
python-ciscoclient Documentation

Release 0.2

David Moreau Simard

April 19, 2016

1	Table of Contents	3
1.1	Installing	3
1.2	Usage	4
1.3	Usage	8

`python-cicoclient` is a client, library, and a CLI interface that can be used to communicate with the ci.centos.org infrastructure provisioning system: [Duffy](#).

It provides:

- A client library for communicating with the `admin.ci.centos.org` REST API
- A wrapper library that leverages the client for doing each API call
- A CLI interface that leverages the wrapper to communicate with the API from the command line
- An [Ansible](#) module that leverages the wrapper to communicate with the API through Ansible.

Table of Contents

1.1 Installing

1.1.1 With pip

Releases of `python-cicoclient` are available on PyPi:

```
pip install python-cicoclient
```

That's it ! There is no extra dependencies or configuration required.

1.1.2 With yum or dnf

A note on dependencies

`python-cicoclient` was developed as part of an effort to streamline and simplify consumption of **ci.centos.org** from the [OpenStack RDO project](#). As such, it's development was heavily influenced by existing OpenStack clients and thus, share a lot of dependencies which are provided by the official RDO mirror.

On EL7

```
yum -y install http://rdoproject.org/repos/openstack-liberty/rdo-release-liberty.rpm
curl -s https://copr.fedoraproject.org/coprs/dmsimard/python-cicoclient/repo/epel-7/dmsimard-python-
yum -y install python-cicoclient
```

On Fedora

```
dnf -y install http://rdoproject.org/repos/openstack-liberty/rdo-release-liberty.rpm
dnf copr enable dmsimard/python-cicoclient
dnf -y install python-cicoclient
```

To get started with `cico`, read the CLI usage or the Ansible usage documentation.

1.2 Usage

In order to be able to use `cico`, you need to use it from a location that has network connectivity to the administrative endpoint, by default this is `http://admin.ci.centos.org:8080/`.

1.2.1 Built-in help

`cico` comes built-in with powerful CLI help that explains available commands, their available arguments and output formatting options thanks to the `cliff` library.

Here's what it looks like:

```
$ cico help
usage: cico [--version] [-v] [--log-file LOG_FILE] [-q] [-h] [--debug]
          [--endpoint <endpoint>] [--api-key <api-key>]

CLI interface to admin.ci.centos.org

optional arguments:
  --version            show program's version number and exit
  -v, --verbose        Increase verbosity of output. Can be repeated.
  --log-file LOG_FILE Specify a file to log output. Disabled by default.
  -q, --quiet          Suppress output except warnings and errors.
  -h, --help           Show help message and exit.
  --debug              Show tracebacks on errors.
  --endpoint <endpoint>
                       Endpoint to the admin.ci.centos.org service. Defaults
                       to: http://admin.ci.centos.org:8080/
  --api-key <api-key> API key to admin.ci.centos.org service. Defaults to
                       environment variable for CICO_API_KEY.

Commands:
  complete      print bash completion command
  help          print detailed help for another command
  inventory     Return a node inventory from the ci.centos.org infrastructure.
  node done     Releases nodes from the ci.centos.org infrastructure for a ssid
  node get      Requests nodes from the ci.centos.org infrastructure
```

If you have installed `python-cicoclient` from a RPM repository, you can also access the complete documentation with `man cico`.

1.2.2 Setting your endpoint

The endpoint defaults to `http://admin.ci.centos.org:8080/`. If you ever need to set this to something else, such as a test environment, you can override the default with the `--endpoint` argument.

1.2.3 Setting your API key

There are two ways of setting your API key when using `cico`. You can either provide it on the command line like so:

```
cico <command> --api-key <key>
```

Or by using the `CICO_API_KEY` environmental variable:


```
export CICO_API_KEY=<key>; cico <command>
```

Some commands, such as `cico inventory` do not require a key to be set. For more information, please refer to the [Duffy documentation](#).

1.2.4 Retrieving node inventory

The `cico inventory` command will allow you to retrieve the node inventory.

- If you do not have an API key configured or if you use the `--all` argument, you will get the list of all nodes.
- If you have an API key configured, you will only get the inventory of nodes that are tied to your API key.
- You can also provide a SSID to only return hosts matching this specific SSID.

Built-in help:

```
$ cico help inventory
usage: cico inventory [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
                    [--max-width <integer>] [--noindent]
                    [--quote {all,minimal,none,nonnumeric}] [--all]
                    [--ssid <ssid>]

Returns a node inventory from the ci.centos.org infrastructure.

optional arguments:
  -h, --help            show this help message and exit
  --all                 Display all nodes, regardless if an API key is used.
  --ssid <ssid>       Only return nodes matching the provided ssid.

output formatters:
  output formatter options

  -f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}
                                the output format, defaults to table
  -c COLUMN, --column COLUMN
                                specify the column(s) to include, can be repeated

table formatter:
  --max-width <integer>
                                Maximum display width, 0 to disable

json formatter:
  --noindent            whether to disable indenting the JSON

CSV Formatter:
  --quote {all,minimal,none,nonnumeric}
                                when to include quotes, defaults to nonnumeric
```

Usage:

```
$ cico inventory
Starting new HTTP connection (1): admin.ci.centos.org
Resetting dropped connection: admin.ci.centos.org
+-----+-----+-----+-----+-----+-----+-----+
| host_id | hostname | ip_address | chassis | used_count | current_state | comment |
+-----+-----+-----+-----+-----+-----+-----+
| 170 | node1.cluster | <obfuscated> | <cluster> | 66 | Deployed | e0c382aa-8a30-11e6-8000-000000000000 |
| 21 | node2.cluster | <obfuscated> | <cluster> | 66 | Deployed | b54cea7a-8a40-11e6-8000-000000000000 |
```

```

|      64 | node3.cluster | <obfuscated> | <cluster> |      67 | Deployed | 3b413756-8967-11e
+-----+-----+-----+-----+-----+-----+-----+
$ cico inventory --ssid b54cea7a-8a40-11e5-b2e3-525400ea212d
Starting new HTTP connection (1): admin.ci.centos.org
Resetting dropped connection: admin.ci.centos.org
+-----+-----+-----+-----+-----+-----+-----+
| host_id | hostname | ip_address | chassis | used_count | current_state | comment
+-----+-----+-----+-----+-----+-----+-----+
|      21 | node2.cluster | <obfuscated> | <cluster> |      66 | Deployed | b54cea7a-8a40-11e
+-----+-----+-----+-----+-----+-----+-----+

```

1.2.5 Requesting nodes

The `cico node get` command will allow you to request one or more nodes. This command requires an API key to be configured.

Built-in help:

```

$ cico help node get
usage: cico node get [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
                  [--max-width <integer>] [--noindent]
                  [--quote {all,minimal,none,nonnumeric}] [--arch <arch>]
                  [--release <release>] [--count <count>]
                  [--retry-count <count>] [--retry-interval <seconds>]

Requests nodes from the ci.centos.org infrastructure

optional arguments:
  -h, --help                show this help message and exit
  --arch <arch>             Requested server architecture. Defaults to x86_64.
  --release <release>      Requested CentOS release. Defaults to 7.
  --count <count>          Requested amount of servers. Defaults to 1.
  --retry-count <count>    Amount of retries to do in case of failure. Defaults
                           to 1.
  --retry-interval <seconds>
                           Wait between subsequent retries. Defaults to 10
                           (seconds).

output formatters:
  output formatter options

  -f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}
                           the output format, defaults to table
  -c COLUMN, --column COLUMN
                           specify the column(s) to include, can be repeated

table formatter:
  --max-width <integer>    Maximum display width, 0 to disable

json formatter:
  --noindent                whether to disable indenting the JSON

CSV Formatter:
  --quote {all,minimal,none,nonnumeric}

```

```
when to include quotes, defaults to nonnumeric
```

Usage:

```
$ cico node get --arch x86_64 --release 7 --count 1 --retry-count 2 --retry-interval 30
Starting new HTTP connection (1): admin.ci.centos.org
Resetting dropped connection: admin.ci.centos.org
Resetting dropped connection: admin.ci.centos.org
SSID for these servers: 8fd381ea-8a46-11e5-b2e3-525400ea212d
+-----+-----+-----+-----+-----+-----+-----+-----+
| host_id | hostname | ip_address | chassis | used_count | current_state | comment | distro |
+-----+-----+-----+-----+-----+-----+-----+-----+
|      117 | node4.cluster | <obfuscated> | cluster |          69 | Ready | - | None |
+-----+-----+-----+-----+-----+-----+-----+-----+
```

1.2.6 Releasing nodes

The `cico node done` command will allow you to release all the nodes tied to a session ID. This command requires an API key to be configured.

Built-in help:

```
$ cico help node done
usage: cico node done [-h] [-f {csv,json,table,value,yaml}] [-c COLUMN]
                    [--max-width <integer>] [--noindent]
                    [--quote {all,minimal,none,nonnumeric}]
                    <ssid>

Releases nodes from the ci.centos.org infrastructure for a ssid

positional arguments:
  <ssid>                SSID of the server pool to release

optional arguments:
  -h, --help            show this help message and exit

output formatters:
  output formatter options

  -f {csv,json,table,value,yaml}, --format {csv,json,table,value,yaml}
                        the output format, defaults to table
  -c COLUMN, --column COLUMN
                        specify the column(s) to include, can be repeated

table formatter:
  --max-width <integer>
                        Maximum display width, 0 to disable

json formatter:
  --noindent            whether to disable indenting the JSON

CSV Formatter:
  --quote {all,minimal,none,nonnumeric}
                        when to include quotes, defaults to nonnumeric
```

Usage:

```
$ cico node done 8fd381ea-8a46-11e5-b2e3-525400ea212d
Starting new HTTP connection (1): admin.ci.centos.org
Resetting dropped connection: admin.ci.centos.org
Resetting dropped connection: admin.ci.centos.org
Released these servers with SSID: 8fd381ea-8a46-11e5-b2e3-525400ea212d
+-----+-----+-----+-----+-----+-----+-----+
| host_id | hostname | ip_address | chassis | used_count | current_state | comment |
+-----+-----+-----+-----+-----+-----+-----+
| 117 | node4.cluster | <obfuscated> | cluster | 69 | Deployed | 8fd381ea-8a46-11e5-
```

1.3 Usage

In order to be able to use the `cico Ansible` module, you need to use it from a location that has network connectivity to the administrative endpoint, by default this is `http://admin.ci.centos.org:8080/`.

The `cico Ansible` module comes packaged with `python-cicoclient`.

To use it, it would be convenient to add the module to your Ansible module library. An example Ansible configuration file can be found inside the [package](#).

1.3.1 Built-in help

The `cico Ansible` module comes built-in with Ansible documentation, you can use `ansible-doc` to access it:

```
$ ansible-doc -M cicoclient/ansible cico
> CICO

  Ansible module to manage ci.centos.org node lifecycle

Options (= is mandatory):

= action
  Action to take (Choices: get, done, list)

- api_key
  API key [Default: CICO_API_KEY environment variable or None]

- arch
  Server architecture (Choices: i386, x86_64) [Default: x86_64]

- count
  Amount of nodes [Default: 1]

- endpoint
  API endpoint [Default: http://admin.ci.centos.org:8080/]

- release
  CentOS release (Choices: 5, 6, 7) [Default: 7]

- retry_count
  Amount of retries to do in case of failure. [Default: 1]

- retry_interval
  Wait (in seconds) between subsequent retries. [Default: 10]
```

```
- ssid
    SessionID, required with action 'done', optional with 'list'.

Requirements: python >= 2.6, python-cicoclient

EXAMPLES:
# Retrieve full inventory
- cico:
    action: list
    register: data

# Retrieve inventory tied to API key
- cico:
    action: list
    api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
    register: data

# Retrieve inventory tied to a SSID
- cico:
    action: list
    ssid: 3e03553f-ae28-4a68-b879-f0fdbf949d5d
    register: data

# Request one CentOS 7 x86_64 node
- cico:
    action: get
    api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
    register: data

# Request one CentOS 7 x86_64 node with increased tolerance failure
- cico:
    action: get
    api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
    retry_count: 3
    retry_interval: 60
    register: data

# Request two CentOS 6 i386 nodes
- cico:
    action: get
    api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
    arch: i386
    release: 6
    count: 2
    register: data

# Release nodes requested in a registered 'get' action
- cico:
    action: done
    api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
    ssid: data.ssid

# Release nodes for a specific ssid
- cico:
    action: done
    api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
    ssid: 3e03553f-ae28-4a68-b879-f0fdbf949d5d
```

```
MAINTAINERS: David Moreau Simard <dms@redhat.com>
```

1.3.2 Retrieving node inventory

The `cico` inventory action will allow you to retrieve the node inventory.

- If you do not provide an API key, you will get the list of all nodes.
- If you provide an API key, you will only get the inventory of nodes that are tied to your API key.
- You can also provide a SSID to only return hosts matching this specific SSID.

Example:

```
# Retrieve full inventory
- cico:
  action: list
  register: data

# Retrieve inventory tied to API key
- cico:
  action: list
  api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
  register: data

# Retrieve inventory tied to a SSID
- cico:
  action: list
  ssid: 3e03553f-ae28-4a68-b879-f0fdbf949d5d
  register: data
```

1.3.3 Requesting nodes

The `cico` get action will allow you to request one or more nodes. This command requires an API key to be configured.

Example:

```
# Request one CentOS 7 x86_64 node
- cico:
  action: get
  api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
  register: data

# Request one CentOS 7 x86_64 node with increased tolerance failure
- cico:
  action: get
  api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
  retry_count: 3
  retry_interval: 60
  register: data

# Request two CentOS 6 i386 nodes
- cico:
  action: get
  api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
  arch: i386
  release: 6
```

```
count: 2
register: data
```

1.3.4 Releasing nodes

The `cico done` action command will allow you to release all the nodes tied to a session ID. This command requires an API key to be configured.

Example:

```
# Release nodes requested in a registered 'get' action
- cico:
  action: done
  api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
  ssid: data.results.ssid

# Release nodes for a specific ssid
- cico:
  action: done
  api_key: 723ef3ce-4ea4-4e8d-9c8a-20a8249b2955
  ssid: 3e03553f-ae28-4a68-b879-f0fdbf949d5d
```