

## Working with Edge Nodes

Date published: 2020-08-14

Date modified: 2025-09-01

# CLOUDERA

# Legal Notice

© Cloudera Inc. 2025. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

- Cloudera Operational Database 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
- Application deployment on Cloudera Operational Database.....6**
  - Configuring an on-prem instance to connect to Cloudera Operational Database.....7
    - Connecting an on-prem instance to the Cloudera Operational Database on AWS.....7
    - Connecting an on-prem instance to the Cloudera Operational Database on Azure.....7
    - Connecting an on-prem instance to the Cloudera Operational Database on GCP.....7
  - Configuring DNS.....8
  - Verifying the DNS configuration.....8
  - Configuring Kerberos.....8
  - Configuring JWT authentication for HBase client..... 9

## Cloudera Operational Database 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 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 Cloudera Operational Database 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 Cloudera Operational Database 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 Cloudera Operational Database cluster that acts as an individual node type and not as a separate Cloudera Data Hub cluster. The edge node automatically synchronizes with the Cloudera Operational Database 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 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 Cloudera Operational Database cluster.

When you create an edge node in the Cloudera Operational Database cluster, it acts as an edge node type and not as a separate edge node Cloudera 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 Cloudera 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
```

## Viewing edge nodes

You can view the list of available edge nodes on a Cloudera Operational Database 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--lqs4ckni9gwnj112955e22",
      "discoveryFQDN": "cod--lqs4ckni9gwnj-edge1.odx-i2dr.xcu2-8y8x
.dev.cldr.work",
      "privateIp": "10.124.64.24",
      "publicIp": "N/A"
    },
    {
      "instanceId": "cod--lqs4ckni9gwnj112955e21",
      "discoveryFQDN": "cod--lqs4ckni9gwnj-edge2.odx-i2dr.xcu2-8y8x.
dev.cldr.work",
      "privateIp": "10.124.64.26",
      "publicIp": "N/A"
    },
    {
      "instanceId": "cod--lqs4ckni9gwnj112955e20",
      "discoveryFQDN": "cod--lqs4ckni9gwnj-edge0.odx-i2dr.xcu2-8y
8x.dev.cldr.work",
      "privateIp": "10.124.64.23",
      "publicIp": "N/A"
    },
    {
      "instanceId": "cod--lqs4ckni9gwnj112955e23",
      "discoveryFQDN": "cod--lqs4ckni9gwnj-edge3.odx-i2dr.xcu2-8y8
x.dev.cldr.work",
      "privateIp": "10.124.64.25",
      "publicIp": "N/A"
    }
  ]
}
```

```
  ],  
  "environmentName": "odx-i2dr46"  
}
```

## Adding an edge node

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

### Procedure

1. Launch the CDP CLI tool.
2. Use the following command to add edge nodes into the Cloudera Operational Database 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 Cloudera Operational Database cluster. You can use the instance ID of an edge node to delete it from the Cloudera Operational Database cluster.

### Procedure

1. Launch the CDP CLI tool.
2. Use the following command to delete an existing node from a Cloudera Operational Database 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
```

## Application deployment on Cloudera Operational Database

Know how to deploy your applications on your Cloudera Operational Database cluster.

You can create an edge node inside your Cloudera Operational Database instance. Deploy the edge node cluster in the same environment as the Cloudera Operational Database instance to ensure that the security groups and data ingress rules that apply to the Cloudera Operational Database instance must also apply to the edge node cluster.

For more information on creating an edge node, see *Creating an edge node*.

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 Cloudera Operational Database in *Compile an application for your database*.

#### Related Information

[Creating an edge node](#)

[Client connectivity information](#)

[Compile an application for your database](#)

## Configuring an on-prem instance to connect to Cloudera Operational Database

You can configure an on-prem instance or node to connect to the Cloudera Operational Database deployed on an AWS, Azure, or GCP environment to run applications using Apache HBase Java API or the Apache Phoenix thick JDBC driver. When configuring a node, you have to configure a DNS and Kerberos

In Cloudera Operational Database, by default, your database is not accessible to the public internet due to security groups on the subnets of the VPC in which the database is deployed and the ingress rules of the VPC itself. Configuring an on-prem instance correctly can run applications using the Apache HBase Java API or the Apache Phoenix thick JDBC driver.

Ensure that the following prerequisites are met before connecting the on-prem instance to the Cloudera Operational Database.

- You have the required permission to log into the Cloudera Operational Database on AWS, Azure, or GCP configured with Cloudera.
- You have the required permission to launch EC2 nodes in your AWS account.
- You have a basic understanding of the VPC and subnets created for use with Cloudera.

### Connecting an on-prem instance to the Cloudera Operational Database on AWS

Learn how to connect an on-prem instance to the Cloudera Operational Database deployed on an AWS environment.

To connect an on-prem instance to the Cloudera Operational Database on AWS, see the AWS documentation, *Connect your VPC to other networks*.

#### Related Information

[Connect your VPC to other networks](#)

### Connecting an on-prem instance to the Cloudera Operational Database on Azure

Learn how to connect an on-prem instance to the Cloudera Operational Database deployed on an Azure environment.

To connect an on-prem instance to the Cloudera Operational Database on Azure, see the Microsoft documentation, *Connect an on-premises network to a Microsoft Azure virtual network*.

#### Related Information

[Connect an on-premises network to a Microsoft Azure virtual network](#)

### Connecting an on-prem instance to the Cloudera Operational Database on GCP

Learn how to connect an on-prem instance to the Cloudera Operational Database deployed on a GCP environment.

To connect an on-prem instance to the Cloudera Operational Database on GCP, see the Google documentation, *Cloud Interconnect overview*.

#### Related Information

[Cloud Interconnect overview](#)

## Configuring DNS

You must configure your instance or node to perform forward and reverse DNS lookups with your Cloudera Operational Database. Obtain the private IP address of your Cloudera environment and configure your instance to resolve hostnames from your Cloudera Operational Database. An instance must resolve the hostnames from your Cloudera Operational Database.

### About this task

Each Cloudera environment acts as its own DNS nameserver. Obtain your environment's private IP address information and configure `resolv.conf` on your instance to list your Cloudera environment's private IP address as a nameserver.

### Procedure

1. Go to the Cloudera Management Console.
2. Click Environments and select your environment from the list.
3. Click Summary.
4. Find the FreeIPA section and copy the contents of the Private IP field.
5. Add the private IP address as a nameserver to the `/etc/resolv.conf` file in your instance. You can do this using the `cat resolv.conf nameserver [***NAMESERVER IP ADDRESS***]` command.



**Note:** When an HBase client application running on the server cannot be set up with reverse DNS lookup and can only perform forward DNS, then the `hbase.unsafe.client.kerberos.hostname.disable.reversedns` property can be set to `true` at the client configuration so that the HBase client connects to the HBase cluster through SASL Kerberos using the hostname of the principal and skips the reverse DNS lookup.

## Verifying the DNS configuration

After configuring the nameserver you must verify that your instance can now resolve DNS names.

### Procedure

1. Navigate to Cloudera Control Plane Cloudera Management Console .
2. Click Environments and select your environment from the list.
3. Click Summary.
4. Find the FreeIPA section and copy the content of the Private IP field.
5. Open a terminal.
6. Run the following command: `$ nslookup [***FULLY QUALIFIED DOMAIN NAME***]`  
If your DNS is set up correctly, this command returns an address for the name you provided.

## Configuring Kerberos

All Cloudera Operational Databases (CODs) are secured with Kerberos-based authentication, which means only authorized users can connect to your database. All HBase and Phoenix Thick JDBC clients must have a 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 instance.

### 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 [***CLOUDERA WORKLOAD NAME***]  
Password: [***CLOUDERA WORKLOAD PASSWORD***]
```

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

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

After configuring the Kerberos successfully, you must compile your instance or application against the Cloudera Operational Database database. For more information, see *Compiling Applications*.

### Related Information

[Setting the workload password](#)

[Cloudera workload user](#)

[Installing CDP CLI beta](#)

[Compile an application for your COD database](#)

## 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 CDP 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 Cloudera environments.
- Ensure that you have `COD_JWT_AUTH` entitlement enabled for your HBase client.

**Procedure**

1. On the Cloudera Operational Database 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
```

### Download URL ⓘ

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

### HBase Client Configuration URL ⓘ

```
curl -f -o "hbase-config.zip" -u "csso_l xxxxxxxxxxxxxx" 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": "eyJraWQiOiJjMDBjNmRlNGE1MjIyYTklIiwidHlwIjo... ",
  "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,eyJraWQiOiJjMDBjNmRlNGE1MjIyYTklIiwidHlwIjo...
```

### 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](#)