Ceph Demo: Store and Retrieve an Object

Overview

1. Clone the git repository
2. Set up dev environment
3. Set up a test cluster
4. Create a Ceph pool
5. Upload a file as an object to that pool
6. List & retrieve objects in the pool

1. Clone the git repository

- Go to wherever you want your git repo to be (i.e. your home directory) and clone the repository:

```
[lflores@folio01 ~]$ git clone https://github.com/ceph/ceph.git
```

- Navigate to your ceph repository:

```
[lflores@folio01 ~]$ cd ceph
[lflores@folio01 ceph]$ pwd
/home/lflores/ceph
```

2. Set up dev environment

- Install dependencies:

```
[lflores@folio01 ceph]$ ./install-deps.sh
```

- Run build directory script:

```
[lflores@folio01 ceph]$ ./do_cmake.sh
```

- Compile the code:

```
[lflores@folio01 build]$ ninja vstart-base cephfs cython_cephfs
```

3. Set up a test (“vstart”) cluster

- Run the vstart script:

```
[lflores@folio01 build]$ RGW=0 MDS=0 ../src/vstart.sh --debug --new -x --localhost --bluestore --without-dashboard
```

- Check cluster health. This is like the “home page” of the cluster:

```
[lflores@folio01 build]$ ./bin/ceph -s
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH ***
2024-09-17T00:21:12.867+0000 7f617193e640 -1 WARNING: all dangerous and experimental features are enabled.
2024-09-17T00:21:12.872+0000 7f617193e640 -1 WARNING: all dangerous and experimental features are enabled.
  cluster:
    id:     87c188ff-724a-4c80-b841-c0f061bc20dd
    health: HEALTH_OK

  services:
    mon: 3 daemons, quorum a,b,c (age 2m)
    mgr: x(active, since 2m)
    osd: 3 osds: 3 up (since 100s), 3 in (since 113s)

  data:
    pools:   1 pools, 1 pgs
    objects: 2 objects, 449 KiB
    usage:   3.0 GiB used, 300 GiB / 303 GiB avail
    pgs:     1 active+clean
```

4. Create a Ceph Pool

- Create a new pool where you will store your object:

```
[lflores@folio01 build]$ ./bin/ceph osd pool create my-pool
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH ***
2024-09-17T03:38:14.627+0000 7f3765a3b640 -1 WARNING: all dangerous and experimental features are enabled.
2024-09-17T03:38:14.638+0000 7f3765a3b640 -1 WARNING: all dangerous and experimental features are enabled.
pool 'my-pool' created
```

- Check that the pool exists:

```
[lflores@folio01 build]$ ./bin/ceph osd pool ls
*** DEVELOPER MODE: setting PATH, PYTHONPATH and LD_LIBRARY_PATH ***
2024-09-17T03:40:14.449+0000 7f7865c50640 -1 WARNING: all dangerous and experimental features are enabled.
2024-09-17T03:40:14.458+0000 7f7865c50640 -1 WARNING: all dangerous and experimental features are enabled.
.mgr
my-pool
```

5. Upload a file as an object to that pool

- Create a text file:

```
[lflores@folio01 build]$ echo "Hello, world!" > hello-world.txt
[lflores@folio01 build]$ cat hello-world.txt
Hello, world!
```

- Upload the text file to the pool. Here, we use the `rados` command to upload the file as an object into the pool. This command uploads `hello-world.txt` to pool “my-pool” as the object “my-object”.

```
[lflores@folio01 build]$ ./bin/rados put my-object --pool=my-pool hello-world.txt
```

6. List & Retrieve Objects in the Pool

- List the objects currently stored in the pool:

```
[lflores@folio01 build]$ ./bin/rados ls --pool=my-pool
2024-09-17T03:49:49.495+0000 7fda1b23e000 -1 WARNING: all dangerous and experimental features are enabled.
2024-09-17T03:49:49.500+0000 7fda1b23e000 -1 WARNING: all dangerous and experimental features are enabled.
2024-09-17T03:49:49.500+0000 7fda1b23e000 -1 WARNING: all dangerous and experimental features are enabled.
my-object
```

- Retrieve the object's contents, saving it to a new text file:

```
[lflores@folio01 build]$ ./bin/rados get my-object --pool=my-pool retrieved-hello-world.txt
```

- Verify the new file's contents:

```
[lflores@folio01 build]$ cat retrieved-hello-world.txt
Hello, world!
```