Tobiko

Release 0.1.1

Jun 07, 2019

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 scenarios 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 Required Packages

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

For instance on RHEL Linux you could type:

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

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 and setuptools are installed and up-to date:

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

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

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

1.1.3 Get Tobiko

Get Tobiko source code using Git:

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

1.1.4 Configure Tobiko Credentials

In order to run the tests successfully you'll need to set up OpenStack credentials. You can do it in one of below ways:

- *Set Tobiko Credentials Via Environment Variables*
- *Set Tobiko Credentials Via tobiko-conf 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
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* adding a section like below:

```
[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

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

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:

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

Get an image for Nova instances created by Tobiko:

```
wget -c http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img
tox -e venv -- openstack image create cirros \
  --file cirros-0.4.0-x86_64-disk.img \
  --disk-format qcow2 \
  --container-format bare \
  --public
```

Create a flavor to be used with above image:

```
tox -e venv -- openstack flavor create --vcpus 1 --ram 64 --disk 1 m1.tiny
```

Create an SSH key file to be used to ssh to Nova server instances:

```
ssh-keygen -f ~/.ssh/id_rsa -P ''
```

Add reference to above resources into your *tobiko.conf* file:

```
[nova]
image = cirros
flavor = m1.tiny
key_file=~/.ssh/id_rsa
```

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

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

1.1.5 Run Test Cases

Finally run Tobiko scenario test cases using Tox:

```
tox -e scenario
```

List resources stacks created by test cases:

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

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 soruce 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

# 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
```

(continues on next page)

(continued from previous page)

```
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-glanceclient \
    python-novaclient \
    python-neutronclient
```

You need to make sure ref:*authentication-environment-variables* are properly set:

```
source openstackrc
openstack image list
openstack flavor list
openstack network list
```

Get an image for Nova instances created by Tobiko:

```
wget http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img
openstack image create cirros \
  --file cirros-0.4.0-x86_64-disk.img \
  --disk-format qcow2 \
  --container-format bare \
  --public
```

Create a flavor to be used with above image:

```
openstack flavor create --vcpus 1 --ram 64 --disk 1 m1.tiny
```

Create an SSH key file to be used to ssh to Nova server instances:

```
ssh-keygen -f ~/.ssh/id_rsa -P ''
```

Add reference to above resources into your *tobiko.conf* file:

```
[nova]
image = cirros
flavor = m1.tiny
key_file=~/.ssh/id_rsa
```

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

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

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/
```

1.5 Tobiko Faults Execution Guide

This document describes how to execute faults with 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 configure Tobiko please read *Tobiko Configuration Guide*.

1.5.1 Requirements

In order to be able faults with Tobiko you need an RC file for your OpenStack hosts (not the instances which run on OpenStack hosts) Using this RC file, Tobiko will be able to generate an os-faults configuration for you automatically. If you already have os-faults configuration file, you don't need this requirement.

1.5.2 CLI

In order to restart openvswitch service, run the following command:

tobiko-fault "restart openvswitch service"

1.5.3 Python API

You can also use faults in your tests. Warning: running a fault in a test while other tests are running in parallel might have negative affect on your other tests.

from tobiko.fault.executor import FaultExecutor fault = FaultExecutor() fault.execute("restart openvswitch service")

1.5.4 Missing services & containers

What to do if the service or the container I'm trying to control is not part of os-faults configuration? In that case please submit a patch to Tobiko to add it to tobiko/fault/templates/os-faults.yml.j2 template.

CHAPTER 2

Project Contributor Guide

CHAPTER 3

Tobiko Framework Reference Guide

- genindex
- search