Using temporary URLs

Even though an object might be stored in a private container, you may still grant temporary access to it. This is known as a *temporary* URL, or TempURL.

Prerequisites

In order to manage TempURLs, be sure that you have installed and configured the `swift` command-line interface (CLI). There is presently no way to create TempURLs with the `openstack` CLI.

Also, ensure that you have configured a private container, i.e. one with an empty Read ACL. The examples in this how-to guide assume that your container is named `private-container`.

Setting a TempURL shared secret

In order to be able to create TempURLs, you must first create a shared secret at the account level. You should create a secret that is hard to guess, such as one generated by a utility like `pwgen`:

```
TEMP_URL_KEY=`pwgen 32 1`
```

To set the account-level secret, proceed with the following command:

OpenStack CLI

```
$ openstack object store account set --property Temp-URL-Key=${TEMP_URL_KEY}
```

Swift CLI

```
$ swift post -m Temp-Url-Key:${TEMP_URL_KEY}
```

Note that since this an account-level setting, you invoke `swift post` without a

container or object
name.
The TempURL secret is not encrypted or hashed; you can read it back at the account level with the following command:

OpenStack CLI **Swift CLI**

```
$ openstack object store account show
+------------+------------------------------------------------+
| Field      | Value                                          |
+------------+------------------------------------------------+
| Account    |
AUTH_30a7768a0ffc40359d6110f21a6e7d88
|
| Bytes      | 24                                             |
| Containers | 2                                              |
| Objects    | 2                                              |
| properties | temp-url-
key='tooNgeiNgieJe6bohg7teik8eiDeeMai' |
+------------+------------------------------------------------+

$ swift stat
                                Account:
AUTH_30a7768a0ffc40359d6110f21a6e7d88
                             Containers: 2
                                Objects: 2
                                  Bytes: 24
Objects in policy "default-placement-bytes": 0
  Bytes in policy "default-placement-bytes": 0
   Containers in policy "default-placement": 2
      Objects in policy "default-placement": 2
        Bytes in policy "default-placement": 24
                        Meta Temp-Url-Key:
tooNgeiNgieJe6bohg7teik8eiDeeMai
                             X-Timestamp:
1670245963.98328
               X-Account-Bytes-Used-Actual: 8192
                             X-Trans-Id:
tx00000fbce1bedc1e2b138-00638dee4b-301ddeb-
default
                  X-Openstack-Request-Id:
tx00000fbce1bedc1e2b138-00638dee4b-301ddeb-
default
                           Accept-Ranges: bytes
                            Content-Type: text/plain;
charset=utf-8
```

ect

vate container, select a duration
elow uses 1 hour (3,600 seconds).

- the HTTP method for which the TempURL should apply (usually `GET`),
- the TempURL lifetime, in seconds,
- the full path to the object including
- the `/v1` prefix,
- the account identifier starting with `AUTH_`,

- the container name,
- the object name,
- the TempURL key.

When specified in this way, the command returns a path similar to the following:

```
$ swift tempurl GET 3600 \
    /v1/AUTH_30a7768a0ffc40359d6110f21a6e7d88/private-container/testobj.txt     \
    tooNgeiNgieJe6bohg7teik8eiDeeMai
/v1/AUTH_30a7768a0ffc40359d6110f21a6e7d88/private-container/testobj.txt?
temp_url_sig=995d136bf2a8b1140d4b26886c9a8fc73bfb6c0d&temp_url_expires=1670250048
```

Accessing objects via their TempURL

You must then use your freshly generated TempURL path as the path in a URL pointing to the object. This will enable you to fetch the object using a simple HTTP client, like `curl`:

```
$ curl 'https://swift-fra1.citycloud.com:8080/swift/v1/
AUTH_30a7768a0ffc40359d6110f21a6e7d88/private-container/testobj.txt?
temp_url_sig=995d136bf2a8b1140d4b26886c9a8fc73bfb6c0d&temp_url_expires=1670250048
hello world
```

If you (or someone else) were to attempt to fetch the same URL *after* its lifetime expired, they would be met with an HTTP 401 error:

```
$ curl -i 'https://swift-fra1.citycloud.com:8080/swift/v1/
AUTH_30a7768a0ffc40359d6110f21a6e7d88/private-container/testobj.txt?
temp_url_sig=995d136bf2a8b1140d4b26886c9a8fc73bfb6c0d&temp_url_expires=1670250048
HTTP/1.1 401 Unauthorized
content-length: 12
x-trans-id: tx0000001113c5020d8a1de-00638df0ea-301ddeb-default
x-openstack-request-id: tx0000001113c5020d8a1de-00638df0ea-301ddeb-default
accept-ranges: bytes
content-type: text/plain; charset=utf-8
date: Mon, 05 Dec 2022 14:23:54 GMT
```

Last update: 2022-12-21

Created: 2022-12-05

Authors: Florian Haas