attempt to
# build the wheel from this repo/branch. Instead, we want python-tempestconf␣
↪to get built
# from the stable release defined in global requirements.
# tempest_tempestconf_git_repo: https://opendev.org/openinfra/python-
↪tempestconf
tempest_tempestconf_git_install_branch: master
tempest_tempestconf_git_constraints:
  - "--constraint {{ tempest_upper_constraints_url }}"
tempest_tempestconf_pip_packages:
  - "{{ (tempest_tempestconf_git_repo is defined) | ternary('git+' ~ (tempest_
↪tempestconf_git_repo | default('https://opendev.org/openinfra/python-
↪tempestconf')) ~ '@' ~ tempest_tempestconf_git_install_branch ~ '
↪#egg=python_tempestconf', 'python_tempestconf') }}"
tempest_tempestconf_profile:
  debug: true
  create: true
  os-cloud: "{{ tempest_cloud_name }}"
  out: "{{ tempest_workspace }}/etc/tempest.conf"
  network-id: "{{ tempest_neutron_public_network_id }}"
  overrides: "{{ tempest_tempestconf_profile_overrides }}"
tempest_tempestconf_profile_extras: {}
tempest_tempestconf_profile_overrides: "{{ tempest_tempest_conf_overrides }}"

# Stackviz tarball url
stackviz_pip_install_args: "{{ pip_install_options | default('--isolated') }}"
stackviz_tarball: "https://tarballs.opendev.org/openstack/stackviz/dist/
↪stackviz-latest.tar.gz"
stackviz_venv_tag: "{{ venv_tag | default('untagged') }}"
```

(continues on next page)

(continued from previous page)

```
stackviz_venv_bin: "/openstack/venvs/stackviz-{{ stackviz_venv_tag }}/bin"
tempest_run_stackviz: true
```

1.4.4 os_tempest configuration

This page shows all of the variables which can be set in order to control the behaviour of `os_tempest` role and provides examples on how to do so.

For a list of all variables with a default value set, please, refer to the this page.

Set the name of the cloud

`os-tempest` uses named cloud credentials so it requires the name of the cloud the role will be executed against. The name is provided to `os-tempest` via the `tempest_cloud_name` variable. In order to use named clouds a `clouds.yaml` file needs to be present on the **target host**. `clouds.yaml` file needs to be stored at one of the supported locations, see here For more information about named clouds, please, follow to the os-client-config official documentation

> **Warning:** `clouds.yaml` file has to be present on the target host - the host `os_tempest` is gonna be executed against.

Resource creation

Tempest requires some openstack resources(like flavors and images) in order to function properly. It is possible to choose which resources should be created or to skip resource creation at all. Below example shows how to use already existing public network and images.

```
tempest_default_role_resources: true
tempest_public_net_create: false
tempest_neutron_public_network_id: <network_id>
tempest_images_create: false
tempest_glance_image_id_1: <image_id>
tempest_glance_image_id_2: <image_id>
```

python-tempestconf

python-tempestconf is a tool which generates a `tempest.conf` file necessary for Tempest execution. For more information about the tool, please, follow its official documentation.

If you want `os_tempest` to execute `python-tempestconf`, prior to Tempest execution in order to generate `tempest.conf` file, set *tempest_use_tempestconf* variable to true:

```
tempest_use_tempestconf: true
```

More information about `python-tempestconf` arguments can be found here.

The best way how to pass any arguments to `python-tempestconf` is using its profile feature.

`os_tempest` provides *tempest_tempestconf_profile* variable for setting desired python-tempestconfs arguments. For example, if you wanted to define **debug** to *true*, **os-cloud** to *demo* and override output of `python-tempestconf` to `/my/location/tempest.conf`, it would be done by:

```
tempest_tempestconf_profile:
  debug: true
  os-cloud: demo
  out: /my/location/tempest.conf
```

- search