# Tobiko

***Release 0.2.1.dev667***

**Sep 25, 2020**

# Contents

CHAPTER 1

# Tobiko User Guide

## 1.1 Tobiko Quick Start Guide

### 1.1.1 Document Overview

This document describes how to install execute Tobiko test cases using Tox.

**See also**

To install Tobiko inside a virutalenv please read *Tobiko Installation Guide*.

To configure Tobiko please read *Tobiko Configuration Guide*.

To run Tobiko scenario test cases please look at *Tobiko Test Cases Execution Guide*.

### 1.1.2 Install Dependencies

#### Install Basic Python Packages

Make sure Git and Python 3 are installed on your system.

For instance on RedHat Linux / Fedora:

```
sudo yum install -y git python3 which
```

Check your Python 3 version is greater than 3.6:

```
python3 --version
```

Make sure pip is installed and up-to date:

```
curl https://bootstrap.pypa.io/get-pip.py | sudo python3
```

Check installed Pip version:

```
python3 -m pip --version
```

Make sure basic Python packages are installed and up-to-date:

```
sudo python3 -m pip install --upgrade setuptools wheel virtualenv tox six
```

Check installed Tox version:

```
tox --version
```

#### Clone the Tobiko repository

Clone the Tobiko repository using Git:

```
git clone https://opendev.org/x/tobiko.git
cd tobiko
```

#### Install Missing Binary Packages

Install required binary packages:

```
tools/install-bindeps.sh
```

### 1.1.3 Configure Logging Options

Test cases load most of the configuration parameters from an INI configuration file, typically found at one of the following locations:

- ./tobiko.conf (Tobiko source files directory)
- ~/.tobiko/tobiko.conf
- /etc/tobiko/tobiko.conf

Create it in the Tobiko source directory with the following (or as your preferences). Example:

```
[DEFAULT]
debug = True
log_file = tobiko.log
```

The file 'tobiko.log' is the default file where test cases and the Python framework are going to write their logging messages. By setting debug as 'true' you ensure that messages with the lowest logging level are written there (DEBUG level). The log_file location specified above is relative to the tobiko.conf file location. In this example it is the Tobiko source files' directory itself.

### 1.1.4 Configure Tobiko Credentials

In order to run the OpenStack test cases you'll need to set up Keystone credentials. You can do it in one of following ways:

- *Set Tobiko Credentials from clouds.yaml file*
- *Set Tobiko Credentials Via Environment Variables*
- *Set Tobiko Credentials Via tobiko-conf File*

### Set Tobiko Credentials from clouds.yaml file

Make sure that in any one of below locations there is a valid OpenStack clouds file containing valid Keystone credentials:

- Tobiko source files directory
- ~/.config/openstack
- /etc/openstack

Finally, you will need to specify which credentials Tobiko should pick up via 'OS_CLOUD' environment variable or by specifying the cloud_name in tobiko.conf file (section 'keystone', option 'cloud_name').

### Specify 'OS_CLOUD' environment variable

Ensure *OS_CLOUD* environment variable is defined before executing Tobiko test cases:

```
export OS_CLOUD=<cloud_name>
```

Please choose a valid cloud_name from your clouds.yaml file.

### Specify cloud_name in tobiko.conf file

Create file *tobiko.conf* in Tobiko sources folder adding a section like below:

```
[keystone]
cloud_name = <cloud_name>
```

Please choose a valid cloud_name from your clouds.yaml file.

### Set Tobiko Credentials Via Environment Variables

**See also**

For more details about supported environment variables please read *Authentication Environment Variables* section.

You can use an existing shell RC file that is valid for Python OpenStack client

```
source openstackrc
```

An example of 'openstackrc' file could looks like below:

```
export OS_IDENTITY_API_VERSION=3
export OS_AUTH_URL=https://my_cloud:13000/v3
export OS_USERNAME=admin
export OS_PASSWORD=secret
```

(continues on next page)

(continued from previous page)

```
export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
```

#### Set Tobiko Credentials Via tobiko.conf File

**See also**

For more details about supported configuration options please read *Autentication Configuration* section.

Create a file at *~/.tobiko/tobiko.conf* and add a section as in the example below (Or add to your existing file):

```
[keystone]
api_version = 3
auth_url = http://my_cloud:13000/v3
username = admin
password = secret
project_name = admin
user_domain_name = Default
project_domain_name = Default
```

#### Setup Required Resources

A public Neutron network is required To be able to execute Tobiko scenario test cases by creating a floating IP port on it.

To execute commands from a virtualenv created by Tox you can type as below:

```
tox -e venv -- <your-commands>
```

You need to make sure ref:*authentication-environment-variables* are properly set so you can list available public netoworks:

```
tox -e venv -- openstack network list
```

If there is any valid public network, you need to create one before running Tobiko OpenStack test cases. Please refer to the Neutron documentation for additional information.

If there is a valid public network for creating floating-IP ports on, Tobiko tests cases will automatically use it. To explicitly select a network, please add a reference to the network in *tobiko.conf* file:

```
[neutron]
floating_network = public
```

### 1.1.5 Running Test Cases

#### Running Scenario Test Cases

Scenario test cases are used to create workloads that simulate real-world use of OpenStack. They create networks, virtual machines, ports, routers, etc. They also test validate that these workloads functioning.

Running Tobiko scenario test cases using Tox (may take some time to complete (minutes)):

```
tox -e scenario
```

To list Heat stacks and Glance images created by test cases:

```
tox -e venv -- openstack image list
tox -e venv -- openstack stack list
```

Scenario test cases are also used to check that previously created resources are still up and working as expected. To ensure test cases will not create those resources again we can set *TOBIKO_PREVENT_CREATE* environment variable before re-running test cases:

```
TOBIKO_PREVENT_CREATE=yes tox -e scenario
```

### Cleanning Up Tobiko Workloads

Once Tobiko test cases have been executed, we may want to clean up all workloads remaining on the cloud so that we restore it to its original state.

### Cleanning Up Heat Stacks

Because Tobiko is using Heat stacks for orchestrating the creation of most of the resources, deleting all stacks created with Tobiko will clean up almost all resources:

```
tox -e venv -- bash -c 'openstack stack list -f value -c ID | xargs openstack stack␣
↪delete'
```

### Cleanning Up Glance Images

Because Heat doen't support creation of Glance images, Tobiko implemented some specific fixtures to download images from the Web and upload them to the Glance service:

```
tox -e venv -- bash -c 'openstack image list -f value -c ID | xargs openstack image␣
↪delete'
```

### Running Disruptive Test Cases

Disruptive test cases are used for testing that after inducing some critical disruption to the operation of the cloud, the services return working as expected after a while. To execute them you can type:

```
tox -e faults
```

The faults induced by these test cases could be cloud nodes reboot, OpenStack services restart, virtual machines migrations, etc.

Please note that while scenario test cases are being executed in parallel (to speed up test case execution), disruptive test case are only executed sequentially. This is because the operations executed by such cases could break some functionality for a short time and alter the regular state of the system which may be assumed by other test cases to be executed.

### Running the Tobiko Workflow

Scenario and disruptive test cases, being executed in a specific sequence could be used to uncover more issues with the cloud than disruptive test cases alone.

- First ensure there are workloads properly running by running scenario test cases:

```
tox -e scenario
```

**Note**

As second step we may, instead, update or upgrade OpenStack nodes.

- Next we could execute disruptive test cases to "stress" the cloud:

```
tox -e faults
```

- Finally we might re-run scenario test cases to check thateverything is still running as expected:

```
TOBIKO_PREVENT_CREATE=yes tox -e scenario
```

### Test Cases Report Files

After executing test cases we can view the results in greater detail via a small set of files:

- **test_results.html**: A user-browseable HTML view of test case results
- **test_results.log**: a log file with logging traces collected from every individual test case
- **test_results.subunit**: the original subunit binary file generated by test runner
- **test_results.xml**: an XML Junit file to be used, for example, to show test cases result by Jenkins CI server

The names of the above files can be changed from the default value (*test_results*) to a custom one by setting the *TOX_REPORT_NAME* environment variable.

**Legend**

*{toxinidir}* stand for the Tobiko source files directory.

*{envname}* is the name of the Tox enviroment to be executed (IE scenario, faults, etc.)

The above files are saved into a folder that can be specified with *TOX_REPORT_DIR* environment variable.

By default the full path of the report directory is made from the below:

```
{toxinidir}/report/{envname}
```

## 1.2 Tobiko Installation Guide

### 1.2.1 Document Overview

This document describes how to install Tobiko inside a Python Virtualenv.

**See also**

For a quick and simpler start you can jump to the *Tobiko Quick Start Guide*.

To configure Tobiko please read *Tobiko Configuration Guide*.

To run Tobiko scenario test cases please look at *Tobiko Test Cases Execution Guide*.

### 1.2.2 Install Tobiko Using Virtualenv

Make sure Gcc, Git and base Python packages are installed on your system.

For instance on RHEL Linux 7.6 or CentOS 7 you could type:

```
sudo yum install -y gcc git python python-devel wget
```

For instance on RHEL Linux 8 or CentOS 8 you could type:

```
sudo dnf install -y gcc git python3 python3-devel wget
sudo alternatives --set python /usr/bin/python3
```

Make sure pip is installed and up-to date:

```
wget https://bootstrap.pypa.io/get-pip.py
sudo python get-pip.py
PIP=$(which pip)
```

Make sure setuptools, virtualenv and wheel are installed and up to date:

```
sudo $PIP install --upgrade setuptools virtualenv wheel
```

Get Tobiko source code using Git and enter into Tobiko source folder:

```
git clone https://opendev.org/x/tobiko.git
cd tobiko
```

To install Tobiko and its dependencies is safer to create a clean Virtualenv where to install it. Create a Virtualenv and activate it:

```
virtualenv .tobiko-env
source .tobiko-env/bin/activate
```

Install Tobiko and its requirements:

```
pip install \
    -c https://opendev.org/openstack/requirements/raw/branch/master/upper-constraints.
↪txt \
    .
```

### 1.2.3 What's Next

To know how to configure Tobiko please read *Tobiko Configuration Guide*.

## 1.3 Tobiko Configuration Guide

### 1.3.1 Document Overview

This document describes how to configure Tobiko.

**See also**

For a quick and simpler start you can jump to the *Tobiko Quick Start Guide*.

To install Tobiko inside a virutalenv please read *Tobiko Installation Guide*.

To run Tobiko scenario test cases please look at *Tobiko Test Cases Execution Guide*.

### 1.3.2 Configure Tobiko Framework

In order to make sure Tobiko tools can connect to OpenStack services via Rest API configuration parameters can be passed either via environment variables or via a ini configuration file (referred here as *tobiko.conf*). Please look at *Authentication Methods* for more details.

To be able to execute scenario test cases there some OpenStack resources that has to be created before running test cases. Please look at *Setup Required Resources* for more details.

#### tobiko.conf

Tobiko tries to load *tobiko.conf* file from one of below locations:

- current directory:

```
./tobiko.conf
```

- user home directory:

```
~/.tobiko/tobiko.conf
```

- system directory:

```
/etc/tobiko/tobiko.conf
```

#### Configure Logging

Tobiko can configure logging system to write messages to a log file. You can edit below options in *tobiko.conf* to enable it as below:

```
[DEFAULT]
# Whenever to allow debugging messages to be written out or not
debug = true

# Name of the file where log messages will be appended.
log_file = tobiko.log
```

(continues on next page)

(continued from previous page)

```
# The base directory used for relative log_file paths.
log_dir = .
```

### Authentication Methods

Tobiko uses OpenStack client to connect to OpenStack services.

### Authentication Environment Variables

To configure how Tobiko can connect to services you can use the same environment variables you would use for OpenStack Python client CLI.

Currently supported variables are:

```
# Identity API version
export OS_IDENTITY_API_VERSION=3

# URL to be used to connect to OpenStack Irentity Rest API service
export OS_AUTH_URL=http://10.0.0.109:5000/v3

# Authentication username (name or ID)
export OS_USERNAME=admin
export OS_USER_ID=...

# Authentication password
export OS_PASSWORD=...

# Project-level authentication scope (name or ID)
export OS_PROJECT_NAME=admin
export OS_TENANT_NAME=admin
export OS_PROJECT_ID=...
export OS_TENANT_ID=...

# Domain-level authorization scope (name or ID)
export OS_DOMAIN_NAME=Default
export OS_DOMAIN_ID=...

# Domain name or ID containing user
export OS_USER_DOMAIN_NAME=Default
export OS_USER_DOMAIN_ID=...

# Domain name or ID containing project
export OS_PROJECT_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_ID=...

# ID of the trust to use as a trustee user
export OS_TRUST_ID=...
```

### Autentication Configuration

You can also configure the same authentication parameters by editing 'keystone' section in *tobiko.conf* file. For example:

```
[keystone]
# Identity API version
api_version = 3

# URL to be used to connect to OpenStack Irentity Rest API service
auth_url=http://10.0.0.109:5000/v3

# Authentication username (name or ID)
username = admin

# Authentication password
password = ...

# Project-level authentication scope (name or ID)
project_name = admin

# Domain-level authorization scope (name or ID)
domain = default

# Domain name or ID containing user
user_domain_name = default

# Domain name or ID containing prject
project_domain_name = default

# ID of the trust to use as a trustee user
trust_id = ...
```

### Proxy Server Configuration

The first thing to make sure is Tobiko can reach OpenStack services. In case OpenStack is not directly accessible from where test cases or Tobiko CLI are executed it is possible to use an HTTP proxy server running on a network that is able to reach all OpenStack Rest API service. This can be performed by using below standard environment variables:

```
export http_proxy=http://<proxy-host>:<proxy-port>/
export https_proxy=http://<proxy-host>:<proxy-port>/
export no_proxy=127.0.0.1,...
```

For convenience it is also possible to specify the same parameters via *tobiko.conf*:

```
[http]
http_proxy = http://<proxy-host>:<proxy-port>/
https_proxy = http://<proxy-host>:<proxy-port>/
no_proxy = 127.0.0.1,...
```

Because Tobiko test cases could execute local commands (like for example ping) to reach network services we have to specify in tobiko.conf file a shell (like OpenSSH client) to be used instead of the default local one ('/bin/sh'):

```
[shell]
command = /usr/bin/ssh <proxy-host>
```

Please make sure it is possible to execute commands on local system without having to pass a password:

```
/usr/bin/ssh <proxy-host> echo 'Yes it works!'
```

To archive it please follow one of the many guides available on Internet .

#### Setup Required Resources

To be able to execute Tobiko scenario test cases there some OpenStack resources that has to be created before running test cases.

Install required Python OpenStack clients:

```
pip install --upgrade \
    -c https://opendev.org/openstack/requirements/raw/branch/master/upper-constraints.
↪txt \
    python-openstackclient \
    python-neutronclient
```

You need to make sure *Authentication Environment Variables* are properly set:

```
source openstackrc
openstack network list
```

Add reference to the network where Tobiko should create floating IP instances in *tobiko.conf* file:

```
[neutron]
floating_network = public
```

#### Skipping resources creation

In some cases, for example when Tobiko is run after upgrade of cloud, it may be expected that resources used for tests should be already created. Tobiko should not try to create resources than and just run tests using what is already created. To configure Tobiko to not create test resources, environment variable `TOBIKO_PREVENT_CREATE` can be used:

```
export TOBIKO_PREVENT_CREATE=True
```

If this is set to `True` or `1` then Tobiko will not try to create resources like VMs, networks, routers or images and just run validation of what is exists in the cloud already.

### 1.3.3 What's Next

To know how to run Tobiko scenario test cases you can look at *Tobiko Test Cases Execution Guide*

## 1.4 Tobiko Test Cases Execution Guide

This document describes how to execute Tobiko scenario test cases.

**See also**

For a quick and simpler start you can jump to the *Tobiko Quick Start Guide*.

To install Tobiko inside a virutalenv please read *Tobiko Installation Guide*.

To configure Tobiko please read *Tobiko Configuration Guide*.

### 1.4.1 Prepare Your System

Before running Tobiko test cases you need to be sure you are doing it from Tobiko source files folder and that you have actived a Virtualenv where Tobiko and its requirements are installed. Please refers to *Tobiko Installation Guide* and *Tobiko Configuration Guide* to know how to setup your system before running test cases.

### 1.4.2 Run Scenario Test Cases

To run test cases you need a test runner able to execute Python test cases. Test cases delivered with Tobiko has been tested using stestr

From Tobiko source folder you can run scenario test cases using below command:

```
stestr run --test-path tobiko/tests/scenario/
```

CHAPTER 2

# Project Contributor Guide

CHAPTER 3

# Tobiko Framework Reference Guide

- genindex
- search

CHAPTER 4

# Configuration Reference

## 4.1 Tobiko Configuration Options

This section provides a list of all configuration options for Tobiko. These are auto-generated from Tobiko code when this documentation is built.

### 4.1.1 Configuration Reference

#### tobiko.conf

#### centos

**image_name**

**Type** string

**Default** <None>

Default centos image name

**image_url**

**Type** string

**Default** <None>

Default centos image URL

**image_file**

**Type** string

**Default** <None>

Default centos image filename

**container_format**

**Type** string

**Default** <None>

Default centos container format

**disk_format**

**Type** string

**Default** <None>

Default centos disk format

**username**

**Type** string

**Default** <None>

Default centos username

**password**

**Type** string

**Default** <None>

Default centos password

### cirros

**image_name**

**Type** string

**Default** <None>

Default cirros image name

**image_url**

**Type** string

**Default** <None>

Default cirros image URL

**image_file**

**Type** string

**Default** <None>

Default cirros image filename

**container_format**

**Type** string

**Default** <None>

Default cirros container format

**disk_format**

**Type** string

**Default** <None>

Default cirros disk format

**username**

**Type** string

**Default** `<None>`

Default cirros username

**password**

**Type** string

**Default** `<None>`

Default cirros password

### glance

**image_dir**

**Type** string

**Default** `~/.tobiko/cache/glance/images`

Default directory where to look for image files

### http

**http_proxy**

**Type** string

**Default** `<None>`

HTTP proxy URL for Rest APIs

**https_proxy**

**Type** string

**Default** `<None>`

HTTPS proxy URL for Rest APIs

**no_proxy**

**Type** string

**Default** `<None>`

Don't use proxy server to connect to listed hosts

### keystone

**api_version**

**Type** integer

**Default** `<None>`

Identity API version

**auth_url**

**Type** string

**Default** <None>

Identity service URL

**username**

**Type** string

**Default** <None>

Username

**project_name**

**Type** string

**Default** <None>

Project name

**password**

**Type** string

**Default** <None>

Password

**domain_name**

**Type** string

**Default** <None>

Domain name

**user_domain_name**

**Type** string

**Default** <None>

User domain name

**project_domain_name**

**Type** string

**Default** <None>

Project domain name

**project_domain_id**

**Type** string

**Default** <None>

Project domain ID

**trust_id**

**Type** string

**Default** <None>

Trust ID for trust scoping.

**cloud_name**

**Type** string

**Default** `<None>`

Cloud name used pick authentication parameters from clouds.*

**clouds_file_dirs**

**Type** list

**Default** `['.', '~/.config/openstack', '/etc/openstack']`

Directories where to look for clouds files

**clouds_file_names**

**Type** list

**Default** `['clouds.yaml', 'clouds.yml', 'clouds.json']`

Clouds file names

### neutron

**floating_network**

**Type** string

**Default** `<None>`

Network for creating floating IPs

**ipv4_cidr**

**Type** string

**Default** `10.100.0.0/16`

The CIDR block to allocate IPv4 subnets from

**ipv4_prefixlen**

**Type** integer

**Default** `24`

The mask bits for IPv4 subnets

**ipv4_dns_nameservers**

**Type** list

**Default** `['8.8.8.8', '8.8.4.4']`

The CIDR block to allocate IPv4 subnets from

**ipv6_cidr**

**Type** string

**Default** `2001:db8::/48`

The CIDR block to allocate IPv6 subnets from

**ipv6_prefixlen**

**Type** integer

**Default** `64`

The mask bits for IPv6 subnets

**ipv6_dns_nameservers**

**Type** list

**Default** `<None>`

The CIDR block to allocate IPv4 subnets from

**custom_mtu_size**

**Type** integer

**Default** `1350`

Customized maximum transfer unit size Notes:

- MTU values as small as 1000 has been seen breaking networking binding due to an unknown cause.
- Too big MTU values (like greater than 1400) may be refused during network creation

### nova

**key_file**

**Type** string

**Default** `~/.ssh/id_rsa`

Default SSH key to login to server instances

### octavia

**check_interval**

**Type** integer

**Default** `5`

Interval to check for status changes, in seconds.

**check_timeout**

**Type** integer

**Default** `360`

Timeout, in seconds, to wait for a status change.

### ping

**count**

**Type** integer

**Default** `1`

Number of ICMP messages to wait before ending ping command execution

**deadline**

**Type** integer

**Default** `5`

Max seconds waited from ping command before self terminating himself

**`fragmentation`**

**Type** string

**Default** `<None>`

If False it will not allow ICMP messages to be delivered in smaller fragments

**`interval`**

**Type** string

**Default** `1`

Seconds of time interval between consecutive before ICMP messages

**`packet_size`**

**Type** integer

**Default** `<None>`

Size in bytes of ICMP messages (including headers and payload)

**`timeout`**

**Type** integer

**Default** `300.0`

Maximum time in seconds a sequence of ICMP messages is sent to a destination host before reporting as a failure

### shell

**`command`**

**Type** string

**Default** `/bin/sh -c`

Default shell command used for executing local commands

**`sudo`**

**Type** string

**Default** `sudo`

Default sudo command used for executing commands as superuser or another user

### ssh

**`debug`**

**Type** boolean

**Default** `False`

Logout debugging messages of paramiko library

**command**

**Type** string

**Default** `/usr/bin/ssh`

Default SSH client command

**port**

**Type** string

**Default** `<None>`

Default SSH port

**username**

**Type** string

**Default** `<None>`

Default SSH username

**config_files**

**Type** list

**Default** `['ssh_config']`

Default user SSH configuration files

**key_file**

**Type** string

**Default** `~/.ssh/id_rsa`

Default SSH private key file

**allow_agent**

**Type** boolean

**Default** `False`

Set to False to disable connecting to the SSH agent

**compress**

**Type** boolean

**Default** `False`

Set to True to turn on compression

**timeout**

**Type** floating point

**Default** `15.0`

SSH connect timeout in seconds

**connection_attempts**

**Type** integer

**Default** `120`

Maximum number of connection attempts to be tried before timeout

**connection_interval**

**Type** floating point

**Default** `5.0`

Minimal seconds to wait between every failed SSH connection attempt

**connection_timeout**

**Type** integer

**Default** `600.0`

Time before stopping retrying establishing an SSH connection

**proxy_jump**

**Type** string

**Default** `<None>`

Default SSH proxy server

**proxy_command**

**Type** string

**Default** `<None>`

Default proxy command

### testcase

**timeout**

**Type** floating point

**Default** `<None>`

Timeout (in seconds) used for interrupting test case execution

### topology

**nodes**

**Type** list

**Default** `<None>`

List of hostname nodes

**key_file**

**Type** string

**Default** `<None>`

Default SSH key to login to cloud nodes

**username**

**Type** string

**Default** `<None>`

Default username for SSH login

**port**

**Type** string

**Default** `<None>`

Default port for SSH login

**ip_version**

**Type** string

**Default** `<None>`

**Valid Values** '', 4, 6

Limit connectivity to cloud to IPv4 o IPv6

### tripleo

**undercloud_ssh_hostname**

**Type** string

**Default** `undercloud-0`

hostname or IP address to be used to connect to undercloud host

**undercloud_ssh_port**

**Type** integer

**Default** `<None>`

TCP port of SSH server on undercloud host

**undercloud_ssh_username**

**Type** string

**Default** `stack`

Username with access to stackrc and overcloudrc files

**undercloud_ssh_key_filename**

**Type** string

**Default** `~/.ssh/id_rsa`

SSH key filename used to login to Undercloud node

**undercloud_rcfile**

**Type** list

**Default** `['~/stackrc']`

Undercloud RC filename

**overcloud_ssh_port**

**Type** integer

**Default** `<None>`

TCP port of SSH server on overcloud hosts

**overcloud_ssh_username**

**Type** string

**Default** `heat-admin`

Default username used to connect to overcloud nodes

**overcloud_ssh_key_filename**

**Type** string

**Default** `~/.ssh/id_overcloud`

SSH key filename used to login to Overcloud nodes

**overcloud_rcfile**

**Type** list

**Default** `['~/overcloudrc', '~/qe-Cloud-0rc']`

Overcloud RC filenames

**overcloud_ip_version**

**Type** integer

**Default** `<None>`

Default IP address version to be used to connect to overcloud nodes

**overcloud_network_name**

**Type** string

**Default** `<None>`

Name of network used to connect to overcloud nodes

### ubuntu

**image_name**

**Type** string

**Default** `<None>`

Default ubuntu image name

**image_url**

**Type** string

**Default** `<None>`

Default ubuntu image URL

**image_file**

**Type** string

**Default** `<None>`

Default ubuntu image filename

**container_format**

**Type** string

**Default** <None>

Default ubuntu container format

**disk_format**

**Type** string

**Default** <None>

Default ubuntu disk format

**username**

**Type** string

**Default** <None>

Default ubuntu username

**password**

**Type** string

**Default** <None>

Default ubuntu password

### 4.1.2 Sample Configuration Files

#### Sample tobiko.conf

This sample configuration can also be viewed in the raw format.

```
[DEFAULT]


[centos]

#
# From tobiko
#

# Default centos image name (string value)
#image_name = <None>

# Default centos image URL (string value)
#image_url = <None>

# Default centos image filename (string value)
#image_file = <None>

# Default centos container format (string value)
#container_format = <None>

# Default centos disk format (string value)
#disk_format = <None>

# Default centos username (string value)
#username = <None>
```

(continues on next page)

(continued from previous page)

```
# Default centos password (string value)
#password = <None>


[cirros]

#
# From tobiko
#

# Default cirros image name (string value)
#image_name = <None>

# Default cirros image URL (string value)
#image_url = <None>

# Default cirros image filename (string value)
#image_file = <None>

# Default cirros container format (string value)
#container_format = <None>

# Default cirros disk format (string value)
#disk_format = <None>

# Default cirros username (string value)
#username = <None>

# Default cirros password (string value)
#password = <None>


[glance]

#
# From tobiko
#

# Default directory where to look for image files (string value)
#image_dir = ~/.tobiko/cache/glance/images


[http]

#
# From tobiko
#

# HTTP proxy URL for Rest APIs (string value)
#http_proxy = <None>

# HTTPS proxy URL for Rest APIs (string value)
#https_proxy = <None>

# Don't use proxy server to connect to listed hosts (string value)
#no_proxy = <None>
```

(continues on next page)

(continued from previous page)

```
[keystone]

#
# From tobiko
#

# Identity API version (integer value)
#api_version = <None>

# Identity service URL (string value)
#auth_url = <None>

# Username (string value)
#username = <None>

# Project name (string value)
#project_name = <None>

# Password (string value)
#password = <None>

# Domain name (string value)
#domain_name = <None>

# User domain name (string value)
#user_domain_name = <None>

# Project domain name (string value)
#project_domain_name = <None>

# Project domain ID (string value)
#project_domain_id = <None>

# Trust ID for trust scoping. (string value)
#trust_id = <None>

# Cloud name used pick authentication parameters from clouds.* (string value)
#cloud_name = <None>

# Directories where to look for clouds files (list value)
#clouds_file_dirs = .,~/.config/openstack,/etc/openstack

# Clouds file names (list value)
#clouds_file_names = clouds.yaml,clouds.yml,clouds.json


[neutron]

#
# From tobiko
#

# Network for creating floating IPs (string value)
#floating_network = <None>

# The CIDR block to allocate IPv4 subnets from (string value)
```

(continues on next page)

(continued from previous page)

```
#ipv4_cidr = 10.100.0.0/16

# The mask bits for IPv4 subnets (integer value)
#ipv4_prefixlen = 24

# The CIDR block to allocate IPv4 subnets from (list value)
#ipv4_dns_nameservers = 8.8.8.8,8.8.4.4

# The CIDR block to allocate IPv6 subnets from (string value)
#ipv6_cidr = 2001:db8::/48

# The mask bits for IPv6 subnets (integer value)
#ipv6_prefixlen = 64

# The CIDR block to allocate IPv4 subnets from (list value)
#ipv6_dns_nameservers = <None>

# Customized maximum transfer unit size
# Notes:
#  - MTU values as small as 1000 has been seen breaking networking binding due
# to an unknown cause.
#  - Too big MTU values (like greater than 1400) may be refused during network
# creation (integer value)
#custom_mtu_size = 1350


[nova]

#
# From tobiko
#

# Default SSH key to login to server instances (string value)
#key_file = ~/.ssh/id_rsa


[octavia]

#
# From tobiko
#

# Interval to check for status changes, in seconds. (integer value)
#check_interval = 5

# Timeout, in seconds, to wait for a status change. (integer value)
#check_timeout = 360


[ping]

#
# From tobiko
#

# Number of ICMP messages to wait before ending ping command execution (integer
# value)
```

(continues on next page)

(continued from previous page)

```
#count = 1

# Max seconds waited from ping command before self terminating himself (integer
# value)
#deadline = 5

# If False it will not allow ICMP messages to be delivered in smaller fragments
# (string value)
#fragmentation = <None>

# Seconds of time interval between consecutive before ICMP messages (string
# value)
#interval = 1

# Size in bytes of ICMP messages (including headers and payload) (integer
# value)
#packet_size = <None>

# Maximum time in seconds a sequence of ICMP messages is sent to a destination
# host before reporting as a failure (integer value)
#timeout = 300.0


[shell]

#
# From tobiko
#

# Default shell command used for executing local commands (string value)
#command = /bin/sh -c

# Default sudo command used for executing commands as superuser or another user
# (string value)
#sudo = sudo


[ssh]

#
# From tobiko
#

# Logout debugging messages of paramiko library (boolean value)
#debug = false

# Default SSH client command (string value)
#command = /usr/bin/ssh

# Default SSH port (string value)
#port = <None>

# Default SSH username (string value)
#username = <None>

# Default user SSH configuration files (list value)
#config_files = ssh_config
```

(continues on next page)

(continued from previous page)

```
# Default SSH private key file (string value)
#key_file = ~/.ssh/id_rsa

# Set to False to disable connecting to the SSH agent (boolean value)
#allow_agent = false

# Set to True to turn on compression (boolean value)
#compress = false

# SSH connect timeout in seconds (floating point value)
#timeout = 15.0

# Maximum number of connection attempts to be tried before timeout (integer
# value)
#connection_attempts = 120

# Minimal seconds to wait between every failed SSH connection attempt (floating
# point value)
#connection_interval = 5.0

# Time before stopping retrying establishing an SSH connection (integer value)
#connection_timeout = 600.0

# Default SSH proxy server (string value)
#proxy_jump = <None>

# Default proxy command (string value)
#proxy_command = <None>


[testcase]

#
# From tobiko
#

# Timeout (in seconds) used for interrupting test case execution (floating
# point value)
#timeout = <None>


[topology]

#
# From tobiko
#

# List of hostname nodes (list value)
#nodes = <None>

# Default SSH key to login to cloud nodes (string value)
#key_file = <None>

# Default username for SSH login (string value)
#username = <None>
```

(continues on next page)

(continued from previous page)

```
# Default port for SSH login (string value)
#port = <None>

# Limit connectivity to cloud to IPv4 o IPv6 (string value)
# Possible values:
# '' - <No description provided>
# 4 - <No description provided>
# 6 - <No description provided>
#ip_version = <None>


[tripleo]

#
# From tobiko
#

# hostname or IP address to be used to connect to undercloud host (string
# value)
#undercloud_ssh_hostname = undercloud-0

# TCP port of SSH server on undercloud host (integer value)
#undercloud_ssh_port = <None>

# Username with access to stackrc and overcloudrc files (string value)
#undercloud_ssh_username = stack

# SSH key filename used to login to Undercloud node (string value)
#undercloud_ssh_key_filename = ~/.ssh/id_rsa

# Undercloud RC filename (list value)
#undercloud_rcfile = ~/stackrc

# TCP port of SSH server on overcloud hosts (integer value)
#overcloud_ssh_port = <None>

# Default username used to connect to overcloud nodes (string value)
#overcloud_ssh_username = heat-admin

# SSH key filename used to login to Overcloud nodes (string value)
#overcloud_ssh_key_filename = ~/.ssh/id_overcloud

# Overcloud RC filenames (list value)
#overcloud_rcfile = ~/overcloudrc,~/qe-Cloud-0rc

# Default IP address version to be used to connect to overcloud nodes  (integer
# value)
#overcloud_ip_version = <None>

# Name of network used to connect to overcloud nodes (string value)
#overcloud_network_name = <None>


[ubuntu]

#
# From tobiko
```

(continues on next page)

(continued from previous page)

```
#

# Default ubuntu image name (string value)
#image_name = <None>

# Default ubuntu image URL (string value)
#image_url = <None>

# Default ubuntu image filename (string value)
#image_file = <None>

# Default ubuntu container format (string value)
#container_format = <None>

# Default ubuntu disk format (string value)
#disk_format = <None>

# Default ubuntu username (string value)
#username = <None>

# Default ubuntu password (string value)
#password = <None>
```