Phobos tutorial – Advanced Commands

Guillaume Courrier
Sebastien Gougeaud

This document has been written and tested with Phobos 1.94.1. It assumes that the reader has access to a VM with Phobos installed and QuadstorVTL to manage tapes. Each exercise assumes that the previous ones were done.

For those exercises, a database was specially set up on the VM, please use it by changing the database connect string in the configuration file `/etc/phobos.conf`.

1 Labelling media

Using `phobos21` *database*

1. Create then add to the database a set of 10 directories such as:
   - they are named `tuto{1..10}`;
   - the first two get the tag `cache`;
   - the last five get the tag `archive`;
   - the last two also get the tag `read-only`.
2. Format all the directories and unlock only the directories 1, 3 and 7.
3. Try to launch the following commands, and tell why some of them fail.

```
1 phobos put /etc/hosts tuto1
2 phobos put --tags cache /etc/hosts tuto2
3 phobos put --tags archive /etc/hosts tuto3
4 phobos put --tags read-only /etc/hosts tuto4
5 phobos put --tags archive,read-only /etc/hosts tuto5
```

4. After unlocking the directory 9, does the fifth command succeed?

2 Updating labels

Using `phobos22` *database*

1. List the available tapes.

2. Update the tapes tags such as:
    - tag `old-gen` on `LTO7` tapes;
    - tag `to-format` on tapes tagged as `corrupted`.
3. Check the tags update succeed.

3 Updating access

Using `phobos23` *database*

1. Modify the directory accesses such as:
    - we can write on the directory tagged `fast`;
    - we can retrieve the object `tuto1`;
    - we can delete the object `tuto2`.
2. Remove all the accesses of all directories.

4 Locating resources

Using `phobos24` *database*

1. List the available tapes and directories.
2. For each of them, launch a `locate` command and tell why the command gives you this result.
3. What will happen if a `get` command target an object located on the `P00001L7` tape?

5 Locating objets

Still using `phobos24` *database*

1. List the objects.
2. For each of them, launch a `locate` command and tell why the command gives you this result.
3. What will happen if a `get` command target the object `on-distant-dir`? The object `on-both-tapes`?

6 Updating an object

Using `phobos26` *database*

1. List the objects.
2. Update the object `to-update` by pushing the file `/etc/hosts`.
3. List the deprecated versions using the option `--deprecated` to check if the overwritten version still exists.
4. Retrieve the version 1 of the object `to-update`.

7 Adding and deleting objets

Using `phobos27` *database*

1. Create and add one directory to the Phobos system. Format and unlock it.
2. Add an objet to the system.
3. Delete it using the `del` command.
4. List the deleted object using the option `--deprecated`.
5. Cancel the deletion.
6. Check the object is alive using the `list` command.

8 Managing objects

Using `phobos28` *database*

1. List the objects.
2. Delete the object `to-remove`.
3. List the deleted/deprecated objects.
4. Cancel the deletion of the objects `removed2` and `removed3`.
5. Why the operation did not work for `removed3`?
6. Retrieve the contents of `removed3` and insert it back in the system.

9 Managing aliases

Using `phobos29` *database*

1. Before launching `phobosd`, open the configuration file and look at the definition of the default alias.
2. Define two new aliases:
   - `fast-archive` – 1 replica, using `fast` tagged directories;
   - `mirror-cache` – 3 replicas, using `directories`.
3. Create 4 directories, with one tagged as `fast`, format and unlock them.
4. Add two objects to the Phobos system, one for each alias.
5. Check the alias was well-considered, by listing the extents.
6. Repeat the last two questions with the following parameters, without defining an alias: 2 replicas, on `raid` tagged directories.