Advanced Cloudbreak Configuration

Date of Publish: 2019-02-06

https://docs.hortonworks.com/

Contents

External Cloudbreak database

Refer to this section if you would like to configure an external database for Cloudbreak. This is required for all production deployments.

By default, Cloudbreak uses an embedded PostgreSQL database to persist data related to Cloudbreak configuration, setup, and so on. For a production deployment, you must configure an external database. For supported databases and configuration steps, refer to the following documentation:

Supported databases

An embedded PostgreSQL 9.6.1 database is used by Cloudbreak by default. If you would like to use an external database for Cloudbreak, you may use one of the supported database types and versions.

The following database types and versions are supported:

Database type	Supported version
External PostgreSQL	9.6.1 or above
External MySQL	Not supported
External MariaDB	Not supported
External Oracle	Not supported
External SQL Server	Not supported

Configure external Cloudbreak database

Perform these steps to configure Cloudbreak with an existing external database, other than the embedded PostgreSQL database instance that Cloudbreak uses by default.

Steps

1. On your Cloudbreak host machine, set the following environment variables according to the settings of your external database:

```
export DATABASE_HOST=my.database.host
export DATABASE_PORT=5432
export DATABASE_USERNAME=admin
export DATABASE_PASSWORD=Admin123!
```

2. On your external database, create three databases: cbdb, uaadb, periscopedb. You can create these databases using the createdb utility with the following commands:

```
createdb -h $DATABASE_HOST -p $DATABASE_PORT -U $DATABASE_USERNAME cbdb
createdb -h $DATABASE_HOST -p $DATABASE_PORT -U $DATABASE_USERNAME uaadb
createdb -h $DATABASE_HOST -p $DATABASE_PORT -U $DATABASE_USERNAME
 periscopedb
```

For more information related to the "createdb" command refer to the PostgreSQL documentation.

Note:

Alternatively, you can log in to the management interface of your external database and execute "create database" commands directly. Refer to PostgreSQL documentation for more information.

3. Set the following variables in your Cloudbreak Profile file. Modify the database parameters according to your external database.

```
export DATABASE_HOST=my.database.host
export DATABASE_PORT=5432
export DATABASE_USERNAME=admin
export DATABASE_PASSWORD=Admin123!

export CB_DB_PORT_5432_TCP_ADDR=$DATABASE_HOST
export CB_DB_PORT_5432_TCP_PORT=$DATABASE_PORT
export CB_DB_ENV_USER=$DATABASE_USERNAME
export CB_DB_ENV_PASS=$DATABASE_PASSWORD
export CB_DB_ENV_DB=cbdb

export PERISCOPE_DB_PORT_5432_TCP_ADDR=$DATABASE_HOST
export PERISCOPE_DB_PORT_5432_TCP_PORT=$DATABASE_PORT
export PERISCOPE_DB_ENV_USER=$DATABASE_USERNAME
export PERISCOPE_DB_ENV_PASS=$DATABASE_PASSWORD
export PERISCOPE_DB_ENV_DB=periscopedb
export PERISCOPE_DB_ENV_SCHEMA=public

export IDENTITY_DB_URL=$DATABASE_HOST:$DATABASE_PORT
export IDENTITY_DB_USER=$DATABASE_USERNAME
export IDENTITY_DB_PASS=$DATABASE_PASSWORD
export IDENTITY_DB_NAME=uaadb
```

4. Restart Cloudbreak application by using the cbd restart command.

After performing these steps, your external database will be used for Cloudbreak instead of the built-in database.

Postrequisites

If your external database uses SSL, you must also perform the steps described in Configure an SSL certificate for an external Cloudbreak database.

If you would like to migrate your existing data (such as blueprints, recipes, and so on) from the embedded database to the external one, then after completing these steps, you should also create a backup of your original database and then restore it in the external database.

Related Information

PostgreSQL: createdb

PostgreSQL: create database

Configure an SSL certificate for an external Cloudbreak database

Configure an SSL certificate for an external Cloudbreak database

Perform these steps to configure Cloudbreak with an existing external database that uses SSL.

Prerequisites

Configure an existing external database for Cloudbreak as described in Configure external Cloudbreak database. Once done, perform the following steps.

Steps

1. Obtain your database's SSL certificate:
   - If your database instance runs on AWS, obtain the certificate from the link provided in the following AWS documentation: Using SSL with a PostgreSQL DB Instance.
   - If your database instance runs on Azure, obtain the certificate from the link provided in the following Azure documentation: Configure SSL connectivity in Azure Database for PostgreSQL.

- If your database instance runs on GCP, you should provide your existing certificate or create a new certificate as described in the following GCP documentation: Connect to your Cloud SQL instance without encryption.

2. Access your Cloudbreak VM via SSH.
3. Download or copy the certificate to the certs directory in your Cloudbreak deployment directory. Ny default, this is /var/lib/cloudbreak-deployment/certs.
4. Set the following variables in your Profile file:

Variable	Description
PERISCOPE_DB_ENV_SSL	Default false. Set to true to enable SSL.
PERISCOPE_DB_ENV_CERT_FILE	Default empty. Set this to the location of your certificate relative to your certs directory within the Cloudbreak deployment directory. For example if your certificate is in /var/lib/cloudbreak-deployment/certs/root.crt set this variable to root.crt.
CB_DB_ENV_SSL	Default false. Set to true to enable SSL.
CB_DB_ENV_CERT_FILE	Default empty. Set this to the location of your certificate relative to your certs directory within the Cloudbreak deployment directory. For example if your certificate is in /var/lib/cloudbreak-deployment/certs/root.crt set this variable to root.crt.

This example assumes that root.crt is the name of the certificate file:

```
export PERISCOPE_DB_ENV_SSL=true
export PERISCOPE_DB_ENV_CERT_FILE-=root.crt
export CB_DB_ENV_SSL=true
export CB_DB_ENV_CERT_FILE=root.crt
```

5. Next, you should make changes in the uaa.yml file located in the Cloudbreak deployment directory. To make changes in the uaa.yml file, you should create a new file called uaa-changes.yml; This file will be used to regenerate your database settings in the uaa.yml file. The steps are:

   a. Open your uaa.yml file and copy the entire “database” entry (which should include values such as driverClassName, maxactive, password, url, and username).
   b. In the Cloudbreak deployment directory, create a new file called uaa-changes.yml.
   c. Paste the copied content to the uaa-changes.yml file.
   d. Update the value of the “url” by adding the following (replacing root.crt with the actual name of your certificate file):

```
?
ssl=true&sslfactory=org.postgresql.ssl.SingleCertValidatingFactory&sslfactoryarg=fi
certs/root.crt
```

   e. Save the uaa-changes.yml file.

   After the update your uaa-changes.yml file should look similar to:

```
database:
  driverClassName: org.postgresql.Driver
  maxactive: 200
  password: ${IDENTITY_DB_PASS}
  url: jdbc:postgresql://${IDENTITY_DB_URL}/${IDENTITY_DB_NAME}?
ssl=true&sslfactory=org.postgresql.ssl.SingleCertValidatingFactory&sslfactoryarg=file
certs/root.crt
  username: ${IDENTITY_DB_USER}
```

6. Run the following to stop cbd:

```
cbd kill
```

7. Run the following to regenerate the uaa.yml file with the SSL information provided earlier:

```
cbd regenerate
```

8. Run the following to start Cloudbreak:

```
cbd start
```

Cloudbreak may take a few minutes to start.

Related Information

Configure external Cloudbreak database
Using SSL with a PostgreSQL DB Instance (AWS)
Configure SSL connectivity in Azure Database for PostgreSQL (Azure)
Connect to your Cloud SQL instance without encryption (GCP)

LDAP/AD for Cloudbreak

Refer to this section if you would like to configure Cloudbreak to use an LDAP/AD.

By default Cloudbreak uses an internal system as the user store for authentication (enabled by using CloudFoundry UAA). If you would like to configure LDAP or Active Directory (AD) external authentication, you must:

1. Collect the information about your LDAP/AD setup.
2. Configure Cloudbreak to work with that LDAP/AD setup.

LDAP/AD information

Review this section to determine what information related to your existing LDAP/AD you should provide to Cloudbreak.

In order to use the LDAP/AD with Cloudbreak, you must provide the information related to your existing LDAP/AD. The following table details the properties and values that you need to know about your LDAP/AD environment in order to use the LDAP/AD with Cloudbreak:

Parameter	Description	Example
base		
url	The LDAP url with port	ldap://10.0.3.128:389/
userDn	Enter the root Distinguished Name to search in the directory for users.	cn=Administrator,ou=srv,dc=hortonworks,dc=local
password	Enter your root Distinguished Name password.	MyPassword1234!
searchBase	Enter your LDAP user search base. This defines the location in the directory from which the LDAP search begins.	ou=Users,dc=hortonworks,dc=local
searchFilter	Enter the attribute for which to conduct a search on the user base.	mail={0}
groups		
searchBase	Enter your LDAP group search base. This defines the location in the directory from which the LDAP search begins.	ou=Groups,dc=hortonworks,dc=local
groupSearchFilter	Enter the attribute for which to conduct a search on the group base.	member={0}

Configuring Cloudbreak for LDAP/AD

There are two parts to configuring Cloudbreak for LDAP/AD: configuring LDAP/AD user authentication for Cloudbreak and configuring LDAP/AD group authorization for Cloudbreak.

Configure user authentication

After obtaining your LDAP/AD information, configure LDAP/AD user authentication for Cloudbreak.

Steps

1. On the Cloudbreak host, browse to /var/lib/cloudbreak-deployment.
2. Create a new yml file called uaa-changes.yml.
3. In the yml file enter the following using your LDAP/AD information.

```
spring_profiles: postgresql,ldap

ldap:
  profile:
    file: ldap/ldap-search-and-bind.xml
  base:
    url: ldap://10.0.3.138:389
    userDn: cn=Administrator,ou=srv,dc=hortonworks,dc=local
    password: 'mypassword'
    searchBase: ou=Users,dc=hortonworks,dc=local
    searchFilter: mail={0}
  groups:
    file: ldap/ldap-groups-map-to-scopes.xml
    searchBase: ou=Groups,dc=hortonworks,dc=local
    searchSubtree: false
    maxSearchDepth: 1
    groupSearchFilter: member={0}
    autoAdd: true
```

   If using LDAPS, use an LDAPS URL such as ldaps://10.0.3.138:636.
4. Save the file and restart Cloudbreak.

Troubleshooting

If you are using LDAPS and the authentication is not working, check the logs of the identity service on the Cloudbreak host:

```
cbd logs identity
```

A message similar to the following means UAA could not connect to the LDAP server because it could not validate its certificate:

```
Caused by: sun.security.validator.ValidatorException: PKIX path building
 failed: sun.security.provider.certpath.SunCertPathBuilderException: unable
 to find valid certification path to requested target
```

To resolve this issue, turn off certificate validation by adding the following lines under the ldap attribute in the uaa-changes.yml file:

```
ssl:
    skipverification: true
```

Next, save the file and restart Cloudbreak.

Related Information

LDAP/AD information

Configure group authorization

Once user authentication is configured, you should configure which group(s) can access Cloudbreak.

Users (once authenticated) will be granted permission to access Cloudbreak and use the capabilities of Cloudbreak based on their group member. The following describes how to create (i.e. execute-and-map) a group authorization and how to remove (i.e. delete-mapping) an authorization. You should select one set of instructions that is appropriate for your use case:

- If using the default embedded PostgreSQL database for Cloudbreak, perform the steps under "Embedded database".
- If using an external database instance for Cloudbreak databases, perform the steps under "External database".

Embedded database

Use these steps if you are using the default embedded PostgreSQL database for Cloudbreak.

To create a group authorization, execute the following (for example: to add “Analysts” group):

```
cbd util execute-ldap-mapping cn=Analysts,ou=Groups,dc=hortonworks,dc=local
```

To remove a group authorization, execute the following (for example: to remove “Analysts” group):

```
cbd util delete-ldap-mapping cn=Analysts,ou=Groups,dc=hortonworks,dc=local
```

External database

Use these steps if you are using an external database instance for Cloudbreak databases.

To create a group authorization:

1. Connect to the external database instance.
2. Select the uaadb database.
3. Compose a valid INSERT statement by replacing [$REPLACE-WITH-REAL-DATA] with a correct group identifier:

```
INSERT INTO external_group_mapping (group_id, external_group, added,
 origin)
SELECT id, 'CN=[$REPLACE-WITH-REAL-DATA],OU=[$REPLACE-WITH-
REAL-DATA], DC=[$REPLACE-WITH-REAL-DATA],DC=[$REPLACE-WITH-
REAL-DATA],DC=com', current_timestamp, 'ldap' from groups where
 displayname like 'cloudbreak%' or displayname like 'periscope%' or
 displayname='sequenceiq.cloudbreak.user';
```

4. Execute the INSERT statement.

To remove a group authorization:

1. Connect to the external database instance.
2. Select the uaadb database.
3. Compose a valid DELETE statement by replacing [$REPLACE-WITH-REAL-DATA] with a correct group identifier:

```
DELETE FROM external_group_mapping external_group = 'CN=[$REPLACE-
WITH-REAL-DATA],OU=[$REPLACE-WITH-REAL-DATA], DC=[$REPLACE-WITH-REAL-
DATA],DC=[$REPLACE-WITH-REAL-DATA],DC=com';
```

4. Execute the DELETE statement.

Outbound internet access and proxy

This section provides information on the outbound network destinations for Cloudbreak, and instructions on how to configure Cloudbreak to use a proxy for outbound access (if required).

Depending on your enterprise requirements, you may have limited or restricted outbound network access and/or require the use of an internet proxy. Installing and configuring Cloudbreak, as well as creating cloud resources and clusters on those resources requires outbound network access to certain destinations, and in some cases must go through a proxy.

Scenario	Documentation
My environment has limited outbound internet access	Refer to the documentation related to outbound network access destinations for information on network rules.
My environment requires use of a proxy for outbound internet access	Refer to the documentation for using a proxy for information on using a proxy with Cloudbreak.

Outbound network access destinations

Review this section to find out which specific outbound destinations must be available in order to install and configure Cloudbreak in an environment with limited outbound network access.

To install and configure Cloudbreak, specific outbound destinations must be available. The following outbound destinations must be available:

Destination	Description
*.docker.io	Obtain the Docker images for Cloudbreak.
• raw.githubusercontent.com • github.com • s3.amazonaws.com • *.cloudfront.net	Obtain Cloudbreak dependencies.
cloudbreak-imagecatalog.s3.amazonaws.com	The default Cloudbreak image catalog used for VMs. Refer to Custom images documentation for more information on image catalogs.

Once Cloudbreak is installed and configured, you will need the following outbound destinations available in order to communicate with the cloud provider APIs to obtain cloud resources for clusters.

Cloud provider	Cloud provider API destinations
Amazon Web Services	*.amazonaws.com
Microsoft Azure	• *.microsoftonline.com • *.windows.net • *.azure.com
Google Cloud Platform	• accounts.google.com • *.googleapis.com

To install the cluster software, you can:

a) use the public hosted repositories provided by Hortonworks, or

b) specify your own local hosted repositories when you create a cluster.

If you choose to (a) use the public hosted repositories, be sure to allow outbound access to the following destinations:

- private-repo-1.hortonworks.com
- public-repo-1.hortonworks.com

Related Information

Using custom images

Using a proxy

Refer to this section if your environment requires all internet traffic to go through an internet proxy.

In some cases, your environment requires all internet traffic to go through an internet proxy. This section describes:

- How to set up Cloudbreak to use a proxy
- How to configure your cluster hosts to use a proxy

Configure Cloudbreak to use a proxy

Use the following steps if you would like to set up Cloudbreak to use your proxy.

Steps

1. After downloading and installing Cloudbreak, configure the Docker daemon to use proxy by adding the following to the Docker service file:

```
Environment="HTTP_PROXY=http://my-proxy-host:my-proxy-port"
 "NO_PROXY=localhost,127.0.0.1"
```

   For example:

```
vi /etc/systemd/system/docker.service ->
 Environment="HTTP_PROXY=http://10.0.2.237:3128"
 "NO_PROXY=localhost,127.0.0.1"
```

   For more information refer to Docker documentation.
2. Ensure that ports 9443 and 8443 are handled as SSL connections in the proxy config.
3. Configure proxy settings in the Profile file by setting the following variables:

```
HTTP_PROXY_HOST=your-proxy-host
HTTPS_PROXY_HOST=your-proxy-host
PROXY_PORT=your-proxy-port
PROXY_USER=your-proxy-user
PROXY_PASSWORD=your-proxy-password
#NON_PROXY_HOSTS
#HTTPS_PROXYFORCLUSTERCONNECTION=false
```

   For example:

```
HTTP_PROXY_HOST=10.0.2.237
HTTPS_PROXY_HOST=10.0.2.237
PROXY_PORT=3128
PROXY_USER=squid
PROXY_PASSWORD=squid
#NON_PROXY_HOSTS
#HTTPS_PROXYFORCLUSTERCONNECTION=false
```

Related Information

Control Docker with systemdb (Docker)

Configuring clusters to use a proxy

Use the following guidelines to find out how to set up your clusters to use a proxy.

What base image are you using?	Where are the platform repositories?	What to do
Default	Public	Use Register a proxy for clusters
Default	Local	Use Register a proxy for clusters
Custom	Public	Set up the proxy on your custom image OR use Register a proxy for clusters.
Custom	Local	Not required. Skip this section.

You can define a proxy configuration as an external source in Cloudbreak web UI or CLI, and then (optionally) specify to configure that proxy configuration on the hosts that are part of the cluster during cluster create. Refer to Register a proxy for clusters section below for more information.

Register a proxy for clusters

Cloudbreak allows you to save your existing proxy configuration information as an external source so that you can provide the proxy information to multiple clusters that you create with Cloudbreak. The steps are:

1. Register your proxy in Cloudbreak web UI or CLI.
2. Once the proxy has been registered with Cloudbreak, it will show up in the list of available proxies when creating a cluster under advanced External Sources > Configure Proxy. You should select it for every cluster that you create with Cloudbreak.

Steps

1. From the navigation pane, select External Sources > Proxy Configurations.
2. Select Register Proxy Configuration.
3. Provide the following information:

Parameter	Description	Example
Name	Enter the name to use when registering this database to Cloudbreak. This is not the database name.	my-proxy
Description	Provide description.	
Protocol	Select HTTP or HTTPS.	HTTPS
Server Host	Enter the URL of your proxy server host.	10.0.2.237
Server Port	Enter proxy server port.	3128
Username	Enter the username for the proxy.	testuser
Password	Enter the password for the proxy.	MyPassword123

4. Click REGISTER to save the configuration.
5. The proxy will now show up when creating a cluster under advanced External Sources > Configure Proxy. You should select it each time you create a cluster.

Advanced proxy setup scenarios

In some cases, Cloudbreak using the proxy might vary depending on your Cloudbreak and cluster deployment.

This section describes two scenarios:

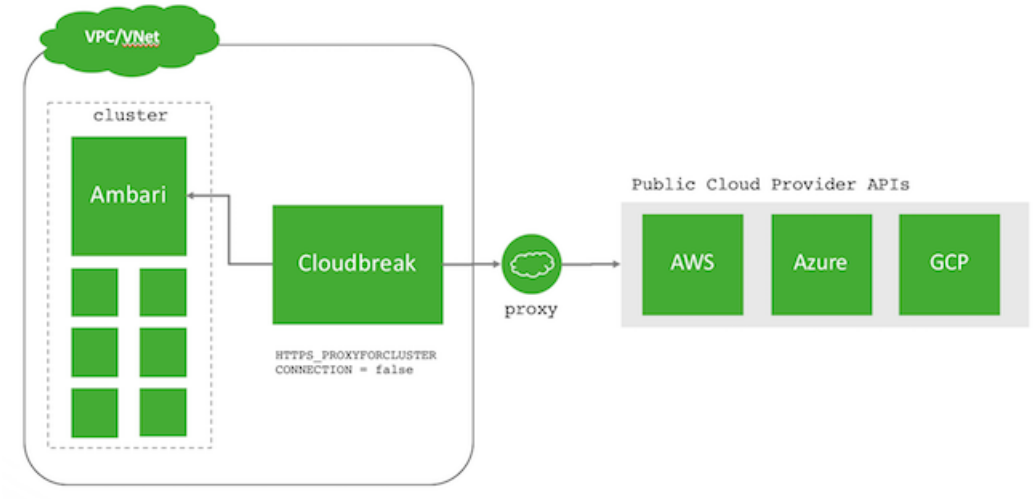
- Scenario 1: Cloudbreak needs to go through a proxy to access the cloud provider APIs (and other public internet resources) but can talk to the cluster hosts directly.
- Scenario 2: Cloudbreak needs to go through a proxy to access the cloud provider APIs (and other public internet resources) and the cluster hosts.

Scenario 1

In this scenario, Cloudbreak can resolve and communicate with the Ambari Server in the cluster hosts directly. For example, this can be a scenario where Cloudbreak is deployed in the same VPC/VNet as the clusters and will not go through the proxy. However, Cloudbreak will communicate to the public cloud provider APIs via the proxy.

To configure this scenario, set this setting in your Profile file:

```
HTTPS_PROXYFORCLUSTERCONNECTION = false
```

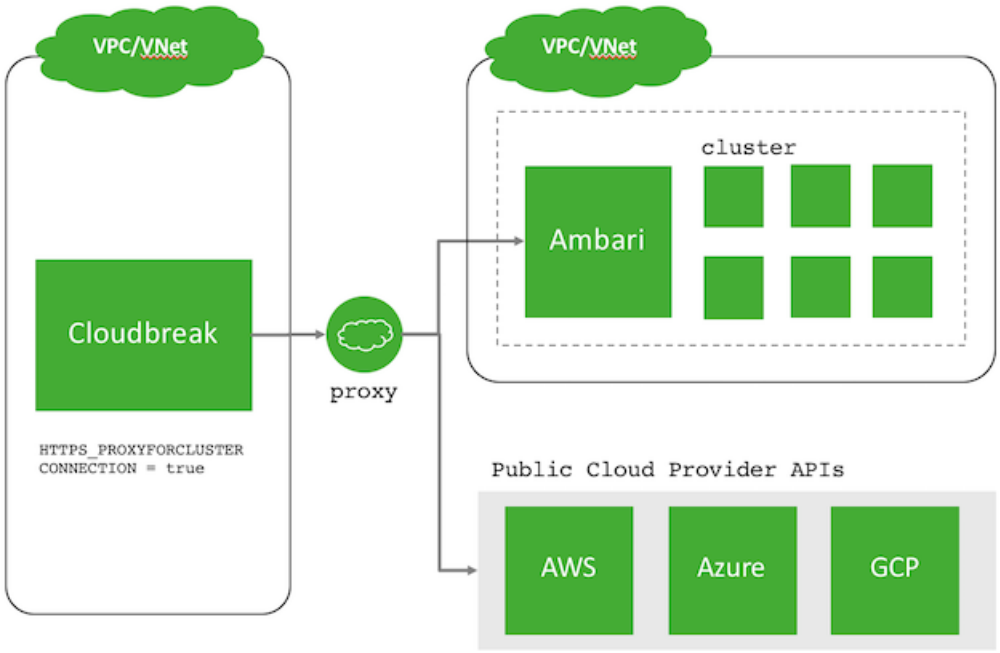


Scenario 2

In this scenario, Cloudbreak will connect to the Ambari Server through the configured proxy. For example, this can be a scenario where Cloudbreak is deployed to a different VPC/VNet than the cluster and must go through a proxy. Communication to the public cloud provider APIs also is via the proxy.

To configure this scenario, set this setting in your Profile file:

```
HTTPS_PROXYFORCLUSTERCONNECTION = true
```



Restrict inbound access to clusters

We recommend that after launching Cloudbreak you set CB_DEFAULT_GATEWAY_CIDR in your Profile file. When you launch a cluster, and Cloudbreak proposes security groups, this CIDR will be used for the Cloudbreak to the cluster master node (i.e. the host with Ambari Server) with this IP. This limits access from Cloudbreak to this cluster name for ports 9443 and 22 for Cloudbreak communication and management of the cluster.

Steps

1. Set CB_DEFAULT_GATEWAY_CIDR to the CIDR address range which is used by Cloudbreak to communicate with the cluster:

   ```
   export CB_DEFAULT_GATEWAY_CIDR=14.15.16.17/32
   ```

   Or, if your Cloudbreak communicates with the cluster through multiple addresses, set multiple addresses separated with a comma:

   ```
   export CB_DEFAULT_GATEWAY_CIDR=14.15.16.17/32,18.17.16.15/32
   ```

2. If Cloudbreak has already been started, restart it using cbd restart.
3. When CB_DEFAULT_GATEWAY_CIDR is set, two additional rules are added to your Ambari node security group: (1) port 9443 open to your Cloudbreak IP, and (2) port 22 open to your Cloudbreak IP. You can view and edit these default rules in the create cluster wizard.

Related Information

Default cluster security groups

Use SSL certificate for Cloudbreak

By default Cloudbreak is configured with a self-signed certificate for access via HTTPS. This is sufficient for many deployments such as trials, development, testing, or staging. However, for production deployments, you should obtain and configure a trusted certificate.

Follow these steps to configure Cloudbreak to use your own trusted certificate.

Prerequisites

To use your own certificate, you must have:

- A resolvable fully qualified domain name (FQDN) for the controller host IP address. For example, this can be configured in Amazon Route 53.
- A valid SSL certificate for this fully qualified domain name. The certificate can be obtained from a number of certificate providers.

Steps

1. SSH to the Cloudbreak host instance:

   ```
   ssh -i mykeypair.pem cloudbreak@[CONTROLLER-IP-ADDRESS]
   ```

2. Make sure that the target fully qualified domain name (FQDN) which you plan to use for Cloudbreak is resolvable:

   ```
   nslookup [TARGET-CONTROLLER-FQDN]
   ```

For example:

```
nslookup hdcloud.example.com
```

3. Browse to the Cloudbreak deployment directory and edit the Profile file:

```
vi /var/lib/cloudbreak-deployment/Profile
```

4. Replace the value of the PUBLIC_IP variable with the TARGET-CONTROLLER-FQDN value:

```
PUBLIC_IP=[TARGET-CONTROLLER-FQDN]
```

5. Copy your private key and certificate files for the FQDN onto the Cloudbreak host. These files must be placed under /var/lib/cloudbreak-deployment/certs/traefik/ directory.

Note:

File permissions for the private key and certificate files can be set to 600.

File	Example
PRIV-KEY-LOCATION	/var/lib/cloudbreak-deployment/certs/traefik/hdcloud.example.com.key
CERT-LOCATION	/var/lib/cloudbreak-deployment/certs/traefik/hdcloud.example.com.crt

6. Configure TLS details in your Profile by adding the following line at the end of the file.

Note:

Notice that CERT-LOCATION and PRIV-KEY-LOCATION are file locations from Step 5, starting at the /certs/... path.

```
export CBD_TRAEFIK_TLS=”[CERT-LOCATION],[PRIV-KEY-LOCATION]”
```

For example:

```
export CBD_TRAEFIK_TLS="/certs/traefik/hdcloud.example.com.crt,/certs/
traefik/hdcloud.example.com.key"
```

7. Restart Cloudbreak deployer:

```
cbd restart
```

8. Using your web browser, access the Cloudbreak UI using the new resolvable fully qualified domain name.
9. Confirm that the connection is SSL-protected and that the certificate used is the certificate that you provided to Cloudbreak.

Access from custom domains

Cloudbreak deployer, which uses UAA as an identity provider, supports multitenancy. In UAA, multitenancy is managed through identity zones. An identity zone is accessed through a unique subdomain. For example, if the standard UAA responds to https://uaa.10.244.0.34.xip.io, a zone on this UAA can be accessed through a unique subdomain https://testzone1.uaa.10.244.0.34.xip.io.

If you want to use a custom domain for your identity or deployment, add the UAA_ZONE_DOMAIN line to your Profile:

```
export UAA_ZONE_DOMAIN=my-subdomain.example.com
```

This variable is necessary for UAA to identify which zone provider should handle the requests that arrive to that domain.

Moving Cloudbreak instance

Refer to this section if you would like to transfer a Cloudbreak instance from one host to another.

Determine which of the following scenarios applies to your Cloudbreak deployment and follow the steps listed for that scenario.

Embedded PostgreSQL database

If you are using the embedded PostgreSQL database (provided by default):

1. Back up current Cloudbreak database data.
2. Launch a new Cloudbreak instance and start Cloudbreak. For steps, refer to Cloudbreak installation documentation for your cloud platform.
3. Populate the new Cloudbreak instance database with the dump from the original Cloudbreak instance on the new host.
4. Modify Cloudbreak Profile.

External PostgreSQL database

If you are using an external PostgreSQL database:

1. Launch a new Cloudbreak instance and start Cloudbreak. For steps, refer to Cloudbreak installation documentation for your cloud platform.
2. Modify Cloudbreak Profile.

Back up Cloudbreak database

If you are using the embedded PostgreSQL database, back up current Cloudbreak database data prior to upgrading or migrating your Cloudbreak instance.

Steps

1. On your Cloudbreak host machine, execute the following command to enter the container of the database:

```
docker exec -it cbreak_commondb_1 bash
```

   If it is not running, start the database container by using the docker start cbreak_commondb_1 command.
2. Create three database dumps (cbdb, uaadb, periscopedb):

```
pg_dump -Fc -U postgres cbdb > cbdb.dump
pg_dump -Fc -U postgres uaadb > uaadb.dump
pg_dump -Fc -U postgres periscopedb > periscopedb.dump
```

3. Quit from the container with shortcut CTRL+d.
4. Save the previously created dumps to the host instance:

```
docker cp cbreak_commondb_1:/cbdb.dump ./cbdb.dump
docker cp cbreak_commondb_1:/uaadb.dump ./uaadb.dump
docker cp cbreak_commondb_1:/periscopedb.dump ./periscopedb.dump
```

Populate database with dump from original Cloudbreak instance

If you are using the embedded PostgreSQL database, perform these steps to populate databases with information from the Cloudbreak server.

Steps

1. Copy the saved database files from the backup to the new Cloudbreak server host.
2. Copy the dump files into the database container with the following commands. Modify the location as necessary (The example below assumes that the files are in /tmp):

```
docker cp /tmp/cbdb.dump cbreak_commondb_1:/cbdb.dump
docker cp /tmp/uaadb.dump cbreak_commondb_1:/uaadb.dump
docker cp /tmp/periscopedb.dump cbreak_commondb_1:/periscopedb.dump
```

3. Execute the following command to stop the container:

```
docker stop cbreak_identity_1
```

4. Execute the following command to enter the container of the database:

```
docker exec -it cbreak_commondb_1 bash
```

5. Execute the following commands:

```
psql -U postgres
drop database uaadb;
drop database cbdb;
drop database periscopedb;
create database uaadb;
create database cbdb;
create database periscopedb;
```

Note:

If you get ERROR: database "uaadb" is being accessed by other users error, ensure that cbreak_identity_1 container is not running and then retry dropping uaadb.

6. Exit the PostgreSQL interactive terminal.

```
\q
```

7. Restore the databases from the original backups:

```
pg_restore -U postgres -d periscopedb periscopedb.dump
pg_restore -U postgres -d cbdb cbdb.dump
pg_restore -U postgres -d uaadb uaadb.dump
```

8. Quit from the container with the shortcut CTRL+d.

Modify Cloudbreak Profile

Perform these steps to ensure that your new Profile file is correctly set up.

Steps

1. Ensure that the following parameter values match in the origin and target Profile files and modify Profile file of the target environment if necessary:

```
export UAA_DEFAULT_USER_EMAIL=admin@example.com
export UAA_DEFAULT_SECRET=cbsecret
```

```
export UAA_DEFAULT_USER_PW=cbuser
```

2. Restart Cloudbreak application by using the cbd restart command.

   After performing these steps the migration is complete. To verify, log in to the UI of your new Cloudbreak instance and make sure that it contains the information from your old instance.

Disable providers

If you are planning to use Cloudbreak with a specific cloud provider or a specific set of cloud providers, you may want to disable the remaining providers. For example, if you are planning to use Cloudbreak with Azure only, you may want to disable AWS, Google Cloud, and OpenStack.

Steps

1. Navigate to the Cloudbreak deployment directory and edit Profile. For example:

   ```
   cd /var/lib/cloudbreak-deployment/
   vi Profile
   ```

2. Add the following entry, setting it to the provider that you would like to see. For example, if you would like to see Azure only, set this to “AZURE”:

   ```
   export CB_ENABLEDPLATFORMS=AZURE
   ```

   Accepted values are:

   - AZURE
   - AWS
   - GCP
   - OPENSTACK

   Any combination of platforms can be used; for example if you would like to see AWS and OpenStack, then use:

   ```
   export CB_ENABLEDPLATFORMS=AWS,OPENSTACK
   ```

   If you want to reverse the change and see all providers, then either delete CB_ENABLEDPLATFORMS from the Profile or add the following:

   ```
   export CB_ENABLEDPLATFORMS=AZURE,AWS,GCP,OPENSTACK
   ```

3. Restart Cloudbreak by using cbd restart.

Modify default Cloudbreak ports

By default, Cloudbreak uses ports 80 (HTTP) and 443 (HTTPS) to access the Cloudbreak server (for the web UI and for the CLI). To change these port numbers, you must edit the Profile file on your Cloudbreak host.

Cloudbreak should not be running when you change the port numbers. Edit Profile either before you start Cloudbreak the first time or stop Cloudbreak before editing the file.

Steps

1. Navigate to the Cloudbreak deployment directory (typically /var/lib/cloudbreak-deployment) and open the Profile file with a text editor.
2. Add one or both of the following parameters, setting them to the port numbers that you want to use:

   ```
   export PUBLIC_HTTP_PORT=111
   ```

```
export PUBLIC_HTTPS_PORT=222
```

3. Start or restart Cloudbreak by using cbd start or cbd restart.
4. This change affects Cloudbreak CLI configuration. When configuring the CLI, you must provide these ports as part of the server URL. For example:

```
cb configure --server http://cb.server.address:111 --username
 test@hortonworks.com
cb configure --server https://cb.server.address:222 --username
 test@hortonworks.com
```

Related Information

Configure Cloudbreak CLI

Modify Cloudbreak credential

The option to modify an existing Cloudbreak credential is useful if you need to make changes in your credential but you have running clusters that were created by using that credential.

Note:

The value of the “Name” parameter cannot be changed. The values of sensitive parameters will not be displayed and you will have to reenter them.

Steps

1. In the Cloudbreak web UI, select Credentials from the navigation pane.
2. Click on

next to the credential that you want to edit.
3. When done making changes, click Save to save your changes.

Set default Cloudbreak credential

If using multiple Cloudbreak credentials, you can select one credential and use it as default for creating clusters. This default credential will be pre-selected in the create cluster wizard.

Steps

1. In the Cloudbreak web UI, select Credentials from the navigation pane.
2. Click Set as default next to the credential that you would like to set as default.
3. Click Yes to confirm.

Alternatively, you can perform the same steps from the Settings page.

Set default cluster wizard view

If using multiple Cloudbreak credentials, you can select one credential and use it as default for creating clusters. This default credential will be pre-selected in the create cluster wizard.

Steps

1. In the Cloudbreak web UI, select Settings from the navigation pane.

2. Under Clusters > Default Cluster Wizard View select the basic or advanced view.

Set up SMTP email notifications

If you want to configure email notification, configure SMTP parameters in your Profile.

Note:

In order to use this configuration, your email server must use SMTP.

The default values of the SMTP parameters are:

```
export CLOUDBREAK_SMTP_SENDER_USERNAME=
export CLOUDBREAK_SMTP_SENDER_PASSWORD=
export CLOUDBREAK_SMTP_SENDER_HOST=
export CLOUDBREAK_SMTP_SENDER_PORT=25
export CLOUDBREAK_SMTP_SENDER_FROM=
export CLOUDBREAK_SMTP_AUTH=true
export CLOUDBREAK_SMTP_STARTTLS_ENABLE=true
export CLOUDBREAK_SMTP_TYPE=smtp
```

For example:

```
export CLOUDBREAK_SMTP_SENDER_USERNAME='myemail@gmail.com'
export CLOUDBREAK_SMTP_SENDER_PASSWORD='Mypassword123'
export CLOUDBREAK_SMTP_SENDER_HOST='smtp.gmail.com'
export CLOUDBREAK_SMTP_SENDER_PORT=25
export CLOUDBREAK_SMTP_SENDER_FROM='myemail@gmail.com'
export CLOUDBREAK_SMTP_AUTH=true
export CLOUDBREAK_SMTP_STARTTLS_ENABLE=true
export CLOUDBREAK_SMTP_TYPE=smtp
```

Note:

The example assumes that you are using gmail. You should use the settings appropriate for your SMTP server.

If your SMTP server uses SMTPS, you must set the protocol in your Profile to smtps:

```
export CLOUDBREAK_SMTP_TYPE=smtps
```

Import HDP and HDF images to OpenStack

An OpenStack administrator can perform these steps to add the Cloudbreak deployer image to your OpenStack deployment. Perform these steps for each image.

Note:

Importing prewarmed and base HDP and HDF images is no longer required, because if these images are not imported manually, Cloudbreak will import them once you attempt to create a cluster.

The following images can be imported:

Image	Operating system	Location
Prewarmed HDP 3.1 image	centos7	http://public-repo-1.hortonworks.com/HDP/cloudbreak/cb-hdp-31-1901151759.img
Prewarmed HDP 2.6 image	centos7	http://public-repo-1.hortonworks.com/HDP/cloudbreak/cb-hdp-26-1808062221.img

Image	Operating system	Location
Prewarmed HDF 3.3 image	centos7	http://public-repo-1.hortonworks.com/HDP/cloudbreak/cb-hdp-33-1901152007.img
Base image	centos7	http://public-repo-1.hortonworks.com/HDP/cloudbreak/cb-hdp--1901151721.img
Base image	ubuntu16	http://public-repo-1.hortonworks.com/HDP/cloudbreak/cb-hdp--1808030959.img

Steps

1. Download the image to your local machine. For example:

```
curl -O https://public-repo-1.hortonworks.com/HDP/cloudbreak/cb-
hdp-26-1805171052.img
```

2. Set the following environment variables for the OpenStack image import:

```
export CB_LATEST_IMAGE=cb-hdp-26-1805171052.img
export CB_LATEST_IMAGE_NAME=cb-hdp-26-1805171052.img
export OS_USERNAME=your_os_user_name
export OS_AUTH_URL=your_authentication_url
export OS_TENANT_NAME=your_os_tenant_name
```

3. Import the new image into your OpenStack:

```
glance image-create --name "$CB_LATEST_IMAGE_NAME" --file
 "$CB_LATEST_IMAGE" --disk-format qcow2 --container-format bare --progress
```

After performing the import, you should be able to see these images among your OpenStack images.

Cloudbreak Profile

Cloudbreak deployer configuration is based on environment variables. Refer to this section to review available Profile configuration options.

During startup, Cloudbreak deployer tries to determine the underlying infrastructure and then sets required environment variables with appropriate default values. If these environment variables are not sufficient for your use case, you can set additional environment variables in your Profile file.

Secure the Profile file

Before starting Cloudbreak for the first time, configure the Profile file as directed below.

Changes are applied during startup so a restart (cbd restart) is required after each change.

1. Execute the following command in the directory where you want to store Cloudbreak-related files:

```
echo export PUBLIC_IP=[the ip or hostname to bind] > Profile
```

2. After you have a base Profile file, add the following custom properties to it:

```
export UAA_DEFAULT_SECRET='[custom secret]'
export UAA_DEFAULT_USER_EMAIL='[default admin email address]'
export UAA_DEFAULT_USER_PW='[default admin password]'
export UAA_DEFAULT_USER_FIRSTNAME='[default admin first name]'
export UAA_DEFAULT_USER_LASTNAME='[default admin last name]'
```

Cloudbreak has additional secrets which by default inherit their values from `UAA_DEFAULT_SECRET`. Instead of using the default, you can define different values in the Profile for each of these service clients:

```
export UAA_CLOUDBREAK_SECRET='[cloudbreak secret]'
export UAA_PERISCOPE_SECRET='[auto scaling secret]'
export UAA_ULUWATU_SECRET='[web ui secret]'
export UAA_SULTANS_SECRET='[authenticator secret]'
```

You can change these secrets at any time, except `UAA_CLOUDBREAK_SECRET` which is used to encrypt sensitive information at database level. `UAA_DEFAULT_USER_PW` is stored in plain text format, but if `UAA_DEFAULT_USER_PW` is missing from the Profile, it gets a default value. Because default password is not an option, if you set an empty password explicitly in the Profile Cloudbreak deployer will ask for password all the time when it is needed for the operation.

```
export UAA_DEFAULT_USER_PW=' '
```

In this case, Cloudbreak deployer wouldn't be able to add the default user, so you have to do it manually by executing the following command:

```
cbd util add-default-user
```

Check available Profile variables

Cloudbreak includes a command that allows you to check all available environment variables.

To see all available environment variables with their default values, use the following command:

```
cbd env show
```

Set Profile variables

You can set Profile variables by using the following steps.

Steps

1. To set environment variables relevant for Cloudbreak deployer, add them to a file called Profile located in the Cloudbreak deployment directory (typically /var/lib/cloudbreak-deployment). The Profile file is sourced, so you can use the usual syntax to set configuration values:

   ```
   export MY_VAR=some_value
   export MY_OTHER_VAR=another_value
   ```

2. After changing a property, you must regenerate the config file and restart the application by using cbd restart.

Create environment-specific Profiles

If you would like to use a different versions of Cloudbreak for prod and qa profile, you must create two environment specific configurations that can be sourced.

For example:

- Profile.prod
- Profile.qa

For example, to create and use a prod profile, you need to:

1. Create a file called Profile.prod

2. Write the environment-specific export DOCKER_TAG_CLOUDBREAK=0.3.99 into Profile.prod to specify Docker image.
3. Set the environment variable: CBD_DEFAULT_PROFILE=prod

To use the prod specific profile once, set:

```
CBD_DEFAULT_PROFILE=prod cbd some_commands
```

To permanently use the prod profile, set export CBD_DEFAULT_PROFILE=prod in your .bash_profile.

Add tags in Profile (AWS)

You can optionally define custom tags for your AWS resources deployed by Cloudbreak.

- If you want just one custom tag for Cloudbreak-provisioned resources, set this variable in the Profile:

  export CB_AWS_DEFAULT_CF_TAG=mytagcontent

  In this example, the name of the tag will be CloudbreakId and the value will be mytagcontent.
- If you prefer to customize the tag name, set this variable:

  export CB_AWS_CUSTOM_CF_TAGS=mytagname:mytagvalue

  In this example the name of the tag will be mytagname and the value will be mytagvalue.
- You can specify a list of tags with a comma separated list:

  export CB_AWS_CUSTOM_CF_TAGS=tag1:value1,tag2:value2,tag3:value3

Related Information

Tags

Modifying UAA_DEFAULT_SECRET

You should not modify the UAA_DEFAULT_SECRET (secret key) Profile variable.

Modifying the secret key value in the Profile file causes problems when upgrading Cloudbreak: When a Cloudbreak upgrade is initiated, Cloudbreak cannot decrypt the encrypted data from its database causing the startup process to crash due to a database migration error.

SmartSense telemetry

Help us make a better product by opting in to automatically send information to Hortonworks. This includes enabling Hortonworks SmartSense and sending performance and usage info. As you use the product, SmartSense measures and collects information and then sends these information bundles to Hortonworks.

Disable bundle upload for Cloudbreak and new clusters

Perform these steps to disable bundle upload for Cloudbreak and new clusters.

Steps

Note:

Do not perform these steps when you have clusters currently in the process of being deployed. Wait for all clusters to be deployed.

1. SSH into the Cloudbreak host.
2. Edit /var/lib/cloudbreak-deployment/Profile.

3. Change CB_SMARTSENSE_CONFIGURE to false:

```
export CB_SMARTSENSE_CONFIGURE=false
```

4. Restart Cloudbreak:

```
cd /var/lib/cloudbreak-deployment
cbd restart
```

Disable bundle upload for an existing cluster

Perform these steps to disable bundle upload for existing clusters.

Steps

1. SSH into the master node for the cluster.
2. Edit /etc/hst/conf/hst-server.ini.
3. Change [gateway] configuration to false:

```
[gateway]
enabled=false
```

4. Restart the SmartSense Server:

```
hst restart
```

5. (Optional) Disable SmartSense daily bundle capture:
   - SmartSense is scheduled to capture a telemetry bundle daily. With the bundle upload disabled, the bundle will still be captured but just saved locally (i.e. not uploaded).
   - To disable the bundle capture, execute the following:

```
hst capture-schedule -a pause
```

6. Repeat on all existing clusters.