

Layout and conventions

Test files and directories follow a few conventions, which makes it easy to create (or expect) certain interactions between tests and scenarios.

All tests are in the `tests` directory. Scenarios are defined in `tests/functional/` and use the following convention for directory structure:

```
tests/functional/<distro>/<distro version>/<scenario name>/
```

For example: `tests/functional/centos/7/journal-collocation`

Within a test scenario there are a few files that define what that specific scenario needs for the tests, like how many OSD nodes or MON nodes. Tls

At the very least, a scenario will need these files:

- `Vagrantfile`: must be symlinked from the root directory of the project
- `hosts`: An Ansible hosts file that defines the machines part of the cluster
- `group_vars/all`: if any modifications are needed for deployment, this would override them. Additionally, further customizations can be done. For example, for OSDs that would mean adding `group_vars/osds`
- `vagrant_variables.yml`: Defines the actual environment for the test, where machines, networks, disks, linux distro/version, can be defined.

Conventions

Python test files (unlike scenarios) rely on paths to *map* where they belong. For example, a file that should only test monitor nodes would live in `ceph-ansible/tests/functional/tests/mon/`. Internally, the test runner (`py.test`) will *mark* these as tests that should run on a monitor only. Since the configuration of a scenario already defines what node has a given role, then it is easier for the system to only run tests that belong to a particular node type.

The current convention is a bit manual, with initial path support for:

- `mon`
- `osd`
- `mds`
- `rgw`
- `journal_collocation`
- `all/any` (if none of the above are matched, then these are run on any host)

testinfra