

Working with a public Swift container

Prerequisites

In order to create a Swift container, be sure that you have **installed and configured** the required command-line interface (CLI) tools.

Creating the container

To create a public container (that is, one whose contents can be accessed without credentials), use the following command:

OpenStack CLI **Swift CLI**

```
$ openstack container create --public public-
container
+-----+-----+
+-----+-----+
| account          | container      | x-
trans-id          |               |
+-----+-----+
+-----+-----+
| AUTH_30a7768a0ffc40359d6110f21a6e7d88 |
public-container |
tx00000d4f7d958e3e0c9aa-00638dc6ac-300de11-
default |
+-----+-----+
+-----+-----+
```

Public container ACL configuration

```
$ swift post --read-acl ".r:*,.rlistings" public-
container
```

With your current set of credentials, use this command:
This command produces no output.

OpenStack CLI **Swift CLI**

```
$ openstack
container list
+-----+
+
|
| Name
|
+-----+
+
| private-
container |
| public-
container |
+-----+
```

```
+-----+
+
```

```
$ swift list
private-
container
public-
container
```

For detailed information about an individual container, you can also use the `show` command. Observe that the Read access control list (ACL) contains the entry `.r:*,.rlistings`, which enables read access to all objects in a container, and to a list of objects included in the container.

OpenStack CLI Swift CLI

```
openstack container show public-container
+-----+-----+
| Field      | Value                                |
+-----+-----+
| account    | AUTH_30a7768a0ffc40359d6110f21a6e7d88 |
| bytes_used | 0                                    |
| container  | public-container                    |
| object_count | 0                                    |
| read_acl   | .r:*,.rlistings                    |
| storage_policy | default-placement                |
+-----+-----+

$ swift stat public-container
Account:
AUTH_30a7768a0ffc40359d6110f21a6e7d88
Container: public-container
Objects: 0
Bytes: 0
Read ACL: .r:*,.rlistings
Write ACL:
Sync To:
Sync Key:
X-Timestamp: 1670235997.87682
X-Container-Bytes-Used-Actual: 0
X-Storage-Policy: default-placement
X-Storage-Class: STANDARD
Last-Modified: Mon, 05 Dec 2022
10:26:37 GMT
X-Trans-Id:
tx00000cd9e7c26095ab862-00638dc78a-301ddeb-
default
X-Openstack-Request-Id:
tx00000cd9e7c26095ab862-00638dc78a-301ddeb-
default
Accept-Ranges: bytes
Content-Type: text/plain; charset=utf-8

$ echo "hello world" > testobj.txt
```

local test file:

Then, upload the file (as a Swift object) into your container, and read back its metadata:

OpenStack CLI Swift CLI

```
$ openstack object create public-container
testobj.txt
+-----+-----+-----+
| object   | container | etag   |
+-----+-----+-----+
| testobj.txt | public-container |
6f5902ac237024bdd0c176cb93063dc4 |
+-----+-----+-----+
```

```
$ openstack object show public-container
testobj.txt
+-----+-----+
| Field      | Value                               |
+-----+-----+
| account    | AUTH_30a7768a0ffc40359d6110f21a6e7d88 |
| container  | public-container                     |
| content-length | 12                                |
| content-type | text/plain                         |
| etag       | 6f5902ac237024bdd0c176cb93063dc4    |
| last-modified | Mon, 05 Dec 2022 10:28:09
GMT      |
| object     | testobj.txt                        |
+-----+-----+
```

```
$ swift upload public-container testobj.txt
testobj.txt

$ swift stat public-container testobj.txt
Account:
AUTH_30a7768a0ffc40359d6110f21a6e7d88
Container: public-container
Object: testobj.txt
Content Type: text/plain
Content Length: 12
Last Modified: Mon, 05 Dec 2022 10:28:09
GMT
ETag:
6f5902ac237024bdd0c176cb93063dc4
Accept-Ranges: bytes
X-Timestamp: 1670236089.75015
X-Trans-Id:
tx0000075bca59e9149bc53-00638dc7fa-301ddeb-
default
X-Openstack-Request-Id:
tx0000075bca59e9149bc53-00638dc7fa-301ddeb-
default
```

container, you can use the following

OpenStack CLI

Swift CLI

```
$ openstack object  
save --file - private-  
container  
testobj.txt  
hello world
```

The `--file -` option prints the file contents to stdout. If instead you want to save the object's content to a local file, use `--file <filename>`.

If you omit the `--file` argument altogether, `openstack object save` will create a local file named like the object you are downloading (in this case, `testobj.txt`).

```
$ swift download -  
o - private-  
container  
testobj.txt  
hello world
```

The `-o` option prints the file contents to stdout. If instead you want to save the object's content to a local file, use `-o <filename>`.

If you omit the `-o` argument altogether, `swift download` will create

a local file named

like the object you

However, this being a public container, you can *also* retrieve your object using any regular HTTP/HTTPS client, using a public URL. This URL is composed as follows:

1. the Swift API's base URL, which differs by Cleura Cloud region (`https://swift-<region>.citycloud.com:<port>/swift/v1/`),
2. the container's account string, starting with `AUTH_`,
3. the container name (in our example, `public-container`),
4. the object name (in our example, `testobj.txt`).

Rather than composing the public URL manually, you can also retrieve it by parsing the CLI's debug output:

OpenStack CLI Swift CLI

```
$ openstack object show --debug public-  
container testobj.txt 2>&1 \  
| grep -o "https://.*testobj.txt"  
https://swift-fra1.citycloud.com:8080/swift/  
v1/  
AUTH_30a7768a0ffc40359d6110f21a6e7d88/  
public-container/testobj.txt  
https://swift-fra1.citycloud.com:8080  
"HEAD /swift/v1/  
AUTH_30a7768a0ffc40359d6110f21a6e7d88/  
public-container/testobj.txt  
https://swift-fra1.citycloud.com:8080/swift/  
v1/  
AUTH_30a7768a0ffc40359d6110f21a6e7d88/  
public-container/testobj.txt
```

```
$ swift stat --debug public-container  
testobj.txt 2>&1 \  

```

```
$ curl https://swift-fra1.citycloud.com:8080/swift/v1/  
AUTH_30a7768a0ffc40359d6110f21a6e7d88/public-container/testobj.txt  
hello world  
  
public-container/testobj.txt
```

you can fetch the object's contents
le uses `curl` :

Last update: 2022-11-29

Created: 2022-12-05

Authors: Florian Haas