Cloudera DataFlow for Data Hub 7.3.1

Setting up your Edge Management cluster

Date published: 2019-12-16
Date modified: 2024-12-10

CLOUDERA

https://docs.cloudera.com/

Legal Notice

Contents

Checking prerequisites

Before you start creating your Edge Flow Management Data Hub Cluster, you need to ensure that you have set up the environment properly and have all the necessary accesses to use CDP Public Cloud. Use this checklist to verify that you meet all the requirements before you start creating the cluster.

- You have CDP login credentials.
- You have an available CDP environment.

    When you register your environment, make sure that the correct security access settings are configured. You need to enable SSH access and specify SSH key so that you can generate certificates for the agents. For more information on creating a CDP environment, see:

    - Working with AWS environments
    - Working with Azure environments
    - Working with GCP environments

- You have a running Data Lake. For more information on the Data Lake service in CDP environment, see Introduction to Data Lakes.

    ⚠️ **Important:** Make sure that the Runtime version of the Data Lake cluster matches the Runtime version of the Data Hub cluster that you are about to create. If these versions do not match, you may encounter warnings and/or errors.

- You have a CDP username and the predefined resource role of this user is EnvironmentAdmin.
- Your CDP user is synchronized to the CDP Public Cloud environment.

If you need more information about CDP basics, see Getting started as a user.

Creating your cluster

If you meet all prerequisites, you are ready to create a managed and secured Edge Flow Management cluster in CDP Public Cloud by using the prescriptive cluster definition available in Technical Preview.

Procedure

1. Log into the CDP web interface.
2. Navigate to Management Console Environments and select the environment where you want to create a cluster.

3. Click Create Data Hub.

The Provision Data Hub page is displayed:

4. Select Cluster Definition.

5. Select the appropriate Edge Flow Management cluster definition from the Cluster Definition dropdown depending on which cloud privider you are using.

There are three template options available:

- Edge Flow Management Light Duty for AWS
- Edge Flow Management Light Duty for Azure
- Edge Flow Management Light Duty for GCP

The cluster template referenced in the selected cluster definition determines which services are included in the cluster. The list of services is automatically displayed below the selected cluster definition name. It shows that the cluster definition contains the Edge Flow Manager.

6. Provide a cluster name and add tags you might need.

> **Note:**
>
> The name must be between 5 and 40 characters, it must start with a letter, and should only include lowercase letters, numbers, and hyphens.

You can define tags that will be applied to your cluster-related resources on your cloud provider account. For more information, see *Tags*.

7. Use the Configure Advanced Options section to customize the infrastructure settings.

For more information on these options, see the *Advanced cluster options* for your cloud environment.

8. Click Provision Cluster.

Results

The new Data Hub cluster appears on the Data Hubs tab of the Clusters page. You can follow the status of the provisioning process in the Status column. When your cluster is ready, its status changes to Running.

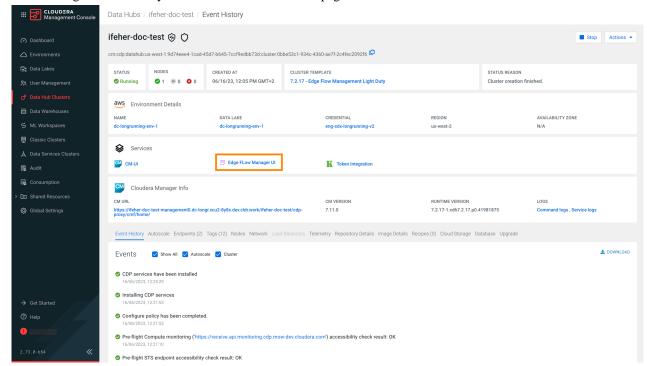# After creating your cluster

The cluster you have created using the Edge Flow Management cluster definition is secured by default, and it is integrated with Knox SSO.

You can access the EFM UI from the Services section of the Data Hub cluster page. Click the CEM icon or the Edge Flow Manager UI link and you are redirected to the EFM page.

The user who creates the Data Hub cluster is automatically designated as an administrator in EFM and gains immediate access to the UI.

EFM now integrates with CDP User Management and synchronizes all available users and groups. Besides the cluster creator user, all users who belong to the admin group also gain administrator rights, providing access to all EFM features. Users who do not belong to the admin group can still log in, but need access rights granted by an administrator before they can access the data in EFM.

For secure the agent-to-EFM communication, generate and use appropriate certificates. You also need to add the agents that you want to manage with EFM.

Generating certificates for MiNiFi agents

To secure the communication between agents and Edge Flow Manager, you need to generate and use proper certificates.

About this task

Edge Flow Manager is a secured application, which has to be bootstrapped with the initial admin identity. The initial admin is the person who is able to assign roles and manage permissions in Edge Flow Manager. In the Technical Preview, the initial admin is the workload user of the person who deploys the Data Hub. For more information about authentication and authorization, see *Access control bootstrapping*.

While the user traffic accessing the UI utilizes Knox, the agents running outside of the CDP deployment need to access Edge Flow Manager directly. To enable this, you have to open a port for the agents on the host where Edge

Flow Manager is deployed. By default, this port is 10090, used by Cloudera Edge Management components for C2 Protocol.

You do not have to generate the certificates from the agent host. You can generate them on any host that has access to the management node. When created, you can copy the certificates to the appropriate agent host.

In test environments it is not necessary to create different certificates for all agents. The same certificate can be configured for all agents. However, in production environments it is highly recommended to create a certificate for each agent.

Generating certificates with this approach is similar to adding a node to the cluster using Cloudera Manager.

> **Note:** Agents using these certificates are considered to be the members of the cluster managed by Cloudera Manager. Use your certificates with care and protect them from illegal access.

MiNiFi agents need to set up mutual TLS (mTLS) for C2 communication to be able to communicate with Edge Flow Manager. For information on MiNiFi Java Agent authentication, see *Securing MiNiFi Java Agent*. For information on MiNiFi C++ Agent authentication, see *Securing MiNiFi C++ Agent*.

In CDP Public Cloud, certificates are managed by Cloudera Manager, acting as a certificate authority. All certificates are generated by Cloudera Manager, there is no option to use custom certificates.

> **Note:**
>
> In the Technical Preview version of Cloudera Edge Management for CDP Public Cloud, you have to set up agent security manually. In later versions there will be an option to set up agent security using Edge Flow Manager.

Before you begin

- You have a running Cloudera Edge Management Public Cloud cluster
- SSH access is configured to the management node of the cluster
- You have an SSH user with keypair that has sudo privileges
- You have the host name of the Edge Management cluster's management node
- An external node is available from which you are able to SSH into the Edge Management cluster's management node

Procedure

1. Create a working directory on your external node that has SSH access to your Edge Flow Management cluster.

2. Save the following script to the previously created working directory, and name it create_certs.sh.

```
#!/bin/bash
set -eo pipefail

# input parameters
SSH_USER=$1
SSH_KEY=$2
CM_HOST=$3
AGENT_FQDN=$4

EXAMPLE_USAGE="Example usage: ./create_certs.sh sshUserName ~/.ssh/userKe
y.pem host0.company.site agent-x.company.site"

[[ -z "$SSH_USER" ]] && echo "SSH User parameter is missing. $EXAMPLE_US
AGE" && exit 1
[[ -z "$SSH_KEY" ]] && echo "SSH Key parameter is missing. $EXAMPLE_USAGE"
 && exit 1
[[ -z "$CM_HOST" ]] && echo "Cloudera Manager parameter is missing. $EXA
MPLE_USAGE" && exit 1
[[ -z "$AGENT_FQDN" ]] && echo "Agent FQDN parameter is missing. $EXAMPLE_
USAGE" && exit 1
```

```
KEYSTORE_PASSWORD=$(hexdump -vn16 -e'4/4 "%08X" 1 "\n"' /dev/urandom | tr
'[:upper:]' '[:lower:]')

# constants
GENERATED_CREDENTIALS_ARCHIVE=credentials.tar
GENERATED_CREDENTIALS_REMOTE_PATH="/tmp/$GENERATED_CREDENTIALS_ARCHIVE"
CM_SITE_PACKAGES="/opt/cloudera/cm-agent/lib/python3.8/site-packages"
ORIGINAL_CERTMANAGER_BASE_DIR="/etc/cloudera-scm-server/certs"
CUSTOM_CERTMANAGER_BASE_DIR="/root/certs"
CERT_PASSWORDS_DIR="$CUSTOM_CERTMANAGER_BASE_DIR/private"
GLOBAL_KEY_PASSWORD_FILE="$CERT_PASSWORDS_DIR/.global_key_password"
GLOBAL_TRUSTSTORE_PASSWORD_FILE="$CERT_PASSWORDS_DIR/.global_truststore
_password"

rm -rf "$AGENT_FQDN"
mkdir "$AGENT_FQDN"

remote_ssh_command=$(cat << EOF
sudo \cp -n -R $ORIGINAL_CERTMANAGER_BASE_DIR $CUSTOM_CERTMANAGER_BASE_DI
R;
sudo /opt/rh/rh-python38/root/bin/python -c "import site; site.addsitedir
('$CM_SITE_PACKAGES'); import cmf.tools.cert; passwd = cmf.tools.cert.re
ad_obfuscated_password('$GLOBAL_TRUSTSTORE_PASSWORD_FILE'); print(passwd
);"
sudo rm -f $GLOBAL_KEY_PASSWORD_FILE;
sudo /opt/rh/rh-python38/root/bin/python -c "import site; site.addsitedir(
'$CM_SITE_PACKAGES'); import cmf.tools.cert; cmf.tools.cert.write_obfusc
ated_password('$GLOBAL_KEY_PASSWORD_FILE', '$KEYSTORE_PASSWORD');";
sudo /opt/cloudera/cm-agent/bin/certmanager --location "$CUSTOM_CERTMANA
GER_BASE_DIR" gen_node_cert --output "$GENERATED_CREDENTIALS_REMOTE_PATH"
 --rotate "$AGENT_FQDN";
sudo chmod 666 "$GENERATED_CREDENTIALS_REMOTE_PATH";
EOF
)

ssh -i "$SSH_KEY" -o StrictHostKeyChecking=no "$SSH_USER"@"$CM_HOST" "$rem
ote_ssh_command" > "$AGENT_FQDN/cm-auto-in_cluster_trust.pw" 2> /dev/null
scp -r -i "$SSH_KEY" -o "StrictHostKeyChecking=no" "$SSH_USER"@"$CM_HOST":
"$GENERATED_CREDENTIALS_REMOTE_PATH" "$AGENT_FQDN/" 2> /dev/null
tar -xf "$AGENT_FQDN/$GENERATED_CREDENTIALS_ARCHIVE" -C "$AGENT_FQDN"
echo "MiNiFi-Java KeyStore File":
ls -alh "$AGENT_FQDN/cm-auto-host_keystore.jks"
echo "MiNiFi-Java TrustStore File:"
ls -alh "$AGENT_FQDN/cm-auto-in_cluster_truststore.jks"
echo "MiNiFi-CPP Client certificate":
ls -alh "$AGENT_FQDN/cm-auto-host_key_cert_chain.pem"
echo "MiNiFi-CPP Client private key":
ls -alh "$AGENT_FQDN/cm-auto-host_key.pem"
echo "MiNiFi-CPP CA certificate"
ls -alh "$AGENT_FQDN/cm-auto-in_cluster_ca_cert.pem"
echo "KeyStore / HostKey Password: sensitive data, please check for it in
 $AGENT_FQDN/cm-auto-host_key.pw"
echo "TrustStore Password: sensitive data, please check for it in $AGENT_
FQDN/cm-auto-in_cluster_trust.pw"

rm -f "$AGENT_FQDN/cm-auto-global_cacerts.pem" "$AGENT_FQDN/cm-auto-globa
l_truststore.jks" "$AGENT_FQDN/$GENERATED_CREDENTIALS_ARCHIVE" "$AGENT_F
QDN/cm-auto-host_cert_chain.pem"
```

3. Make the script executable.

```
chmod +x create_certs.sh
```

4. Run the script with the following parameters:

```
./create_certs.s
h **[ssh_user}** **[ssh_private_key]** **[management_node_host_name]** **[agent_fqdn]
```

For example:

```
./create_certs.sh adminuser ~/.ssh/adminuser.pem management-node.company
.site.com agent-1.company.site.com
```

The script should print a similar output:

```
credentials.tar


                                                       100%  420KB 222.0KB/s
   00:01
MiNiFi-Java KeyStore File:
-rw-------@ 1 user   group    5.2K Apr 24 13:33 agent-1.company.site.com/cm-
auto-host_keystore.jks
MiNiFi-Java TrustStore File:
-rw-r-----@ 1 user   group    2.3K Apr 24 13:19 agent-1.company.site.com/cm-
auto-in_cluster_truststore.jks
MiNiFi-CPP Client certificate:
-rw-------@ 1 user   group    7.1K Apr 24 13:33 agent-1.company.site.com/cm-
auto-host_key_cert_chain.pem
MiNiFi-CPP Client private key:
-rw-------@ 1 user   group    2.5K Apr 24 13:33 agent-1.company.site.com/cm-
auto-host_key.pem
MiNiFi-CPP CA certificate
-rw-r-----@ 1 user   group    3.0K Apr 24 13:19 agent-1.company.site.com/cm-
auto-in_cluster_ca_cert.pem
KeyStore / HostKey Password: sensitive data, please check for it in agent-
1.company.site.com/cm-auto-host_key.pw
TrustStore Password: sensitive data, please check for it in agent-1.compa
ny.site.com/cm-auto-in_cluster_trust.pw
```

A directory is created with the same name as the agent's FQDN, provided as a parameter for the script. The directory contains all the necessary keystores and certificates for configuring mTLS authentication.

The keystore and truststore passwords are not printed as they are sensitive information. You can find them in the directory that was created with the following names:

- cm-auto-host_key.pw
- cm-auto-in_cluster_trust.pw

5. Set the agent parameters.

- For the MiNiFi Java Agent:

```
c2.security.truststore.location=/path/to/cm-auto-in_cluster_truststore.j
ks
c2.security.truststore.password=<password_from_cm-auto-in_cluster_tru
st.pw>
c2.security.truststore.type=JKS
c2.security.keystore.location=/path/to/cm-auto-host_keystore.jks
c2.security.keystore.password=<password_from_cm-auto-host_key.pw>
c2.security.keystore.type=JKS
```

- For the MiNiFi C++ Agent:

```
nifi.security.client.certificate=/path/to/cm-auto-host_key_cert_chain.pe
m
nifi.security.client.private.key=/path/to/cm-auto-host_key.pem
```

```
nifi.security.client.pass.phrase=/path/to/cm-auto-host_key.pw
nifi.security.client.ca.certificate=/path/to/cm-auto-in_cluster_ca_ce
rt.pem
```

Note:

Although the parameter is called Agent FQDN, it is not mandatory to use the agent's domain name. You can use any other string. Keep in mind that the string you provide will be the common name (CN) in the generated certificate.

Adding agents to your cluster

When your cluster has been created successfully, you can add agents that you want to manage with EFM. Agents are deployed outside of CDP Public Cloud, so follow the standard agent deployment instructions:

Java Agents

Installing the MiNiFi Java Agent

C++ agents

Installing the MiNiFi C++ Agent

Note:

Make sure that you point the agents to heartbeat to the Data Hub EFM deployment.