

Install Cloudbreak CLI 2

## Installing Cloudbreak CLI

**Date of Publish:** 2019-05-28



<https://docs.hortonworks.com/>

# Contents

<b>Installing Cloudbreak CLI.....</b>	<b>3</b>
Install Cloudbreak CLI.....	3
Configure Cloudbreak CLI.....	3
<b>Advanced CLI configuration.....</b>	<b>4</b>
Configure CLI for use with workspaces.....	4
Add multiple configurations.....	5
Configure default output.....	5
Configure CLI autocomplete.....	6
<b>Web UI options useful for CLI.....</b>	<b>6</b>
Obtain cluster JSON template.....	6
Obtain CLI command.....	7
Obtain cluster blueprint.....	7
<b>Get help.....</b>	<b>7</b>

## Installing Cloudbreak CLI

The Cloudbreak Command Line Interface (CLI) is a tool to help you manage your Cloudbreak cluster instances. This tool can be used to interact with Cloudbreak for automating cluster creation, management, monitoring, and termination.

The CLI is available for Linux, Mac OS X, and Windows. To install and configure it, refer to the following documentation:

### Related Information

[Cloudbreak CLI Reference](#)

## Install Cloudbreak CLI

After you have installed Cloudbreak, the CLI is available for download from the Cloudbreak web UI.

Steps

1. Browse to your Cloudbreak instance and log in to the Cloudbreak web UI.
2. Select Download CLI from the navigation pane.
3. Select your operating system. The CLI is available for Linux, Mac OS X, and Windows.
4. Download the selected bundle to your local machine.
5. Extract the bundle.
6. You can optionally add `cb` to your system path.
7. Run the executable to verify the CLI:

```
cb --version
```

### Related Information

[Configure Cloudbreak CLI](#)

## Configure Cloudbreak CLI

Once you have installed the CLI, you must configure the CLI to work with your Cloudbreak instance.

Steps

1. Use the `cb configure` command to set up the CLI configuration file. The configuration options are:
  - `--server` server address [`$_CB_SERVER_ADDRESS`]
  - `--username` user name (e-mail address) [`$_CB_USER_NAME`]
  - `--password` password [`$_CB_PASSWORD`]

The password configuration is optional. If you do not provide the password, no password is stored in the CLI configuration file. Therefore, you will need to provide the password with each command you execute or via an environment variable.

For example:

```
cb configure --server https://ec2-11-111-111-11.compute-1.amazonaws.com --username atest@cloudera.com
```

2. The CLI configuration file will be saved at `~/cb/config`. The content will look similar to the following:

```
default:
  username: admin@hortonworks.com
```

```
server: https://ec2-11-111-111-11.compute-1.amazonaws.com
```

- Run any command to verify that you can connect to the Cloudbreak instance via CLI. For example:

```
cb cluster list
```



**Note:**

The CLI can look for configuration options from different locations. You can optionally pass the configuration options on each command or from environment variables. The following order is used for the CLI to look for configuration options: Command line, Environment variables and the Configuration file.

**Related Information**

[Install Cloudbreak CLI](#)

[Cloudbreak CLI Reference](#)

[Add multiple configurations](#)

[Configure default output](#)

[Configure CLI autocomplete](#)

[Configure CLI for use with workspaces](#)

## Advanced CLI configuration

This section lists advanced CLI configuration options.

These optional configurations can be performed after installing the CLI:

### Configure CLI for use with workspaces

When using multiple workspaces, you can switch between them in the CLI in order to access workspace-specific resources.

There are two options to access a workspace via the CLI:

Option	When to use
Configure CLI to use a workspace permanently	This method is useful if you would like to use one workspace permanently.
Provide the <code>--workspace</code> flag with each command	This method is useful if you would like to switch between multiple workspaces.

#### Configure CLI to use a workspace permanently

To configure the CLI to use a specific workspace permanently, use:

```
cb configure --username [$VALUE] --password [$VALUE] --server [$VALUE] --workspace [$VALUE]
```

For example:

```
cb configure --username test@cloudera.com --password MyPassword123! --server http://191.118.89.112 --workspace marketing
```

This will permanently save the configuration in `~/cb/config` so you don't need to provide the `--workspace` flag when running CLI commands. Here is an example config file:

```
default:
```

```
username: test@hcloudera.com
password: MyPassword123!
server: http://191.118.89.112
workspace: marketing
```

### Provide the `--workspace` flag with each CLI command

If you do not want to permanently configure a specific workspace in the Profile, you can simply provide the `--workspace <value>` flag with each CLI command. For example, the following command returns all clusters that are part of the workspace called "marketing":

```
cb cluster list --workspace marketing
```

## Add multiple configurations

If you are using multiple profiles for multiple environments, you can configure the CLI to recognize them.

To do that, use the `cb configure` command and pass the name of your environment-specific profile file in the `--profile` parameter.

After running the command, the configuration will be added as a new entry to the config file. For example, running the following command `cb configure --server https://192.167.65.4 --username test@hortonworks.com --profile staging` will add the "staging" entry:

```
default:
  username: test1@cloudera.com
  server: https://192.167.65.4
staging:
  username: test2@cloudera.com
  server: https://192.167.65.4
```

### Example

```
#cb configure --server https://192.167.65.4 --username test2@cloudera.com --
profile staging
INFO: [writeConfigToFile] dir already exists: /Users/rkovacs/.cb
INFO: [writeConfigToFile] writing credentials to file: /Users/rkovacs/.cb/
config
# cat /Users/rkovacs/.cb/config
default:
  username: test1@cloudera.com
  server: https://192.167.65.4
  output: table
staging:
  username: test2@cloudera.com
  server: https://192.167.65.4
```

## Configure default output

You can configure a default output type (json, yaml, or table) for the CLI. By default, JSON format is used in command output.

For example, if you run `cb list-clusters` without specifying output type, the output will be JSON. If you would like to change default output, add it to the config file. For example:

```
default:
  username: test@cloudera.com
  server: https://192.167.65.4
```

```
output: table
```

## Configure CLI autocomplete

The CLI includes an autocomplete option. To use this option, you must run the bash completion function.

Use one of the following commands to configure CLI autocomplete:



**Note:**

Prior to running these commands, the cb must be added to your PATH.

```
eval "$ (cb completion) "
```

```
source <(cb completion)
```

Once configured, type the CLI commands as usual and use the Tab key to automatically complete the commands.

## Web UI options useful for CLI

This section lists web UI and CLI options that are particularly important or useful when using Cloudbreak CLI.

The following UI options are available for CLI users:

### Obtain cluster JSON template

The simplest way to obtain a valid JSON template for your cluster is to get it from the Cloudbreak web UI.

You can do this in two ways:

- From the last page of the create cluster wizard.
- For a running cluster, from cluster details.

#### From create cluster

Once you've provided all the cluster parameters, on the last page of the create cluster wizard, click Show CLI Command to obtain the JSON template.

Click Copy the JSON to copy the content and then use a text editor to edit and save it.

For detailed steps on how to use this option, refer to the [How to generate a cluster template and reuse it with Cloudbreak HCC](#) article.

#### From cluster details

You can obtain the JSON template for a cluster from the cluster details page by selecting Actions > Show CLI Command. This option is available for all clusters that have been initiated, so the cluster does not need to be in the running state to obtain this information. In fact, this option is useful when troubleshooting cluster failures.

Click Copy the JSON to copy the content and then use a text editor to edit and save it.

#### Related Information

[How to generate a cluster template and reuse it with Cloudbreak \(HCC\)](#)

## Obtain CLI command

Cloudbreak web UI includes an option to generate the create command for resources such as credentials, blueprints, clusters, and recipes.

This option is available when creating a resource and for existing resources, from the resource details page.

### From create resource

When creating a resource (credential, blueprint, cluster, or recipe), provide all information and then click Show CLI Command. The UI will display the create CLI command for the resource.

### From resource details

Navigate to credential, blueprint, cluster, or recipe details and click Show CLI Command. The UI will display the create CLI command for the resource.

## Obtain cluster blueprint

You can obtain a cluster blueprint for any running cluster and from the create cluster wizard from the Cloudbreak web UI.

### From create cluster

Once you've provided all the cluster parameters, on the last page of the create cluster wizard, click Show Generated Blueprint to obtain the blueprint.

### From cluster details

You can obtain the blueprint for a cluster from the cluster details page by selecting Actions > Show Generated Blueprint. This option is available for all clusters that have been initiated, so the cluster does not need to be in the running state to obtain this information. In fact, this option is useful when troubleshooting cluster failures.

## Get help

To get CLI help, you can add help to the end of a command. The following will list help for the CLI at the top-level:

```
cb --help
```

or

```
cb --h
```

The following will list help for the create-cluster command, including its command options and global options:

```
cb cluster --help
```

or

```
cb cluster --h
```