



Collibra Data Intelligence Cloud

# Edge Infrastructure

## Collibra Data Intelligence Cloud - Edge Infrastructure

Release date: August 7th, 2022

Revision date: Thu Aug 04, 2022

You can find the most up-to-date technical documentation on our Documentation Center at

[https://productresources.collibra.com/docs/collibra/latest/Content/Edge/to\\_edge.htm](https://productresources.collibra.com/docs/collibra/latest/Content/Edge/to_edge.htm)

# Contents

Contents .....	ii
Introducing Edge .....	viii
Edge components .....	viii
Integration steps .....	viii
Edge security .....	x
Edge components .....	xi
Communication between Edge and Collibra .....	xiv
Authentication to data sources .....	xvi
Storing secrets .....	xvii
Customer Credentials .....	xix
Credentials storage .....	xix
Secret encryption .....	xix
Credential encryption .....	xix
Credentials transfer .....	xx
Platform credentials .....	xx
Data samples in Edge .....	xxi
Edge service repository .....	xxii
Monitoring and logging .....	xxiii
Installing an Edge site .....	xxiv
About an Edge site installation .....	xxv
Properties .....	xxv
Statuses .....	xxvi
Installation directories on K3S .....	xxvii

System requirements of an Edge site .....	xxviii
Software requirements .....	xxviii
Hardware requirements .....	xxviii
Network requirements .....	xxx
EKS requirements .....	xxx
Software requirements .....	xxxii
Hardware requirements .....	xxxii
Network requirements .....	xxxiii
Create an Edge site .....	xxxiv
Prerequisites .....	xxxiv
Steps .....	xxxiv
What's next? .....	xxxv
Configure a forward proxy .....	xxxvi
Steps .....	xxxvi
What's next? .....	xl
Install an Edge site .....	xlii
Prerequisites .....	xlii
Steps .....	xlii
Host hardening .....	xlvii
Prerequisites .....	xlvii
Enable host hardening .....	xlvii
Disable host hardening .....	xlviii
Enable or disable classification on an Edge site .....	xlix
Enable classification .....	xlix
Disable classification .....	l
Reinstall an Edge site using backup and restore .....	li

Upgrade the operating system of an Edge site .....	lii
Steps .....	lii
Troubleshooting .....	lii
Custom Docker Registry .....	liv
Enable private docker registry .....	liv
Configure private docker registry .....	lv
JDBC connections .....	lvi
Data sources supported by Edge .....	lvii
Create a JDBC connection .....	lviii
Available Catalog connectors .....	lviii
Edit a JDBC connection .....	lix
Available Catalog connectors .....	lix
Delete a JDBC connection .....	lx
Prerequisites .....	lx
Steps .....	lx
Use keys to access a database .....	lxi
Edge capabilities .....	lxii
About Edge capabilities .....	lxiii
Capability templates .....	lxiii
Capability template structure .....	lxiii
Page layout .....	lxiv
About Edge capabilities connecting to data sources .....	lxvi
Connection types .....	lxvi
Add an Edge capability to an Edge site .....	lxvii
Prerequisites .....	lxvii
Steps .....	lxvii

What's next? .....	lxxiv
Edit an Edge capability of an Edge site .....	lxxv
Prerequisites .....	lxxv
Steps .....	lxxv
Delete an Edge capability from an Edge site .....	lxxxiii
Prerequisites .....	lxxxiii
Steps .....	lxxxiii
Edge Jobs page .....	lxxxiv
Cancel jobs .....	lxxxv
Prerequisites .....	lxxxv
Steps .....	lxxxv
Maintaining Edge sites .....	lxxxvii
Running Edge tools .....	lxxxviii
Prepare the Edge tools on K3S .....	lxxxviii
Overview Edge commands on K3S .....	lxxxviii
Prepare Edge tools on EKS .....	lxxxix
Overview Edge commands on EKS .....	xc
Edit an Edge site .....	xcii
Prerequisites .....	xcii
Steps .....	xcii
Update Edge user password .....	xciii
Steps .....	xciii
Update the outbound proxy configuration .....	xciv
Steps .....	xciv
Help file of the script .....	xciv
Back up an Edge site .....	xcv

Restore an Edge site .....	xcvi
Delete an Edge site .....	xcviii
Prerequisites .....	xcviii
Steps .....	xcviii
Troubleshooting Edge .....	c
General troubleshooting Edge .....	ci
Use an explicit resolv.conf file for Edge .....	ciii
Edge logging .....	civ
Edge diagnostics file .....	civ
Edge infrastructure log files .....	civ
Metadata connector log files .....	cv
Edge system monitoring .....	cv
Create an Edge diagnostics file .....	cvii
Prerequisites .....	cvii
Steps .....	cvii
What's next? .....	cviii
Create Metadata connector log files .....	cix
Prerequisites .....	cix
Steps .....	cix
Prerequisites .....	cx
Steps .....	cx
Enable debug logging for Edge infrastructure logs .....	cxii
Prerequisites .....	cxii
Steps .....	cxii
Disable OpenTelemetry .....	cxiii
Disable OpenTelemetry at installation time .....	cxiii

Edge FAQ ..... cxiv

# Introducing Edge

Edge is a cluster of Linux servers for accessing and processing data close to where it resides. It helps to connect to data sources and process information within your data landscape.

Edge enables Collibra Data Intelligence Cloud to [safely](#) connect to your data sources hosted in an on-premise or cloud environment. It processes the data source information on the Edge site and sends the process results to Collibra Data Intelligence Cloud.

## Edge components

Edge consists of three main components:

- An Edge configuration page in Collibra Data Intelligence Cloud to create and install Edge sites.
- An Edge integration capability repository that resides on the Collibra Platform and contains all capabilities that can run on an Edge site.
- An [Edge site](#) that is installed close to a data source in the customer's environment, whether it's in the cloud or on the customer's premises.

## Integration steps

The following table shows which steps you have to take to set up Edge.

Step	Description	Required permissions
1	<a href="#">Create</a> an Edge site via Collibra Data Intelligence Cloud Settings.	You have a global role with the Manage Edge sites global permission in Collibra Data Intelligence Cloud.

Step	Description	Required permissions
2	<p><a href="#">Install</a> the Edge site close to the data source you want to access.</p> <p>You can only install an Edge site on a Linux system that meets the necessary <a href="#">system requirements</a>.</p>	<p>You have a global role with the Install Edge sites global permission in Collibra Data Intelligence Cloud.</p>
3	<p><a href="#">Update</a> the credentials of the Edge site user.</p>	<p>You have a global role with the Connect Edge sites to Collibra global permission in Collibra Data Intelligence Cloud.</p>

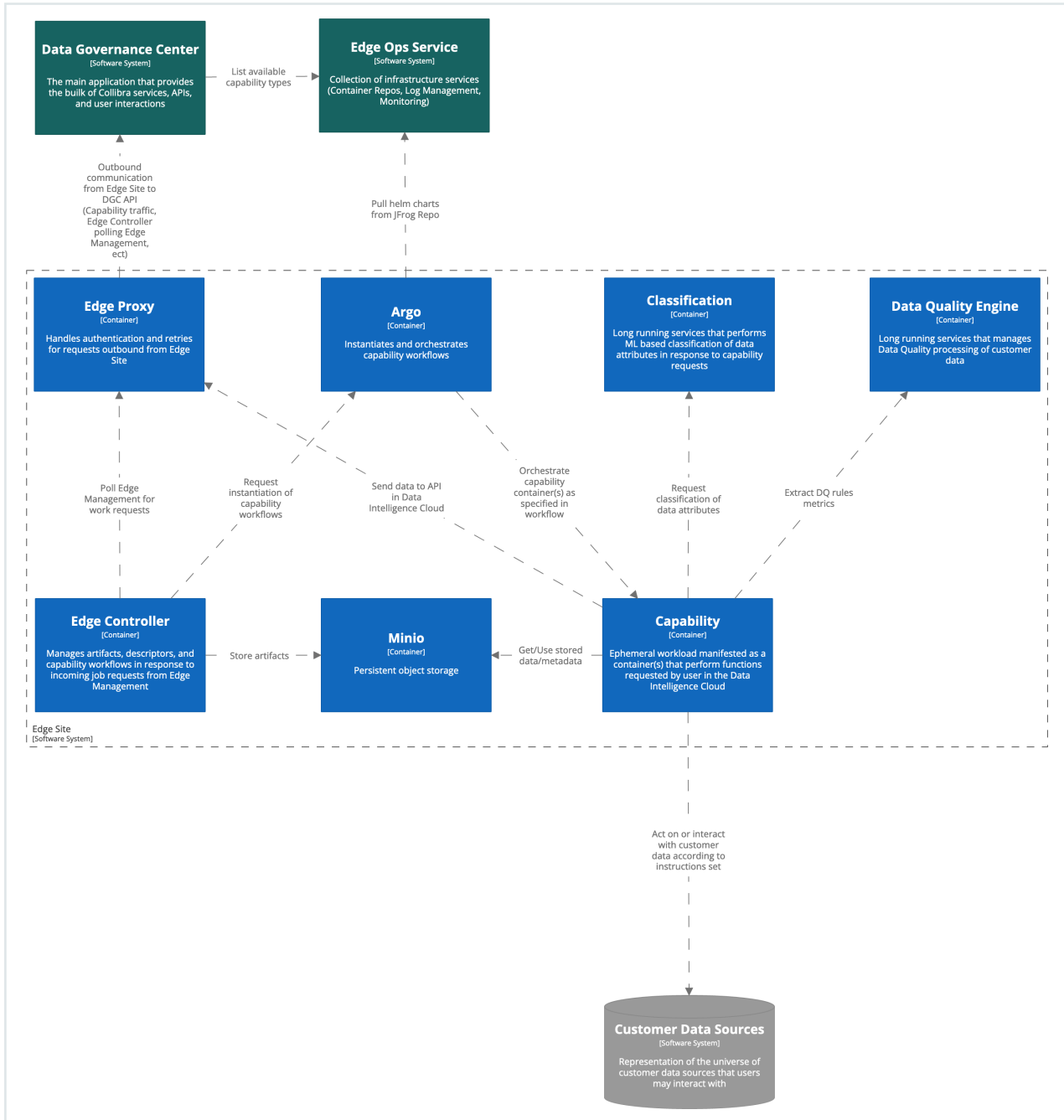
# Edge security

Edge is built with security first approach. All communication channels are secured by TLS 1.2 and all endpoints outside Edge are accessible only via authentication. Edge does not send or store any customer data, its purpose is to host capabilities that process the data in its own environment and to send only processing results to Collibra Data Intelligence Cloud.

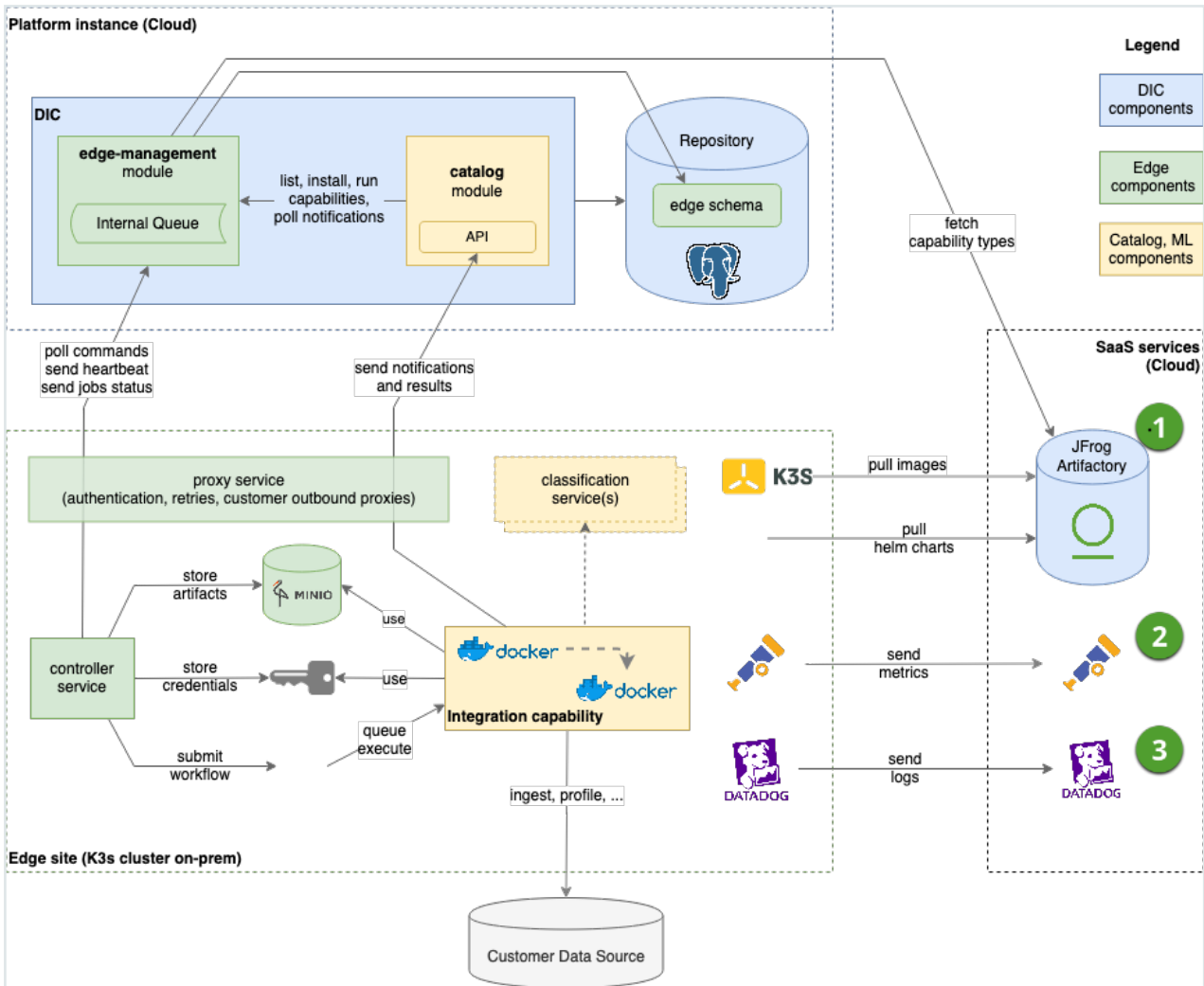


# Edge components

In the following overview, you see a detailed overview of the different communication flows between the Edge components and what they are responsible for.



In the following overview, you see the interaction between Edge and the other systems such as Colibra Data Intelligence Cloud or third-party components.



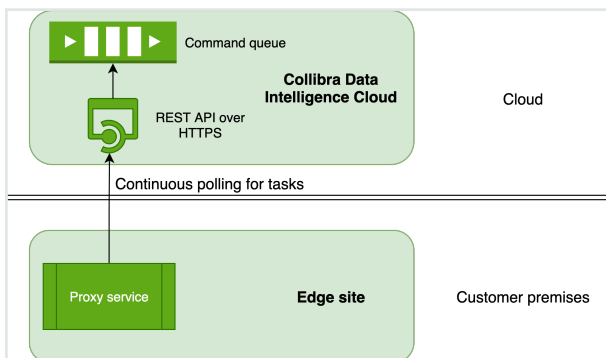
Edge requires access to the following servers:

Server	Communication
JFrog Artifactory	This is needed in order to download Helm Charts and Docker Images that are running onEdge.
OpenTelemetry Backbone	This is needed in order to upload various Edge related metrics.

Server	Communication
DataDog	<p>This is needed in order to upload logs from all Edge components:</p> <ul style="list-style-type: none"><li>• Core edge components</li><li>• Edge capabilities, for example, ingestion, profiling, lineage, classification, quality.</li></ul>

# Communication between Edge and Collibra

Edge does not need incoming access - it executes tasks as commands stored in the Edge management queue in Collibra. Communication to Collibra uses basic authentication over TLS 1.2. A user account for communicating to Collibra is generated on each Edge site installer download, and it is unique for each Edge site. It is possible to change the password of this user account through a [documented procedure](#).



- Edge sites always use REST API endpoints to establish connections.
- Edge does not store data after the data is processed on the Edge site, not even sample data.
- Edge requires access to a Collibra server. It is needed for:
  - Reading a request queue, a queue with jobs which need to be run on Edge.
  - Sending back results of jobs.
- Edge manages Collibra Data Intelligence Cloud and data source credentials. This has the following consequences:
  - Credentials are not accessible outside of Edge.
  - Credentials used on an Edge site are encrypted with a key that is [secured](#) in Collibra.
  - Credentials of data sources and Collibra can be updated if necessary.
- All configuration parameters, files or strings marked as secret, are stored on the Edge site encrypted with a public key that resides in Collibra. The private part of that key is encrypted with a public key from the Edge site. As a result, secrets can only be decrypted with both key pairs, one residing on the Edge site and the other on Collibra.

- An Edge site communicates in a secure way with your Collibra environment using certificates, issued by a Collibra-chosen Certificate Authority (CA). However, if there is a forward proxy server between the Edge site and Collibra, you have to use the [proxy server's CA](#).

# Authentication to data sources

Edge connections and capabilities use different ways to connect to data sources. The required level of privileges or security greatly depends on the data source type and supported Catalog Connectors.

Collibra regularly adds and certifies Catalog connectors. To understand the authentication methods and the level of security, consult the Catalog connector documentation.

# Storing secrets

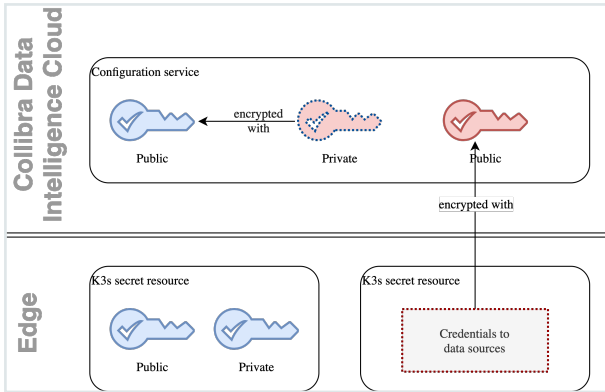
Secrets for connections and capabilities are stored solely on the Edge site. While at rest, secrets are using envelope encryption where the secret is encrypted by a key, which on its turn is encrypted by another key.

The Edge native encryption mechanism is based on two RSA key pairs. They are stored in the following places:

Keys	DIC server	Edge server	Purpose
Public red key	Yes	No	Used to encrypt credentials
Private red key	Yes (encrypted using public blue key)	No	Used to decrypt credentials
Public blue key	Yes	Yes	Used to encrypt red private keys
Private blue key	No	Yes	Used to decrypt red private key

The blue key pair is stored as a Kubernetes secret on the Edgeserver so it undergoes a native K3S encryption as described [here](#).

An Edge site owns the Blue key pair, with the Blue private key stored on Edge. Similar to that, Collibra Data Intelligence Cloud owns the Red key pair. Every secret on Edge is encrypted with the private Red key, which is sent to the Edge site for each capability execution, encrypted with the Blue public key. Once on the Edge site, Red private key is decrypted, and secrets needed to execute a connection or a capability are decrypted and injected into the capability container.



**Note** Inside the k8s cluster, all other secrets, for example data source credentials and datadog credentials, are stored encrypted at rest.

# Customer Credentials

## Credentials storage

All sensitive data is stored on Edge and encrypted by the native K3S mechanism. Additionally, all user entered credentials are encrypted using the native Edge encryption mechanism.

## Secret encryption

In the case of Virtual Machine or Bare Metal installations (K3S based), all secrets are encrypted using the native Kubernetes mechanism. The whole state of the cluster, including secrets and ConfigMap, are subject to encryption. The encryption algorithm that is used is AES 256 in CBC mode and PKCS#7 padding, which can be checked by running the following command: `sudo /usr/local/bin/k3s secrets-encrypt status`

The entire database is stored in the `/var/lib/rancher/k3s/server/db/state.db` file which contains the SQLite data.

## Credential encryption

Every value that is marked as **To be encrypted by Edge management** is additionally encrypted by the Edge site specific red public key.

The algorithm for encryption is summarized below:

1. User enters sensitive text either via Web UI or REST API.
2. Text is sent to aCollibra server (Edgemanagement module).
3. The Edge management module retrieves the red public key for the specific site.
4. A new AES 128 symmetric key (encryption key) is generated.
5. The encryption key is used to encrypt the sensitive text.
6. The encryption key itself is encrypted using the red public key.
7. The encrypted encryption key and encrypted text are concatenated and encoded using Base64 encoding to form the Edgesecret.

8. The Edgesecret is then sent directly to the Edge site, where it is stored as a Kubernetes secret.

In short the algorithms used are:

- RSA 2048 in EBC mode and PKCS#1 padding
- AES 128 in EBC mode and PKCS#7 padding

Note AES 128 was selected due to the restriction of the RSA algorithm which can only encrypt 245 bytes.

## Credentials transfer

When the Collibra server (Edge management module) has encrypted the credentials, they are sent to the Edge site using the HTTP TLS 1.2 protocol.

## Platform credentials

Apart from the credentials that users need to enter in order to connect to the data sources, there are also credentials which are needed to access the Collibraserver itself.

These credentials include:

- Collibraserver credentials (username and password, stored in dgc-secret Secret)
  - You can rotate these credentials by using the script: `edge update-dgc-cred`
- DataDog API key (stored in datadog-secret Secret)
  - Rotation is currently not possible. You have to reinstall Edge.
- JFrog credentials (stored in collibra-edge-repo-creds Secret)
  - Rotation is currently not possible. You have to reinstall Edge.

For K3S based installations, the JFrog credentials are also stored in file:

**`/etc/rancher/k3s/registries.yaml`**

Note This file is unencrypted, but it is only accessible by a root user.

# Data samples in Edge

By default, Edge by design, doesn't store any samples. To view sample data for data sources registered via Edge, you can activate a sampling capability. For all details, see [Sample data](#).

Edgecapabilities such as Profiling and Classification use data in memory, after which the data is discarded.

# Edge service repository

To keep Edge synchronized with your Collibra Data Intelligence Cloud version, we deploy core Collibra services and business capabilities in the Collibra repository of your environment. An Edge site uses token-based authentication with read privileges to download services for each release. The authentication and endpoint to access the Collibra repository are stored in the **registries.yaml** file as part of the Edge site installer.

When Collibra builds our Edge components, we run [XRay scans](#) on every build, as well as [Contrast scanning](#) for dependency in runtime scanning. The Xray scans ensure the Edge components remain free of vulnerabilities.

For 2-day vulnerability, you can edit **registries.yaml** and access the registry independently, and download images for Edge to scan them. Currently there is no SLA for vulnerabilities that you may find. The standard support SLAs are applied.

# Monitoring and logging

We monitor and log all interaction between an Edge site and Collibra Data Intelligence Cloud, as well as the Edge site infrastructure health. All logs are kept in the Collibra Datadog account.

**Note** We don't send Catalog connector logs to your environment. These Catalog connector logs are by default turned off. If they are enabled, they are kept on the Edge site itself. If you have Catalog connector issues, you have to extract these logs and send them to Collibra Support via a support ticket.

# Installing an Edge site

An Edge site is a component installed in a customer's environment. Each Edge site has a unique identifier and hosts an Edge capability that can access a data source.

This section contains the information that you need to know to install an Edge site.

# About an Edge site installation

After [creating the Edge sites](#) in Collibra Data Intelligence Cloud, you have to install the Edge software on either K3S or EKS. You typically [install](#) Edge sites within the same secure environment as the relevant data source. A customer usually has several Edge sites, depending on his requirements. The required amount of Edge sites is can be based for example on the number of networks and secure environments but it can also on the technical and legal spread of data sources.

An Edge site can have:

- Zero or more predefined connections to data sources via a JDBC driver.
- One or more integration capabilities to process data on site and send the results to Collibra.

An Edge site is a compute runtime on K3S or EKS, that executes capabilities close to your data but that is configurable from the Collibra Data Intelligence Cloud settings. It has a dedicated unique identifier and handles data sources that it can reach within its network. You can have more than one Edge site, depending on the number of networks, security domains, regions or VPCs that you have.

## Properties

Property	Description
Name	The name of the Edge site. Use a meaningful name, for example NetherlandsDataCentre1. Do not use spaces or special characters.  This field is mandatory and the name must be globally unique.
Status	The status of the Edge site.  The status is automatically shown when you create an Edge site.

Property	Description
ID	The unique ID of the Edge site, which is created automatically when you <a href="#">created</a> the Edge site.
Description	The description of the Edge site. We recommend to put at least basic location information of the Edge site.  This field is mandatory.
Installer and property files	A section where you can download the installer and property files to <a href="#">install</a> an Edge site on a server.  This section is only visible when the Edge site has status <b>To be installed</b> .

## Statuses

The status of an Edge site indicates if the Edge site can be used or not. The status is shown on the **Edge** settings page of the [Collibra settings](#). An Edge site can have one of the following statuses:

Status	Description
To be installed	The Edge site is created, but not <a href="#">installed</a> yet.
Offline	Collibra cannot reach the Edge site. This can be caused by an unsuccessful installation or a lost connection.  See the installation logs for more information.
Unhealthy	Collibra can connect to the Edge site, but some functions don't work correctly. This is typically caused by problems during the installation.  See the installation logs for more information.
Healthy	The Edge site installation was successful.

# Installation directories on K3S

The Edge site installer installs files in the following directories on your host server:

- `/var/lib/rancher/`
- `/var/log/`
- `/etc/`
- `/usr/local/bin/`

# System requirements of an Edge site

If you want to use [Edge](#), you must make sure that the following system requirements are met.

## Software requirements

- You must be able to install the Edge software on the latest version of CentOS/RedHat Enterprise Linux 8.x
- The `sudo` package is installed on the Linux host.
- The user who installs Edge has full sudo access (`ALL=(ALL) ALL`)

**Tip** If you are an early adopter or you use Edge for beta testing purposes, we highly recommend to [disable SELinux](#).

## Hardware requirements

You need the following minimum hardware requirements:

- 64 GB memory
- 16 core CPU with x86\_64 architecture
- You have at least 50 GB of free storage on the partition that will contain `/var/lib/rancher/k3s` or its symlink target.

**Warning** Any data in this location is fully managed by the Edge site, do not save any other data in here as it can be removed by Edge without notification.

- You have mounted at least 500 GB of dedicated storage for the Edge site data on a freely chosen mountpoint, for example `/var/edge/storage`.

```
mkdir -p /var/edge/storage
mkfs.xfs /dev/<block-device-name>
mount /dev/<block-device-name> /var/edge/storage
echo '/dev/<block-device-name> /var/edge/storage xfs
defaults 0 0' >> /etc/fstab
```

**Note** Change `<block-device-name>` to the name of the device that contains the storage.

**Warning** This dedicated storage should not be shared with other services because Edge can delete and overwrite files on this location without notice, so don't use `/home/<username>` or `/var`.

- If you run the Linux server on AWS or GCP, disable the services **`nm-cloud-setup.service`** and **`nm-cloud-setup.timer`**.

```
systemctl disable nm-cloud-setup.service nm-cloud-
setup.timer
reboot
```

**Warning** When new capabilities are added in the future, the hardware requirements may change.

# Network requirements

- An Edge site needs outbound connections to all of the following:
  - The URL of your Collibra Data Intelligence Cloud environment
  - `https://*.datadoghq.com`

**Note** If wildcard whitelisting is not accepted:

- `https://http-intake.logs.datadoghq.com`
- `https://7-36-1-app.agent.datadoghq.com`
- `https://agent-http-intake.logs.datadoghq.com`
- `https://api.datadoghq.com`

- `https://*.repository.collibra.io`

**Note** If wildcard whitelisting is not accepted:

- `https://repository.collibra.io`
- `https://edge-docker-delivery.repository.collibra.io`

- `https://otlp-http.observability.collibra.dev/`
- Your Edge site has to be able to connect to port 443.
- The resolve configuration file of your Linux host has maximum three search domains and two name servers.

**Note** If a firewall is enabled, run the following commands to add the `cni0` and `loopback` interfaces to a trusted zone, so that Kubernetes can use it between its services:

```
firewall-cmd --zone=trusted --change-interface=cni0 --
permanent
firewall-cmd --zone=trusted --change-interface=lo --
permanent
firewall-cmd --reload
```

# EKS requirements

You can install the Edge software on managed Kubernetes clusters.

**Important** A managed Kubernetes cluster must be fully dedicated for Edge, do not use the cluster for other purposes.

- AWS EKS 1.22 (only with `--container-runtime containerd`)
- AWS EKS worker nodes use the EKS optimized Amazon Linux 2 AMI
- EKS cluster has [IRSA enabled](#)
- AWS EKS worker nodes need to be in the same (one) Availability Zone!  
This can be implemented by creating just one node group for the EKS cluster, which limits the subnets to just one subnet, that is one of the subnets of the EKS cluster.

```

module "eks" {
  source          = "terraform-aws-modules/eks/aws"
  version        = "17.24.0"
  cluster_name   = "${var.vpc_name}-${var.cluster_
name}-eks"
  cluster_version = "1.21"
  vpc_id         = var.vpc_id
  subnets       = data.aws_subnet_ids.public_subnet_
ids.ids # Subnets specified must be in at least two
different AZs
  worker_additional_security_group_ids = [aws_security_
group.worker_sg.id]
  enable_irsa    = true # enable iam role for service
account, for later use
  worker_groups = [
    {
      name              = "${var.vpc_name}-${var.cluster_
name}-eks-workers"
      instance_type     = var.worker_type
      asg_desired_capacity = var.instance_count_workers
      key_name          = aws_key_pair.cluster-ssh-
keypair.key_name
      bootstrap_extra_args = "--container-runtime
containerd" # mandatory to run with containerd if on
1.21
      subnets          = [subnet1]
    }
  ]
}

```

```

        # restriction for now to use only 1 subnet due to
EBS tied to AZ
    },
]
map_accounts = [
    data.aws_caller_identity.current.account_id
]

tags = {
    Name = "${var.vpc_name}-${var.cluster_
name}-eks"
}
}

```

## Software requirements

- A Linux server with bash available. This is the server from which you install the Edge software on EKS.

**Tip** This server will also contain the Edge tools.

- Plain cluster\_admin kubectl access to the EKS cluster using its kubeconfig. With this kubeconfig, you must be able to use the kubectl command to communicate with the Kubernetes API server with full cluster access.
- Kubectl client version 1.21.6, supports EKS 1.21 and 1.22.

## Hardware requirements

You need an operational EKS cluster with at least 1 worker node. The cluster must meet the following requirements:

- The total cluster capacity has at least 16 core CPU and 64 GB memory, for example 4 worker nodes with each 4 core CPU and 16 GB.
- Each worker node needs at least 100 GB free disk space to store Docker images.

- Ability to create [EBS](#)-based persistent volumes as a default storage class, at least 500 GB in total.

## Network requirements

- An Edge site needs outbound connections to all of the following:
  - The URL of your Collibra Data Intelligence Cloud environment
  - `https://*.datadoghq.com`

**Note** If wildcard whitelisting is not accepted:

- `https://http-intake.logs.datadoghq.com`
- `https://7-36-1-app.agent.datadoghq.com`
- `https://agent-http-intake.logs.datadoghq.com`
- `https://api.datadoghq.com`

- `https://*.repository.collibra.io`

**Note** If wildcard whitelisting is not accepted:

- `https://repository.collibra.io`
- `https://edge-docker-delivery.repository.collibra.io`

- `https://otlp-http.observability.collibra.dev/`
- Your Edge site has to be able to connect to port 443.
- The resolve configuration file of your Linux host has maximum three search domains and two name servers.

**Note** If a firewall is enabled, run the following commands to add the `cni0` and loopback interfaces to a trusted zone, so that Kubernetes can use it between its services:

```
firewall-cmd --zone=trusted --change-interface=cni0 --
permanent
firewall-cmd --zone=trusted --change-interface=lo --
permanent
firewall-cmd --reload
```

# Create an Edge site



You create an [Edge site](#) to have a processing runtime at your premises.

## Prerequisites

- You have a global role that has the System administration global permission.
- You have a global role that has the Manage Edge sites global permission.
- You have [enabled](#) database registration via Edge in Collibra Console.

**Note** You must restart the Data Governance Center service when you have enabled this option.

## Steps

1. In the main menu, click , then  **Settings**.
  - » The [Collibra settings page](#) opens.
2. Click **Edge**
  - » The Edge sites overview appears.
3. Above the table, to the right, click **Create Edge site**.
  - » The **Create Edge site** wizard starts.
4. Enter the required information.

Field	Description
Name	<p>The name of the Edge site. Use a meaningful name, for example NetherlandsDataCentre1. Do not use spaces or special characters.</p> <p>This field is mandatory and the name must be globally unique.</p>
Description	<p>The description of the Edge site. We recommend to put at least basic location information of the Edge site.</p> <p>This field is mandatory.</p>

5. Click **Create**.

- » The Edge sites overview appears, including the new Edge site with the status **To be installed**.

## What's next?



You can now [install the Edge site](#).

# Configure a forward proxy

For security reasons, it is possible that an [Edge site](#) has to connect via a forward HTTP proxy. In that case, you have to update **proxy.properties** before installing the Edge site.

If the forward proxy server is responsible to decrypt TLS traffic, you also have to use the proxy server's CA certificate during the installation.

## Steps

1. Download the Edge site installer:
  - a. Open an Edge site.
    - a. In the main menu, click , then  **Settings**.
      - » The [Collibra settings page](#) opens.
    - b. Click **Edge**
      - » The Edge sites overview appears.
    - c. In the Edge site overview, click the name of an Edge site.
      - » The Edge site page appears.
  - b. In the **Installer and properties files** section, click **Download**.
  - c. Depending on your operating system and browser, follow the regular steps for downloading files.
    - » The installer file is a TGZ archive that contains the files **proxy.properties**, **properties.yaml** and **registries.yaml**.

**Note** If you download an installer, all previously downloaded installers become invalid.

2. Open the **proxy.properties** file.
3. Uncomment and update the outbound-proxy properties by removing "#" at the beginning of the following lines:

```
#noProxy=<host IP addresses>,<host DNS names>,<k8s-svc-ip-
addresses>,<k8s-pod-ip-addresses>,<others>
#proxyHost=<proxy domain name or IP address>
#proxyPort=<proxy-port>
```

```
#proxyUsername=<proxy username>
#proxyPassword=<proxy password>
```

Setting	Value
noProxy	<p>A comma-separated list of IP or DNS addresses that can bypass the proxy server.</p> <p>This list must include at least the Kubernetes cluster's internal IP addresses and the Kubernetes nodes' IP and DNS addresses.</p> <p>The list may not contain spaces.</p> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>Tip</b> To get the values for this setting, you can <a href="#">use the <code>edge-get-noproxy.sh</code> script</a>, which you can find in the extracted installer directory under <code>/resources/tools</code>. However, make sure that your network administrator reviews these values.</p> </div> <p>where:</p> <ul style="list-style-type: none"> <li>◦ <code>&lt;host-ip-addresses&gt;</code>: for example <code>172.20.0.0/16</code>.</li> <li>◦ <code>&lt;host-dns-names&gt;</code>: for example <code>*.compute.internal</code>.</li> <li>◦ <code>&lt;k8s-svc-ip-addresses&gt;</code>: is by default <code>10.43.0.0/16</code>, but this can differ for other k8s flavors or configurations.</li> <li>◦ <code>&lt;k8s-pod-ip-addresses&gt;</code>: is by default <code>10.42.0.0/16</code>, but this can differ for other k8s flavors or configurations.</li> <li>◦ <code>&lt;others&gt;</code>: other IP addresses that don't need to be proxied. Add at least <code>169.254.169.254</code> for AWS.</li> </ul> <div style="border: 1px solid #ccc; padding: 10px; margin: 10px 0;"> <p><b>Example</b></p> <pre>noProxy=172.20.0.0/16,*.compute.internal,10.43.0.0/16,10.42.0.0/16,169.254.169.254</pre> </div>

Setting	Value
proxyHost	<p>The IP or DNS address of the proxy server.</p> <pre>Example proxyHost=site4- proxy.shared.edge.collibra.dev</pre>
proxyPort	<p>The TCP port of the proxy server. This value must be a quoted string and not an integer value.</p> <pre>Example proxyPort="3128"</pre>
proxyUsername	<p>The username to authenticate at the proxy server.</p> <pre>Example proxyUsername=edge</pre>
proxyPassword	<p>The password to authenticate at the proxy server.</p> <pre>Example proxyPassword=la;fs90jpo4j3rR%</pre>

```
#noProxy=<host IP addresses>,<host DNS names>,<k8s-svc-ip-
addresses>,<k8s-pod-ip-addresses>,<others>
#proxyHost=<proxy domain name or IP address>
#proxyPort=<proxy-port>
#proxyUsername=<proxy username>
#proxyPassword=<proxy password>
```

Setting	Value
noProxy	<p>A comma-separated list of IP or DNS addresses that can bypass the proxy server.</p> <p>This list must include at least the Kubernetes cluster's internal IP addresses and the Kubernetes nodes' IP and DNS addresses.</p> <p>The list may not contain spaces.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>Tip</b> To get the values for this setting, you can <a href="#">use the edge-get-noproxy.sh</a> script. However, make sure that your network administrator reviews these values.</p> </div> <p>where:</p> <ul style="list-style-type: none"> <li>◦ &lt;host-ip-addresses&gt;: for example <code>172.20.0.0/16</code>.</li> <li>◦ &lt;host-dns-names&gt;: for example <code>*.compute.internal</code>.</li> <li>◦ &lt;k8s-svc-ip-addresses&gt;: depends on your EKS installation. Typically this is <code>10.100.0.0/16</code> or <code>172.20.0.0/16</code>.</li> <li>◦ &lt;k8s-pod-ip-addresses&gt;: depends on your EKS installation. Typically they are the same subnets as in the VPC, for example <code>172.20.0.0/16</code>.</li> <li>◦ &lt;others&gt;: other IP addresses that don't need to be proxied, for EKS, always add <code>169.254.169.254..</code></li> </ul> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>Example</b></p> <pre>noProxy=172.20.0.0/16,*.compute.internal,10.100.0.0/16,169.254.169.254</pre> </div>
proxyHost	<p>The IP or DNS address of the proxy server.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>Example</b> <code>proxyHost=site4-proxy.shared.edge.collibra.dev</code></p> </div>

Setting	Value
proxyPort	The TCP port of the proxy server. This value must be a quoted string and not an integer value.  Example <code>proxyPort="3128"</code>
proxyUsername	The username to authenticate at the proxy server.  Example <code>proxyUsername=edge</code>
proxyPassword	The password to authenticate at the proxy server.  Example <code>proxyPassword="la;fs90jpo4j3rR%"</code>

**Important** When you add a new node to a cluster, review and update, if necessary, the `noProxy` and implicitly forward proxy settings, unless the subnet used for nodes and their DNS suffix are added to `noProxy`.

4. If you use a forward proxy that decrypts TLS traffic, a so-called man-in-the-middle proxy or MITM proxy, then on-the-fly TLS certificates that are generated by the MITM proxy, must use the subjectAltName (SAN) extension. To enable Edge via a MITM proxy, perform also the following steps.

If the proxy server does not decrypt the TLS traffic, you can skip the following steps.

- a. Export your proxy server's CA certificate in PEM format.
- b. Save this certificate as **ca.pem** in the same directory as the Edge site installer.

**Note** If you save the certificate in another directory, use the `--ca` argument in the [Edge site installation command](#).

## What's next?

[Install the Edge site](#)

If you want to update the forward proxy afterwards, you can use the [update script](#).

# Install an Edge site

After you have created the [Edge site](#) in Collibra Data Intelligence Cloud, you have to install the Edge software on a server.

## Tip



Every time you download an Edge site installer, the previously downloaded Edge site installer becomes outdated. If you use this outdated installer, the Edge site cannot communicate with Collibra.

## Prerequisites

- You have a global role with the Install Edge sites and the User Administration global permission, for example Edge site administrator
- You have a global role that has the System administration global permission.
- You have [created](#) an Edge site.
- You have [configured the forward proxy](#), if a forward proxy is required for Edge to connect to Collibra, Datadog, OpenTelemetry and jFrog. Contact your network administrator if this is applicable.
- Your server meets all [system requirements](#).

Tip If you are an early adopter or you use Edge for beta testing purposes, we highly recommend to [disable SELinux](#).

## Steps

1. Download the installer:
  - a. Open an Edge site.
    - a. In the main menu, click , then  **Settings**.
      - » The [Collibra settings page](#) opens.
    - b. Click **Edge**
      - » The Edge sites overview appears.
    - c. In the Edge site overview, click the name of an Edge site.
      - » The Edge site page appears.

- b. In the **Installer and properties files** section, click **Download**.

**Tip** When you download the installer, [an Edge user is automatically created](#) in Collibra.

- c. Depending on your operating system and browser, follow the regular steps for downloading files.

» The installer file is a TGZ archive that contains the files **proxy.properties**, **properties.yaml** and **registries.yaml**.

**Warning** If you download an installer, the previously downloaded Edge site installer becomes invalid.

2. Extract the TGZ archive on the server on which you are going to install the Edge site software.

```
tar -xf <edge-site-id>-installer.tgz
```

**Tip** Keep the installer or the content of the extracted installer somewhere safe on your server. It contains various tools that you may need later, for example to troubleshoot issues.

3. Execute the installation. Use the correct path to the mounted storage as described in the [prerequisites](#).

**Important**

- If the Edge site has to connect via a forward HTTP proxy, then first [configure the forward proxy](#) before executing the installation.
- If you can use a custom docker registry instead of the Collibra registry, you can [enable one by following instructions in this overview](#).

## a. Clean installation:

```
sudo sh install-master.sh --storage-path
/path/mounted/storage properties.yaml -r
registries.yaml
```

for example:

```
sudo sh install-master.sh --storage-path
/var/edge/storage properties.yaml -r registries.yaml
```

## b. Installation with classification enabled:

```
sudo sh install-master.sh --storage-path
/path/mounted/storage properties.yaml -r
registries.yaml
```

for example:

```
sudo sh install-master.sh --storage-path
/var/edge/storage properties.yaml -r registries.yaml
--set collibra_
edge.collibra.classification.enabled=true
```

» In the Edge sites overview, you can see the [status](#) of the deployment.

## 4. Run the following commands to verify the status of the installation.

- To ensure that Kubernetes is running and that there is an existing node:

```
sudo /usr/local/bin/kubectl get nodes
```

- To ensure the state of all pods are installed and running:

```
sudo /usr/local/bin/kubectl get pods --all-namespaces
```

**Tip** If you already installed Edge site and you want to enable classification afterwards, see [this article](#).

1. Download the installer:

- a. Open an Edge site.
  - a. In the main menu, click ☰, then ⚙️ **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
- b. In the **Installer and properties files** section, click **Download**.

**Tip** When you download the installer, an Edge user is automatically created in Collibra.

- c. Depending on your operating system and browser, follow the regular steps for downloading files.
  - » The installer file is a TGZ archive that contains the files **proxy.properties**, **properties.yaml** and **registries.yaml**.

**Warning** If you download an installer, the previously downloaded Edge site installer becomes invalid.

2. Extract the TGZ archive on the server on which you are going to install the Edge site.

```
tar -xf <edge-site-id>-installer.tgz
```

**Tip** Keep the installer or the content of the extracted installer somewhere safe on your server. It contains various tools that you may need later, for example to troubleshoot issues.

### 3. Execute the installation.

#### a. Clean installation:

```
./run-installer-job.sh properties.yaml --repositories  
repositories.json --set collibra_  
edge.collibra.minio.persistence.size=120Gi
```

#### b. Installation with classification enabled:

```
./run-installer-job.sh properties.yaml --repositories  
repositories.json --set collibra_  
edge.collibra.minio.persistence.size=120Gi --set  
collibra_edge.collibra.classification.enabled=true
```

» In the Edge sites overview, you can see the [status](#) of the installation.

### 4. Run the following commands to verify the status of the installation.

- To ensure that Kubernetes is running and that there is an existing node:

```
kubectl get nodes
```

- To ensure the state of the installation is either running or finished:

```
kubectl get pods --all-namespaces
```

**Tip** If you already installed Edge site and you want to enable classification afterwards, see [this article](#).

# Host hardening

Each time you start K3S, a KUBECONFIG file is created. This file contains the credentials to access the K3S cluster as an administrator. The KUBECONFIG file is created by default under `/etc/rancher/k3s/k3s.yaml`. For security reasons, we recommend host hardening by making the KUBECONFIG file inaccessible for other users. As long as the host hardening is applied to Edge, you cannot connect to the K3S cluster using `kubectl` or the Edge tools.

In this article, you will learn how to enable and disable the host hardening.

## Prerequisites

- Edge needs to be [installed](#).
- You need root privileges on the server that hosts the Edge site.

## Enable host hardening

1. Sign into the server that hosts your Edge site with root privileges.
2. Open the file `/etc/systemd/system/k3s.service.env` for editing.
3. Add the line `K3S_KUBECONFIG_OUTPUT=/dev/null`.

**Note** If there are other lines, setting other environment variables do not remove them.

4. Restart the K3S service: `systemctl restart k3s`
5. Check if the KUBECONFIG file is empty: `cat /etc/rancher/k3s/k3s.yaml`

**Note** K3S is actually making `/etc/rancher/k3s/k3s.yaml` a symlink to `/dev/null`.

To further increase the security of your server, you can prevent connections to K3S from other sources than localhost.

Limit the access to the following ports other than localhost:

Protocol	Port	Description
TCP	6443	Kubernetes API Server
TCP	10250	Kubelet metrics
UDP	4500	strongSwan
UDP	500	strongSwan

The following commands prevent access to the ports mentioned in the table. Please check with your security team for compliance and for the tools used to filter the traffic before applying these commands.

```
iptables -I INPUT -j DROP -p tcp -m multiport --dports
6443,10250
iptables -I INPUT -j DROP -p udp -m multiport --dports
4500,500
iptables -I INPUT -j ACCEPT -i lo -p tcp -m multiport --dports
6443,10250
iptables -I INPUT -j ACCEPT -i lo -p udp -m multiport --dports
4500,500
```

## Disable host hardening

1. Sign into the server that hosts your Edge site with root privileges.
2. Open the file `/etc/systemd/system/k3s.service.env` for editing.
3. Remove the line `K3S_KUBECONFIG_OUTPUT=/dev/null`.
4. Restart the K3S service: `systemctl restart k3s`
5. Check if the KUBECONFIG file is empty: `cat /etc/rancher/k3s/k3s.yaml`

# Enable or disable classification on an Edge site

If you have an existing Edge site installation without classification, you can enable it afterwards. Similarly, you can disable classification on installation where it is enabled.

**Note** Enabling or disabling classification can take a couple of minutes before the changes are in effect.

## Enable classification

To enable classification on an existing Edge site, deployed on K3S, run this command:

```
DEPLOYMENT_NAME=$(sudo /usr/local/bin/kubectl get deployment
-n collibra-edge -l app.kubernetes.io/component=application-
controller -o name)

sudo /usr/local/bin/kubectl -n collibra-edge exec -it
${DEPLOYMENT_NAME} \
-- bash -c 'argocd admin cluster kubeconfig
https://kubernetes.default.svc \
/tmp/config --namespace collibra-edge ; env
KUBECONFIG=/tmp/config argocd app set collibra-edge --
core -p collibra.classification.enabled=true'
```

To enable classification on an existing Edge site, deployed on EKS, run this command:

```
DEPLOYMENT_NAME=$(kubectl get deployment -n collibra-edge -l
app.kubernetes.io/component=application-controller -o name)

kubectl -n collibra-edge exec -it ${DEPLOYMENT_NAME} \
-- bash -c 'argocd admin cluster kubeconfig
https://kubernetes.default.svc \
/tmp/config --namespace collibra-edge ; env
KUBECONFIG=/tmp/config argocd app set collibra-edge --
core -p collibra.classification.enabled=true'
```

## Disable classification

To disable classification on an existing Edge site, deployed on K3S, run this command:

```
DEPLOYMENT_NAME=$(sudo /usr/local/bin/kubectl get deployment
-n collibra-edge -l app.kubernetes.io/component=application-
controller -o name)

sudo /usr/local/bin/kubectl -n collibra-edge exec -it
${DEPLOYMENT_NAME} \
-- bash -c 'argocd admin cluster kubeconfig
https://kubernetes.default.svc \
/tmp/config --namespace collibra-edge ; env
KUBECONFIG=/tmp/config argocd app set collibra-edge --
core -p collibra.classification.enabled=false'
```

**Tip** The only difference with enabling classification is the last argument being false instead of true.

To disable classification on an existing Edge site, deployed on EKS, run this command:

```
DEPLOYMENT_NAME=$(kubectl get deployment -n collibra-edge -l
app.kubernetes.io/component=application-controller -o name)

kubectl -n collibra-edge exec -it ${DEPLOYMENT_NAME} \
-- bash -c 'argocd admin cluster kubeconfig
https://kubernetes.default.svc \
/tmp/config --namespace collibra-edge ; env
KUBECONFIG=/tmp/config argocd app set collibra-edge --
core -p collibra.classification.enabled=false'
```

**Tip** The only difference with enabling classification is the last argument being false instead of true.

# Reinstall an Edge site using backup and restore

You can reinstall an Edge site by keeping previous Edge site data.

1. **Back up** your current Edge site (optionally): `edge backup -o backup.yaml`
2. If you are reusing the same server as your current Edge site, **uninstall** the current installation: `uninstall-edge.sh`
3. **Download** the new installer and save it on your server.
4. Extract the installer and ensure that your custom setup, for example **proxy.properties** and **ca.pem**, is available as in the previous setup.
5. Reinstall with the new installer, optionally with backup and/or classification:
 

```
./install-master.sh --storage-path <storagepath> -b backup.yaml
--set collibra_edge.collibra.classification.enabled=true
```

1. **Back up** your current Edge site (optionally): `edge backup -o backup.yaml`
2. If you are reusing the same server as your current Edge site, **uninstall** the current installation: `<extracted installer>/resources/tools/installer-job/uninstall-edge-on-managed-k8s.sh`
3. **Download** the new installer and save it on your Linux server that has `kubectl` access to the k8s cluster.
4. Extract the installer and ensure that your custom setup, for example **proxy.properties** and **ca.pem**, is available as in the previous setup.
5. Reinstall with the new installer, optionally with backup and/or classification: `./run-installer-job.sh -b backup.yaml --set collibra_`  
`edge.collibra.classification.enabled=true`

# Upgrade the operating system of an Edge site



When you have a running Edge site, you can safely upgrade the operating system by following the procedure in this article.

## Steps

1. **Back up** the Edge site.  
The backup is not mandatory, but highly recommended in case the upgrade of your OS would fail.
2. Upgrade your OS.
3. Restart the OS.
4. Wait until the Edge site becomes healthy in the Collibra Data Intelligence Cloud user interface.

## Troubleshooting

If the Edge site does not become healthy after the OS upgrade, then **reinstall** the Edge site with a new Edge installer and the backup that you created before the OS upgrade.

1. In Collibra, go to the Edge site you want to reinstall.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the top right corner, click **Actions** → **Redownload installer**.
  - » A new Edge installer is downloaded.

3. Install the Edge site with the backup that you created earlier.

```
install-master.sh properties.yaml --storage-path  
/var/edge/storage properties.yaml -r registries.yaml -b  
/<path to backup file>/edge-backup.yaml
```

4. Wait until the Edge site becomes healthy in the Collibra Data Intelligence Cloud user interface.

# Custom Docker Registry

## Enable private docker registry

Edge supports a private docker registry. Switching to a private docker registry is only possible during installation, and we recommend following the process below:

1. [Backup](#) your Edge site.
2. [Uninstall](#) the Edge site.
3. [Download and reinstall](#) the latest installer.

The installer supports the following installation parameters:

- `--registry-url`, example: `https://edge-docker-delivery.repository.collibra.io`
- `--registry-host`, optional, will automatically be derived from `--registry-url`, that is, `edge-docker-delivery.repository.collibra.io`
- `--registry-pass`, the password to access the docker registry.
- `--registry-user`, the username to access the docker registry.

In order to install Edge with support for a private docker registry, the follow set of additional parameters are required:

- For bare metal installations:
  - `./install-master.sh (...usual parameters...) --registry-url <url private docker registry> --registry-user <username> --registry-pass <password>`
- For EKS installations:
  - `./run-installer-job.sh (...usual parameters...) --registry-url <url private docker registry> --registry-user <username> --registry-pass <password>`

**Note** For the above example, the private docker registry must mirror the current Edge docker repository. Configuring this private repository is out of the scope of this documentation, as we can only provide a way to extract the credentials to the Edge docker repository.

## Configure private docker registry

You need to extract your private repository credentials to the current Edge docker repository before you can begin configuring your private docker registry.

To do that, open the **registries.yaml** file. It should contain something like this:

```
version: 1
configs:
- uri: "edge-docker-delivery.repository.collibra.io"
  username: "user"
  password: "pass"
```

Currently, this file contains a single registry, though in the future it may contain more. It is important that all of these are mirrored in the private docker registry.

# JDBC connections

JDBC connections, or simply connections, define how an [Edge capability](#) accesses a data source.

To [create a connection to your data source](#), you need to select a connection provider, which determines the available properties of the connection, such as the authentication method and connection string and driver.

**Example** If you want to ingest data from an Amazon Redshift data source, you need a specific JDBC driver for Amazon Redshift. You use that driver to create a connection between your Edge site and your Amazon Redshift data source.

**Tip** Collibra provides a selection of certified JDBC drivers on [Collibra Marketplace](#). We highly recommend to only use [JDBC drivers that are certified for Edge](#).



# Data sources supported by Edge

You can [register](#), [profile](#) and [classify](#) several data sources via Edge. Depending on your data source, you can use a Collibra-provided Catalog connector, or your own JDBC driver when you create a [JDBC connection](#).

The following data sources have been tested for registering, profiling and classifying via Edge.

# Create a JDBC connection

You can create a [JDBC connection](#) from an [Edge site](#) to a data source. You can then [register the data source via Edge](#).

**Warning** Currently, Data Catalog does not support registering Oracle and data sources without schemas via Edge.

## Available Catalog connectors

# Edit a JDBC connection

You can edit a [JDBC connection](#), for example if you want to change one of its connection properties. You can then [register the data source via Edge](#).

**Warning** Currently, Data Catalog does not support registering Oracle and data sources without schemas via Edge.

## Available Catalog connectors



# Delete a JDBC connection

You can delete an [JDBC connection](#) from an [Edge site](#) to a data source if you no longer need it.

## Prerequisites

- You have a global role that has the System administration global permission.
- You have a [global role](#) that has the Manage connections and capabilities [global permission](#).
- You have [created](#) and [installed](#) an Edge site.
- You have created a [JDBC connection](#).

## Steps

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the JDBC Connections section, click the name of a JDBC connection.
  - » The **Connection settings** page appears.
3. At the bottom of the page, click **Delete**.
  - » The **Delete confirmation** dialog box appears.
4. Click **Delete Connection**.

# Use keys to access a database

It is possible that, to access a database, the JDBC driver requires a private key. In this case, you have to manually add extra connection properties when you [create a JDBC connection](#).

For example, the Snowflake driver exposes `private_key_file` and `private_key_file-pwd` properties. You can use these connection properties for the connection with Snowflake as shown in the following image.

### Create connection

#### Connection settings

Name \*

Description

Please provide a meaningful description.

Connection provider \*

Generic JDBC connection

Generic connection provider, established JDBC connection with the provided JDBC driver. Uses any authentication scheme supported by the driver

#### Connection parameters

Driver class name \*

The fully qualified name of the jdbc driver

Driver jar \*



The jar file containing the jdbc driver class

Connection string \*

The jdbc connection string

#### Connection properties

Name *	Type *	Encryption *	File upload *
<input type="text" value="private_key_file"/>	<input type="text" value="File"/>	<input type="text" value="To be encrypted by Edge management server"/>	<input type="text" value="snowflake.pk"/> <input type="button" value="Upload"/>
Name *	Type *	Encryption *	Value *
<input type="text" value="private_key_file_pwd"/>	<input type="text" value="Text"/>	<input type="text" value="To be encrypted by Edge management server"/>	<input type="text" value="....."/>

# Edge capabilities

An Edge capability is an application that runs on an [Edge site](#) to extract and process data. It delivers the results to Collibra Data Intelligence Cloud.



# About Edge capabilities

An Edge capability is an application that runs on an [Edge site](#). The Edge capability can access a data source to extract and process data and usually runs as a scheduled job. It delivers the output to Collibra Data Intelligence Cloud in a secure and reliable way.

An Edge capability has a capability template that defines a specific use case, for example data source ingestion.

## Capability templates

A capability template is developed for a specific task on a specific data source type. The capability template also determines which properties are available to configure the Edge capability.

Currently, the following capability templates are available:

- **Catalog JDBC ingestion:** A capability template you use to [register a data source](#) and [synchronize schemas](#) from a data source via a JDBC connection.
- **JDBC Profiling:** A capability template you use to [profile and classify](#) data from a registered data source.
- **Catalog JDBC Sampling:** A capability template you use to [collect and cache sample data](#) from a data source in the Edge site via a JDBC connection.
- **DQ Connector:** A capability template you use to ingest Data Quality user-defined rules, metrics, and dimensions into Collibra Data Catalog.
- **S3 synchronization:** A capability template you use to [connect to Amazon S3](#).

**Important** While these capability templates are available for all customers, the features that you use them for might still be in beta.

## Capability template structure

Each Edge capability template contains the following:

File	Description
A manifest file (YAML)	This file contains the capability metadata and input parameter requirements.
A workflow file (YAML)	This file defines the workflow and binds the parameters to capability containers.
Docker images	One or more Docker images that implement the business logic.

## Page layout

The following image shows the page for adding an edge capability.

Edge sites ▶ site3

### Add capability

#### Capability

Name \*

Description

Capability template \*

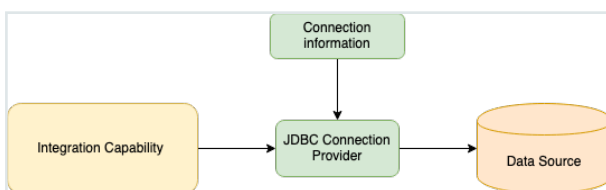
Field	Description	Required
<b>Capability</b>	This section contains the general information about the capability.	
Name	The name of the Edge capability.	✓ Yes

Field	Description	Required
Description	The description of the Edge capability.	× No
Capability template	The capability template, which determines the next available sections.	✓ Yes
<b>Custom properties</b>	Custom properties as required by the capability template.  You see the custom properties after you select a capability template.	✓ Yes

# About Edge capabilities connecting to data sources

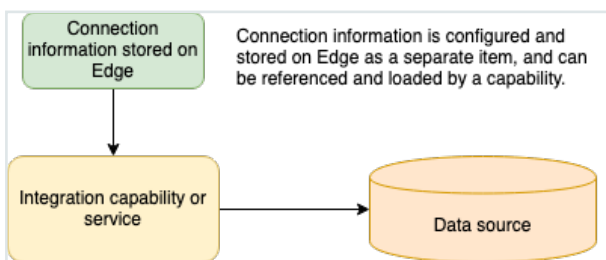
A connection on an Edge site identifies a unique system, whether it's a database, a file share or a REST service.

For JDBC (Java Database Connectivity), you can connect directly from the Edge user interface. Capabilities don't have to resolve different ways to authenticate to the JDBC service. When a capability uses JDBC connection provider libraries to connect, the capability becomes connection agnostic.



If an integration capability does not connect to a JDBC data source, it has to connect on its own, by using the information provided by Edge. The connection information is defined and stored as a Connection instance. The connection properties are shown on the Connections configuration page within Edge user interface.

Capability that does not use JDBC to connect



## Connection types

All supported connection types are bundled in Edge. You cannot add new connection types, for example Tableau or S3.

# Add an Edge capability to an Edge site



After you created and installed an [Edge site](#), you can add an [Edge capability](#) to perform specific tasks on a data source. For example, you can [register a data source](#) using a [JDBC connection](#) that belongs to an Edge capability.

## Prerequisites

- You have a global role that has the System administration global permission.
- You have a global role with the Manage connections and capabilities global permission, for example Edge integration engineer.
- You have a global role with the Register profiling information global permission. (optional)
- You have [created](#) and [installed](#) an Edge site.
- You have created a [JDBC connection](#).

## Steps

**Tip** For more information about all fields in the capability, go to the [online version of the documentation](#).

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. In the tab pane, click **Edge**.
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site with the status **Healthy**.
    - » The Edge site page appears.
2. In the **Capabilities** section, click **Add capability**.

- » The **Add capability** page appears.
3. Enter the required information.

Field	Description	Required
<b>Capability</b>	This section contains the general information about the capability.	
Name	The name of the Edge capability.	✓ Yes
Description	The description of the Edge capability.	✗ No
Capability template	<p>The <a href="#">capability template</a>, which determines the next available sections.</p> <p>Select the Catalog JDBC ingestion capability template to <a href="#">register a data source</a>.</p> <p>Select the JDBC Profiling capability template to <a href="#">profile and classify</a> your data.</p> <p>Select the Catalog JDBC Sampling capability template to <a href="#">view sample data</a>.</p> <p>Select the DQ Connector capability template to ingest Data Quality user-defined rules, metrics, and dimensions into Collibra Data Catalog.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>Important</b> Data Quality is only available in beta. Please create a <a href="#">support ticket</a> to get access.</p> </div> <p>Select the S3 synchronization capability template to connect to <a href="#">Amazon S3</a>.</p>	✓ Yes

Field	Description	Required
<b>Custom properties</b>	Custom properties as required by the capability template. You see the custom properties after you select a capability template.	✓ Yes
<b>Connection</b>	This section contains information to connect to the data source.	
JDBC connection	The <a href="#">connection to the data source</a> .	✓ Yes
JDBC data source type	The data source type of the data source that you want to ingest.	✓ Yes
Support schemas	A text field where you have to enter <i>True</i> to enable database registration of data sources that have no schema. If the data source has schemas, you can ignore this field.  <div style="border-left: 2px solid green; padding-left: 10px; margin-top: 10px;"> <p><b>Tip</b> If the data source does not have a schema, Data Catalog creates a Schema asset with the same name as the full name of the database.</p> </div>	✗ No

Field	Description	Required
Others	<p>This section can contain additional capability properties.</p> <div style="border: 1px solid red; padding: 5px; margin: 5px 0;"> <p>Warning Adding additional properties can have a significant impact on your Edge site. Only add or update them together with Collibra Support.</p> </div> <p>Click <b>Add property</b> to add a property. The possible properties are: Show properties</p>	× No

Name	Description	Type	Encryption	Value
ingest-schem-a-garbage-collector	<p>The garbage collector that is used by the capability. The default garbage collector is - XX:+UseParallelGC. For information about other possibilities, see <a href="#">AZUL documentation</a>.</p>	Text	Not encrypted (plain text)	For example: - XX:+UseSerialGC
ingest-schem-a-requests-cpu	<p>The minimum amount of CPU computing power requested by the capability. The amount is expressed in milliCPU.</p>	Text	Not encrypted (plain text)	For example: 100
ingest-schem-	<p>The maximum amount of</p>	Text	Not encryp-	For example:

Field	Description	Required
DQ	This section contains information about the Collibra Data Quality connection.	
Base URL	Your Collibra Data Quality URL	✓ Yes
Username	The Collibra Data Quality username for this connection.	✓ Yes
Password	The Collibra Data Quality password for this connection.	✓ Yes
Encryption options	Select the type of encryption to use. Default: <i>To be encrypted by Edge management server.</i>	
Issuer of the JWT	If you have selected <i>Encrypted with public key</i> , enter your JWT issuer.	✗ No
Collibra metadata model	This section contains information about where to ingest Data Quality assets.	
DQ Rules domain id	The UUID of the <b>Rulebook Domain</b> for the ingested Data Quality rules.	✓ Yes

Field	Description	Required
DQ Metrics domain id	The UUID of the <b>Business Asset Domain</b> for the ingested Data Quality metrics.	✓ Yes
DQ Dimensions domain id	The UUID of the <b>Business Asset Domain</b> for the ingested Data Quality dimensions.	✓ Yes
Default DQ Dimension name	The default <b>Data Quality Dimension</b> , for example <i>Accuracy, Completeness, Consistency</i> and so on. Default: <i>Completeness</i> .	✓ Yes
DQ Metric classified by DQ Dimension relation type id	The UUID of the <b>Data Quality Metric classified by / classifies Data Quality Dimension</b> relation. If left unspecified, this relation will not be added.	✗ No
Assets are imported in batches of this size	The batch size of the ingestion. Default: <i>5000</i> .	✓ Yes

Field	Description	Required
<b>S3 service account</b>	This section contains the information on how to connect to Amazon S3.	
AWS Connection	The <a href="#">AWS connection</a> to be used.	✓ Yes
IAM role	The IAM role used by AWS Glue <a href="#">crawlers</a> .	✓ Yes
Encryption options	Select the type of encryption used to store the IAM role. Default: <i>To be encrypted by Edge management server.</i>	✓ Yes
<b>General</b>	This section contains general information about logging.  <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Note This section only applies to JDBC capabilities.</p> </div>	
Debug	An option to automatically send Edge infrastructure log files to Collibra Data Intelligence Cloud. By default, this option is set to <i>false</i> .  <div style="background-color: #f0f0f0; padding: 5px; border: 1px solid #ccc;"> <p>Note We highly recommend to only send Edge infrastructure log files to Collibra Data Intelligence Cloud when you have issues with Edge. If you set it to <i>true</i>, it will automatically revert to <i>false</i> after 24h.</p> </div>	✗ No
Log level	An option to determine the verbosity level of Catalog connector log files. By default, this option is set to <i>No logging</i> .	✗ No

4. Click **Save**.
  - » The capability is added to the Edge site.
  - » The fields become read-only.

## What's next?

You can now [register a data source via Edge](#).

# Edit an Edge capability of an Edge site



You can edit an [Edge capability](#) of an [Edge site](#), for example to change the custom properties.

## Prerequisites

- You have a global role that has the System administration global permission.
- You have a global role with the Manage connections and capabilities global permission, for example Edge integration engineer.
- You have a global role with the Register profiling information global permission. (optional)
- You have [created](#) and [installed](#) an Edge site.
- You have added an [Edge capability](#) to the Edge site.

## Steps

**Tip** For more information about all fields in the capability, go to the [online version of the documentation](#).

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. In the tab pane, click **Edge**.
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site with the status **Healthy**.
    - » The Edge site page appears.
2. In the **Capabilities** section, click the name of an Edge capability.
  - » The **Capability** page appears and shows a read-only overview of the capability.
3. Click **Edit**.
4. Enter the required information.

Field	Description	Required
<b>Capability</b>	This section contains the general information about the capability.	
Name	The name of the Edge capability.	✓ Yes
Description	The description of the Edge capability.	✗ No
Capability template	<p>The <a href="#">capability template</a>, which determines the next available sections.</p> <p>Select the Catalog JDBC ingestion capability template to <a href="#">register a data source</a>.</p> <p>Select the JDBC Profiling capability template to <a href="#">profile and classify</a> your data.</p> <p>Select the Catalog JDBC Sampling capability template to <a href="#">view sample data</a>.</p> <p>Select the DQ Connector capability template to ingest Data Quality user-defined rules, metrics, and dimensions into Collibra Data Catalog.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin: 10px 0;"> <p><b>Important</b> Data Quality is only available in beta. Please create a <a href="#">support ticket</a> to get access.</p> </div> <p>Select the S3 synchronization capability template to connect to <a href="#">Amazon S3</a>.</p>	✓ Yes
<b>Custom properties</b>	<p>Custom properties as required by the capability template.</p> <p>You see the custom properties after you select a capability template.</p>	✓ Yes

Field	Description	Required
<b>Connection</b>	This section contains information to connect to the data source.	
JDBC connection	The <a href="#">connection to the data source</a> .	✓ Yes
JDBC data source type	The data source type of the data source that you want to ingest.	✓ Yes
Support schemas	<p>A text field where you have to enter <i>True</i> to enable database registration of data sources that have no schema. If the data source has schemas, you can ignore this field.</p> <div style="border: 1px solid #ccc; background-color: #f9f9f9; padding: 10px; margin-top: 10px;"> <p><b>Tip</b> If the data source does not have a schema, Data Catalog creates a Schema asset with the same name as the full name of the database.</p> </div>	✗ No

Field	Description	Required
Others	<p>This section can contain additional capability properties.</p> <div style="border: 1px solid red; padding: 5px; margin: 5px 0;"> <p>Warning Adding additional properties can have a significant impact on your Edge site. Only add or update them together with Collibra Support.</p> </div> <p>Click <b>Add property</b> to add a property. The possible properties are: Show properties</p>	× No

Name	Description	Type	Encryption	Value
ingest-schem-a-garbage-collector	<p>The garbage collector that is used by the capability.</p> <p>The default garbage collector is - XX:+UseParallelGC.</p> <p>For information about other possibilities, see <a href="#">AZUL documentation</a>.</p>	Text	Not encrypted (plain text)	For example: - XX:+UseSerialGC
ingest-schem-a-requests-cpu	<p>The minimum amount of CPU computing power requested by the capability.</p> <p>The amount is expressed in milliCPU.</p>	Text	Not encrypted (plain text)	For example: 100
ingest-schem-	The maximum amount of	Text	Not encryp-	For example:

Field	Description	Required
DQ	This section contains information about the Collibra Data Quality connection.	
Base URL	Your Collibra Data Quality URL	✓ Yes
Username	The Collibra Data Quality username for this connection.	✓ Yes
Password	The Collibra Data Quality password for this connection.	✓ Yes
Encryption options	Select the type of encryption to use. Default: <i>To be encrypted by Edge management server.</i>	
Issuer of the JWT	If you have selected <i>Encrypted with public key</i> , enter your JWT issuer.	✗ No
Collibra metadata model	This section contains information about where to ingest Data Quality assets.	
DQ Rules domain id	The UUID of the <b>Rulebook Domain</b> for the ingested Data Quality rules.	✓ Yes

Field	Description	Required
DQ Metrics domain id	The UUID of the <b>Business Asset Domain</b> for the ingested Data Quality metrics.	✓ Yes
DQ Dimensions domain id	The UUID of the <b>Business Asset Domain</b> for the ingested Data Quality dimensions.	✓ Yes
Default DQ Dimension name	The default <b>Data Quality Dimension</b> , for example <i>Accuracy, Completeness, Consistency</i> and so on.  Default: <i>Completeness</i> .	✓ Yes
DQ Metric classified by DQ Dimension relation type id	The UUID of the <b>Data Quality Metric classified by / classifies Data Quality Dimension</b> relation. If left unspecified, this relation will not be added.	✗ No
Assets are imported in batches of this size	The batch size of the ingestion.  Default: <i>5000</i> .	✓ Yes

Field	Description	Required
<b>S3 service account</b>	This section contains the information on how to connect to Amazon S3.	
AWS Connection	The <a href="#">AWS connection</a> to be used.	✓ Yes
IAM role	The IAM role used by AWS Glue <a href="#">crawlers</a> .	✓ Yes
Encryption options	Select the type of encryption used to store the IAM role. Default: <i>To be encrypted by Edge management server.</i>	✓ Yes
<b>General</b>	This section contains general information about logging.  <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Note This section only applies to JDBC capabilities.</p> </div>	
Debug	An option to automatically send Edge infrastructure log files to Collibra Data Intelligence Cloud. By default, this option is set to <i>false</i> .  <div style="border: 1px solid #ccc; padding: 5px; background-color: #f9f9f9;"> <p>Note We highly recommend to only send Edge infrastructure log files to Collibra Data Intelligence Cloud when you have issues with Edge. If you set it to <i>true</i>, it will automatically revert to <i>false</i> after 24h.</p> </div>	✗ No
Log level	An option to determine the verbosity level of Catalog connector log files. By default, this option is set to <i>No logging</i> .	✗ No

5. Click **Save**.
  - » The fields become read-only.

# Delete an Edge capability from an Edge site



You can remove an [Edge capability](#) from an [Edge site](#) if you no longer need it.

**Warning** If you delete a JDBC Profiling capability and synchronize previously profiled and classified schemas again, the profiling and classification results are removed.

## Prerequisites

- You have a global role that has the System administration global permission.
- You have a global role with the Manage connections and capabilities global permission, for example Edge integration engineer.
- You have [created](#) and [installed](#) an Edge site.
- You have added an [Edge capability](#) to the Edge site.

## Steps

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the **Capabilities** section, click the name of a capability.
  - » The **Capability** page appears and shows a read-only overview of the capability.
3. Click **Delete**.
4. Click **Delete Capability**.
  - » The capability is deleted from the Edge site.

# Edge Jobs page

The Edge Jobs page gives you an overview of all jobs that are executed by an Edge site.

When **enabled** in Collibra Console, you find the Edge Jobs page in the Collibra Data Intelligence Cloud settings.

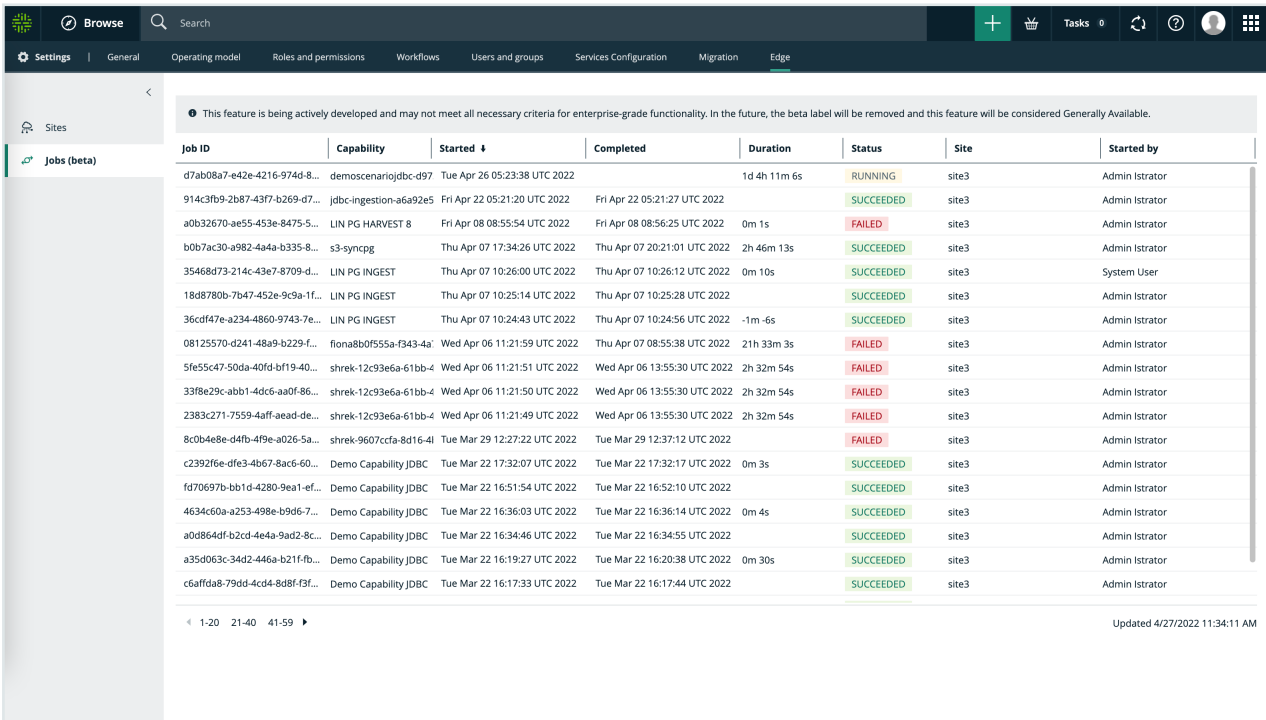
**Important** This is a beta feature.

 **Edge**

Connect to local data sources to extract metadata (structure, profiling stats, quality, classes, lineage...) and show it in Collibra.

Sites **Jobs (beta)**

On the Edge Jobs page, you find an overview of all jobs that are running or have run. Each job is a row in the table and contains basic information such as start and completion date and time, status, Edge site, capability and so on.



Job ID	Capability	Started	Completed	Duration	Status	Site	Started by
d7ab08a7-e42e-4216-974d-8...	demoscenarijdbbc-d97	Tue Apr 26 05:23:38 UTC 2022		1d 4h 11m 6s	RUNNING	site3	Admin Istrator
914c3fb9-2b87-43f7-b269-d7...	jdbc-ingestion-a6a92e5	Fri Apr 22 05:21:20 UTC 2022	Fri Apr 22 05:21:27 UTC 2022		SUCCEEDED	site3	Admin Istrator
a0b32670-ae55-453e-8475-5...	LIN PG HARVEST 8	Fri Apr 08 08:55:54 UTC 2022	Fri Apr 08 08:56:25 UTC 2022	0m 1s	FAILED	site3	Admin Istrator
b0b7ac30-a982-4a4a-b335-8...	s3-synccpg	Thu Apr 07 17:34:26 UTC 2022	Thu Apr 07 20:21:01 UTC 2022	2h 46m 13s	SUCCEEDED	site3	Admin Istrator
35468d73-214c-43e7-8709-d...	LIN PG INGEST	Thu Apr 07 10:26:00 UTC 2022	Thu Apr 07 10:26:12 UTC 2022	0m 10s	SUCCEEDED	site3	System User
18d8780b-7b47-452e-9c9a-1f...	LIN PG INGEST	Thu Apr 07 10:25:14 UTC 2022	Thu Apr 07 10:25:28 UTC 2022		SUCCEEDED	site3	Admin Istrator
36cdf47e-a234-4860-9743-7e...	LIN PG INGEST	Thu Apr 07 10:24:43 UTC 2022	Thu Apr 07 10:24:56 UTC 2022	-1m -6s	SUCCEEDED	site3	Admin Istrator
08125570-d241-48a9-b229-f...	fiona8b0f555a-f343-4a...	Wed Apr 06 11:21:59 UTC 2022	Thu Apr 07 08:55:38 UTC 2022	21h 33m 3s	FAILED	site3	Admin Istrator
5fe55c47-50da-40fd-bf19-40...	shrek-12c936ea-61bb-4	Wed Apr 06 11:21:51 UTC 2022	Wed Apr 06 13:55:30 UTC 2022	2h 32m 54s	FAILED	site3	Admin Istrator
33f8e29c-abb1-4dc6-a0df-86...	shrek-12c936ea-61bb-4	Wed Apr 06 11:21:50 UTC 2022	Wed Apr 06 13:55:30 UTC 2022	2h 32m 54s	FAILED	site3	Admin Istrator
2383c271-7559-4aff-aead-de...	shrek-12c936ea-61bb-4	Wed Apr 06 11:21:49 UTC 2022	Wed Apr 06 13:55:30 UTC 2022	2h 32m 54s	FAILED	site3	Admin Istrator
8c0b4e8e-d4fb-4f9e-a026-5a...	shrek-9607cca-8d16-41	Tue Mar 29 12:27:22 UTC 2022	Tue Mar 29 12:37:12 UTC 2022		FAILED	site3	Admin Istrator
c2392f6e-dfe3-4b67-8ac6-60...	Demo CapabilityJDBC	Tue Mar 22 17:32:07 UTC 2022	Tue Mar 22 17:32:17 UTC 2022	0m 3s	SUCCEEDED	site3	Admin Istrator
fd70697b-bb1d-4280-9ea1-ef...	Demo CapabilityJDBC	Tue Mar 22 16:51:54 UTC 2022	Tue Mar 22 16:52:10 UTC 2022		SUCCEEDED	site3	Admin Istrator
4634c60a-a253-498e-b9d6-7...	Demo CapabilityJDBC	Tue Mar 22 16:36:03 UTC 2022	Tue Mar 22 16:36:14 UTC 2022	0m 4s	SUCCEEDED	site3	Admin Istrator
a0d864df-b2cd-4e4a-9ad2-8c...	Demo CapabilityJDBC	Tue Mar 22 16:34:46 UTC 2022	Tue Mar 22 16:34:55 UTC 2022		SUCCEEDED	site3	Admin Istrator
a35d063c-34d2-446a-b21f-fb...	Demo CapabilityJDBC	Tue Mar 22 16:19:27 UTC 2022	Tue Mar 22 16:20:38 UTC 2022	0m 30s	SUCCEEDED	site3	Admin Istrator
c6affda8-79dd-4cd4-8d8f-f3f...	Demo CapabilityJDBC	Tue Mar 22 16:17:33 UTC 2022	Tue Mar 22 16:17:44 UTC 2022		SUCCEEDED	site3	Admin Istrator



# Cancel jobs


You can cancel an Edge site job which is either running or queued to run.

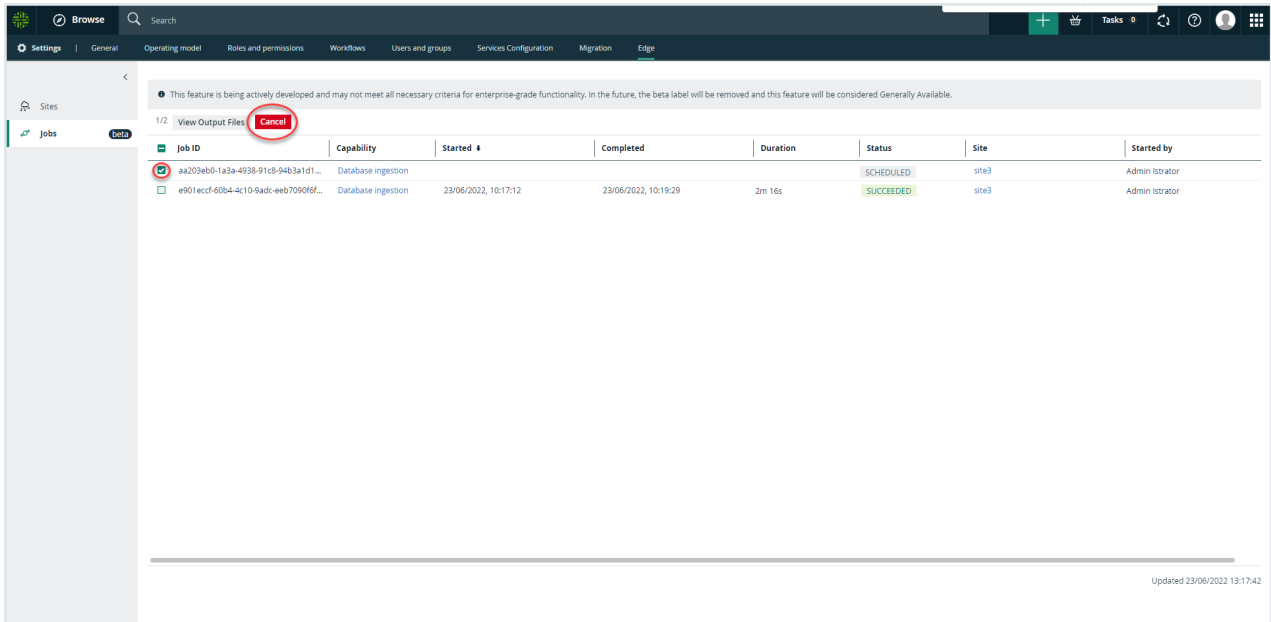
## Prerequisites

- You have a [global role](#) that has the Manage connections and capabilities [global permission](#).
- You have jobs currently running or queued .

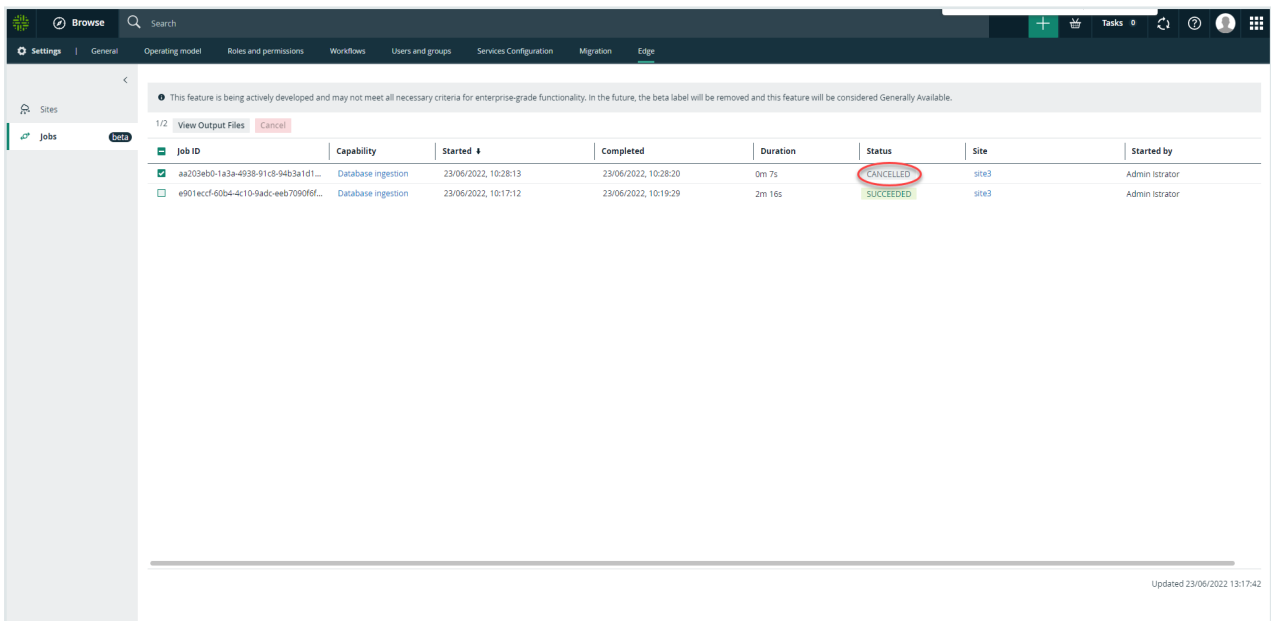
## Steps

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. In the tab pane, click **Edge**.
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site with the status **Healthy**.
    - » The Edge site page appears.
2. Click **Jobs**.
3. Select the checkbox next to the job you would like to cancel.

 **Tip** You can select more than one job at a time.
4. Click **Cancel**.



The job is now canceled, and the status of this job is updated to **CANCELED**.



# Maintaining Edge sites

In this section, you get to know how you can maintain your Edge site installations, such as taking backups or updating credentials.



# Running Edge tools

This section contains an overview on how to use the Edge tools, for example to create a backup of your Edge site.

## Prepare the Edge tools on K3S

On k3s, the Edge tool is downloaded at the end of a successful installation.

Alternatively, you can download it from the cluster:

```
TOOLS_POD=$(sudo /usr/local/bin/kubectl -n collibra-edge get
pod -l edge.collibra.com/contains=tools -o jsonpath='{.items
[0].metadata.name}')

sudo /usr/local/bin/kubectl cp collibra-edge/$TOOLS_POD:edge
/usr/local/bin/edge

sudo chmod +x /usr/local/bin/edge
```

The Edge command is in **/usr/local/bin** on the host, that is your first worker node, so you run the Edge command on the actual host where k3s runs.

## Overview Edge commands on K3S

Edge tool	Command for K3S
Uninstall Edge	<code>/usr/local/bin/uninstall-edge.sh</code>
Create Edge diagnostics file	<ul style="list-style-type: none"> <li>Edge site is not yet installed:  <code>&lt;extracted installer directory&gt;/resources/tools/edge-diagnostics.sh -d &lt;file name&gt;.tgz</code></li> <li>Edge site is up and running:  <code>edge diagnostics -d &lt;file name&gt;.tgz</code></li> </ul>

Edge tool	Command for K3S
Create an Edge site backup	<code>edge backup -o /&lt;path to folder&gt;/&lt;backup-name&gt;.yaml</code>
Set Edge storage cache ttl	<code>edge cachettl --ttl &lt;value in days&gt;</code>
Retrieve logs from a catalog connector	<code>edge catalog-connector --jobid &lt;Edge job ID&gt; \</code> <code>--dst &lt;path to destination&gt;</code>
Update Collibra credentials	<ul style="list-style-type: none"> <li>• Interactive way: <code>edge update-dgc-creds -i</code></li> <li>• Explicit update: <code>edge update-dgc-creds &lt;username&gt; &lt;password&gt; &lt;url collibra environment&gt;</code></li> </ul>
Update forward proxy settings	<code>edge update-outbound-proxy --update-outbound-proxy</code> <code>/path/to/proxy.properties</code>
Get help to set up no_proxy configuration	<ul style="list-style-type: none"> <li>• Edge site is not yet installed: <code>&lt;extracted installer directory&gt;/resources/tools/edge-get-noproxy.sh k3s</code></li> <li>• Edge site is up and running: <code>edge get-noproxy k3s</code></li> </ul>

## Prepare Edge tools on EKS

Edge is installed from a Linux machine that has access to the actual k8s cluster.

There is no automatic download of Edge tool after installation, because we don't want to enforce it in some location. Therefore, you have to download the Edge tool to your Linux machine, for example in your current folder:

```
TOOLS_POD=$(kubectl -n collibra-edge get pod -l
edge.collibra.com/contains=tools -o jsonpath='{.items
[0].metadata.name}')

kubectl cp collibra-edge/$TOOLS_POD:edge edge

chmod +x edge
```

You can now run Edge commands from your current folder.

**Note** As you are not on the worker node itself, you cannot collect worker node diagnostics. If you need these diagnostics, [create a support ticket](#).

## Overview Edge commands on EKS

Edge tool	Command for EKS
Uninstall Edge	<code>&lt;extracted installer directory&gt;/resources/tools/installer-job/uninstall-edge-on-managed-k8s.sh</code>
Create Edge diagnostics file	<ul style="list-style-type: none"> <li>Edge site is not yet installed: <code>&lt;extracted installer directory&gt;/resources/tools/edge-diagnostics.sh -d &lt;file name&gt;.tgz</code></li> <li>Edge site is up and running: <code>edge diagnostics -d &lt;file name&gt;.tgz</code></li> </ul>
Create an Edge site backup	<code>edge backup -o /&lt;path to folder&gt;/&lt;backup-name&gt;.yaml</code>
Set Edge storage cache ttl	<code>edge cachettl &lt;value in seconds&gt;</code>

Edge tool	Command for EKS
Retrieve logs from a catalog connector	<pre>edge catalog-connector --jobid &lt;Edge job ID&gt; \   --dst &lt;path to destination&gt;/&lt;file name&gt;.txt</pre>
Update Collibra credentials	<ul style="list-style-type: none"> <li>• Interactive way:  <pre>edge update-dgc-creds -i</pre> </li> <li>• Explicit update:  <pre>edge update-dgc-creds &lt;username&gt; &lt;password&gt; &lt;url collibra environment&gt;</pre> </li> </ul>
Update forward proxy settings	<pre>edge update-outbound-proxy --update-outbound-proxy /path/to/proxy.properties</pre>
Get help to set up no_proxy configuration	<ul style="list-style-type: none"> <li>• Edge site is not yet installed:  <pre>&lt;extracted installer directory&gt;/resources/tools/edge-get-noproxy.sh eks &lt;clustername&gt;</pre> </li> <li>• Edge site is up and running:  <pre>edge get-noproxy eks &lt;clustername&gt;</pre> </li> </ul>



# Edit an Edge site

You can edit a [Edge site](#) to give it another name or description.

## Prerequisites

- You have [created](#) an Edge site.
- You have a global role that has the System administration global permission.
- You have a global role that has the Manage Edge sites global permission.

## Steps

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the top right corner, click **Actions** → **Edit**.  
The **Edit Edge site** wizard starts.
3. Enter the required information.

Field	Description
Name	The name of the Edge site. Use a meaningful name, for example NetherlandsDataCentre1. Do not use spaces or special characters.  This field is mandatory and the name must be globally unique.
Description	The description of the Edge site. We recommend to put at least basic location information of the Edge site.  This field is mandatory.

4. Click **Save**.
  - » The Edge sites overview appears with the new name and description.

# Update Edge user password

When you [download the Edge site installer](#), a dedicated user is created in Collibra Data Intelligence Cloud. This user has always "Edge" as first name and the "Edge site name" as last name.

A user will be created for each Edge site. This user is deleted when you delete the Edge site.

**Note** The Edge user must have the Connect Edge to Collibra global permission.

## Steps

1. Reset the password of the Edge user in Collibra.
2. Connect to the Edge master node via SSH.
3. Run the following script: `/usr/local/bin/edge update-dgc-creds -i`
4. Enter the username and new password of the Edge user.

# Update the outbound proxy configuration

If you have to change the outbound proxy configuration of a running Edge site, you can use Collibra's outbound proxy update script.

## Steps

1. Find the **proxy.properties** file on the server that you used during the [configuration of the outbound proxy](#).
2. Update the file with the new [property](#) values and save the file.
3. Go to **/usr/local/bin** and run the following command:

```
./edge update-outbound-proxy -u /path/to/proxy.properties
```

## Help file of the script

```
$ /usr/local/bin/edge update-outbound-proxy --help
Collibra Edge Utility for updating Outbound Proxy settings.
Usage:
    edge update-outbound-proxy.sh -h|--help
    edge update-outbound-proxy.sh -g|--generate-template
<filename>
    edge update-outbound-proxy.sh -u|--update-outbound-
proxy <filename>

    -h|--help                - Show help
    -g|--generate-template    - generate template file for
proxy properties in <filename>
    -u|--update-outbound-proxy - update outbound-proxy secret
based on proxy properties <filename>
```

# Back up an Edge site

To avoid losing your Edge site configurations, you can [back up](#) an [Edge site](#). You can use this backup to [restore](#) it later, for example when you want to reinstall an Edge site with a new installer.

**Note** For privacy reasons, Edge site backups remain in your personal environment and are not sent to the cloud.

On the server that runs your Edge site, execute the following command:

```
~$ ./edge backup -o /<path to folder where you want to save the backup file>/<backup-name>.yaml
```

» Edge creates a backup of your Edge site in the selected folder of the command.

**Tip** If the Edge command is not available, you first have to [download](#) it.

On the server from which you manage your EKS cluster, execute the following command:

```
~$ ./edge backup -o /<path to folder where you want to save the backup file>/<backup-name>.yaml
```



» Edge creates a backup of your Edge site in the defined folder of the last command.

**Tip** If the Edge command is not available, you first have to [download](#) it.



# Restore an Edge site

This article walks through how to restore your Edge site. You may want to restore it if you've previously created a [backup](#) or want to reinstall an Edge site with a new installer.

Restoring an Edge site is the same command as installing an Edge site but with an extra argument to use a backup.

1. Optionally, [download a new Edge installer](#).
1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the top right corner, click **Actions** → **Redownload installer**.
  - » A new Edge installer is downloaded.
2. Run the Edge installer and add the backup file as a parameter:

```
install-master.sh properties.yaml --storage-path
/var/edge/storage properties.yaml -r registries.yaml -b /<path
to backup file>/edge-backup.yaml
```

1. Optionally, [download a new Edge installer](#).
1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the top right corner, click **Actions** → **Redownload installer**.
  - » A new Edge installer is downloaded.
2. Run the Edge installer and add the backup file as a parameter:

```
./run-installer-job.sh properties.yaml --repositories
repositories.json --set collibra_
```

```
edge.collibra.minio.persistence.size=120Gi -b /<path to backup  
file>/edge-backup.yaml
```



# Delete an Edge site

You can delete an [Edge site](#) if you no longer need it.

## Prerequisites

- You have [created](#) an Edge site.
- You have a global role that has the System administration global permission.
- You have a global role that has the Manage Edge sites global permission.



## Steps

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the top right corner, click **Actions** → **Delete Edge site**.
  - » The Delete Edge site wizard starts.
3. Click **Delete Edge site**.
  - » The Edge sites overview appears, without the deleted Edge site.
4. On the server that hosts the Edge site, go to `/usr/local/bin` where you can find the uninstall script `uninstall-edge.sh`, then run one of the following commands:

Command	
Delete Edge site but keep its data.  The data consists of drivers, required files for capabilities, and data that was saved by Edge capabilities	<pre>/usr/local/bin/uninstall-edge.sh</pre>

	Command
Delete Edge site and its data.	<pre>/usr/local/bin/uninstall-edge.sh --remove-local-data</pre>
Delete Edge site without confirmation request, for example if you want to delete the site via a script.  You can use this in combination with removing the site data.	<pre>/usr/local/bin/uninstall-edge.sh --remove-local-data --force</pre>

**Warning** When you delete an Edge site, the EBS volumes containing the data are also removed. If you like to keep your data, first back up these EBS volumes.

1. Open an Edge site.
  - a. In the main menu, click , then  **Settings**.
    - » The [Collibra settings page](#) opens.
  - b. Click **Edge**
    - » The Edge sites overview appears.
  - c. In the Edge site overview, click the name of an Edge site.
    - » The Edge site page appears.
2. In the top right corner, click **Actions** → **Delete Edge site**.
  - » The Delete Edge site wizard starts.
3. Click **Delete Edge site**.
  - » The Edge sites overview appears, without the deleted Edge site.
4. On the server from which you manage your EKS cluster, run this command:

```
<extracted installer>/resources/tools/installer-job/uninstall-edge-on-managed-k8s.sh
```

# Troubleshooting Edge

In this section, you find some articles that help you to troubleshoot Edge issues.

# General troubleshooting Edge

The following table shows how to solve issues you encountered while working with Edge. Select the tab of your installation type, k3s or EKS.

Issue	Proposed solution
<p>You get the following error message:</p> <pre>Out of disk space</pre>	<p>You have to restart the Kubernetes pod in Edge.</p> <p>Run the following command:</p> <pre>sudo /usr/local/bin/kubectl delete pod &lt;pod_name&gt; -- namespace &lt;pod_namespace&gt;</pre> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;"> <p><b>Tip</b> For more information about Pods and namespaces, see the <a href="#">Kubernetes documentation</a>.</p> </div>
<p>You get one of the following error message:</p> <ul style="list-style-type: none"> <li>• Cannot allocate memory</li> <li>• Error syncing pod</li> </ul>	<p>You have to restart Edge.</p> <p>Do the following:</p> <ol style="list-style-type: none"> <li>1. Run the following commands to remove all workflows:           <pre>sudo /usr/local/bin/kubectl delete --all workflows -- namespace=collibra- capabilities</pre> <pre>sudo /usr/local/bin/kubectl delete --all workflows -- namespace=collibra-fast</pre> </li> <li>2. Run the following command to reboot Edge:           <pre>sudo reboot</pre> </li> </ol>

Issue	Proposed solution
<p>You get the following error message:</p> <pre>Out of disk space</pre>	<p>You have to restart the Kubernetes pod in Edge.</p> <p>Run the following command:</p> <pre>kubectl delete pod &lt;pod_name&gt; --namespace &lt;pod_namespace&gt;</pre> <p><b>Tip</b> For more information about Pods and namespaces, see the <a href="#">Kubernetes documentation</a>.</p>
<p>You get one of the following error message:</p> <ul style="list-style-type: none"> <li>• Cannot allocate memory</li> <li>• Error syncing pod</li> </ul>	<p>You have to restart Edge.</p> <p>Do the following:</p> <ol style="list-style-type: none"> <li>1. Run the following commands to remove all workflows: <pre>kubectl delete --all workflows --namespace=collibra-capabilities</pre> <pre>kubectl delete --all workflows --namespace=collibra-fast</pre> </li> <li>2. Run the following command to reboot Edge: <pre>sudo reboot</pre> </li> </ol>

# Use an explicit `resolv.conf` file for Edge

**Important** This is only applicable for K3S installations.

The default resolver configuration file `/etc/resolv.conf` is in most cases picked-up by K3S and used successfully, but on Google Cloud Platform, where the default nameserver is `169.254.169.254`, K3S generates another file with a nameserver pointing to `8.8.8.8`.

Your firewall or network configuration may be filtering connections to `8.8.8.8`, in which case the resolver file has to be explicitly configured with a reachable nameserver. If the default file `/etc/resolv.conf` is explicitly configured even on GCP nodes having nameserver `169.254.169.254`, then K3S will successfully use it.

You can also explicitly indicate in `install-master.sh` to use `/etc/resolv.conf` by adding the argument `--resolv-conf </path/to/resolv.conf>`.

## Example

```
sudo sh install-master.sh --storage-path /var/edge/storage
properties.yaml -r registries.yaml --resolv.conf
/etc/resolv.conf
```

# Edge logging

When you encounter an issue in Edge, you can use diagnostic files and log files to provide a lot of indications about the issue. If you want to report a problem to Collibra support, you can include the diagnostics or log files in the support ticket. As a result, Collibra support will be able to determine what went wrong and find a solution to your issue.

You can create a diagnostics file which is a ZIP file with log files and information about the environment in which you install an Edge site. Edge also generates two types of log files that are not included in the diagnostics file:

- [Edge infrastructure log files](#), which are sent to Collibra Data Intelligence Cloud immediately upon creation.
- [Metadata connector log files](#), which can only be stored locally.

## Edge diagnostics file

The Edge diagnostics file is a ZIP file that is created by running the diagnostics script in the Edge site installer folder. The diagnostics script checks amongst others:

- Your operating system setup
- Your firewall settings
- Connectivity information
- Edge cluster logs.

You can send the diagnostics file to Collibra support when you have an issue with the [Edge site installation](#).

## Edge infrastructure log files

Edge infrastructure logs contain Edge infrastructure information, for example Edge status updates, capability information. The logs can be used by Collibra Support to help solve a

general issue with Edge. The log files do not contain any database content or private information.

By default, the Edge infrastructure log files are always enabled on an information level. You can [enable debug level logging](#) per specific capability when you add or edit an Edge capability. As a result, Edge sends infrastructure log with more information about that capability to Collibra Data Intelligence Cloud. Edge infrastructure log files can contain the following information:

- Job execution phases
- The Edge status
- Service updates
- System upgrades

These log files can only be accessed by Collibra Support.

**Note** By default, **Debug** logging for an Edge capability is set to `False`. We highly recommend only enabling the **Debug** logging for an Edge capability if an issue arises.

## Metadata connector log files

Metadata connector log files contain the logs of the JDBC connections between the Edge capability and your data source. These log files can be used by Collibra Support to help solve issues with processing or accessing data. The log files may contain information about your data source.

For security reasons, these log files are not automatically sent to your Collibra Data Intelligence Cloud environment. You can, however, [create the log files](#), save and review them locally and then attach them to a Collibra support ticket.

## Edge system monitoring

The system monitoring, executed via OpenTelemetry, sends the following information to your Collibra environment:

- CPU usage
- Memory usage
- Network statistics

Collibra Support can then analyze this information to troubleshoot potential anomalies. These data are only available to Collibra personnel.

## Verbosity log levels

The verbosity log levels indicate how much information you want to see in the Catalog Connector log files. You can change the verbosity log levels in the Edge capability for which you want to create logs. The following verbosity log levels are available:

Verbosity log level	Description
No logging	The Catalog Connector logs are not created. This is the default.
Low	The Catalog Connector logs contain the following: <ul style="list-style-type: none"> <li>• All connection query logs</li> <li>• Any errors</li> </ul>
Middle	The Catalog Connector logs include the Low logs and: <ul style="list-style-type: none"> <li>• All cache queries</li> <li>• Additional information about the request</li> </ul>
High	The Catalog Connector logs include the Middle logs and: <ul style="list-style-type: none"> <li>• The body of the request</li> <li>• The response</li> </ul>

# Create an Edge diagnostics file

You can create an [Edge diagnostics file](#) to check issues with the [Edge site installation](#) in your environment.

## Prerequisites

- You have [created](#) an Edge site.
- You have [downloaded](#) the Edge installer.

## Steps

### Edge site is not yet installed

Running the diagnostics script without an installed Edge site, checks if your system meets all requirements to install the Edge site.

1. Extract the Edge installer.
2. On the command line, go to the folder with the extracted files.
3. In this folder, go to **resources/tools**.
4. Run the following command to create the diagnostics file:

```
edge-diagnostics.sh --diag-file <file name>.tgz
```

» A TGZ file with the given file name is created and contains all Edge diagnostics file.

### Edge site is already installed

On the command line, run the following command to create the diagnostics file:

```
edge diagnostics --diag-file <file name>.tgz
```

» A TGZ file with the given file name is created and contains all Edge diagnostics file.

**Tip** If the Edge command is not available, you first have to [download](#) it.

## What's next?

You can send the diagnostics file to Collibra support to help you resolve your installation issues.

# Create Metadata connector log files

If you have an issue with a JDBC connection, for example while registering a data source via Edge, you can create the [Metadata connector log files](#) and then save and review them locally. If you create a support ticket, you can attach the reviewed Metadata connector log files to Collibra support to help you with your issue.

## Prerequisites

- You have a global role that has the System administration global permission.
- You have a global role with the Manage connections and capabilities global permission, for example Edge integration engineer.
- You have [created](#) and [installed](#) an Edge site.
- You have created a JDBC connection and an Edge capability.

## Steps

1. Edit the Edge capability that contains the JDBC connection for which you want to create a log file.
  - a. Click the name of the Edge capability to open it.
  - b. Click **Edit**.
  - c. In the **General** section, click the **Log level** drop-down menu.
  - d. Select the log verbosity level.

**Tip** The level must be at least *low*.

- e. Click **Save**.
    - » The fields become read-only.
2. Click **Run** to rerun the Edge capability.
  3. Contact Collibra support to request the Edge job ID of the Edge capability.
  4. Run the following command:

```
./edge catalog-connector --jobid <Edge job ID> --dst <path to destination>
```

- » The log file is created and stored in the predefined destination.

**Tip** If the Edge command is not available, you first have to [download it](#).

## Prerequisites

- You have a Linux host with `kubectl` access to your EKS installation.
- You have `mc` (minio client) installed in `/usr/local/bin`:

```
sudo curl -L "https://dl.min.io/client/mc/release/linux-amd64/mc" -o /usr/local/bin/mc
sudo chmod +x /usr/local/bin/mc
```

## Steps

Execute the following commands:

```
kubectl -n collibra-edge port-forward service/minio 9000:9000 &

MC_ACCESSKEY="$(kubectl get secrets edge-minio -n collibra-edge
-o jsonpath="{.data.accesskey}" | base64 --decode) "

MC_SECRETKEY="$(kubectl get secrets edge-minio -n collibra-edge
-o jsonpath="{.data.secretkey}" | base64 --decode) "

export MC_HOST_edge="http://${MC_ACCESSKEY}:${MC_SECRETKEY}@localhost:9000"

mc cp --quiet --recursive edge/cdata/<jobId> <destination_
directory>

pkill -f "port-forward"
```



# Enable debug logging for Edge infrastructure logs

By default, the Edge infrastructure logs are always enabled on an information level. If you have an issue with [Edge](#) in general, you can enable Edge to create [Edge infrastructure debug log files](#) and send them to Collibra Data Intelligence Cloud. Collibra support uses these log files to solve Edge issues.

## Prerequisites

- You have a global role that has the System administration global permission.
- You have a global role with the Manage connections and capabilities global permission, for example Edge integration engineer.
- You have [created](#) and [installed](#) an Edge site.
- You have created a JDBC connection and an Edge capability.

## Steps

1. In the main menu, click , then  **Settings**.
  - » The [Collibra settings page](#) opens.
2. On the **Settings** page, click **Edge**.
  - » The Edge sites overview appears.
3. Click the site that runs the capability with issues.
  - » The site details page appears.
4. On the **Capabilities** tab, click the name of the Edge capability.
5. Click **Edit**.
6. In the **General** section, click the **Debug** drop-down menu and select *true*.

**Note** This field is by default set to *false*. If you set it to *true*, it will automatically revert to *false* after 24 hours.

7. Click **Save**.
  - » The fields become read-only.

8. Click **Run** to rerun the Edge capability.
  - » The log files are automatically sent to Collibra Data Intelligence Cloud.

# Disable OpenTelemetry

If you reinstalled an Edge site with a new version it is possible that the new setup is not working due to a missing network connectivity. You would have to request for OpenTelemetry to be added again. If this request takes days to be completed, you may want to disable OpenTelemetry to still have a running Edge site.

## Disable OpenTelemetry at installation time

Add the flag `--disable-otel` when you [run the installation script](#).

```
sudo sh install-master.sh --storage-path /var/edge/storage
properties.yaml \
  --disable-otel \
  -r registries.yaml
```

```
./run-installer-job.sh properties.yaml --repositories
repositories.json \
  --set collibra_edge.collibra.minio.persistence.size=120Gi \
  --disable-otel
```

# Edge FAQ

The following table contains the most frequently asked questions about Edge that were not answered anywhere else in the Edge documentation.

Question	Answer
Who benefits from using Edge?	<p>All customers who want to ingest data into Collibra Data Intelligence Cloud benefit from Edge.</p> <p>Some of the benefits for using Edge are:</p> <ul style="list-style-type: none"> <li>• Data is processed in the customer's secure environment and only the process results are sent to Collibra Data Intelligence Cloud.</li> <li>• Edge can automatically anonymize sensitive profiling data before sending it to Collibra Data Intelligence Cloud.</li> <li>• Edge can automatically classify the metadata and send the classification results together with the profiling results to Collibra Data Intelligence Cloud.</li> <li>• Edge enables better profiling performance, because data no longer has to be copied or moved.</li> </ul>
Where can I find Edge API documentation?	<p>You can find the Edge API reference documentation in your Collibra environment on this URL: <a href="https://&lt;your_collibra_platform_url&gt;/edge/docs/index.html">https://&lt;your_collibra_platform_url&gt;/edge/docs/index.html</a></p>



Question	Answer
Does Edge replace the Jobserver?	<p>Customers can choose between Edge and Jobserver.</p> <p>The main differences between Edge and Jobserver are the following:</p> <ul style="list-style-type: none"> <li>• Edge is based on Kubernetes, a distributed runtime, which means: <ul style="list-style-type: none"> <li>◦ It can scale out.</li> <li>◦ It offers built in resource management.</li> <li>◦ It has out of the box high availability.</li> <li>◦ It has reliable delivery of results to Collibra Data Intelligence Cloud.</li> </ul> </li> <li>• Edge is a Collibra service compatible with on-premises as well as cloud environments.</li> <li>• Edge offers continuous delivery of capability types and updates will be installed on a regular basis.</li> <li>• Edge updates are included in Collibra Data Intelligence Cloud releases.</li> </ul> <p>Jobserver features correspond to Edge capabilities, each one is developed and deployed independently of one another. In the future, we will provide a script for migrating features from Jobserver to Edge where applicable.</p>
Can Edge run alongside Jobserver?	<p>Yes. Both can even be installed on the same server as long as the server has enough resources to support both, though we recommend not to run both services on a single server.</p>
What does the Edge architecture look like?	<p>You can see how Edge interacts with other components in <a href="#">this architecture and components overview</a>.</p>

Question	Answer
<p>Can Edge use Kubernetes provided by a Cloud vendor, for example GKE, AKS, Amazon EKS?</p>	<p>When the Edge site is installed in a Cloud environment, it does not use a managed Kubernetes provided by the Cloud vendor, because Kubernetes is already included in the Edge site installation process.</p> <p>Collibra manually manages the cluster on top of EC2 or similar machines in other platforms. In the first releases, we cannot benefit from seamless integration of various Cloud services offered by those platforms, for example, embedded authentication, auto-scaling and databases.</p>
<p>Can Edge be installed on top of a customer's provided Kubernetes cluster?</p>	<p>Yes, you can install Edge on EKS.</p>
<p>Can you migrate an Edge site between K3S and EKS?</p>	<p>No. We don't support the migration of existing sites to a different Kubernetes environment.</p>
<p>Can Edge be installed on Windows servers?</p>	<p>No, you cannot install an Edge site on Windows servers.</p>
<p>What are the supported data sources on Edge?</p>	<p>You can find the list of supported data sources in the <a href="#">Data sources supported by Edge section</a>.</p>
<p>Can you connect using a cloud provider key manager such as AWS Secrets Manager, GCP Secret Manager or Azure Key Vault?</p>	<p>Not at this moment.</p>

Question	Answer
How does Edge connect to Collibra Data Intelligence Cloud?	<p>An Edge site is installed in the customer's environment, close to the data source. The Edge site communicates to Collibra Data Intelligence Cloud and other 3rd party systems using an HTTPS connection.</p> <div data-bbox="687 600 1417 779" style="border: 1px solid #ccc; background-color: #f0f0f0; padding: 10px; margin-top: 10px;"> <p><b>Note</b> The connection between an Edge site and Collibra Data Intelligence Cloud is outbound-only.</p> </div>
Who controls Edge?	<p>Edge is controlled by the customer through local access via the Collibra Data Intelligence Cloud user interface. You can also use local access via the Linux shell for advanced troubleshooting when Edge is unable to connect.</p>
Can an Edge site connect to more than one Collibra environment?	<p>No. Every Edge site belongs and authenticates to only one Collibra Data Intelligence Cloud environment.</p>
Can Edge use customer-provided certificates to connect to Collibra Data Intelligence Cloud?	<p>Currently, we do not support this.</p> <p>Edge is a Collibra product that can run on the customer's on-premises or cloud environment. The authentication between the Edge site and Collibra Data Intelligence Cloud is controlled and secured by Collibra. The <a href="#">keys and credentials</a> are generated when you <a href="#">install the Edge site</a>.</p>
When do internal K3S certificates expire?	<p>The internal K3S certificates expire 12 months after the initial installation. You should restart the K3S-based Edge site in the last 3 months to ensure the internal certificates are rotated. If not, restart K3S or <a href="#">reinstall</a> the Edge site.</p>

Question	Answer
Does Edge support mTLS when connecting to Collibra Data Intelligence Cloud?	Currently, we do not support this.
Is Edge horizontally scalable?	Currently, Edge is not horizontally scalable. You cannot add more nodes.
Can metrics data from an Edge site be sent to Collibra through a private link instead of over the Internet?	No, this data can only be sent over the Internet.