Cloudbreak CLI Reference

Date of Publish: 2018-09-14

http://docs.hortonworks.com

Contents

Cloudbreak CLI reference

This section will help you get started with the Cloudbreak CLI after you have installed and configured it.

Related Information

Installing Cloudbreak CLI

Command structure

The CLI command can contain multiple parts. The first part is a set of global options. The next part is the command. The next part is a set of command options and arguments which could include sub-commands.

```
cb [global options] command [command options] [arguments...]
```

Command output

You can control the output from the CLI using the –output argument. The possible output formats include:

- JSON (json)
- YAML (yaml)
- Formatted table (table)

For example:

```
cb cluster list --output json
```

```
cb clusters list --output yaml
```

```
cb cluster list --output table
```

Commands

The following commands are available via Cloudbreak CLI.

audit describe

Displays details of a specific audit event.

Required options

--audit-id <value> Id of an audit event (You can obtain this ID from the output of the cb audit list)

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Displays details of the audit event:

```
cb audit describe --audit-id 196452
{
  "Audit": {
    "auditId": 196452,
    "duration": 27,
    "operation": {
      "account": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
      "cloudbreakId": "qa-cloudbreak-280dev355-7d59d87cdc-d8845",
      "cloudbreakVersion": "2.8.0-dev.355",
      "eventType": "REST",
      "resourceId": 3782,
      "resourceType": "rdsconfigs",
      "timestamp": 1534288440363,
      "userId": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
      "userName": "dbialek@hortonworks.com",
      "zonedDateTime": "2018-08-14T23:14:00.363Z"
    },
    "rawRestEvent": {
      "duration": 27,
      "operation": {
        "account": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
        "cloudbreakId": "qa-cloudbreak-280dev355-7d59d87cdc-d8845",
        "cloudbreakVersion": "2.8.0-dev.355",
        "eventType": "REST",
        "resourceId": 3782,
        "resourceType": "rdsconfigs",
        "timestamp": 1534288440363,
        "userId": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
        "userName": "test@hortonworks.com",
        "zonedDateTime": "2018-08-14T23:14:00.363Z"
      },
      "restCall": {
        "duration": 27,
        "restRequest": {
          "body":
 "eyJjb25uZWN0aW9uVVJMIjoiamRiYzpwb3N0Z3Jlc3FsOi8vaGl2ZWRiLmNyamR1amtheWJndi51cy1lYXN0LT
          "headers": {
            "connection": "close",
            "content-length": "225",
            "content-type": "application/json",
            "host": "qa-cloudbreak-
canary.qa.svc.l42scl.hortonworks.com:8080",
            "trackingid": "0ef86b9a-6505-4474-b6e8-7ab7fec9bc03"
          },
          "mediaType": "application/json",
          "method": "POST",
          "requestUri": "http://qa-cloudbreak-
canary.qa.svc.l42scl.hortonworks.com:8080/cb/api/v1/rdsconfigs/user"
        },
        "restResponse": {
          "body":
 "eyJuYW1lIjoiYW1iYXJpLXJkIiwiY29ubmVjdGlvblVSTCI6ImpkYmM6cG9zdGdyZXNxbDovL2hpdmVkYi5jcr
```

```
          "headers": {
            "Content-Type": "application/json"
          },
          "mediaType": "application/json",
          "statusCode": 200,
          "statusText": "OK"
        }
      },
      "status": "OK - 200",
      "type": "StructuredRestCallEvent"
    },
    "status": "OK - 200"
  }
}
```

audit list

Lists all audit events for a specific resource.

Sub-commands

blueprint List audit for blueprints

cluster List audit for clusters

credential List audit for credentials

database List audit for database configurations

imagecatalog List audit for image catalogs

ldap List audit for LDAP/AD configurations

recipe List audit for recipes

Required options

--resource-id <value> Id of a resource (You can obtain this ID from the output of cb <resource> describe)

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists all audit events for a specific RDS configuration with a resource id of 3782:

```
cb audit list database --resource-id 3782
[
  {
    "Audit": {
      "auditId": 196452,
```

```
      "duration": 27,
      "operation": {
        "account": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
        "cloudbreakId": "qa-cloudbreak-280dev355-7d59d87cdc-d8845",
        "cloudbreakVersion": "2.8.0-dev.355",
        "eventType": "REST",
        "resourceId": 3782,
        "resourceType": "rdsconfigs",
        "timestamp": 1534288440363,
        "userId": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
        "userName": "test@hortonworks.com",
        "zonedDateTime": "2018-08-14T23:14:00.363Z"
      },
      "rawRestEvent": {
        "duration": 27,
        "operation": {
          "account": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
          "cloudbreakId": "qa-cloudbreak-280dev355-7d59d87cdc-d8845",
          "cloudbreakVersion": "2.8.0-dev.355",
          "eventType": "REST",
          "resourceId": 3782,
          "resourceType": "rdsconfigs",
          "timestamp": 1534288440363,
          "userId": "48c9ee87-f5f4-4749-97ca-085ac7bb1a1c",
          "userName": "dbialek@hortonworks.com",
          "zonedDateTime": "2018-08-14T23:14:00.363Z"
        },
        "restCall": {
          "duration": 27,
          "restRequest": {
            "body":
 "eyJjb25uZWN0aW9uVVJMIjoiamRiYzpwb3N0Z3Jlc3FsOi8vaGl2ZWRiLmNyamR1amtheWJndi51cy1lYXN0LT
            "headers": {
              "connection": "close",
              "content-length": "225",
              "content-type": "application/json",
              "host": "qa-cloudbreak-
canary.qa.svc.l42scl.hortonworks.com:8080",
              "trackingid": "0ef86b9a-6505-4474-b6e8-7ab7fec9bc03"
            },
            "mediaType": "application/json",
            "method": "POST",
            "requestUri": "http://qa-cloudbreak-
canary.qa.svc.l42scl.hortonworks.com:8080/cb/api/v1/rdsconfigs/user"
          },
          "restResponse": {
            "body":
 "eyJuYW1lIjoiYW1iYXJpLXJkIiwiY29ubmVjdGlvblVSTCI6ImpkYmM6cG9zdGdyZXNxbDovL2hpdmVkYi5jcn
            "headers": {
              "Content-Type": "application/json"
            },
            "mediaType": "application/json",
            "statusCode": 200,
            "statusText": "OK"
          }
        },
        "status": "OK - 200",
        "type": "StructuredRestCallEvent"
      },
      "status": "OK - 200"
    }
  }
]
```

blueprint create

Adds a new blueprint from a file or from a URL.

Sub-commands

from-url Creates a blueprint by downloading it from a URL location

from-file Creates a blueprint by reading it from a local file

Required options

Options required for from-url sub-command:

--name <value> Name for the blueprint

--url <value> URL location of the Ambari blueprint JSON file

Options required for from-file sub-command:

--name <value> Name for the blueprint

--file <value> Location of the Ambari blueprint JSON file on the local machine

Options

--description <value> Description of the resource

--datalake Marks the blueprint with "Data Lake Ready" tag

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace where to create the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Adds a blueprint from a URL:

```
cb blueprint create from-url --url https://someurl.com/test.bp --name test1
```

Adds a blueprint from a local file:

```
cb blueprint create from-file --file /Users/test/Documents/blueprints/
test.bp --name test2
```

Adds a blueprint from a local file and marks it as "Data Lake Ready":

```
cb blueprint create from-file --file /Users/test/Documents/blueprints/
testdl.bp --name testdl --datalake
```

blueprint delete

Deletes an existing blueprint.

Required options

--name <value> Blueprint name

Options

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes a blueprint called "testbp":

```
cb blueprint delete --name "testbp"
```

blueprint describe

Describes an existing blueprint.

Required options

--name <value> Blueprint name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Describes a blueprint called "bptest1":

```
cb blueprint describe --name "EDW-ETL: Apache Hive, Apache Spark 2"
{
  "Name": "EDW-ETL: Apache Hive, Apache Spark 2",
  "Description": "Useful for ETL data processing with Hive and Spark",
  "StackName": "HDP",
  "StackVersion": "2.6",
  "HostgroupCount": "3",
  "Tags": "DEFAULT",
  "BlueprintTextAsBase64":
 "eyJCbHVlcHJpbnRzIjp7ImJsdWVwcmludF9uYW1lIjoiaGRwMjYtZXRsLWVkdy1zcGFyazIiLCJzdGFja19uY
  "ID": "97"
```

```
}
```

blueprint list

Lists available blueprints.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists existing blueprints:

```
cb blueprint list
[
  {
    "Name": "EDW-Analytics: Apache Hive 2 LLAP, Apache Zeppelin",
    "Description": "Useful for EDW analytics using Hive LLAP",
    "StackName": "HDP",
    "StackVersion": "2.6",
    "HostgroupCount": "3",
    "Tags": "DEFAULT"
  },
  {
    "Name": "Data Science: Apache Spark 2, Apache Zeppelin",
    "Description": "Useful for data science with Spark and Zeppelin",
    "StackName": "HDP",
    "StackVersion": "2.6",
    "HostgroupCount": "3",
    "Tags": "DEFAULT"
  },
  {
    "Name": "EDW-ETL: Apache Hive, Apache Spark 2",
    "Description": "Useful for ETL data processing with Hive and Spark",
    "StackName": "HDP",
    "StackVersion": "2.6",
    "HostgroupCount": "3",
    "Tags": "DEFAULT"
  },
  {
    "Name": "Flow Management: Apache NiFi",
    "Description": "Useful for data-flow management with Apache NiFi",
    "StackName": "HDF",
    "StackVersion": "3.1",
    "HostgroupCount": "2",
```

```
    "Tags": "DEFAULT"
  },
  {
    "Name": "my-hdf-test",
    "Description": "",
    "StackName": "HDF",
    "StackVersion": "3.1",
    "HostgroupCount": "3",
    "Tags": "USER_MANAGED"
 }
]
```

cloud availability-zones

Lists all availability zones available in the specified cloud provider region.

Required options

--credential <value> Name of the credential

--region <value> Name of the region

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists availability zones in the us-west-2 (Oregon) region on the AWS account identified by the credential called “aws-cred”:

```
cb cloud availability-zones --credential aws-cred --region us-west-2
[
  {
    "Name": "us-west-2a"
  },
  {
    "Name": "us-west-2b"
  },
  {
    "Name": "us-west-2c"
  }
]
```

cloud regions

Lists the available cloud provider regions.

Required options

--credential <value> Name of the credential

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists regions available on the AWS account identified by the credential called “aws-cred”:

```
cb cloud regions --credential aws-cred
[
  {
    "Name": "ap-northeast-1",
    "Description": "Asia Pacific (Tokyo)"
  },
  {
    "Name": "ap-northeast-2",
    "Description": "Asia Pacific (Seoul)"
  },
  ...
```

cloud volumes

Lists the available cloud provider volume types.

Sub-commands

aws Lists the available aws volume types

azure Lists the available azure volume types

gcp Lists the available gcp volume types

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists volumes available on AWS:

```
cb cloud volumes aws
[
  {
    "Name": "ephemeral",
    "Description": "Ephemeral"
  },
  {
    "Name": "gp2",
    "Description": "General Purpose (SSD)"
  },
  {
    "Name": "st1",
    "Description": "Throughput Optimized HDD"
  },
  {
    "Name": "standard",
    "Description": "Magnetic"
  }
]
```

cloud instances

Lists the available cloud provider instance types.

Required options

--credential <value> Name of the credential

--region <value> Name of the region

Options

--availability-zone <value> Name of the availability zone

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists instance types available in the us-west-2 (Oregon) region on the AWS account identified by the credential called “aws-cred”:

```
cb cloud instances --credential aws-cred --region us-west-2
```

```
  {
    "Name": "c3.2xlarge",
    "Cpu": "8",
    "Memory": "15.0",
    "AvailabilityZone": "us-west-2b"
  },
  {
    "Name": "c3.4xlarge",
    "Cpu": "16",
    "Memory": "30.0",
    "AvailabilityZone": "us-west-2b"
  },
  ...
```

cluster change-ambari-password

Changes Ambari password.

Required options

--name <value> Cluster name

--old-password <value> Old Ambari password

--new-password <value> New Ambari password

--ambari-user <value> Ambari user

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Changes password for Ambari user called “admin” for a cluster called “test1234”:

```
cb cluster change-ambari-password --name test1234 --old-password 123456 --
new-password Ambari123456 --ambari-user admin
```

cluster change-image

Changes default image used for newly added cluster nodes.

Required options

--name <value> Cluster name

--image-id <value> ID of the image that you would like to use as default

Options

--imagecatalog <value> Name of an existing image catalog

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Changes default image for the cluster called "mycluster":

```
cb cluster change-image --name mycluster --imageid 98c75afc-d6fb-417e-7957-
f0c43200f1e4
```

cluster create

Creates a new cluster based on a JSON template.

Required options

--cli-input-json <value> User provided file in JSON format

Options

--name <value> Name for the cluster

--description <value> Description of resource

--input-json-param-password <value> Password for the cluster and Ambari

--wait Wait for the operation to finish. No argument is required

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace where to create the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Creates a cluster called “testcluster” based on a local JSON file called “mytemplate.json” located in the /Users/test/Documents directory:

```
cb cluster create --name testcluster --cli-input-json /Users/test/Documents/
mytemplate.json
```

cluster delete

Deletes an existing cluster.

Required options

--name <value> Cluster name

Options

--force Force the operation

--wait Wait for the operation to finish. No argument is required

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes a cluster called "test1234":

```
cb cluster delete --name test1234
```

cluster describe

Describes an existing cluster.

Required options

--name <value> Cluster name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Returns a JSON file describing an existing cluster called “test1234”:

```
cb cluster describe --name test1234
```

The command returns JSON output which due to space limitation was not captured in the example.

cluster generate-attached-cluster-template

Generates a template for a cluster that can be attached to a data lake.

Required options

--source-cluster <value> Data lake cluster to which the attached cluster will be attached

--blueprint-name <value> Name of the blueprint

Options

--blueprint-file <value> Location of the blueprint file

--cloud-storage <value> Type of the cloud storage [wasb/WASB, adls/ADLS, s3/S3, gcs/GCS]

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Generates a template for a cluster (1) that can be attached to the data lake called "my-datalake" and (2) that uses the test-bp blueprint:

```
cb cluster generate-attached-cluster-template --source-cluster my-datalake
 --blueprint-name test-bp
```

cluster generate-template

Generates a provider-specific cluster template in JSON format.

Sub-commands

aws new-network Generates an AWS cluster JSON template with new network

aws existing-network Generates an AWS cluster JSON template with existing network

aws existing-subnet Generates an AWS cluster JSON template with existing network and subnet

azure new-network Generates an Azure cluster JSON template with new network

azure existing-subnet Generates an Azure cluster JSON template with existing network and subnet

gcp new-network Generates an GCP cluster JSON template with new network

gcp existing-network Generates an GCP cluster JSON template with existing network

gcp existing-subnet Generates an GCP cluster JSON template with existing network and subnet

gcp legacy-network Generates an GCP cluster JSON template with legacy network without subnets

openstack new-network Generates an OS cluster JSON template with new network

openstack existing-network Generates an OS cluster JSON template with existing network

openstack existing-subnet Generates an OS cluster JSON template with existing network and subnet

Required options

None

Options

--blueprint-name <value> Name of the blueprint

--blueprint-file <value> Location of the blueprint file

--cloud-storage <value> Type of the cloud storage [wasb/WASB, adls/ADLS, s3/S3, gcs/GCS]

--with-custom-domain Adds custom domain configuration to the template

--with-tags Adds user-defined tags configuration to the template

--with-image Adds image-catalog configuration to the template

--with-kerberos-managed Adds Cloudbreak managed Kerberos configuration to the template

--with-kerberos-mit Adds existing MIT Kerberos configuration to the template

--with-kerberos-ad Adds existing Active Directory Kerberos configuration to the template

--with-kerberos-custom Adds custom Kerberos configuration to the template

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Options for AWS only:

--with-default-encryption Default encryption for AWS instances which can use a default key

--with-custom-encryption Custom key encryption for AWS instances which can use your custom key

Options for Azure only:

N/A

Options: GCP only:

--with-raw-encryption Custom encryption for GCP instances which can use your raw key

--with-rsa-encryption Custom key encryption for GCP instances which can use your rsa key

--with-kms-encryption Custom key encryption for GCP instances which can use your kms key

Options for OpenStack only:

N/A

Parameters

Template parameters to fill in the generated template:

- userName: Name of the Ambari user

- password: Password of the Ambari user
- name: Name of the cluster
- region: Region of the cluster
- availabilityZone: Availability zone of the cluster, on AZURE it is the same as the region
- blueprintName: Name of the selected blueprint
- credentialName: Name of the selected credential
- instanceGroups.group: Name of the instance group
- instanceGroups.nodeCount: Number of nodes in the group
- instanceGroups.template.instanceType: Name of the selected template
- instanceGroups.template.volumeCount: Number of volumes
- instanceGroups.template.volumeSize: Size of Volumes in GB
- stackAuthentication.publicKey: Public key

Examples

Generates a cluster template for AWS where a new network is created for the cluster:

```
cb cluster generate-template aws new-network
```

cluster generate-reinstall-template

Generates a cluster template that you can use to reinstall the cluster if installation went fail.

Required options

--blueprint-name <value> Name of the blueprint

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Generates a reinstall template that can be later used with the cb cluster reinstall command:

```
cb cluster generate-reinstall-template --blueprint-name "EDW-ETL: Apache
 Hive 1.2.1, Apache Spark 2.1"
```

cluster list

Lists all clusters which are currently associated with the Cloudbreak instance.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists available clusters:

```
cb cluster list
[
  {
    "Name": "test1234",
    "Description": "",
    "CloudPlatform": "AZURE",
    "StackStatus": "UPDATE_IN_PROGRESS",
    "ClusterStatus": "REQUESTED"
  }
]
```

Lists available clusters, with output in a table format:

```
cb cluster list --output table
+----------+-------------+---------------+--------------------
+---------------+
|   NAME   | DESCRIPTION | CLOUDPLATFORM |    STACKSTATUS     |
 CLUSTERSTATUS |
+----------+-------------+---------------+--------------------
+---------------+
| test1234 |             | AZURE         | UPDATE_IN_PROGRESS | REQUESTED
  |
+----------+-------------+---------------+--------------------
+---------------+
```

cluster reinstall

Reinstalls a cluster based on the JSON template generated by the cb cluster reinstall command.

Required options

--name <value> Cluster name

--cli-input-json <value> User-provided file with JSON cluster skeleton

Options

--blueprint-name <value> Name of the blueprint

--kerberos-password <value> Kerberos password

--kerberos-principal <value> Kerberos principal

--wait Wait for the operation to finish. No argument is required

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Reinstalls the cluster called test1234 by using the cluster-skeleton JSON cluster template:

```
cb cluster reinstall --name test1234 --cli-input-json /Users/test/Documents/
cluster-skeleton.JSON
```

cluster repair

Repairs a cluster if cluster installation failed by removing, or removing and replacing failed nodes. You must specify the cluster name and the host group with the failed nodes.

Required options

--name <value> Cluster name

--host-groups <value> Comma separated list of host groups where the failed nodes should be repaired

Options

--remove-only The failed nodes will be removed (rather than “repaired” by removing and replacing)

--wait Wait for the operation to finish. No argument is required

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Removes and replaces failed nodes on worker1 and worker 2 host groups:

```
cb cluster repair --name test1234 --host-groups worker1,worker2
```

Removes failed nodes on worker1 host group without replacing them:

```
cb cluster repair --name test1234 --host-groups worker1 --remove-only
```

cluster retry

Retries the process if cluster or stack provisioning failed.

Required options

--name <value> Cluster name

Options

--wait Wait for the operation to finish. No argument is required

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Retries a cluster called "test1234":

```
cb cluster retry --name test1234
```

cluster scale

Scales a cluster by adding or removing nodes.

Required options

--name <value> Cluster name

--group-name <value> Name of the group to scale

--desired-node-count <value> Desired number of nodes

Options

--wait Wait for the operation to finish. No argument is required

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Scales a cluster called "test1234" by adding 3 nodes to the worker host group:

```
cb cluster scale --name test1234 --group-name worker --desired node-count 3
```

cluster start

Starts a cluster which has previously been stopped.

Required options

--name <value> Cluster name

Options

--wait Wait for the operation to finish. No argument is required

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Starts a cluster called "test1234":

```
cb cluster start --name test1234
```

cluster stop

Stops a cluster.

Required options

--name <value> Cluster name

Options

--wait Wait for the operation to finish. No argument is required

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Stops a cluster called "test1234":

```
cb cluster stop --name test1234
```

cluster sync

Synchronizes a cluster with the cloud provider.

Required options

--name <value> Cluster name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Syncs a cluster called "test1234":

```
cb cluster sync --name test1234
```

configure

Configures the Cloudbreak server address and credentials used to communicate with this server.

Required options

--server <value> Server address [$CB_SERVER_ADDRESS]

--username <value> User name (e-mail address) [$CB_USER_NAME]

Options

--password <value> Password [$CB_PASSWORD]

--workspace <value> Name of an existing workspace

--profile <value> Select a config profile to use [$CB_PROFILE]

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

This example configures the server address with username and password:

```
cb configure --server https://ec2-11-111-111-11.compute-1.amazonaws.com --
username admin@hortonworks.com --password MySecurePassword123
```

This example configures the server address with username but without a password:

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

credential create

Creates a new Cloudbreak credential.

Sub-commands

aws role-based Creates a new AWS credential

aws key-based Creates a new AWS credential

aws-gov role-based Creates a new AWSGov credential

aws-gov key-based Creates a new AWSGov credential

azure app-based Creates a new app-based Azure credential

gcp p12-based Creates a new gcp credential with a P12 key (deprecated)

gcp json-based Creates a new J credential with a JSON key

openstack keystone-v2 Creates a new OpenStack credential

openstack keystone-v3 Creates a new OpenStack credential

Required options

Options required for aws role-based sub-command:

--name <value> Name for the credential

--role-arn <value> IAM Role ARN of the role used for Cloudbreak credential

Options required for aws key-based sub-command:

--name <value> Name for the credential

--access-key <value> AWS Access Key

--secret-key <value> AWS Secret Key

Options required for aws-gov role-based sub-command:

--name <value> Name for the credential

--role-arn <value> IAM Role ARN of the role used for Cloudbreak credential

Options required for aws-gov key-based sub-command:

--name <value> Name for the credential

--access-key <value> AWS Access Key

--secret-key <value> AWS Secret Key

Options required for azure app-based sub-command:

--name <value> Name for the credential

--subscription-id <value> Subscription ID from your Azure Subscriptions

--tenant-id <value> Directory ID from your Azure Active Directory > Properties

--app-id <value> Application ID of your app from your Azure Active Directory > App Registrations

--app-password <value> Your application key from app registration's Settings > Keys

Options required for gcp p12-based sub-command:

--name <value> Name for the credential

--project-id <value> Project ID from your GCP account

--service-account-id <value> Your GCP Service account ID from IAM & Admin > Service accounts

--service-account-private-key-file <value> P12 key from your GCP service account

Options required for gcp json-based sub-command:

--name <value> Name for the credential

--service-account-json-file <value> JSON key from your GCP service account

Options required for openstack keystone-v2 sub-command:

--name <value> Name for the credential

--tenant-user <value> OpenStack user name

--tenant-password <value> OpenStack password

--tenant-name <value> OpenStack tenant name

--endpoint <value> OpenStack endpoint

Options required for openstack keystone-v3 sub-command:

--name <value> Name for the credential

--tenant-user <value> OpenStack user name

--tenant-password <value> OpenStack password

--user-domain <value> OpenStack user domain

--endpoint <value> OpenStack endpoint

Options

--description <value> Description of the resource

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace where to create the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Additionally, the following option is available for OpenStack Keystone2 and Keystone3:

--facing <value> API facing. One of: public, admin, internal

Additionally, the following options are available for OpenStack Keystone3:

--project-domain-name <value> OpenStack project domain name

--project-name <value> OpenStack project name

--domain-name <value> OpenStack domain name

--keystone-scope <value> OpenStack keystone scope. One of: default, domain, project

Examples

Creates a role-based credential on AWS:

```
cb credential create aws role-based --name my-credential1 --role-arn
 arn:aws:iam::517127065441:role/CredentialRole
```

Creates a key-based credential on AWS:

```
cb credential create aws key-based --name my-credential2 --access-key
 ABDVIRDFV3K4HLJ45SKA --secret-key D89L5pOPM+426Rtj3curKzJEJL3lYoNcP8GvguBV
```

Creates a role-based credential on AWSGov:

```
cb credential create aws-gov role-based --name my-credential1 --role-arn
 arn:aws:iam::517127065441:role/CredentialRole
```

Creates a key-based credential on AWSGov:

```
cb credential create aws-gov key-based --name my-credential2 --access-key
 ABDVIRDFV3K4HLJ45SKA --secret-key D89L5pOPM+426Rtj3curKzJEJL3lYoNcP8GvguBV
```

Creates an app-based credential on Azure:

```
cb credential create azure app-based --name my-credential3 --
subscription-id b8e7379e-568g-55d3-na82-45b8d421e998 --tenant-id
 c79n5399-3231-65ba-8dgg-2g4e2a40085e --app-id 6d147d89-48d2-5de2-eef8-
b89775bbfcg1 --app-password 4a8hBgfI52s/C8R5Sea2YHGnBFrD3fRONfdG8w7F2Ua=
```

Creates a credential on Google Cloud by using a key in JSON format :

```
cb credential create gcp json-based --name my-credential4  --service-
account-json-file /Users/test/3fff57a6f68e.json
```

Creates a role-based credential on OpenStack with Keystone-v2:

```
cb credential create openstack keystone-v2 --name my-credential5 --tenant-
user test --tenant-password MySecurePass123 --tenant-name test --endpoint
 http://openstack.test.organization.com:5000/v2.0
```

credential delete

Deletes an existing Cloudbreak credential.

Required options

--name <value> Credential name

Options

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes a credential called "testcred":

```
cb credential delete --name testcred
```

credential describe

Describes an existing credential.

Required options

--name <value> Credential name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Describes a credential called "testcred":

```
cb credential describe --name testcred
{
  "Name": "testcred",
  "Description": "",
  "CloudPlatform": "AZURE",
  "ID": "1702"
}
```

credential list

Lists existing Cloudbreak credentials.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists credentials:

```
cb credential list
[
  {
    "Name": "testcred",
    "Description": "",
    "CloudPlatform": "AZURE"
  }
]
```

Lists credentials, with output formatted in a table format:

```
cb credential list --output table
+---------+-------------+---------------+
|  NAME   | DESCRIPTION | CLOUDPLATFORM |
+---------+-------------+---------------+
| armcred |             | AZURE         |
+---------+-------------+---------------+
```

credential modify

Modifies an existing Cloudbreak credential.

Note:

The --name parameter is used to identify the credential that is being modified, and therefore its value cannot be modified.

Sub-commands

aws role-based Modifies an AWS role-based credential

aws key-based Modifies an AWS key-based credential

aws-gov role-based Modifies an AWS role-based credential

aws-gov key-based Modifies an AWS key-based credential

azure app-based Modifies an app-based Azure credential

gcp p12-based Modifies a Google Cloud credential that uses a P12 key

gcp json-based Modifies a Google Cloud credential that uses a JSON key

openstack keystone-v2 Modifies an OpenStack v2 credential

openstack keystone-v3 Modifies an OpenStack v3 credential

Required options

Options required for aws role-based sub-command:

--name <value> Credential name

--role-arn <value> IAM Role ARN of the role used for Cloudbreak credential

Options required for aws key-based sub-command:

--name <value> Credential name

--access-key <value> AWS Access Key

--secret-key <value> AWS Secret Key

Options required for aws-gov role-based sub-command:

--name <value> Credential name

--role-arn <value> IAM Role ARN of the role used for Cloudbreak credential

Options required for aws-gov key-based sub-command:

--name <value> Credential name

--access-key <value> AWS Access Key

--secret-key <value> AWS Secret Key

Options required for azure app-based sub-command:

--name <value> Credential name

--subscription-id <value> Subscription ID from your Azure Subscriptions

--tenant-id <value> Directory ID from your Azure Active Directory > Properties

--app-id <value> Application ID of your app from your Azure Active Directory > App Registrations

--app-password <value> Your application key from app registration’s Settings > Keys

Options required for gcp p12-based sub-command:

--name <value> Credential name

--project-id <value> Project ID from your GCP account

--service-account-id <value> Your GCP Service account ID from IAM & Admin > Service accounts

--service-account-private-key-file <value> P12 key from your GCP service account

Options required for gcp json-based sub-command:

--name <value> Credential name

--service-account-json-file <value> JSON key from your GCP service account

Options required for openstack keystone-v2 sub-command:

--name <value> Credential name

--tenant-user <value> OpenStack user name

--tenant-password <value> OpenStack password

--tenant-name <value> OpenStack tenant name

--endpoint <value> OpenStack endpoint

Options required for openstack keystone-v3 sub-command:

--name <value> Credential name

--tenant-user <value> OpenStack user name

--tenant-password <value> OpenStack password

--user-domain <value> OpenStack user domain

--endpoint <value> OpenStack endpoint

Options

--description <value> Description of the resource

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Additionally, the following option is available for OpenStack Keystone2 and Keystone3:

--facing <value> API facing. One of: public, admin, internal

Additionally, the following options are available for OpenStack Keystone3:

--project-domain-name <value> OpenStack project domain name

--project-name <value> OpenStack project name

--domain-name <value> OpenStack domain name

--keystone-scope <value> OpenStack keystone scope. One of: default, domain, project

Examples

Modifies a role-based AWS credential:

```
cb credential modify aws role-based --name my-credential1 --role-arn
 arn:aws:iam::517127065441:role/CredentialRole
```

Modifies a key-based AWS credential:

```
cb credential modify aws key-based --name my-credential2 --access-key
 ABDVIRDFV3K4HLJ45SKA --secret-key D89L5pOPM+426Rtj3curKzJEJL3lYoNcP8GvguBV
```

Modifies a role-based AWSGov credential:

```
cb credential modify aws-gov role-based --name my-credential1 --role-arn
 arn:aws:iam::517127065441:role/CredentialRole
```

Modifies a key-based AWSGov credential:

```
cb credential modify aws-gov key-based --name my-credential2 --access-key
 ABDVIRDFV3K4HLJ45SKA --secret-key D89L5pOPM+426Rtj3curKzJEJL3lYoNcP8GvguBV
```

Modifies an app-based Azure credential:

```
cb credential modify azure app-based --name my-credential3 --
subscription-id b8e7379e-568g-55d3-na82-45b8d421e998 --tenant-id
```

```
 c79n5399-3231-65ba-8dgg-2g4e2a40085e --app-id 6d147d89-48d2-5de2-eef8-
b89775bbfcg1 --app-password 4a8hBgfI52s/C8R5Sea2YHGnBFrD3fRONfdG8w7F2Ua=
```

Modifies a Google Cloud credential that uses a P12 key:

```
cb credential modify gcp --name my-credential4 --project-id test-proj --
service-account-id test@test-proj.iam.gserviceaccount.com --service-account-
private-key-file /Users/test/3fff57a6f68e.p12
```

Modifies a Google Cloud credential that uses a JSON key:

```
cb credential modify gcp --name my-credential5 --service-account-json-file /
Users/test/3fff57a6f68e.json
```

Modifies a role-based OpenStack credential which uses Keystone-v2:

```
cb credential modify openstack keystone-v2 --name my-credential5 --tenant-
user test --tenant-password MySecurePass123 --tenant-name test --endpoint
 http://openstack.test.organization.com:5000/v2.0
```

database create

Registers an existing external database with Cloudbreak.

Sub-commands

mysql Registers a MySQL database configuration

oracle11 Registers an Oracle 11 database configuration

oracle12 Registers an Oracle 12 database configuration

postgres Registers a Postgres database configuration

Required options

--name <value> Name for the database

--db-username <value> Username for the JDBC connection

--db-password <value> Password for the JDBC connection

--url <value> JDBC connection URL in the form of jdbc:db-type://address:port/db

--type <value> Name if the service that will use the database (AMBARI, DRUID, HIVE, OOZIE, RANGER, SUPERSET, or other custom type)

If using MySQL and Oracle, the --connector-jar-url value <value> parameter is required in all cases except the following: If you are using a custom image and you already placed the JAR file on the machine, then this parameter is not required.

Options

--description <value> Description for the database

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Registers an existing Postgres database called “test-postgres” with Cloudbreak:

```
cb database create postgres --name testpostgres  --type HIVE --url
 jdbc:postgresql://test-db.cic6nusrpqec.us-west-2.rds.amazonaws.com:5432/
testdb --db-username testuser --db-password MySecurePassword123
```

The connection URL includes three components db-type://address:port/db:

- Database type “jdbc:postgresql”
- Endpoint “test-db.cic6nusrpqec.us-west-2.rds.amazonaws.com:5432”
- Port “5432”
- Database name “testdb”

Registers an existing MySQL database called “testmysql” with Cloudbreak:

```
cb database create mysql --name testmysql --type OOZIE --url jdbc:mysql://
test-db.cic6nusrpqec.us-west-2.rds.amazonaws.com:5432/testdb  --db-username
 test --db-password test --connector-jar-url http://example-page/driver-
file.JAR
```

database delete

Unregisters a previously registered database with Cloudbreak. It does not delete the database instance.

Required options

--name <value> Database registration name

Options

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes a database called "testdatabase":

```
cb database delete --name testdatabase
```

database list

Lists all available database registrations.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists existing database registrations:

```
cb database list
```

Lists existing database registrations, with output presented in a table format:

```
cb database list --output table
```

database test

Test database connection.

Sub-commands

by-name Tests a stored database configuration identified by its name

by-params Tests database connection parameters

Required options

Options required for by-name sub-command:

--name <value> Database registration name

Options required for by-params sub-command:

--db-username <value> Username to use for the JDBC connection

--db-password <value> Password to use for the JDBC connection

--url <value> JDBC connection URL in the form of jdbc:db-type://address:port/db

--type <value> Type of database (the service name that will use the database

Options

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Tests connection to a previously registered database called “testpostgres”:

```
database test --name testpostgres
```

Tests connection to a database based on connection parameters provided:

```
cb database test by-params --type HIVE --url jdbc:postgresql://test-
db.cic6nusrpqec.us-west-2.rds.amazonaws.com:5432/testdb --db-username
 testuser --db-password MySecurePassword123
```

imagecatalog create

Registers a new custom image catalog based on the URL provided.

Required options

--name <value> Name for the image catalog

--url <value> URL location of the image catalog JSON file

Options

--description <value> Description for the recipe

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace where to create the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Registers an image catalog called “mycustomcatalog” which is available at https://example.com/myimagecatalog.json:

```
cb imagecatalog create --name mycustomcatalog --url https://example.com/
myimagecatalog.json
```

imagecatalog delete

Deletes a previously registered custom image catalog. It does not delete any cloud provider resources that you created as a prerequisite for creating the Cloudbreak credential.

Required options

--name <value> Image catalog name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes an image catalog called “mycustomcatalog”:

```
cb imagecatalog delete --name mycustomcatalog
```

imagecatalog describe

Displays details of a specific image catalog.

Required options

--name <value> Name of an image catalog

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Displays details of the cloudbreak-default image catalog:

```
cb imagecatalog describe --name cloudbreak-default
{
  "Name": "cloudbreak-default",
  "Default": true,
  "URL": "https://cloudbreak-imagecatalog.s3.amazonaws.com/v2-rc-cb-image-
catalog.json"
}
```

imagecatalog images

Lists images from the specified image catalog available for the specified cloud provider.

Sub-commands

describe aws Provides detailed information about an AWS image

describe azure Provides detailed information about an Azure image

describe gcp Provides detailed information about a GCP image

describe openstack Provides detailed information about an OpenStack image

aws Lists available aws images

azure Lists available azure images

gcp Lists available gcp images

openstack Lists available openstack images

cluster-upgrade Lists images that are valid for upgrading the default image of the cluster

Required options

--imagecatalog <value> Name of the imagecatalog

--imageid <value> ID of the image (Only for describe commands)

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Returns date, description, Ambari version, and image ID for all AWS images from an image catalog called “myimagecatalog”:

```
cb imagecatalog images aws --imagecatalog cloudbreak-default
[
  {
    "Date": "2017-10-13",
    "Description": "Cloudbreak official base image",
    "Version": "2.6.0.0",
    "ImageID": "44b140a4-bd0b-457d-b174-e988bee3ca47"
  },
  {
    "Date": "2017-11-16",
    "Description": "Official Cloudbreak image",
    "Version": "2.6.0.0",
    "ImageID": "3c7598a4-ebd6-4a02-5638-882f5c7f7add"
  }
]
```

Returns date, description, Ambari version, image ID, OS, OS type, image names in provider regions and package versions for the specified GCP image from an image catalog called “cloudbreak-default”:

```
cb imagecatalog images describe gcp --imagecatalog cloudbreak-default --
imageid 4c05c0b8-c3f8-461d-5bdc-97b0f5324ee3
[
  {
    "Date": "2018-07-19",
    "Description": "Official Cloudbreak image",
    "Version": "2.6.2.2",
    "ImageID": "4c05c0b8-c3f8-461d-5bdc-97b0f5324ee3",
    "OS": "centos7",
    "OSType": "redhat7",
    "Images": {
      "gcp": {
        "default": "sequenceiqimage/cb-hdp--1807191752.tar.gz"
      }
    },
    "PackageVersions": {
      "kernel": "3.10.0-123.el7,3.10.0-327.36.1.el7,3.10.0-862.9.1.el7",
      "python": "2.7.5-69.el7_5",
      "salt": "2017.7.5-1.el7",
      "salt-bootstrap": "0.13.0-2018-05-03T07:39:07"
    }
  }
]
```

List the available images you can upgrade your cluster to:

```
cb imagecatalog images cluster-upgrade --cluster test-cluster
```

imagecatalog list

Lists default and custom image catalogs registered with Cloudbreak instance.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists existing image catalogs:

```
cb imagecatalog list
[
  {
```

```
    "Name": "mycustomcatalog",
    "Default": false,
    "URL": "https://example.com/imagecatalog.json"
  },
  {
    "Name": "cloudbreak-default",
    "Default": true,
    "URL": "https://s3-eu-west-1.amazonaws.com/cloudbreak-info/v2-dev-cb-
image-catalog.json"
  }
]
```

imagecatalog set-default

Sets the specified image catalog as default.

Required options

--name <value> Image catalog name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Sets “mycustomcatalog” as default:

```
imagecatalog set-default --name mycustomcatalog
```

ldap create

Registers an existing LDAP with Cloudbreak.

Required options

--name <value> Name for the LDAP

--ldap-server <value> Address of the LDAP server (e.g. ldap://10.0.0.1:384)

--ldap-domain <value> LDAP domain (e.g. ad.cb.com)

--ldap-bind-dn <value> LDAP bind DN (e.g. CN=Administrator,CN=Users,DC=ad,DC=cb,DC=com)

--ldap-bind-password <value> LDAP bind password

--ldap-directory-type <value> LDAP directory type (LDAP or ACTIVE_DIRECTORY)

--ldap-user-search-base <value> LDAP user search base (e.g. CN=Users,DC=ad,DC=cb,DC=com)

--ldap-user-name-attribute <value> LDAP user name attribute

--ldap-user-object-class <value> LDAP user object class

--ldap-group-member-attribute <value> LDAP group member attribute

--ldap-group-name-attribute <value> LDAP group name attribute

--ldap-group-object-class <value> LDAP group object class

--ldap-group-search-base <value> LDAP group search base (e.g. OU=scopes,DC=ad,DC=cb,DC=com)

Options

--ldap-admin-group <value> LDAP group of administrators

--description <value> Description for the LDAP

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace where to create the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

ldap delete

Deletes selected LDAP registration from Cloudbreak. It does not delete the LDAP.

Required options

--name <value> LDAP name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes the LDAP called "testldap" from Cloudbreak:

```
cb ldap delete --name testldap
```

ldap list

Lists all available LDAPs.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists existing LDAPs:

```
cb ldap list
```

Lists existing LDAPs, with output presented in a table format:

```
cb ldap list --output table
```

mpack create

Registers an existing management pack with Cloudbreak.

Required options

--name <value> Name for the mpack

--url <value> URL that points to the location of the mpack tarball

Options

--purge Purge existing resources specified in purge-list

--purge-list <value> Comma-separated list of resources to purge (stack-definitions,service-definitions,mpacks). By default (stack-definitions,mpacks) will be purged

--force Force install management pack

--description <value> Description for the LDAP

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace where to create the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Registers a new mpack without purging:

```
cb mpack create --name test-hdp-search --url http://public-
repo-1.hortonworks.com/HDP-SOLR/hdp-solr-ambari-mp/solr-service-
mpack-3.0.0.tar.gz
```

mpack delete

Deletes selected management registration from Cloudbreak. It does not delete the management pack.

Required options

--name <value> Management pack name

Options

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes an mpack called "testmpack":

```
cb mpack delete --name testmpack
```

mpack list

Lists all available management packs.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists all currently registered management packs and provides information about each:

```
cb mpack list
[
  {
    "Name": "hdp-search-3",
    "Description": "",
    "URL": "http://public-repo-1.hortonworks.com/HDP-SOLR/hdp-solr-ambari-
mp/solr-service-mpack-3.0.0.tar.gz",
    "Purge": "false",
    "PurgeList": "",
    "Force": "false"
  }
]
```

proxy create

Registers an existing proxy with Cloudbreak.

Required options

--name <value> Name for the proxy

--proxy-host <value> Hostname or IP address of the proxy

--proxy-port <value> Port of the proxy

Options

--proxy-protocol <value> Protocol for the proxy (http or https) (default: “http”)

--proxy-user <value> User for the proxy if basic auth is required

--proxy-password <value> Password for the proxy if basic auth is required

--description <value> Description for the proxy

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

proxy delete

Unregisters a previously registered proxy with Cloudbreak. It does not delete the proxy.

Required options

--name <value> Proxy registration name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes a proxy registration called "testproxy":

```
cb proxy delete --name testproxy
```

proxy list

Lists all proxies that were previously registered with Cloudbreak.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists existing proxy registrations:

```
cb proxy list
```

Lists existing proxy registrations, with output presented in a table format:

```
cb proxy list --output table
```

recipe create

Adds a new recipe from a file or from a URL.

Sub-commands

from-url Creates a recipe by downloading it from a URL location

from-file Creates a recipe by reading it from a local file

Required options

Options required for from-url sub-command:

--name <value> Name for the recipe

--execution-type <value> Type of execution [pre-ambari-start, pre-termination, post-ambari-start, post-cluster-install]

--url <value> URL location of the Ambari blueprint JSON file

Options required for from-file sub-command:

--name <value> Name for the recipe

--execution-type <value> Type of execution [pre-ambari-start, pre-termination, post-ambari-start, post-cluster-install]

--file <value> Location of the Ambari blueprint JSON file

Options

--description <value> Description for the recipe

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Adds a new recipe called “test1” from a URL:

```
cb recipe create from-url --name "test1" --execution-type post-ambari-start
 --url http://some-site.com/test.sh
```

Adds a new recipe called “test2” from a file:

```
cb recipe create from-url --name "test2" --execution-type post-ambari-start
 --file /Users/test/Documents/test.sh
```

recipe delete

Deletes an existing recipe.

Required options

--name <value> Recipe name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes a recipe called "test-recipe":

```
cb recipe delete --name test-recipe
```

recipe describe

Describes an existing recipe.

Required options

--name <value> Recipe name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Describes a recipe called “test”:

```
cb recipe describe --name test
{
  "Name": "test",
  "Description": "",
  "ExecutionType": "POST"
}
```

Describes a recipe called “test”, with output presented in a table format:

```
cb describe-recipe --name test --output table
+------+-------------+----------------+------+
| NAME | DESCRIPTION | EXECUTION TYPE |  ID  |
+------+-------------+----------------+------+
| test |             | POST           | 1810 |
+------+-------------+----------------+------+
```

recipe list

Lists all available recipes.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists existing recipes:

```
cb recipe list
[
  {
    "Name": "test",
    "Description": "",
    "ExecutionType": "POST"
  }
]
```

Lists existing recipes, with output presented in a table format:

```
cb recipe list --output table
+------+-------------+-------------------+
| NAME | DESCRIPTION | EXECUTION TYPE    |
+------+-------------+-------------------+
| test |             | POST-AMBARI-START |
+------+-------------+-------------------+
```

user list

Lists all LDAP/AD users.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists all LDAP/AD users:

```
cb user list
[
  {
    "User": {
      "id": 20,
      "userId": "test1@example.com"
    }
  },
  {
    "User": {
      "id": 21,
      "userId": "test2@example.com"
    }
  }
]
```

workspace add user

Adds user to a workspace and assigns access permissions.

Sub-commands:

read Assigns read access permission to the user

read-write Assigns read and write access permission to the user

Required options

--name <value> Name of the workspace

--user-id <value> ID of the user

Options

--description <value> Description for the user

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Adds jdean@example.com to a workspace called “marketing” with read-write access permissions:

```
cb workspace add-user read-write --name "marketing" --user-id
 "jdean@example.com"
```

workspace create

Creates a new workspace.

Required options

--name <value> Name for the workspace

Options

--description <value> Description for the recipe

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace where to create the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Adds a new workspace called “engineering”:

```
cb workspace create --name "engineering"
```

workspace delete

Deletes an existing workspace.

Required options

--name <value> Workspace name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Deletes a workspace called "marketing-data":

```
cb workspace delete --name marketing-data
```

workspace describe

Provides detailed information about an existing workspace.

Required options

--name <value> Workspace name

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Describes a workspace called “marketing-data”:

```
cb org describe --name marketing-data
```

workspace list

Lists all workspaces that the user running the command is a member of.

Required options

None

Options

--output <value> Supported formats: json, yaml, table (default: “json”) [$CB_OUT_FORMAT]

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Lists all workspaces that the user running the command is a member of:

```
cb workspace list
```

workspace remove-user

Removes a user from a workspace.

Required options

--name <value> Name of the workspace

--user-id <value> ID of the user

Options

--server <value> Cloudbreak server address [$CB_SERVER_ADDRESS]

--username <value> Cloudbreak user name (e-mail address) [$CB_USER_NAME]

--password <value> Cloudbreak password [$CB_PASSWORD]

--workspace <value> Name of the workspace holding the resource

--profile <value> Selects a config profile to use [$CB_PROFILE]

--auth-type <value> Authentication method to use. Values: oauth2, basic [$CB_AUTH_TYPE]

Examples

Removes the user jdean@example.com from a workspace called "test-org":

```
cb workspace remove-user --name "test-org" --user-id "jdean@example.com"
```

Debugging

To use debugging mode, pass the --debug option.

Checking CLI Version

To check CLI version, use cb --version.