Running Tests

Although tests run continuously in CI, a lot of effort was put into making it easy to run in any environment, as long as a couple of requirements are met.

Dependencies

There are some Python dependencies, which are listed in a `requirements.txt` file within the `tests/` directory. These are meant to be installed using Python install tools (pip in this case):

```
pip install -r tests/requirements.txt
```

For virtualization, either libvirt or VirtualBox is needed (there is native support from the harness for both). This makes the test harness even more flexible as most platforms will be covered by either VirtualBox or libvirt.

Running a scenario

Tests are driven by `tox`, a command line tool to run a matrix of tests defined in a configuration file (`tox.ini` in this case at the root of the project).

For a thorough description of a scenario see Test Scenarios.

To run a single scenario, make sure it is available (should be defined from `tox.ini`) by listing them:

```
tox -l
```

In this example, we will use the `luminous-ansible2.4-xenial_cluster` one. The harness defaults to `VirtualBox` as the backend, so if you have that installed in your system then this command should just work:

```
tox -e luminous-ansible2.4-xenial_cluster
```

And for libvirt it would be:

```
tox -e luminous-ansible2.4-xenial_cluster -- --provider=libvirt
```

Warning:

Depending on the type of scenario and resources available, running these tests locally in a personal computer can be very resource intensive.

Note:

Most test runs take between 20 and 40 minutes depending on system resources

The command should bring up the machines needed for the test, provision them with ceph-ansible, run the tests, and tear the whole environment down at the end.

The output would look something similar to this trimmed version:

```
luminous-ansible2.4-xenial_cluster create: /Users/alfredo/python/upstream/ceph-ansible
luminous-ansible2.4-xenial_cluster installdeps: ansible==2.4.2, -r/Users/alfredo/pytho
luminous-ansible2.4-xenial_cluster runtests: commands[0] | vagrant up --no-provision -
Bringing machine 'client0' up with 'virtualbox' provider...
Bringing machine 'rgw0' up with 'virtualbox' provider...
```

```
Bringing machine 'mds0' up with 'virtualbox' provider...
Bringing machine 'mon0' up with 'virtualbox' provider...
Bringing machine 'mon1' up with 'virtualbox' provider...
Bringing machine 'mon2' up with 'virtualbox' provider...
Bringing machine 'osd0' up with 'virtualbox' provider...
...
```

After all the nodes are up, ceph-ansible will provision them, and run the playbook(s):

```
...
PLAY RECAP *********************************************************************
client0                    : ok=4    changed=0    unreachable=0    failed=0
mds0                       : ok=4    changed=0    unreachable=0    failed=0
mon0                       : ok=4    changed=0    unreachable=0    failed=0
mon1                       : ok=4    changed=0    unreachable=0    failed=0
mon2                       : ok=4    changed=0    unreachable=0    failed=0
osd0                       : ok=4    changed=0    unreachable=0    failed=0
rgw0                       : ok=4    changed=0    unreachable=0    failed=0
...
```

Once the whole environment is all running the tests will be sent out to the hosts, with output similar to this:

```
luminous-ansible2.4-xenial_cluster runtests: commands[4] | testinfra -n 4 --sudo -v --
============================= test session starts ==============================
platform darwin -- Python 2.7.8, pytest-3.0.7, py-1.4.33, pluggy-0.4.0 -- /Users/alfre
cachedir: ../../../../.cache
rootdir: /Users/alfredo/python/upstream/ceph-ansible/tests, inifile: pytest.ini
plugins: testinfra-1.5.4, xdist-1.15.0
[gw0] darwin Python 2.7.8 cwd: /Users/alfredo/python/upstream/ceph-ansible/tests/funct
[gw1] darwin Python 2.7.8 cwd: /Users/alfredo/python/upstream/ceph-ansible/tests/funct
[gw2] darwin Python 2.7.8 cwd: /Users/alfredo/python/upstream/ceph-ansible/tests/funct
[gw3] darwin Python 2.7.8 cwd: /Users/alfredo/python/upstream/ceph-ansible/tests/funct
[gw0] Python 2.7.8 (v2.7.8:ee879c0ffa11, Jun 29 2014, 21:07:35)  -- [GCC 4.2.1 (Apple
[gw1] Python 2.7.8 (v2.7.8:ee879c0ffa11, Jun 29 2014, 21:07:35)  -- [GCC 4.2.1 (Apple
[gw2] Python 2.7.8 (v2.7.8:ee879c0ffa11, Jun 29 2014, 21:07:35)  -- [GCC 4.2.1 (Apple
[gw3] Python 2.7.8 (v2.7.8:ee879c0ffa11, Jun 29 2014, 21:07:35)  -- [GCC 4.2.1 (Apple
gw0 [154] / gw1 [154] / gw2 [154] / gw3 [154]
scheduling tests via LoadScheduling

../../../tests/test_install.py::TestInstall::test_ceph_dir_exists[ansible:/mon0]
../../../tests/test_install.py::TestInstall::test_ceph_dir_is_a_directory[ansible:/mon
../../../tests/test_install.py::TestInstall::test_ceph_conf_is_a_file[ansible:/mon0]
../../../tests/test_install.py::TestInstall::test_ceph_dir_is_a_directory[ansible:/mon
[gw2] PASSED ../../../tests/test_install.py::TestCephConf::test_ceph_config_has_mon_ho
../../../tests/test_install.py::TestInstall::test_ceph_conf_exists[ansible:/mon1]
[gw3] PASSED ../../../tests/test_install.py::TestCephConf::test_mon_host_line_has_corr
../../../tests/test_install.py::TestInstall::test_ceph_conf_is_a_file[ansible:/mon1]
[gw1] PASSED ../../../tests/test_install.py::TestInstall::test_ceph_command_exists[ans
../../../tests/test_install.py::TestCephConf::test_mon_host_line_has_correct_value[ans
...
```

Finally the whole environment gets torn down:

```
luminous-ansible2.4-xenial_cluster runtests: commands[5] | vagrant destroy --force
==> osd0: Forcing shutdown of VM...
==> osd0: Destroying VM and associated drives...
==> mon2: Forcing shutdown of VM...
==> mon2: Destroying VM and associated drives...
==> mon1: Forcing shutdown of VM...
==> mon1: Destroying VM and associated drives...
==> mon0: Forcing shutdown of VM...
==> mon0: Destroying VM and associated drives...
==> mds0: Forcing shutdown of VM...
==> mds0: Destroying VM and associated drives...
==> rgw0: Forcing shutdown of VM...
==> rgw0: Destroying VM and associated drives...
==> client0: Forcing shutdown of VM...
```

```
    ==> client0: Destroying VM and associated drives...
```

And a brief summary of the scenario(s) that ran is displayed:

```
    _____________________________________________ summary ______________________________
      luminous-ansible2.4-xenial_cluster: commands succeeded
      congratulations :)
```