CloudConnect 2.5.x
CLI

Installation and User Guide

November 2017

Document Revision: 0.1

Important Notice

This document is delivered subject to the following conditions and restrictions:

- This guide contains proprietary information belonging to Elastifile Inc. Such information is supplied solely for the purpose of assisting explicitly and properly authorized users of Elastifile Inc. products.
- No part of contents may be used for any other purpose, disclosed to any person or firm, translated or reproduced by any means, electronic and mechanical, without the express prior written permission of Elastifile Inc..
- The text and graphics are for the purpose of illustration and reference only, based on the current version of the product(s) described in this document.
- The software described in this document is furnished under a license agreement. The software may be used or copied only in accordance with the terms of that agreement.
- Information in this document is subject to change without notice. Corporate and individual names and data used in examples herein are fictitious unless otherwise noted.
- Elastifile Inc. makes no warranty of any kind with regard to this printed material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Elastifile Inc. shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.
- Brand or product names are trademarks of their respective companies or organizations.

Table of Contents

1. Introduction

1.1 Document Scope

This document contains the explanations and procedures you require to efficiently install and use CloudConnect®.

1.2 Related Documentation

Refer to the following documents for additional information:

- *CloudConnect API User Guide*

1.3 Customer Support

You can contact Elastifile customer support at support@elastifile.com.

2. Overview

2.1 Introduction to CloudConnect

CloudConnect efficiently replicates file system snapshots to cloud object stores and restores file system snapshots to local file systems. CloudConnect works with cloud object stores, including Amazon S3 and Google Cloud Storage (GCS).

The following image shows the high-level relationship between a file system, CloudConnect, and a cloud object store.

Cloud object storage services allow users to create buckets in which objects are stored. CloudConnect stores file systems snapshots in buckets formatted using the CloudConnect File System (CCFS). These formatted buckets are called "CCStores."

A snapshot of a file system is copied to a CCStore using a version create operation. Each version create operation creates a new version in the CCStore. A CCStore can contain multiple versions. Versions can be either replicated contents of the same file system at different points in time, or contents of many different file systems.

The contents of a version can be restored from the cloud to a file system using a version checkout operation.

Operations such as creating or checking out versions involve the transfer of large amounts of data. Such operations take time, so CloudConnect enables you to track and manage these operations while they are being performed.

3. Installing CloudConnect

This section describes how to install and configure CloudConnect.

You need to request the CloudConnect installation tar file from Elastifile.

3.1 Requirements

The installer file must be run from a CentOS 7.3 machine with at least 10GB of free space.

The CloudConnect machine should have the following resources:

- 8GB RAM
- 64GB of free space
- 4 CPUs

3.2 Installation Procedure

1. On the Linux machine, create a directory and save the tar archive in it.
2. Untar the archive:

```
tar -xzvf <archive>
```

3. Change directory to the new directory created when untarring the archive.
4. Run the installer file:

```
./install.sh
```

5. When installation has completed, access the machine by entering its IP address or given DNS name URL in your browser.

If you do not know the CloudConnect IP address, use: ip a.

4. Basic Operations

4.1 elcc-cli Command and Path

To execute high-level CLI commands, use the elcc-cli command with the following syntax:

```
$ elcc-cli <resource-type> action
```

4.2 Setting User Credentials

Accessing commands requires user credentials. The default user credentials are:

username: admin

password :changeme

You can set user credentials in the CREDENTIALS environment variable or by specifying them in a command.

To set user credentials in the environment variable use the following syntax:

```
$ export CREDENTIALS=<username>:<password>
```

For example, to set credentials using the CREDENTIALS environment variable:

```
$ export CREDENTIALS=admin:changeme
```

To set user credentials with a command use the following syntax:

```
$ elcc-cli --auth username:password <command ...>
```

For example:

```
$ elcc-cli --auth admin:changeme <command ...>
```

4.3 Getting Help

You can use the CLI to get help on CloudConnect resources.

To get information about available resource types use the following syntax:

```
$ elcc-cli help
```

To get information about the actions available for a specific type of resource use the following syntax:

```
$ elcc-cli <resource-type> help
```

For example, to get a list of actions that can be performed on CCStores:

```
$ elcc-cli ccstore help
```

To get information about a specific action on a specific resource type use the following syntax:

```
$ elcc-cli <resource-type> help <command>
```

For example, to get information on the connect action to a CCStore

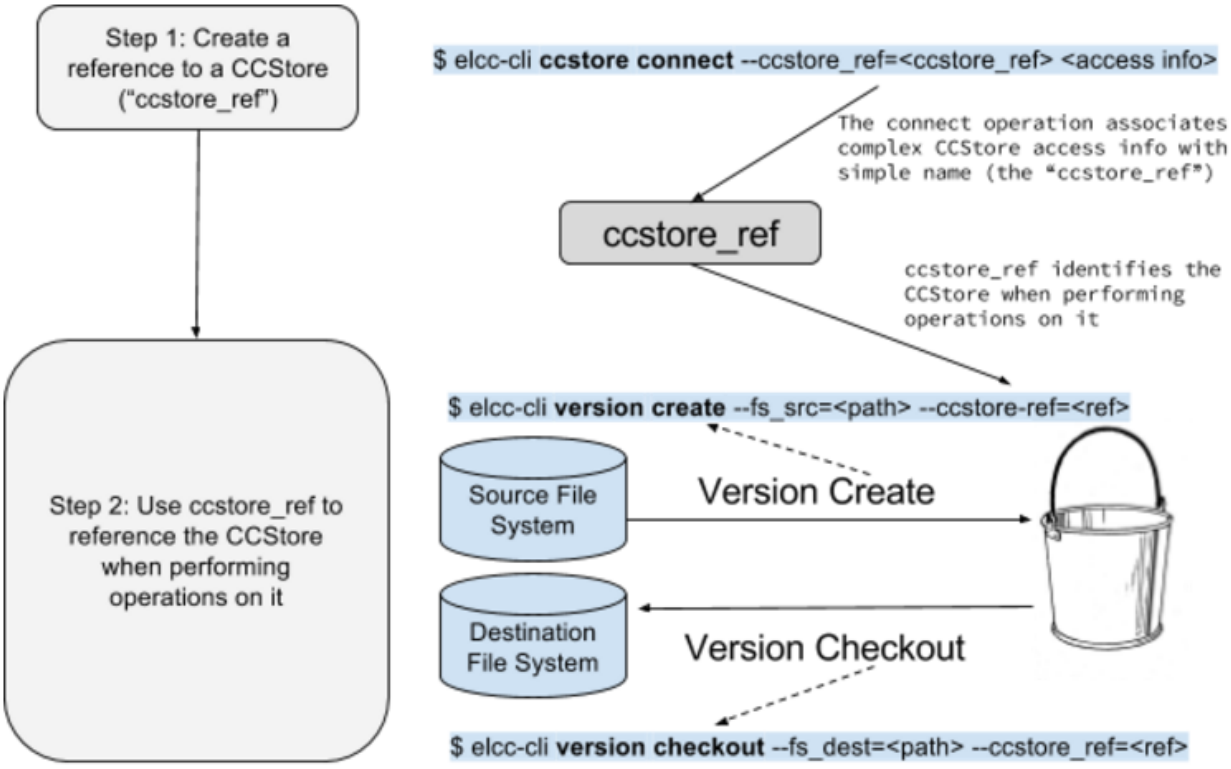
```
$ elcc-cli ccstore help connect
```

4.4 Basic Operational Flow

The CloudConnect basic operational flow consists of two steps:

1. Creating a reference to a CCStore.
2. Using the reference to perform operations.

The following image shows creating a reference and two operations using the reference.



You can use underscores ("_") or hyphens ("-") in parameter names interchangeably. For example, "--ccstore_ref" and "--ccstore-ref" are both valid.

5. Creating and Connecting to a CCStore

You can create and connect to CCStores using the connect command.

The following parameters are used with this command:

- --ccstore-ref - the identifier to assign to this connection
- --object-store - cloud storage service
- --bucket-name - name of the bucket in the cloud storage service

For bucket naming restrictions, see your cloud provider's guidelines.

- --create-if-not-found - defines if the bucket should be created in the cloud if not found
- --region - region in the cloud storage service where the bucket is located
- --access-key - the username or access key for cloud providers that use username/password or access key/secret key
- --secret-key - the password or secret key for cloud providers that use username/password or access key/secret key
- --endpoint - the endpoint URL to use for the object service
- --gc-credentials-filename - Google Cloud Platform credentials file to use (in /elastifile/ccweb/elcc_gc_credentials)

The parameters you need to provide with the command depend on which platform the CCStore resides, as shown in the following table:

Provider	cloud_service parameter value	Additional required parameters
Google Cloud Platform	gcp	ccstore_ref, bucket-name, gc_credentials_ filename
AWS S3	s3	ccstore_ref, bucket-name, access-key, secret-key
SwiftStack	swift	ccstore_ref, bucket-name, access-key, secret-key
IBM Cloud Object Storage	cleversafe	ccstore_ref, bucket-name, endpoint, access-key, secret-key
Generic S3	s3_generic	ccstore_ref, bucket-name, endpoint, access-key, secret-key
Dell EMC ECS	ecs	ccstore_ref, bucket-name, endpoint, access-key, secret-key
Cloudian Hyperstore	hyperstore	ccstore_ref, bucket-name, endpoint, access-key, secret-key
HGST ActiveScale	activescale	ccstore_ref, bucket-name, endpoint, access-key, secret-key

5.1 Creating CCStore Examples

5.1.1 Creating a CCStore on GCS

Before creating a CCStore on GCS, you must:

1. Use the Google Cloud Platform Console to create a service account JSON key (see https://cloud.google.com/docs/authentication#oauth). The JSON key is automatically downloaded to your machine.
2. Copy the JSON credentials file to:

   `/elastifile/ccweb/elcc_gc_credentials/`
3. Make sure the JSON credentials file is securely stored.

To create a CCStore on GCS, use the following syntax:

elcc-cli ccstore connect --cloud_service=gc --gc_credentials_filename=<service account key filename> --bucket-name=<CCStore> --ccstore_ref=<reference name> --create-if-not-found=true

- You must enclose values with special characters in quotes.
- If the CCStore exists and is empty, it is formatted as a CCStore.
- If the CCStore exists, is not empty, and is formatted as a CSStore, the reference is created.

For example, to create a CCStore with a service account key filename "google_credentials.json", CCStore name "cc-elastifile-com" and a reference name "myccstore":

elcc-cli ccstore connect --cloud_service=gc --gc_credentials_filename=google_credentials.json --bucket-name=cc-elastifile-com --ccstore_ref=myccstore --create-if-not-found=true

5.1.2 Creating a CCStore on Amazon S3

To create a CCStore on Amazon S3, use the following syntax:

$ elcc-cli ccstore connect --cloud-service=s3 --access-key=<account name> --secret-key=<secret key> --bucket-name=<bucket name> --ccstore_ref=<reference name> --create-if-not-found=true

- The secret key may contain special characters, and therefore should be enclosed in quotes.
- If the CCStore exists and is empty, it is formatted as a CCStore.
- If the CCStore exists, is not empty, and is formatted as a CSStore, the reference is created.

For example, to creates a CCStore with a bucket name "mybucket-cc-elastifile-com", a reference name "myccstore", and using the access key "ASDKJASKDJ" and secret key "KJHDA654632+/ASIUEBN44":

```
$ elcc-cli ccstore connect --ccstore_--cloud-service=s3 --access-key=ASDKJASKDJ --secret-key="KJHDA654632+/ASIUEBN44" --bucket-name=mybucket-cc-elastifile-com ref=myccstore --create-if-not-found=true
```

5.2 Connecting to a CCStore

The connect command is also used to connect to existing stores, but without the switch: --create-if-not-found=true.

For example, to connect to a CCStore with a service account key filename "google_credentials.json", CCStore name "cc-elastifile-com" and a reference name "myccstore":

```
elcc-cli ccstore connect --cloud_service=gc --gc_credentials_filename=google_credentials.json --bucket-name=cc-elastifile-com --ccstore_ref=myccstore
```

6. Using a CCStore

You can connect to CCStores, show CSStore connections, create versions, show operations, and show statistics.

6.1 Showing CCStore Connections

You can show a list of connected CCStores.

The list of connected CCStores does not include all CCStores you can access, only those connected.

To show a list of connected CCStores use the following syntax:

```
$ elcc-cli ccstore list
```

6.2 Creating a New Version

You can use the version create command to create a new version of any local path in a CCStore.

The following is required to create a CCStore:

- local path - the folder to upload
- reference name - CCStore reference from the connect command

The command returns immediately and outputs progress information, including:

- status - status of the operation (for example OK indicating the operating is working).
- PID - process Id of the initial phase of the version create operation.
- UUID - a unique name identifying the version create operation.
- Id - a number to identify this version create operation in subsequent commands, for example an abort comment.

For example, the following image and table shows the version create operation progress information.

```
status:   ok
pid:      7971
uuid:     morning-water-9865
id:       3
```

Progress Information	Description
status	Operation is in progress and **OK**.
pid	Initial phase is in progress and the process Id is **7971**.
uuid	The operation's unique name is **morning-water-9865**.
id	The operation's process Id is **3** for use in other commands, for example an abort.

To create a new version using a local path use the following syntax:

```
$ elcc-cli version create --fs-src=<local path> --ccstore-ref=<reference name>
```

For example, to create a new version of a folder named "/tmp/for_upload" in the CCStore with the reference name "zebra12":

```
$ elcc-cli version create --fs-src=/tmp/for_upload --ccstore-ref=zebra12
```

6.3 Showing Operations

You can show a list of operations, including:

- only operations in progress
- operations in progress and finished successfully
- operations in progress and finished successfully or failed
- finished successfully

To show only operations in progress use the following syntax:

```
$ elcc-cli operation list
```

To show operations in progress and finished successfully use the following syntax:

```
$ elcc-cli operation list --include-complete=true
```

To show operations in progress and finished successfully or failed, use the following syntax:

```
$ elcc-cli operation list --include-complete=true --include-failed=true
```

To show operations that finished successfully use the following syntax:

```
$ elcc-cli operation list --include-complete=true --include-running=false
```

6.4 Showing Statistics for an Operation

You can show statistics for an operation in progress using the operation's ID.

To show statistics for an operation in progress use the following syntax:

```
$ elcc-cli operation show --id=<operation Id>
```

For example, to show statistics the operation Id 4:

```
$ elcc-cli operation show --id=4
```

6.5 Restoring Versions from a CCStore to a Local Path

You can restore the latest version or a specific version from a CCStore to a local path.

If the local path does not exist, it is created.

To list all existing versions in a CCStore use the following syntax:

```
$ elcc-cli version list --ccstore-ref=<reference name>
```

For example, to show information about all the versions currently in the CCstore referenced by the name "zebra12":

```
$ elcc-cli version list --ccstore-ref=zebra12
```

To restore the latest version from a CCStore to a local path use the following syntax:

```
$ elcc-cli version checkout --fs_dest=<local path> --ccstore-ref=<reference name>
```

For example, to the restore the latest version of the CCStore referenced by the name "zebra12" to the local path "/tmp/output":

```
$ elcc-cli version checkout --fs_dest=/tmp/output --ccstore-ref=zebra12
```

To restore a specific version from a CCStore to a local path use the following syntax:

```
$ elcc-cli version checkout --fs_dest=/tmp/output --ccstore-ref=zebra12 --version-id=<version id>
```

For example, to restore version 3 of the CCStore referenced by the name "zebra12" to the local path "/tmp/output":

```
$ elcc-cli version checkout --fs_dest=/tmp/output --ccstore-ref=zebra12 --version-id=3
```

7. Mounting CCStores from the Cloud

You can install a special version of CloudConnect on a cloud-based VM for efficient viewing of CCstore contents.

This cloud version of CloudConnect contains the following components:

- Low-level CLI
- FUSE (Filesystem in Userspace) interface

7.1 VM Requirements

The VM must have the following configuration:

- Centos 7.1 or higher
- 8GB RAM
- Four CPUs
- 64GB storage minimum for only viewing CCStore contents
- Identity and API access: select the service account with access permissions to the required CCStores.

If you want to checkout the CCStore, the VM must have sufficient storage to store the CCStore contents taking into account data reduction.

7.2 Installation Procedure

To install the CloudConnect cloud version:

1. Transfer the following RPMs to the VM.
2. Install the RPMs as follows:
   - yum install ./jdk-8u60-linux-x64.rpm
   - yum install ./elcc-0.0.10-777.06b8c9075ac3.el7.centos.noarch.rpm

7.3 Viewing CCStores

To view a CCStore, use the following syntax:

```
$ fuse-elcc version ccstore-ref FUSE-mount-point
```

For example:

```
fuse-elcc -v hybrid-elastifile-com/Version_main/1492704231169976 hybrid-elastifile-com /mnt5
```