

Troubleshooting Cloudbreak 2

Troubleshooting Cloudbreak

Date of Publish: 2019-05-28



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

Contents

Getting help.....	4
HCC.....	4
Flex subscription.....	4
Configure SmartSense.....	4
Register and manage Flex subscriptions.....	5
Use Flex subscription for a cluster.....	5
Use Flex subscription for Cloudbreak node.....	5
More Cloudbreak resources.....	5
Create a troubleshooting bundle.....	5
Checking Cloudbreak logs.....	6
Cloudbreak logs.....	6
Saltstack logs.....	7
Ambari logs.....	7
Recipe logs.....	7
Troubleshooting Cloudbreak.....	8
Invalid PUBLIC_IP in CBD Profile.....	8
Cbd cannot get VM's public IP.....	8
Docker does not start and returns an error.....	8
Permission or connection problems.....	9
Creating cbreak_sultans_1 ... Error.....	9
Copy the JSON button does not work in Firefox.....	11
UnicodeEncodeError when starting Cloudbreak.....	11
Cloudbreak not working after VM restart.....	11
Troubleshooting Cloudbreak on AWS.....	12
User is not authorized to perform: sts:AssumeRole.....	12
Troubleshooting Cloudbreak on Azure.....	12
Resource was not found.....	12
You don't have permissions to assign roles on Azure.....	12
Credential creation errors on Azure.....	13
Troubleshooting cluster creation.....	14
Configure communication via private IPs on AWS.....	14
Cannot access Oozie web UI.....	14
Failed to retrieve the server's certificate.....	15
Quota limitations.....	15
Connection timeout when ports are not open.....	15
Blueprint errors.....	16
Recipe errors.....	16

Troubleshooting Cloudbreak CLI..... 17

Getting help

If you need help with Cloudbreak, you have two options:

Option	Description
Hortonworks Community Connection	This is free optional support via Hortonworks Community Connection (HCC).
Hortonworks Flex Support Subscription	This is paid Hortonworks enterprise support.

HCC

You can register for optional free community support at [Hortonworks Community Connection](#), where you can browse articles and previously answered questions and ask questions of your own.

When posting questions related to Cloudbreak, make sure to use the “Cloudbreak” tag.

Flex subscription

You can optionally use your existing Hortonworks [flex support subscription\(s\)](#) to cover the Cloudbreak node and clusters managed by it.



Note:

You must have an existing SmartSense ID and a Flex subscription. For general information about the Hortonworks Flex Support Subscription, visit the Hortonworks Support page at <https://hortonworks.com/services/support/enterprise/>.

The general steps are:

1. Configure Smart Sense in your Profile file.
2. Register your Flex subscription in the Cloudbreak web UI. You can register and manage multiple Flex subscriptions.
3. When creating a cluster, in the General Configuration > Flex Subscription, select the Flex subscription that you want to use for the cluster.
4. Use the Flex subscription to cover your Cloudbreak instance.

Configure SmartSense

To configure SmartSense in Cloudbreak, enable SmartSense and add your SmartSense ID to the Profile.

Add the following variables to the Profile:

```
export CB_SMARTSENSE_CONFIGURE=true
export CB_SMARTSENSE_ID=YOUR-SMARTSENSE-ID
```

For example:

```
export CB_SMARTSENSE_CONFIGURE=true
export CB_SMARTSENSE_ID=A-00000000-C-00000000
```

You can do this in one of the two ways:

- When initiating Cloudbreak deployer
- After you’ve already initiated Cloudbreak deployer. If you choose this option, you must restart Cloudbreak using `cbd restart`.

Register and manage Flex subscriptions

Once you log in to the Cloudbreak web UI, you can manage your Flex subscriptions from the Cloudbreak web UI.

You can manage your Flex subscriptions from the Settings page > Flex Subscriptions.

You can:

- Register a new Flex subscription
- Set a default Flex subscription (“Default”)
- Select a Flex subscription to be used for the Cloudbreak node (“Use for controller”)
- Delete a Flex subscription

Use Flex subscription for a cluster

When creating a cluster, you can select the Flex subscription that you would like to use for the cluster.

You can do this from the General Configuration page by using the Flex Subscription option.

Use Flex subscription for Cloudbreak node

You can use a Flex subscription for Cloudbreak node.

To use a Flex subscription for Cloudbreak node, on the Settings page, in the Flex Subscriptions section, check the “Use for controller option” for the selected Flex ID.

More Cloudbreak resources

Cloudbreak documentation only includes the information related to deploying Cloudbreak and Cloudbreak-managed clusters. If you are looking for information related to Ambari, HDP, or HDF, check out the following resources.

Resource	Description
Hortonworks documentation	During cluster create process, Cloudbreak automatically installs Ambari and sets up a cluster for you. After this deployment is complete, refer to the Ambari documentation and HDP documentation for help.
Hortonworks tutorials	Use Hortonworks tutorials to get started with Apache Spark, Apache Hive, Apache Zeppelin, and more.
Apache documentation	In addition to Hortonworks documentation, refer to the Apache Software Foundation documentation to get information on specific Hadoop services.
Ambari blueprints	Learn about Ambari blueprints. Ambari blueprints are a declarative definition of a Hadoop cluster that Ambari can use to create Hadoop clusters.
Cloudbreak project	Visit the Hortonworks website to see Cloudbreak-related news and updates.
Apache Ambari project	Learn about the Apache Ambari project. Apache Ambari is an operational platform for provisioning, managing, and monitoring Apache Hadoop clusters. Ambari exposes a robust set of REST APIs and a rich web interface for cluster management.

Create a troubleshooting bundle

Cloudbreak includes the `cbd create-bundle` command that allows you to export Cloudbreak logs into a bundle. When filing a support case ticket, you should use this command to generate a bundle and then attach the bundle to the support case ticket.

Steps

1. Access the Cloudbreak VM via SSH.
2. Navigate to your Cloudbreak deployment directory, typically `/var/lib/cloudbreak-deployment/`:

```
/var/lib/cloudbreak-deployment/
```

3. Run the following command:

```
cbd create-bundle
```

This generates a file called `cbd_export_timestamp.tar.gz`.

Optionally you can customize the file name by specifying its name by running this command instead:

```
cbd create-bundle my-bundle-name
```

This command collects all Cloudbreak logs and configurations (such as Cloudbreak configurations, firewall and iptables configurations, os type, and open ports). After collection, all data is anonymized and compressed into a tar.gz archive. During the bundle creation process temporary folders are created and then removed.

4. After the process finished, you can find the bundle in the location where you issued the command. Copy the bundle onto your machine, for example by using the secure copy command (`scp`).
5. When filing a support case ticket, attach the bundle to the ticket.

Checking Cloudbreak logs

When troubleshooting, you can access the following Cloudbreak logs.

Cloudbreak logs

When installing Cloudbreak using a pre-built cloud image, the Cloudbreak deployer location and the `cbd` root folder is `/var/lib/cloudbreak-deployment`. You must execute all `cbd` actions from the `cbd` root folder as "cloudbreak" user.



Note:

Your `cbd` root directory may be different if you installed Cloudbreak on your own VM.

Aggregated logs

Cloudbreak consists of multiple microservices deployed into Docker containers. To check aggregated service logs, use the following commands:

`cbd logs` shows all service logs.

`cbd logs | tee cloudbreak.log` allows you to redirect the input into a file for sharing these logs.

Individual service logs

To check individual service logs, use the following commands:

`cbd logs cloudbreak` shows Cloudbreak logs. This service is the backend service that handles all deployments.

`cbd logs uluwatu` shows Cloudbreak web UI logs. Uluwatu is the UI component of Cloudbreak.

`cbd logs identity` shows Identity logs. Identity is responsible for authentication and authorization.

`cbd logs periscope` shows Periscope logs. Periscope is responsible for triggering autoscaling rules.

Docker logs

The same logs can be accessed via Docker commands:

docker logs cbreak_cloudbreak_1 shows the same logs as `cbd logs cloudbreak`.

Cloudbreak logs are rotated and can be accessed later from the Cloudbreak deployment folder. Each time you restart the application via `cbd restart` a new log file is created with a timestamp in the name (for example, `cbreak-20170821-105900.log`).

**Note:**

There is a symlink called `cbreak.log` which points to the latest log file. Sharing this symlink does not share the log itself.

Saltstack logs

Cloudbreak uses Saltstack to install Ambari and the necessary packages for the HDP/HDF provisioning. Salt Master always runs alongside the Ambari Server node. Each instance in the cluster runs a Salt Minion, which connects to the Salt Master.

There can be multiple Salt Masters if the cluster is configured to run in HA (High Availability) mode and in this case each Salt Minion connects to each Salt Master.

Cloudbreak also uses SaltStack to execute user-provided customization scripts called “recipes”.

Salt Master and Salt Minion logs can be found at the following location: `/var/log/salt`

Ambari logs

Cloudbreak uses Ambari to orchestrate the installation of the different HDP/HDF components. Each instance in the cluster runs an Ambari agent which connects to the Ambari server. Ambari server is declared by the user during the cluster installation wizard.

Ambari server logs

Ambari server logs can be found on the nodes where Ambari server is installed in the following locations:

`/var/log/ambari-server/ambari-server.log`

`/var/log/ambari-server/ambari-server.out`

Both files contain important information about the root cause of a certain issue so it is advised to check both.

Ambari agent logs

Ambari agent logs can be found on the nodes where Ambari agent is installed in the following locations:

`/var/log/ambari-agent/ambari-agent.log`

Recipe logs

Cloudbreak supports “recipes” - user-provided customization scripts that can be run prior to or after cluster installation. It is the user’s responsibility to provide an idempotent well tested script. If the execution fails, the recipe logs can be found at `/var/log/recipes` on the nodes on which the recipes were executed.

It is advised, but not required to have an advanced logging mechanism in the script, as Cloudbreak always logs every script that are run. Recipes are often the sources of installation failures as users might try to remove necessary packages or reconfigure services.

Troubleshooting Cloudbreak

This section includes common errors related to launching Cloudbreak and steps to resolve them.

Invalid PUBLIC_IP in CBD Profile

Invalid PUBLIC_IP error when starting Cloudbreak.

Error: Invalid PUBLIC_IP error when starting Cloudbreak.

Solution: The PUBLIC_IP property must be set in the Profile file or else you won't be able to log in to the Cloudbreak web UI. If you are migrating your instance, check the Profile file to make sure that after the start the value of the PUBLIC_IP property remains valid. If editing the IP, make sure to restart Cloudbreak by using `cbd restart`.

Cbd cannot get VM's public IP

When starting Cloudbreak, `cbd` cannot get VM's public IP.

Error: When starting Cloudbreak, `cbd` cannot get VM's public IP.

Solution: By default, the `cbd` tool tries to get the VM's public IP to bind Cloudbreak web UI to it. But if `cbd` cannot get the IP address during the initialization, you must set it manually. To solve the issue, check your Profile and if PUBLIC_IP is not set, add the PUBLIC_IP variable and set it to the public IP of the VM. For example:

```
export PUBLIC_IP=192.134.23.10
```

Next, restart Cloudbreak by using `cbd restart`.

Docker does not start and returns an error

Failed to start Docker Application Container Engine.

Error: In newer versions of Docker (1.12+) for CentOS, `overlay2` is not a supported filesystem and Docker might not start with the following message:

```
-- Unit docker.service has begun starting up.
Aug 06 12:47:01 ip-172-31-24-242 dockerd-current[1622]:
  time="2018-08-06T12:47:01.431682737-04:00" level=warning msg="could not
change group /var/run/docker.sock
STORAGE_DRIVER=devicemapper
Aug 06 12:47:01 ip-172-31-24-242 dockerd-current[1622]:
  time="2018-08-06T12:47:01.433641166-04:00" level=info msg="libcontainerd:
new containerd process, pid: 16
Aug 06 12:47:02 ip-172-31-24-242 dockerd-current[1622]:
  time="2018-08-06T12:47:02.440563079-04:00" level=warning msg="overlay2: the
backing xfs filesystem is for
Aug 06 12:47:02 ip-172-31-24-242 dockerd-current[1622]: Error starting
daemon: SELinux is not supported with the overlay2 graph driver on this
kernel. Either boo
Aug 06 12:47:02 ip-172-31-24-242 systemd[1]: docker.service: main process
exited, code=exited, status=1/FAILURE
Aug 06 12:47:02 ip-172-31-24-242 systemd[1]: Failed to start Docker
Application Container Engine.
-- Subject: Unit docker.service has failed
```

Solution: To resolve this error, open the following files in an editor:

```
/etc/sysconfig/docker-storage  
/etc/sysconfig/docker-storage-setup
```

For example:

```
vi /etc/sysconfig/docker-storage  
vi /etc/sysconfig/docker-storage-setup
```

Next, in each of the files, change "overlay2" to "devicemapper".

Permission or connection problems

An error related to permission or connection problems when starting Cloudbreak.

Error: An error related to permission or connection problems when starting Cloudbreak.

Solution: The most common reason for this type of error is that SELinux is enabled. To solve it:

1. Disable SELINUX:

```
setenforce 0  
sed -i 's/SELINUX=enforcing/SELINUX=disabled/g' /etc/selinux/config
```

2. Ensure the SELinux is not turned on afterwards:

```
sestatus | grep -i mode  
Current mode:                permissive  
Mode from config file:       permissive
```

Creating cbreak_sultans_1 ... Error

Cannot create container for service sultans

Error: cbd start returns the following error:

```
Creating cbreak_sultans_1 ... error  
  
ERROR: for cbreak_sultans_1 Cannot create container for service sultans:  
unknown log opt 'max-size' for journald log driver  
Creating cbreak_consul_1  
Creating cbreak_logrotate_1 ... error  
Creating cbreak_periscope_1 ... error  
Creating cbreak_mail_1 ... error  
Creating cbreak_haveged_1 ... error  
  
ERROR: for cbreak_mail_1 Cannot create container for service mail: unknown  
log opt 'max-size' for journald log driver  
  
Creating cbreak_uluwatu_1 ... error  
  
Creating cbreak_smartsense_1 ... error  
  
Creating cbreak_consul_1 ... error  
Creating cbreak_identity_1 ... error  
  
ERROR: for cbreak_identity_1 Cannot create container for service identity:  
unknown log opt 'max-file' for journald log driver  
Creating cbreak_logsink_1 ... error
```

```
Creating cbreak_commondb_1 ... error

ERROR: for cbreak_commondb_1 Cannot create container for service commondb:
unknown log opt 'max-size' for journald log driver

ERROR: for haveged Cannot create container for service haveged: unknown log
opt 'max-size' for journald log driver

ERROR: for uluwatu Cannot create container for service uluwatu: unknown log
opt 'max-size' for journald log driver

ERROR: for consul Cannot create container for service consul: unknown log
opt 'max-size' for journald log driver

ERROR: for commondb Cannot create container for service commondb: unknown
log opt 'max-size' for journald log driver

ERROR: for logrotate Cannot create container for service logrotate: unknown
log opt 'max-size' for journald log driver

ERROR: for periscope Cannot create container for service periscope: unknown
log opt 'max-size' for journald log driver

ERROR: for sultans Cannot create container for service sultans: unknown log
opt 'max-size' for journald log driver

ERROR: for mail Cannot create container for service mail: unknown log opt
'max-size' for journald log driver

ERROR: for logsink Cannot create container for service logsink: unknown log
opt 'max-size' for journald log driver

ERROR: for smartsense Cannot create container for service smartsense:
unknown log opt 'max-size' for journald log driver

ERROR: for identity Cannot create container for service identity: unknown
log opt 'max-file' for journald log driver
Encountered errors while bringing up the project.
```

Solution: This error means that your [Docker logging drivers](#) are not configured correctly. To resolve the issue, on your Cloudbreak VM:

1. Check the Docker Logging Driver configuration:

```
docker info | grep "Logging Driver"
```

If it is set to “Logging Driver: journald”, you must set it to “json-file”.

2. Open the docker file for editing:

```
vi /etc/sysconfig/docker
```

3. Edit the following part of the file so that it looks like below (showing log-driver=json-file):

```
# Modify these options if you want to change the way the docker daemon
runs
OPTIONS='--selinux-enabled --log-driver=json-file --signature-
verification=false'
```

4. Restart Docker:

```
systemctl restart docker
systemctl status docker
```

Copy the JSON button does not work in Firefox

When using the Copy the JSON or Copy the Command button with Firefox, the content does not get copied.

Error: When using the Show CLI Command > Copy the JSON or Copy the Command button with Firefox, the content does not get copied if adblock plugin or other advertise blocker plugins are present.

Solution: To resolve this issue, use a browser without an adblock plugin.

UnicodeEncodeError when starting Cloudbreak

UnicodeEncodeError when starting Cloudbreak.

Error:

```
{root@sldifrdwbo02:/sldifrdwbo02/solr/cloudbreak # ./cbd start
generating docker-compose.yml
generating uaa.yml
Initialize and migrate databases
Starting cbreak_commondb_1 ... done
Pulling uluwatu (hortonworks/hdc-web:2.8.0)...
Trying to pull repository registry.access.redhat.com/hortonworks/hdc-
web ...
Trying to pull repository docker.io/hortonworks/hdc-web ...
2.8.0: Pulling from docker.io/hortonworks/hdc-web
Trying to pull repository registry.fedoraproject.org/hortonworks/hdc-
web ...
Pulling repository registry.fedoraproject.org/hortonworks/hdc-web
Trying to pull repository quay.io/hortonworks/hdc-web ...
Pulling repository quay.io/hortonworks/hdc-web
Trying to pull repository registry.centos.org/hortonworks/hdc-web ...
Pulling repository registry.centos.org/hortonworks/hdc-web
Trying to pull repository docker.io/hortonworks/hdc-web ...
2.8.0: Pulling from docker.io/hortonworks/hdc-web
Traceback (most recent call last):
File "logging/init.py", line 861, in emit
File "logging/init.py", line 734, in format
File "logging/init.py", line 469, in format
UnicodeEncodeError: 'ascii' codec can't encode character u'\xab' in position
 3000: ordinal not in range(128)
Logged from file main.py, line 80
}}
```

Solution: Try setting the following variables in the Profile, one by one, and see if Cloudbreak starts or not.

```
export LANG=C.UTF-8
export LANG=en_US.UTF-8
export PYTHONIOENCODING=utf8
```

Cloudbreak not working after VM restart

After VM restart, Cloudbreak web UI is not working.

Error: After VM restart, Cloudbreak web UI is not working.

Solution: After VM restart, you must SSH to the VM and manually start Cloudbreak from the directory in which it was deployed (typically /var/lib/cloudbreak-deployer/):

```
cd /var/lib/cloudbreak-deployer/
```

```
cbd start
```

Troubleshooting Cloudbreak on AWS

The following section lists common issues related to launching Cloudbreak on AWS and steps to resolve them.

User is not authorized to perform: sts:AssumeRole

User: arn:aws:iam:::user/assume-only-user is not authorized to perform: sts:AssumeRole.

Error: User: arn:aws:iam:::user/assume-only-user is not authorized to perform: sts:AssumeRole

Solution: This error occurs when the Cloudbreak instance is not authorized to use the role that you are trying to register as part of the Cloudbreak credential creation. The most common reason for this error is that when creating the CredentialRole AWS IAM role you did not provide the AWS ID. Refer to the [Create CredentialRole](#) documentation for correct steps.

Troubleshooting Cloudbreak on Azure

The following section lists common issues related to launching Cloudbreak on Azure and steps to resolve them.

Resource was not found

Resource was not found error when deploying Cloudbreak on Azure.

Error: Resource /subscriptions/.../resourceGroups//providers/Microsoft.Network/virtualNetworks/cbdeployerVnet/subnets/cbdeployerSubnet referenced by resource /subscriptions/.../resourceGroups/Manulife-ADLS/providers/Microsoft.Network/networkInterfaces/cbdeployerNic was not found. Please make sure that the referenced resource exists, and that both resources are in the same region.

Solution: The most common reason for this error is that you did not provide the Vnet RG Name (last parameter in the template). To solve this, when launching Cloudbreak, under “Vnet RG Name” provide the name of the resource group in which the selected VNet is located. If using a new VNet, enter the same resource group name as in “Resource group”.

You don't have permissions to assign roles on Azure

Permissions errors when user is not authorized to perform role assignment,

Error: When fulfilling prerequisites for an app-based Cloudbreak credential, you must register an application and assign the Contributor role to it.

If you are authorized to perform role assignment, you will get the following error:

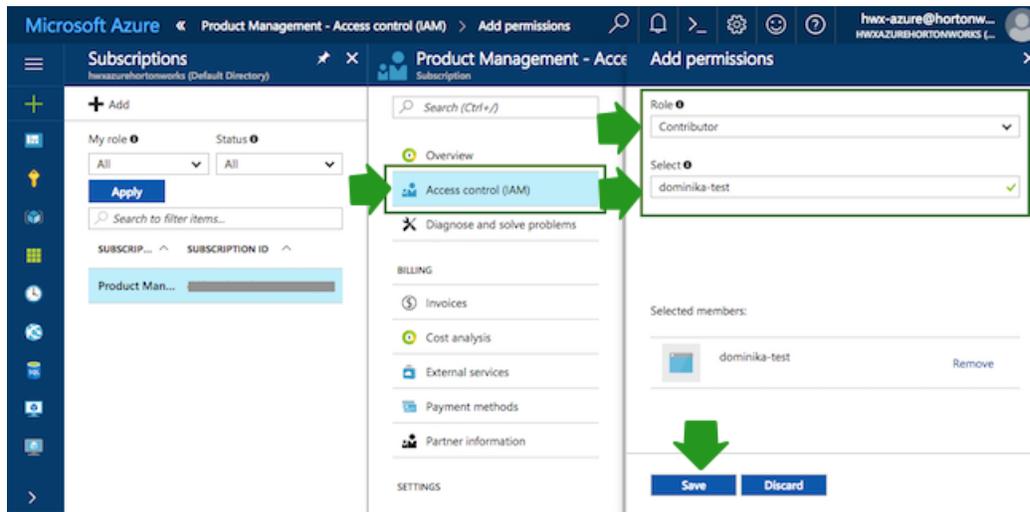
```
You don't have enough permissions to assign roles, please contact with your administrator
```

If you skip the role assignment step you will get the following error when creating an app-based credential:

```
Failed to verify the credential:  
Status code 403, {"error":{"code":"AuthorizationFailed","message":  
"The client 'someid' with object id 'someid' does not have  
authorization to perform action
```

```
'Microsoft.Storage/storageAccounts/read' over scope '/subscriptions/someid'. "}}
```

Solution: To solve the problem, ask your Azure administrator to perform the step of assigning the “Contributor” role to your application:



Credential creation errors on Azure

The following errors may occur during credential creation.

Role already exists

Error: Role already exists in Azure with the name: CloudbreakCustom50

Solution: This error occurs because you specified that you want to create a new role for Cloudbreak credential, but an existing role with the same name already exists in Azure. To solve this, you should either rename the role during credential creation or select the Reuse existing custom role option.

Role does not exist

Error: Role does not exist in Azure with the name: CloudbreakCustom60

Solution: This error occurs because you specified that you want to reuse an existing role for your Cloudbreak credential, but that particular role does not exist in Azure. To solve this, you should either rename the new role during the credential creation to match the existing role’s name or select the Let Cloudbreak create a custom role option.

Role does not have enough privileges

Error: CloudbreakCustom 50 role does not have enough privileges to be used by Cloudbreak!

Solution: This error occurs because you specified that you want to reuse an existing role for your Cloudbreak credential, but that particular role does not have the necessary privileges for Cloudbreak cluster management. To solve this, you should either select an existing role with enough privileges or select the Let Cloudbreak create a custom role option. The necessary action set for Cloudbreak to be able to manage the clusters includes: "Microsoft.Compute/*", "Microsoft.Network/*", "Microsoft.Storage/*", "Microsoft.Resources/*"

Client does not have authorization

Error: Failed to verify credential: Status code 403, {"error":{"code":"AuthorizationFailed", "message":"The client 'X' with object id 'z' does not have authorization to perform action 'Microsoft.Storage/storageAccounts/read' over scope 'subscriptions/...'}}

Solution: This means that your Azure account does not have sufficient permissions to create a Cloudbreak credential. If you get this error during interactive credential creation, ensure that your Azure account has Microsoft.Authorization/*Write permission; Otherwise contact your Azure administrator to either give your account that permission or create the necessary resources for the app-based credential creation method.

Cloud not validate publickey certificate

Error: Could not validate publickey certificate [certificate: 'fdfdfsf'], detailed message: Corrupt or unknown public key file format

Solution: The syntax of your SSH public key is incorrect. You must correct the syntax of your SSH key. For information about the correct syntax, refer to [this](#) page.

Troubleshooting cluster creation

The following section lists common issues related to cluster creation.

Configure communication via private IPs on AWS

Cloudbreak uses public IP addresses when communicating with cluster nodes. On AWS, you can configure it to use private IPs instead of public IPs by setting the CB_AWS_VPC variable in the Profile.



Note:

This configuration is available for AWS only. Do not use it for other cloud providers.

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 AWS VPC identifier where you have deployed Cloudbreak:

```
export CB_AWS_VPC=your-VPC-ID
```

For example:

```
export CB_AWS_VPC=vpc-e261a185
```

3. Restart Cloudbreak by using `cbd restart`.

Cannot access Oozie web UI

Ext JS is GPL licensed software and is no longer included in builds of HDP 2.6. Because of this, the Oozie WAR file is not built to include the Ext JS-based user interface unless Ext JS is manually installed on the Oozie server.

If you add Oozie using Ambari 2.6.1.0 to an HDP 2.6.4 or greater stack, no Oozie UI will be available by default. Therefore, if you plan to use Oozie web UI with Ambari 2.6.1.0 and HDP 2.6.4 or greater, you must manually install Ext JS on the Oozie server host.

You can install Ext JS by adding the following PRE-AMBARI-START recipe:

```
export EXT_JS_VERSION=2.2-1  
export OS_NAME=centos6  
wget http://public-repo-1.hortonworks.com/HDP-UTILS-GPL-1.1.0.22/repos/  
$OS_NAME/extjs/extjs-$EXT_JS_VERSION.noarch.rpm
```

```
rpm -ivh extjs-$EXT_JS_VERSION.noarch.rpm
```

Make the following changes to the script:

- Change the EXT_JS_VERSION to the specific ExtJS version that you want to use.
- Change the OS_NAME to the name of the operating system. Supported values are: centos6, centos7, centos7-ppc.

The general steps are:

1. Be sure to review and agree to the Ext JS license prior to using this recipe.
2. Create a PRE-AMBARI-START recipe. For instructions on how to create a recipe, refer to [Add recipes](#).
3. When creating a cluster, choose this recipe to be executed on all host groups of the cluster.

Failed to retrieve the server's certificate

Cloudbreak-managed cluster fails with "Infrastructure creation failed. Reason: Failed to retrieve the server's certificate".

Error: Cloudbreak-managed cluster fails with "Infrastructure creation failed. Reason: Failed to retrieve the server's certificate".

Solution: The most common reasons for this error are related to using your own custom image. If you are using your own custom image:

- If using the CLI, you must send the imageId explicitly in the CLI in the cluster template request.
- There was a breaking change between Cloudbreak 2.4 and 2.7 versions. If you burned your image for Cloudbreak 2.4, and you would like to use Cloudbreak 2.7 or newer, you must burn a new image.

Quota limitations

Each cloud provider has quota limitations on various cloud resources, and these quotas can usually be increased on request. If there is an error message in Cloudbreak stating that there are no more available EIPs (Elastic IP Address) or VPCs, you need to request more of these resources.

To see the limitations visit the cloud provider's site:

- [AWS service limits](#)
- [Azure subscription and service limits, quotas, and constraints](#)
- [GCP resource quotas](#)

Example error on OpenStack

Cluster creation fails with the following error:

```
Infrastructure creation failed. Reason: Failed to create the
stack for CloudContext{id=3689, name='test-exisitngnetwork',
platform='StringType{value='OPENSTACK'}', owner='e0307f96-bd7d-4641-8c8f-
b95f2667d9c6'} due to: Resource CREATE failed: ResourceInError:
resources.ambari_volume_master_0_0: Went to status error due to "Unknown"
```

This may mean that the volumes that you requested exceed volumes available on your cloud provider account. When creating a cluster, on the advanced Hardware and Storage page of the create cluster wizard, try reducing the amount of requested storage. If you need more storage, try using a different region or ask your cloud provider admin to increase the resource quota for volumes.

Connection timeout when ports are not open

A common reason for connection timeout is security group misconfiguration.

In the cluster installation wizard, you must specify on which node you want to run the Ambari server. Cloudbreak communicates with this node to orchestrate the installation. A common reason for connection timeout is security group misconfiguration. Cloudbreak allows configuring different security groups for the different instance groups; however, there are certain requirements for the Ambari server node. For more information, refer to [Default cluster security group](#) documentation.

Blueprint errors

The following sections list common blueprint-related issues.

Invalid services and configurations

Ambari blueprints are a declarative definition of a cluster. With a blueprint, you specify a stack, the component layout, and the configurations to materialize a Hadoop cluster instance via a REST API without having to use the Ambari cluster install wizard.

Cloudbreak supports any type of blueprints, which is a common source of errors. These errors are only visible once the core infrastructure is up and running and Cloudbreak tries to initiate the cluster installation through Ambari. Ambari validates the blueprint and rejects it if it's invalid.

For example, if there are configurations for a certain service like Hive but Hive as a service is not mapped to any host group, the blueprint is invalid.

To fix these type of issues, edit your blueprint and then reinstall your cluster. Cloudbreak web UI has support for this so the infrastructure does not have to be terminated.

There are some cases when Ambari cannot validate your blueprint beforehand. In these cases, the issues are only visible in the Ambari server logs. To troubleshoot, check Ambari server logs.

Wrong HDP/HDF version

In the blueprint, only the major and minor HDP/HDF version should be defined (for example, "2.6"). If wrong version number is provided, the following error can be found in the logs:

```
5/15/2017 12:23:19 PM testcluster26 - create failed: Cannot use the
specified Ambari stack: HDPRepo
{stack='null'; utils='null'}
. Error: org.apache.ambari.server.controller.spi.NoSuchResourceException:
The specified resource doesn't exist: Stack data, Stack HDP 2.6.0.3 is not
found in Ambari metainfo
```

For correct blueprint layout, refer to the [Ambari cwiki](#) page.

Recipe errors

The following sections list common recipe related issues.

Recipe execution times out

If the scripts are taking too much time to execute, the processes will time out, as the threshold for all recipes is set to 15 minutes. To change this threshold, you must override the default value by adding the following to the cbd Profile file:

```
export CB_JAVA_OPTS=" -Dcb.max.salt.recipe.execution.retry=90"
```

This property indicates the number of tries for checking if the scripts have finished with a sleep time (i.e. the wait time between two polling attempts) of 10 seconds. The default value is 90. To increase the threshold, provide a number greater than 90. You must restart Cloudbreak after changing properties in the Profile file.

Recipe execution fails

It often happens that a script cannot be executed successfully because there are typos or errors in the script. To verify this you can check the recipe logs at `/var/log/recipes`. For each script, there will be a separate log file with the name of the script that you provided on the Cloudbreak web UI.

Troubleshooting Cloudbreak CLI

The following section lists common issues related to Cloudbreak CLI.

Special characters in blueprint name cause an error

When registering a blueprint via blueprint create CLI command, if the name of the blueprint includes one or more of the following special characters `@#$%|:&*`, you will get an error similar to:

```
cb blueprint create from-url --name test@# --url https://myurl.com/
myblueprint.bp
[1] 7547
-bash: application.yml: command not found
-bash: --url: command not found
~ # integration-test # 1 # time="2018-02-01T12:56:44+01:00" level="error"
msg="the following parameters are missing: url\n"
```

Solution: When using special characters in a blueprint name, make sure to use quotes; for example `"test@#"`:

```
cb blueprint create from-url --name "test@#" --url https://myurl.com/
myblueprint.bp
```