

How-to deploy and configure teuthology framework

1. Teuthology overview

Teuthology is an automation test framework for Ceph, written on Python. It is used to run the vast majority of its tests and was developed due to the unique requirements of testing such a highly distributed system with active kernel development meant that none of existing framework could do the job.

The general mode of teuthology operation is to remotely orchestrate operations on remote hosts over SSH. A typical job consists of multiple nested tasks, each of which performs operations on a remote host over the network.

This paper describes process of deployment and configuration of Teuthology framework on Ubuntu 14.0 LTS (Ceph official recommended to use Ubuntu).

1.1 Paddles

Paddles is a RESTful based API to store and report back on test results from Ceph tests.

Paddles is an open source project:

<https://github.com/ceph/paddles>

The project relies on the pulpito project.

1.2 Pulpito

Pulpito is a dashboard for graphical representation of the Ceph test runs and results.

Pulpito is an open source project:

<https://github.com/ceph/pulpito>

You can check the official Pulpito interface via the link below:

<http://pulpito.ceph.com/>

The project relies on the paddles project.

1.3 Supervisor

Supervisor is a client/server system that allows its users to control a number of processes on UNIX-like operating systems. We are using supervisor for management paddles and pulpito processes.

1.4 Server NTP

We are using server NTP for time synchronization.

1.5 GitBuilder

You can check the official Ceph GitBuilder interface via the link below:

<http://gitbuilder.ceph.com/>

GitBuilder retrieves the latest version of your project from its git repository, then builds the most recent versions of all the tags and all the branches in the entire repository. If any of them have build failures, it automatically does a "git bisect" style operation to try to track down the most recent version that compiled successfully. With that information, you should be able to identify exactly which commit caused the problem, as well who exactly was responsible.

2. Installation, deployment and configuration of environment for Teuthology framework

Installation, deployment and configuration of the automation test framework Teuthology for Ceph.

I would like to stress that this article is based on the Ubuntu 14.0 LTS system, which is a very common system, is also the official recommendation of the Ceph.

This paper, contains step by step instructions which will help you install, deploy and configure Teuthology automation test framework.

2.2 Installation of dependencies

Install all dependencies by executing a command:

```
$ sudo apt-get install git python-dev python-virtualenv postgresql \
    postgresql-contrib postgresql-server-dev-all supervisor
```

```
$ sudo apt-get -y install python-pip libevent-dev python-libvirt beanstalkd
```

```
$ sudo apt-get -y install qemu-utils libev-dev libvirt-dev \
    libmysqlclient-dev libffi-dev libyaml-dev
```

2.3 Create postgres user and database for Paddles

Create user by executing a simple command:

```
$ sudo -u postgres createuser paddles -P
```

Please note that you should set password for database

Create database by executing a simple command:

```
$ sudo -u postgres createdb paddles
```

2.4 Installation, deployment and configuration of Paddles

Please note that you can also follow the official tutorial to install, or relatively simple.
<https://github.com/ceph/paddles>

Prerequisite:

Please install and configure the Ubuntu system, both on virtual or physical machine, the machine should be able to access the Internet.

Navigate to home directory.

```
$ cd ~
```

Clone paddles repository.

```
$ git clone https://github.com/ceph/paddles.git
```

Navigate to directory where paddles repository have been cloned.

```
$ cd paddles/
```

Inside the repository, create a virtualenv.

```
$ virtualenv ./virtualenv
```

Create a copy of the configuration template.

```
$ cp config.py.in config.py
```

Edit config.py to reflect your hostnames, database info, etc.

```
$ vim config.py
```

Please note that you should change logging section of the config.py in the following line:
'sqlalchemy': {'level': 'WARN'}, → 'sqlalchemy': {'level': 'WARN', 'handlers': ['console']},

Activate the virtualenv.

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

Install required python packages.

```
$ pip install -r requirements.txt
```

Run.

```
$ python setup.py develop
```

Populate the database tables.

```
$ pecan populate config.py
```

Create a copy of the alembic configuration template.

```
$ cp alembic.ini.in alembic.ini
```

Edit alembic.ini to reflect your database information.

```
$ vim alembic.ini
```

Tell alembic that you have the latest database version.

```
$ alembic stamp head
```

Create a copy of the configuration template.

```
$ sudo cp supervisord_paddles.conf /etc/supervisor/conf.d/paddles.conf
```

Edit paddles.py to reflect your paddles info, etc.

```
$ sudo nano /etc/supervisor/conf.d/paddles.conf
```

Restart supervisor service.

```
$ sudo /etc/init.d/supervisor restart
```

Check status of the processes which is run by via supervisor.

```
$ sudo supervisorctl status
```

Output example.

```
$ supervisorctl status
paddles                                RUNNING    pid 12653, uptime 0:02:13
```

Check if paddles web interface is available.

2.5 Installation, deployment and configuration of Pulpito

Please note that you can also follow the official tutorial to install, or relatively simple.
<https://github.com/ceph/pulpito>

Navigate to home directory.

```
$ cd ~
```

Clone pulpito repository.

```
$ git clone https://github.com/ceph/pulpito.git
```

Navigate to directory where have been cloned pulpito repository.

```
$ cd pulpito/
```

Inside the repository, create a virtualenv.

```
$ virtualenv ./virtualenv
```

Create a copy of the configuration template

```
$ cp config.py.in prod.py
```

Edit prod.py to reflect your paddles configuration

```
$ vim prod.py
```

Activate the virtualenv.

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

Install required python packages.

```
$ pip install -r requirements.txt
```

Create a copy of the configuration template

```
$ sudo cp supervisord_pulpito.conf /etc/supervisor/conf.d/pulpito.conf
```

Edit pulpito.conf to reflect your pulpito info.

```
$ sudo vim /etc/supervisor/conf.d/pulpito.conf
```

Restart supervisor service.

```
$ sudo supervisorctl restart
```

Check status of the processes which is run by via supervisor.

```
$ sudo supervisorctl status
```

Output example.

```
$ sudo supervisorctl status
paddles          RUNNING      pid 12986, uptime 0:00:16
pulpito          RUNNING      pid 12987, uptime 0:00:16
```

2.6 Installation, deployment and configuration of Teuthology framework

Create teuthology configuration file which reflects required configuration. For example please visit the [link](#).

```
$ sudo vim /etc/teuthology.yaml
```

Create user teuthology.

```
$ sudo adduser teuthology
```

Create user teuthworker.

```
$ sudo adduser teuthworker
```

Add the user teuthology to the group sudo.

```
$ sudo usermod -a -G sudo teuthology
```

Add the user teuthworker to the group sudo.

```
$ sudo usermod -a -G sudo teuthworker
```

Copy ssh key for granting access to environment without password.

```
$ sudo cp -rf /home/ubuntu/.ssh /home/teuthology/
```

copy ssh key for granting access to environment without password.

```
$ sudo cp -rf /home/ubuntu/.ssh /home/teuthworker/
```

Grand privileges on ssh keys.

```
$ sudo chown -R teuthology:teuthology /home/teuthology/.ssh
```

Grand privileges on ssh keys.

```
$ sudo chown -R teuthworker:teuthworker /home/teuthworker/.ssh
```

Switch user to teuthology.

```
$ su teuthology
```


Navigate to user home directory.

```
$ cd ~/
```

Create directory named src.

```
$ mkdir ~/src
```

Clone teuthology repository.

```
$ git clone https://github.com/ceph/teuthology.git src/teuthology_master
```

Navigate to directory where have been cloned teuthology repository.

```
$ cd src/teuthology_master/
```

Run.

```
$ ./bootstrap
```

Navigate to user home directory.

```
$ cd ~/
```

Switch user to teuthworker.

```
$ su teuthworker
```

Navigate to user home directory.

```
$ cd ~
```

Create directory named src.

```
$ mkdir ~/src
```

Clone teuthology repository.

```
$ git clone https://github.com/ceph/teuthology.git src/teuthology_master
```

Navigate to directory where have been cloned teuthology repository.

```
$ cd src/teuthology_master/
```

Run.

```
$ ./bootstrap
```

Navigate to home directory.

```
$ cd ~/
```

Create directory named bin.

```
$ mkdir ~/bin
```

Download script which starts worker.

```
$ wget -O ~/bin/worker_start \
https://raw.githubusercontent.com/ceph/teuthology/master/docs/_static/worker_start.sh
```

Add environment variable to profile configuration file.

```
$ echo 'PATH="$HOME/src/teuthology_master/virtualenv/bin:$PATH"' >>
~/.profile
```

Activate profile settings.

```
$ source ~/.profile
```

Create directory for log storing.

```
$ mkdir -p ~/archive/worker_logs
```

Grant required privileges on directory.

```
$ chmod +x bin/worker_start
```

Start one worker named magna.

```
$ worker_start magna 1
```

```
$ touch ~/.vault_pass.txt
```

3 Submitting slave node.

Download script which help you to configure slave nodes for Teuthology.

```
$ wget  
https://raw.githubusercontent.com/ceph/teuthology/master/docs/_static/create_nodes.py
```

Edit create_nodes.py to reflect slave nodes hostnames, etc.

```
$ vim create_nodes.py
```

Please note that all slave nodes hostnames should be resolved via /etc/hosts.

Run.

```
$ python create_nodes.py
```

Please note that before running test you should connect to all slave nodes via ssh on behalf of teuthworker user.

```
$ ssh ubuntu@<NODE_NAME>
```