COS Command Line Utilities

COS provides two sets of command line utilities (CLIs) for use on a separate Linux installation to leverage the APIs exposed by the COS node:

- cos-swift – provides command-line access to the Swift API
- cos-swauth – provides command-line access to the Swauth API

This section explains how to install and use these CLI utilities.

Note
For details on the Swift and Swauth APIs, see the *Cisco Cloud Object Storage Release 3.18.1 API Guide*.

Hardware Prerequisites

Before installing the COS CLI utilities, you must have the following hardware available:

- A COS node, release 2.1.1 or later, staged in either standalone or cluster mode
- A CENTOS 6 Linux workstation on which to install the CLI utilities

Installing the CLI Utilities

Perform the following steps to install the COS CLI utilities.

Note
In the examples shown below, commands that you enter are shown in bold, while system responses are shown in normal type. System responses are typical examples, and may vary by installation.

**Step 1**
Copy the COS 3.18.1 ISO image to the Linux workstation that will host the CLI utilities.

```
ll cos_repo-3.18.1.iso
-rw-r--r-- 1 root root 250613760 Dec 12 20:39 cos_repo-3.18.1.iso
```

**Step 2**
Mount the ISO image.

```
mount -o loop cos_repo-3.18.1.iso /mnt/cdrom
ls /mnt/cdrom
```

```
cos-3.18.1-0b8-local.repo local_repo_setup Packages repodata
```
Step 3  Install the utilities on the local workstation.

  .local_repo_setup

Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
  * base: mirror.cogentco.com
  * extras: mirrors.advancedhosters.com
  * updates: centos.mirror.nac.net
base                 3.7 kB 00:00
base/group_gz        216 kB 00:00
base/filelists_db    6.1 MB 00:00
base/other_db        2.8 MB 00:00
cos-COS-0b8          3.6 kB 00:00 ...
cos-COS-0b8/group_gz 536 B 00:00 ...
cos-COS-0b8/filelists_db 66 kB 00:00 ...
cos-COS-0b8/primary_db 43 kB 00:00 ...
cos-COS-0b8/other_db 20 kB 00:00 ...
extras               3.4 kB 00:00
extras/filelists_db   31 kB 00:00
extras/prestodelta   605 B 00:00
extras/other_db      37 kB 00:00
updates               3.4 kB 00:00
updates/filelists_db 1.5 MB 00:00
updates/prestodelta  194 kB 00:00
updates/other_db     19 MB 00:02
Metadata Cache Created
Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
  * base: mirror.cogentco.com
  * extras: mirrors.advancedhosters.com
  * updates: centos.mirror.nac.net
repo id  repo name               status
  base CentOS-6 - Base          6,518
  cos-COS-0b8 Cisco Cloud Object Store COS 57
  extras CentOS-6 - Extras     37
  updates CentOS-6 - Updates   748
repolist: 7,360

Step 4  Run the YUM installation for the COS CLI utilities.

  yum install cos-cli

Loaded plugins: fastestmirror, security
Loading mirror speeds from cached hostfile
  * base: mirror.cogentco.com
  * extras: mirrors.advancedhosters.com
  * updates: centos.mirror.nac.net
Setting up Install Process
Resolving Dependencies
---> Running transaction check
---> Package cos-cli.x86_64 0:3.18.1-cos0.1 will be installed
---> Finished Dependency Resolution

Dependencies Resolved

========================================================================================================
Package Arch  Version  Repository             Size
========================================================================================================
Installing:
  cos-cli   x86_64  COS-cos0.1  cos-COS-0b8   142 k
Using the CLI Utilities

**cos swauth**

The cos-swauth CLI utility is used to manage authentication accounts and users in a COS cluster. This utility has the following form:

```
cos-swauth [-a <auth-ip>] [-u <admin-user>] [-k <admin-key>] [-h/- –help <subcommand>]
[-v/- –verbose] [subcommand <options>]
```

You can use the following parameters with this utility:

- **–a <auth-ip>**
  - The IP address of the COS authentication service.
  - The IP address can also be specified by using the COS_AUTH_IP environment variable.

- **–u <admin-user>**
  - Name of the user authorizing the command.
  - When unspecified, the user name defaults to .super_admin.
  - The user name, except in the case of the super admin, must be specified in the form `<account>:<user>`.
  - The user name can also be specified by using the COS_ADMIN_USER environment variable.

- **–k <admin-key>**
  - The authorization key (password) of the user.
  - The key can also be specified by using the COS_ADMIN_KEY environment variable.

- **–h/- –help <subcommand>**
  - Display help information for the specified subcommand.
Using the CLI Utilities

- `–v|--verbose`
  - Display detailed information as part of the output.

- subcommand <options>
  - The subcommands include `–l/list`, `–sa/set-account`, `–su/set-user`, `–g/get`, `–i/info`, `–d/delete`.
  - Each subcommand is associated with a set of relevant options. Table E-1 lists the subcommands and their options.

Table E-1  
\textit{cos-swauth Subcommands and Their Options}

<table>
<thead>
<tr>
<th>Subcommand/Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>–l/list</code></td>
<td>No account specified List the accounts for the specified admin user.</td>
</tr>
<tr>
<td><code>&lt;account&gt;</code></td>
<td>List the users belonging to the specified account.</td>
</tr>
<tr>
<td><code>–sa/set-account</code></td>
<td>&lt;account&gt; Create or update the specified account.</td>
</tr>
<tr>
<td>`–s</td>
<td>--suffix`</td>
</tr>
<tr>
<td>&lt;account&gt;</td>
<td>If a suffix is not provided, a random UUID will be used.</td>
</tr>
<tr>
<td>`–e</td>
<td>--endpoint`</td>
</tr>
<tr>
<td><code>&lt;name&gt;</code></td>
<td>If the IP address is omitted, the specified endpoint will be deleted.</td>
</tr>
<tr>
<td><code>[IP]</code></td>
<td>For the reserved endpoint name \textit{default}, the name of the default endpoint for the service type must be specified instead of the IP address.</td>
</tr>
<tr>
<td>`–t</td>
<td>--type`</td>
</tr>
<tr>
<td><code>&lt;user&gt;</code></td>
<td>Create or update the details of the specified user.</td>
</tr>
<tr>
<td>`–K</td>
<td>--key`</td>
</tr>
<tr>
<td><code>&lt;user key&gt;</code></td>
<td>This option can be used by admin or reseller-admin users to assign a password to a new user. It can also be used by existing users to change their password.</td>
</tr>
<tr>
<td>`–A</td>
<td>--admin`</td>
</tr>
<tr>
<td></td>
<td>Account admins can create, delete, or modify users within an account, and have admin privilege in the services associated with the account.</td>
</tr>
<tr>
<td></td>
<td>The user privileges default to `–N</td>
</tr>
</tbody>
</table>
Using the CLI Utilities

The cos-swift CLI utility can be used to manage storage accounts, storage containers, and storage objects in a COS cluster. This utility has the following form:

```
cos-swift [-t <auth-token>] [-a <storage-url>] [-v/-v--verbose][-h/-h--help <subcommand>] [subcommand <options>]
```

You can use the following parameters with this utility:

- `-t <auth-token>`
  - The authentication token returned by the `cos-swauth get` operation.
Using the CLI Utilities

• The authentication token can also be specified by using the COS_AUTH_TOKEN environment variable.
  • –a <storage-url>
    – The storage URL returned by the `cos-swauth get` operation.
    – The storage URL can also be specified by using the COS_STORAGE_URL environment variable.
  • –v/–verbose
    – Display detailed information as part of the output.
  • –h/–help <subcommand>
    – Display help information for the specified subcommand.
  • sub-command <options>
    – The subcommands include `–i/info`, `–l/list`, `–s/set`, `–g/get`, `–u/update`, `–d/delete`.
    – Each subcommand is associated with a set of relevant options. Table E-2 lists the subcommands and their options.

### Table E-2 cos-swift Subcommands and Their Options

<table>
<thead>
<tr>
<th>Subcommand/Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–i/info</td>
<td>No container or object specified. Display metadata for the storage account in the storage URL.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;container name&gt;</code> Display metadata for the specified storage container.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;container name&gt;/&lt;object name&gt;</code> Display metadata for the specified storage object.</td>
</tr>
<tr>
<td>–l/list</td>
<td>No container specified. List the containers of the storage account in the storage URL.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;container name&gt;</code> List the objects of the specified storage container.</td>
</tr>
<tr>
<td></td>
<td><code>&lt;container name&gt;/&lt;object path&gt;</code> Form a prefix by appending a slash to the specified object path if the path does not end with a slash. Then, list the objects of the specified storage container whose names have this prefix.</td>
</tr>
<tr>
<td>–L/–limit &lt;value&gt;</td>
<td>Restrict the number of entries listed to less than or equal to the specified value.</td>
</tr>
<tr>
<td>–m/–marker &lt;value&gt;</td>
<td>List the storage containers or objects with names greater than the specified marker value.</td>
</tr>
<tr>
<td>–e/–end_marker &lt;value&gt;</td>
<td>List the storage containers or objects with names less than the specified marker value.</td>
</tr>
<tr>
<td>–p/–prefix &lt;value&gt;</td>
<td>List the storage containers or objects whose names begin with the specified prefix.</td>
</tr>
<tr>
<td>–d/–delimiter &lt;character&gt;</td>
<td>List the storage containers or objects whose names do not have the specified delimiting character.</td>
</tr>
</tbody>
</table>

**Note** If the `–p/–prefix <value>` option is used, the delimiting character may be present in the specified prefix.
### Table E-2  cos-swift Subcommands and Their Options

<table>
<thead>
<tr>
<th>Subcommand/Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>–V/- –Verbose</td>
<td>If no container name is specified, display the metadata of the containers belonging to the storage account in the storage URL. For each container, the container name, the number of subordinate objects, and the total content size of all subordinate objects are displayed. If a container name is specified, display the metadata of the objects in that storage container. For each object, the object name, MD5 checksum, content type, content size, and modification timestamp are displayed.</td>
</tr>
<tr>
<td>–s/set</td>
<td>Create a container with the specified name.</td>
</tr>
<tr>
<td>&lt;container name&gt;</td>
<td>Create an object with the specified name within the mentioned container.</td>
</tr>
<tr>
<td>–m/- –meta &lt;name&gt;:&lt;value&gt;</td>
<td>Create a custom metadata item with the specified name and value. This option may be repeated multiple times.</td>
</tr>
<tr>
<td>–r/- –read &lt;ACL&gt;</td>
<td>Specify a read access control list (ACL) for a container. Specifying an empty string (&quot;&quot;) as the ACL will cause the existing ACL to be deleted.</td>
</tr>
<tr>
<td>–w/- –write &lt;ACL&gt;</td>
<td>Specify a write access control list (ACL) for a container.</td>
</tr>
<tr>
<td>–f/- –file &lt;name&gt;</td>
<td>Upload the named file as object content. If an existing object is specified, this file will replace prevailing object content.</td>
</tr>
<tr>
<td>–T/- –type &lt;content type&gt;</td>
<td>Specify the content type of an object. The specified content type must be a valid mime type.</td>
</tr>
<tr>
<td>–u/update</td>
<td>No container or object specified. Update the metadata of the storage account in the storage URL.</td>
</tr>
<tr>
<td>&lt;container name&gt;</td>
<td>Update the metadata of the specified container.</td>
</tr>
<tr>
<td>&lt;container name&gt;/&lt;object name&gt;</td>
<td>Update the metadata of the specified object in the container.</td>
</tr>
<tr>
<td>–m/- –meta &lt;name&gt;:&lt;value&gt;</td>
<td>Create a custom metadata item with the specified name and value. This option may be repeated multiple times.</td>
</tr>
<tr>
<td>–r/- –read &lt;ACL&gt;</td>
<td>Specify a read access control list (ACL) for a container. Specifying an empty string (&quot;&quot;) as the ACL will cause the existing ACL to be deleted.</td>
</tr>
<tr>
<td>–w/- –write &lt;ACL&gt;</td>
<td>Specify a write access control list (ACL) for a container.</td>
</tr>
<tr>
<td>–g/get</td>
<td>Retrieve the content of the specified object.</td>
</tr>
<tr>
<td>&lt;container name&gt;/&lt;object name&gt;</td>
<td>Store the retrieved object content in the specified file.</td>
</tr>
<tr>
<td>–f/- –file &lt;name&gt;</td>
<td>If this option is not used and file is not specified, the object content will be directed to stdout.</td>
</tr>
<tr>
<td>–d/delete</td>
<td></td>
</tr>
</tbody>
</table>

Note
### Using the CLI Utilities

**Using the CLI Utilities**

**<container name>** Delete the specified container.

- **Note**: Only an empty container can be deleted.

**<container name>/<object name>** Delete the specified object.

<table>
<thead>
<tr>
<th>Subcommand/Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;container name&gt;</td>
<td>Delete the specified container.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Only an empty container can be deleted.</td>
</tr>
<tr>
<td>&lt;container name&gt;/&lt;object name&gt;</td>
<td>Delete the specified object.</td>
</tr>
</tbody>
</table>