DEVELOPER GUIDE (QUICK)

This guide will describe how to build and test Ceph for development.

DEVELOPMENT

The `run-make-check.sh` script will install Ceph dependencies, compile everything in debug mode and run a number of tests to verify the result behaves as expected.

```
$ ./run-make-check.sh
```

RUNNING A DEVELOPMENT DEPLOYMENT

Ceph contains a script called `vstart.sh` (see also Deploying a development cluster) which allows developers to quickly test their code using a simple deployment on your development system. Once the build finishes successfully, start the ceph deployment using the following command:

```
$ cd ceph/build  # Assuming this is where you ran cmake
$ make vstart
$ ../src/vstart.sh -d -n -x
```

You can also configure `vstart.sh` to use only one monitor and one metadata server by using the following:

```
$ MON=1 MDS=1 ../src/vstart.sh -d -n -x
```

The system creates two pools on startup: *cephfs_data_a* and *cephfs_metadata_a*. Let's get some stats on the current pools:

```
$ bin/ceph osd pool stats
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH ***
pool cephfs_data_a id 1
  nothing is going on

pool cephfs_metadata_a id 2
  nothing is going on

$ bin/ceph osd pool stats cephfs_data_a
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH ***
pool cephfs_data_a id 1
  nothing is going on

$ bin/rados df
POOL_NAME         USED OBJECTS CLONES COPIES MISSING_ON_PRIMARY UNFOUND DEGRADED RD_OPS RD WR
cephfs_data_a        0       0      0      0                  0       0        0      0  0
cephfs_metadata_a 2246      21      0     63                  0       0        0      0  0

total_objects    21
total_used       244G
total_space      1180G
```

Make a pool and run some benchmarks against it:

```
$ bin/rados mkpool mypool
$ bin/rados -p mypool bench 10 write -b 123
```

Place a file into the new pool:

```
$ bin/rados -p mypool put objectone <somefile>
```

```
$ bin/rados -p mypool put objecttwo <anotherfile>
```

List the objects in the pool:

```
$ bin/rados -p mypool ls
```

Once you are done, type the following to stop the development ceph deployment:

```
$ ../src/stop.sh
```

RESETTING YOUR VSTART ENVIRONMENT

The vstart script creates out/ and dev/ directories which contain the cluster's state. If you want to quickly reset your environment, you might do something like this:

```
[build]$ ../src/stop.sh
[build]$ rm -rf out dev
[build]$ MDS=1 MON=1 OSD=3 ../src/vstart.sh -n -d
```

RUNNING A RADOSGW DEVELOPMENT ENVIRONMENT

Set the `RGW` environment variable when running vstart.sh to enable the RadosGW.

```
$ cd build
$ RGW=1 ../src/vstart.sh -d -n -x
```

You can now use the swift python client to communicate with the RadosGW.

```
$ swift -A http://localhost:8000/auth -U test:tester -K testing list
$ swift -A http://localhost:8000/auth -U test:tester -K testing upload mycontainer ceph
$ swift -A http://localhost:8000/auth -U test:tester -K testing list
```

RUN UNIT TESTS

The tests are located in *src/tests*. To run them type:

```
$ make check
```