Cloudera Operational Database ..

Working with Edge Nodes

Date published: 2020-08-14
Date modified: 2024-04-21

Legal Notice

Contents

COD edge node overview.....4

Managing edge nodes..... 4

Creating an edge node.....4
Viewing edge nodes..... 5
Adding an edge node.....6
Deleting an edge node.....6

Deploying applications on Cloudera Operational Database.....6

Configuring Kerberos..... 8
Configuring JWT authentication for HBase client..... 8

COD edge node overview

An edge node is a resource dedicated to access private computing resources on the public cloud. You must configure an edge node in your public cloud environment if you use the Apache HBase Java API or the Apache Phoenix thick JDBC client.

Cloudera Operational Database (COD) cannot be accessed directly by clients and resources on the public internet. The subnet security group and ingress rules of your public cloud providers prevent you from accessing your database from a public network.

If you have enabled a public endpoint access gateway while creating your environment, you can access your COD instance from outside the public cloud through Apache Knox. But, if you use a private subnet in your environment, you must configure a VPN for your client applications to access your COD instance using Apache Knox or use an edge node.

Clients that use HTTP interfaces such as the HBase REST server, Thrift client and server, Phoenix Query Server, SQL over HTTP using Apache Phoenix thin JDBC driver, ODBC driver, Go driver, and Python phoenixdb library, need not use the edge node and can be proxied through the Apache Knox gateway.

However, you must create an edge node to use the Apache HBase Java API or the Apache Phoenix thick JDBC driver.

You can create an edge node in your COD cluster that acts as an individual node type and not as a separate Data Hub cluster. The edge node automatically synchronizes with the COD cluster, which means you do not need to manually configure the node.

Related Information

Configure edge node on AWS

Configure edge node on Azure

Configure edge node on GCP

Managing edge nodes

Cloudera Operational Database (COD) supports creation of an edge node while creating an operational database. You can also modify the number of instances in the edge node type.

You must create an edge node if you plan to use Apache HBase Java API or the Apache Phoenix thick JDBC driver. You can use the edge node either as a gateway node or use it to deploy your applications while working with a COD cluster.

When you create an edge node in the COD cluster, it acts as an edge node type and not as a separate edge node Data Hub cluster.

Related Information

Create database

List edge nodes

Update edge nodes

Creating an edge node

You can create an edge node while creating an operational database in your CDP environment. You can also define the number of nodes to be created.

Procedure

1. Launch the CDP CLI tool.
2. Use the following command to create an edge node.

   `cdp opdb create-database --environment-name` <*CDP_environment_name*> `--database-name` <*database_name*> `--num-edge-nodes` <*number_of_edge_nodes*>

   For example,

   ```
   cdp opdb create-database --environment-name cdp1 --database-name test1 --
   num-edge-nodes 1
   ```

Results

What to do next

Viewing edge nodes

You can view the list of available edge nodes on a COD cluster.

Procedure

1. Launch the CDP CLI tool.
2. Use the following command to list the available edge nodes.

   `cdp opdb list-edge-nodes --environment` <*CDP_environment_name*> `--database` <*database_name*>

   For example,

   ```
   cdp opdb list-edge-nodes --environment odx-i2dr46 --database edge2
   ```

   Sample output:

   ```
   "database": "edge2",
       "edgeNodes": [
           {
               "instanceId": "cod--1qs4ckni9gwnj112955e22",
               "discoveryFQDN": "cod--1qs4ckni9gwnj-edge1.odx-i2dr.xcu2-8y8x
   .dev.cldr.work",
               "privateIp": "10.124.64.24",
               "publicIp": "N/A"
           },
           {
               "instanceId": "cod--1qs4ckni9gwnj112955e21",
               "discoveryFQDN": "cod--1qs4ckni9gwnj-edge2.odx-i2dr.xcu2-8y8x.
   dev.cldr.work",
               "privateIp": "10.124.64.26",
               "publicIp": "N/A"
           },
           {
               "instanceId": "cod--1qs4ckni9gwnj112955e20",
               "discoveryFQDN": "cod--1qs4ckni9gwnj-edge0.odx-i2dr.xcu2-8y
   8x.dev.cldr.work",
               "privateIp": "10.124.64.23",
               "publicIp": "N/A"
           },
           {
               "instanceId": "cod--1qs4ckni9gwnj112955e23",
   ```

```
            "discoveryFQDN": "cod--1qs4ckni9gwnj-edge3.odx-i2dr.xcu2-8y8
x.dev.cldr.work",
            "privateIp": "10.124.64.25",
            "publicIp": "N/A"
        }
    ],
    "environmentName": "odx-i2dr46"
}
```

Results

What to do next

Adding an edge node

You can add additional edge nodes into your existing COD cluster. You can define the number of edge nodes to be added, COD automatically adds the additional nodes into the COD cluster.

Procedure

1. Launch the CDP CLI tool.
2. Use the following command to add edge nodes into the COD cluster.

   `cdp opdb update-edge-nodes --environment` *<CDP_environment_name>* `--database` *<database_name>* `--num-add-edge-nodes` *<number_of_edge_nodes>*

   For example,

   ```
   cdp opdb update-edge-nodes --environment cdp1 --database cod-7215 --num-
   add-edge-nodes 2
   ```

Deleting an edge node

You can delete single or multiple existing edge nodes from a COD cluster. You can use the instance ID of an edge node to delete it from the COD cluster.

Procedure

1. Launch the CDP CLI tool.
2. Use the following command to delete an existing node from a COD cluster.

   `cdp opdb update-edge-nodes --environment` *<CDP_environment_name>* `--database` *<database_name>* `--delete-edge-instances` *<edge_node_instance_ID_1>* *<edge_node_instance_ID_2>*

   For example,

   ```
   cdp opdb update-edge-nodes --environment cdp1 --database cod-7215 --dele
   te-edge-instances cod--1msrexj6oumro108509e10 cod--1msrexj6oumro108509e12
   ```

Deploying applications on Cloudera Operational Database

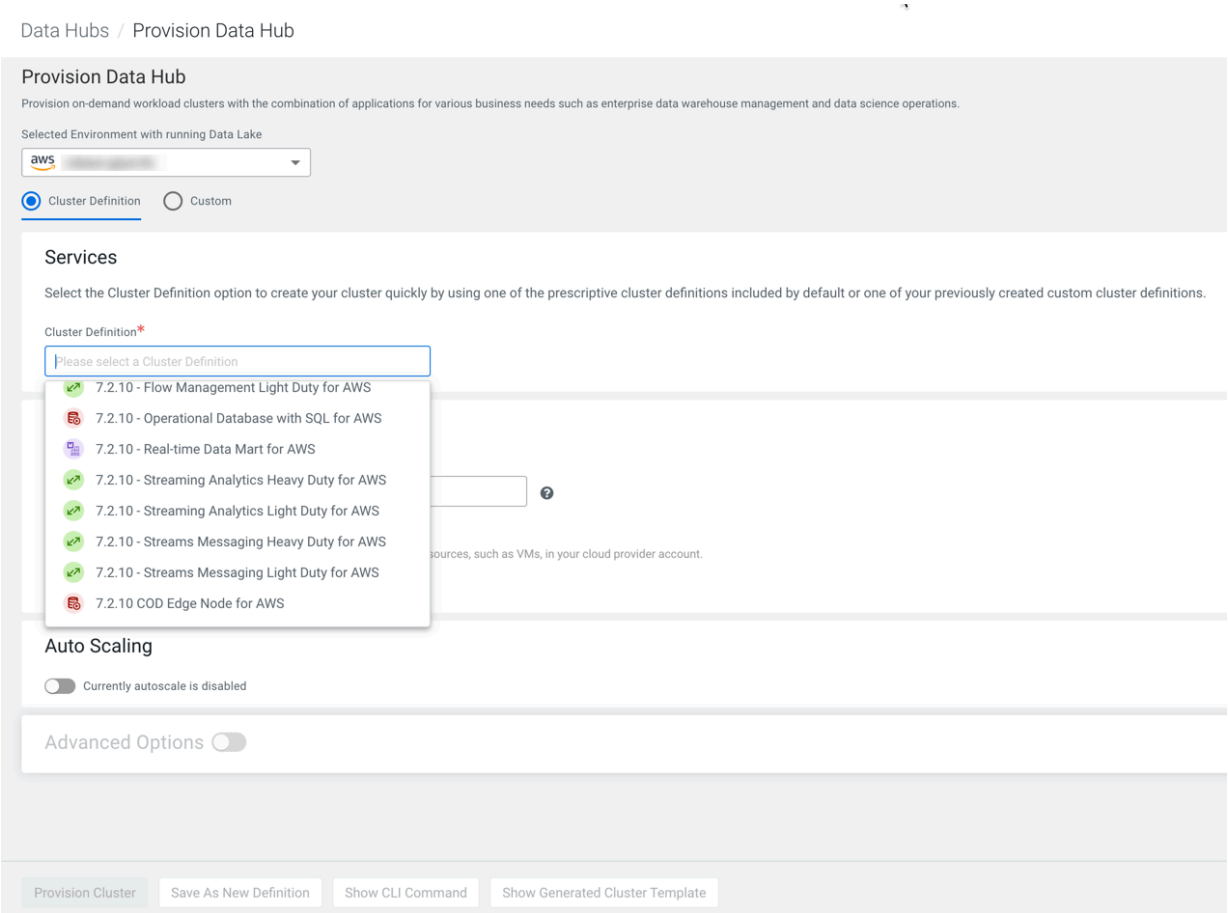
The edge node is a dedicated Data Hub cluster that enables you to communicate with your Cloudera Operational Database (COD) instance and your applications. You can deploy a cluster that works as an edge node to access your

COD instance. Deploy the edge node cluster in the same environment as the COD instance to ensure that the security groups and data ingress rules that apply to the COD instance must also apply to the edge node cluster.

Procedure

1. From the Cloudera Management Console, click Data Hub Clusters.
2. Click Create Data Hub.
3. In the Selected Environment with running Data Lake drop-down list, select the same environment used by your COD instance.
4. Select the Cluster Definition.
5. In the Cluster Definition drop-down list, select the [***RUNTIME VERSION***] COD Edge Node for [***CLOUD PROVIDER NAME***].

   For example, select the 7.2.10 COD Edge Node for AWS cluster template.



6. In the Cluster Name field, provide a cluster name that you can identify later as an edge node of a specific COD instance.
7. Click Provision Cluster.

What to do next

After you deploy the edge node, you can run your applications on this edge node using the Client connectivity information. See how to compile applications for COD in Compile an application for your database.

Configuring Kerberos

All Cloudera Operational Databases (CODs) are secured with Kerberos-based authentication, meaning that only authorized users can connect to your database. All HBase and Phoenix Thick JDBC clients must have a proper Kerberos configuration on the host where they run a client.

Procedure

1. Run the following command to obtain the necessary Kerberos information and a sufficient krb5.conf file encoded with Base64 :

```
$ cdp opdb describe-client-connectivity --environment-name [***YOUR
 ENVIRONMENT***] \
  --database-name [***YOUR DATABASE NAME***] | jq -r \
  '.kerberosConfiguration.krb5Conf' | base64 --decode
```

2. Copy the output of the command.
3. Add the contents into the /etc/krb5.conf file on your edge node.

What to do next

Validate that Kerberos is correctly set up. Use the kinit command to validate that you are able to obtain a Kerberos ticket:

```
$ kinit [***CDP WORKLOAD NAME***]
Password: [***CDP WORKLOAD PASSWORD***]
```

For more information, see *CDP workload user* and *Setting the workload password*.

If you successfully authenticate, you will not receive an error and will be able to validate that you have a ticket using the klist command. For more information, see *Installing CDP CLI beta*.

Related Information

Setting the workload password

CDP workload user

Installing CDP CLI beta

Configuring JWT authentication for HBase client

JWT (JSON Web Token)-based authentication uses an unique identifier and is a standard way of securely transmitting signed information between two parties. Learn how to configure JWT-based authentication for your HBase client.

About this task

JSON Web Token (JWT) is a compact, URL-safe means of representing claims to be transferred between two parties. The claims in a JWT are encoded as a JSON object that is used as the payload of a JSON Web Signature (JWS) structure or as the plaintext of a JSON Web Encryption (JWE) structure, enabling the claims to be digitally signed or integrity protected with a Message Authentication Code (MAC) and/or encrypted. The structure of JWT allows you to verify whether the content is tampered.

To disable JWT authentication for HBase clients, you can use the --disable-jwt-auth option while creating an operational database using COD CLI. Ensure that the COD_JWT_AUTH entitlement is enabled for the HBase client.

For example,

`cdp opdb create-database --environment-name` *myEnvironment* `--database-name` *myDatabase* `--disable-jwt-auth`

Important:

- HBase client tarball contains all the binary dependencies that are required for JWT to function smoothly.
- You cannot perform MapReduce operations such as RowCounter and bulk-load while using JWT authentication.

Before you begin

- CDP CLI must have been configured to access CDP environments.
- Ensure that you have COD_JWT_AUTH entitlement enabled for your HBase client.

Procedure

1. On the COD UI, click on the database and go to Connect HBase Client Tarball JWT Configuration . This section provides you the necessary details in setting up a connection to HBase with a JWT token.

Connect Charts Events

HBase | HBase REST | HBase Client Tarball | Phoenix (Thick)

Usage

You can download the Apache HBase Client Tarball that contains the .
such as HBase Shell or SQLLine.

HBase Version

x.x.x.x.x.x.x.x.x.x.x.x

Download URL

https://cod--xnz769xcwvk3-gateway0.cod-7216.xcu2-8y8x.dev.cldr.w

HBase Client Configuration URL

curl -f -o "hbase-config.zip" -u "csso_l xxxxxxxxxxxx "https://cod--xnz7

> Kerberos Configuration

> Yarn Configuration

JWT Configuration

Download Environment Certificate

Run the following command to create the truststore:

```
keytool -importcert -noprompt -storetype JKS -keystore truststore j
```

hbase-site.xml

```
<!-- TLS -->

<property>
  <name>hbase.client.netty.tls.enabled</name>
```

2. Download the Environment Certificate and run the command as mentioned on the UI to build your own truststore JKS file.
3. Open the HBase client's hbase-site.xml file. The file is usually located in /etc/hbase/conf.

   Download the configuration snippet from the UI or add the following TLS and JWT properties to the hbase-site.xml file filling in the template based on your local configuration.

```
<!-- TLS -->

<property>
    <name>hbase.client.netty.tls.enabled</name>
    <value>true</value>
</property>
<property>
   <name>hbase.rpc.tls.truststore.location</name>
   <value>/path/to/truststore.jks</value>
</property>
<property>
    <name>hbase.rpc.tls.truststore.password</name>
    <value>...</value>
</property>
<property>
    <name>hbase.rpc.tls.truststore.type</name>
    <value>jks</value>
</property>

<!-- JWT -->
<property>
  <name>hbase.client.sasl.provider.extras</name>
  <value>com.cloudera.hbase.security.provider.OAuthBearerSaslClientAuth
enticationProvider</value>
</property>
<property>
  <name>hbase.client.sasl.provider.class</name>
  <value>com.cloudera.hbase.security.provider.OAuthBearerSaslProviderSelec
tor</value>
</property>
<property>
  <name>hbase.client.sasl.oauth.tokenprovider</name>
  <value>com.cloudera.hbase.security.token.FileOAuthBearerTokenProvider</
value>
</property>
<property>
  <name>hbase.client.sasl.oauth.tokenfile</name>
  <value>/path/to/token.txt</value>
</property>
```

   Note: Cloudera recommends that you set this client configurations in hbase-site.xml file using the CDP CLI client instead of Cloudera Manager. If you set using Cloudera Manager, you might encounter problems.

4. Ensure that the following JWT libraries (included in the HBase client tarball) are added on the classpath.

```
cloudera-opdb-jwtauth-client-1.0.0.7.2.16.0-SNAPSHOT.jar
cloudera-opdb-jwtauth-common-1.0.0.7.2.16.0-SNAPSHOT.jar
nimbus-jose-jwt-9.15.2.jar
```

5. Obtain the JWT token from the IAM service or the console authentication service.

```
cdp iam generate-workload-auth-token --workload-name OPDB
```

The command returns a JWT token in a JSON format. For example,

```
{
    "token": "eyJraWQiOiJjMDBjNmRlNGE1MjIyYTk1IiwidHlwIjo...",
    "expireAt": "2022-03-17T17:10:32.472000+00:00"
}
```

6. Copy and paste the base64 encoded token into a TXT file (token.txt) with username. For example,

```
<username>,<token>
cloudbreak,eyJraWQiOiJjMDBjNmRlNGE1MjIyYTk1Iiwid…
```

What to do next

Validate that JWT is correctly set up. Use the following `list` command to validate that you are able to run commands on HBase.

```
bin/hbase shell
hbase> list
```

After successful authentication, you can see the list of available tables in the database.

Related Information

Create database