Collibra

Collibra Data Intelligence Cloud

Edge Infrastructure

Version 2024.02

Collibra

Collibra Data Intelligence Cloud - Edge Infrastructure

Release date: February 4, 2024

Revision date: February 01, 2024

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

https://productresources.collibra.com/docs/collibra/latest/#cshid=edge-infrastructure

Contents

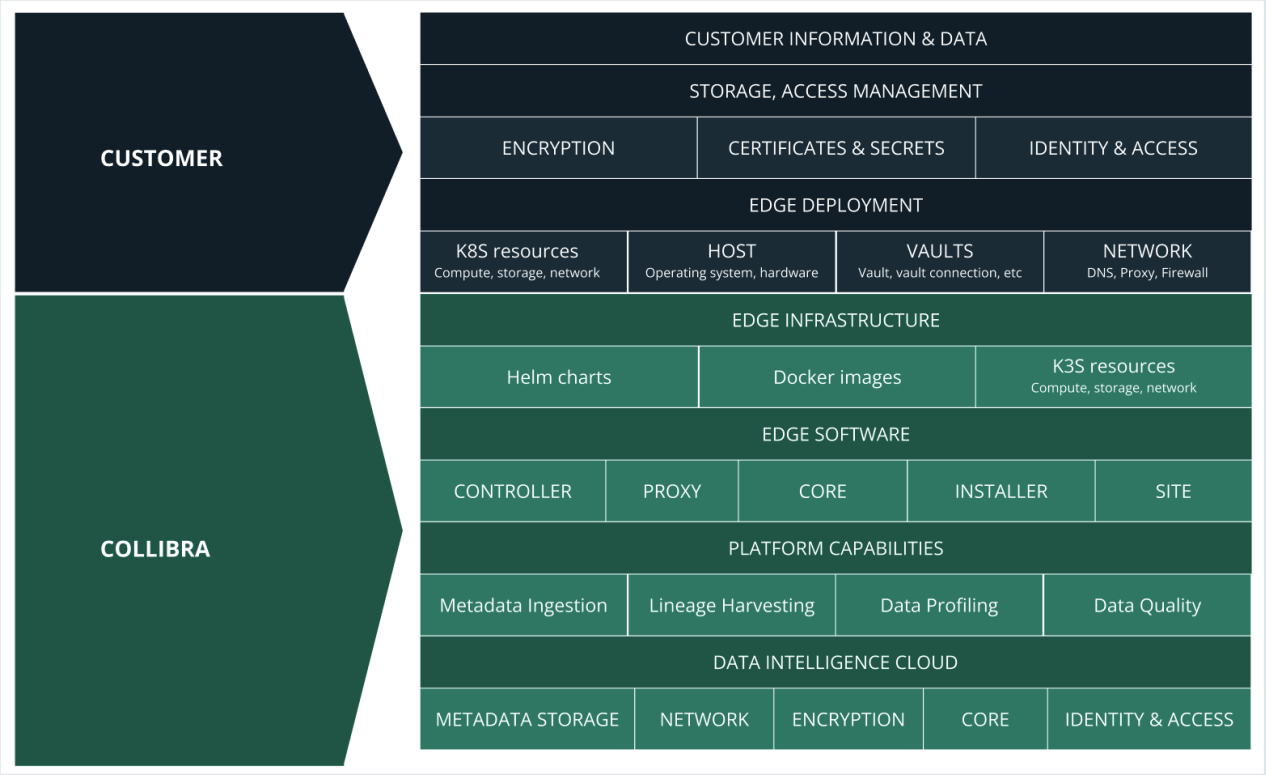
Introducing Edge

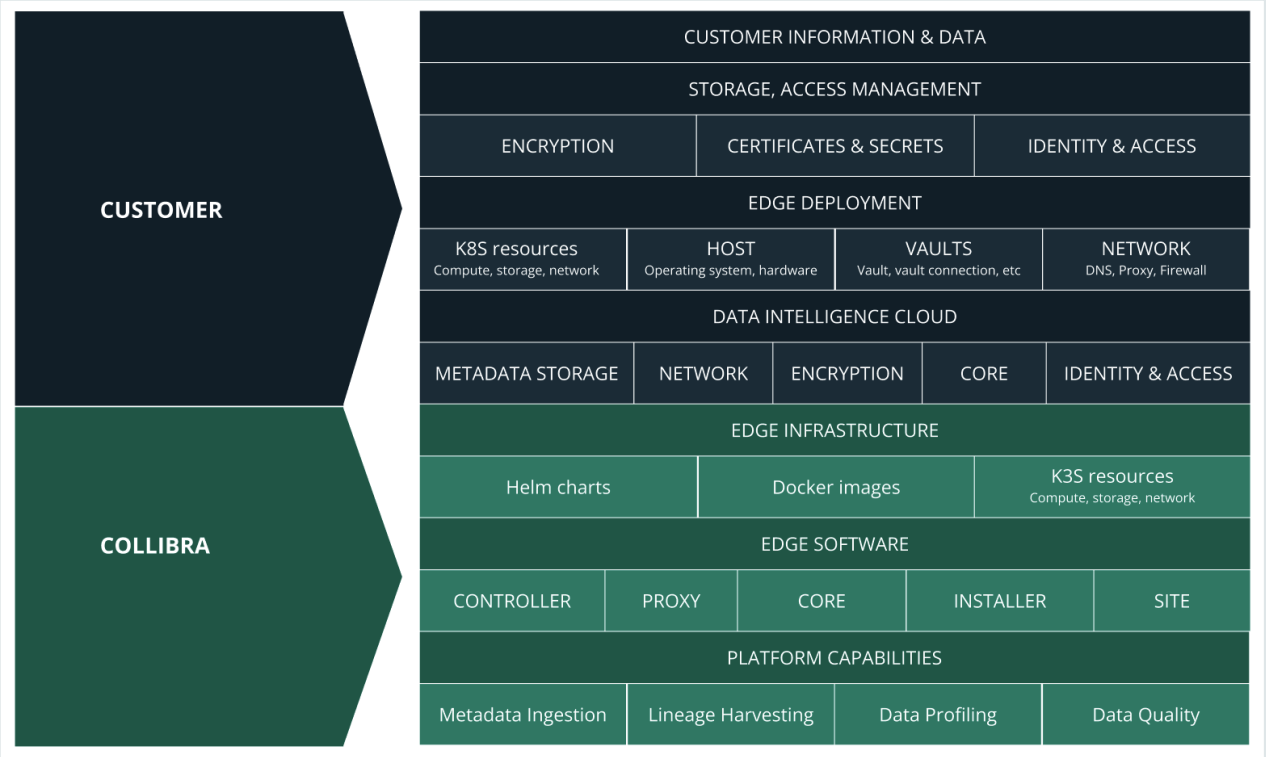
Edge is a cluster of Linux servers for accessing and processing data close to where it resides. It helps to connect to data sources and process information within your data landscape.

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 Responsibilities

The ownership of responsibility over the various Edge components is shared between you and Collibra. The diagram below illustrates which components you are responsible for and have control over, and those which belong to Collibra.


CUSTOMER
COLLIBRA
CUSTOMER INFORMATION & DATA
STORAGE, ACCESS MANAGEMENT
ENCRYPTION
CERTIFICATES & SECRETS
IDENTITY & ACCESS
EDGE DEPLOYMENT
K8S resources
Compute, storage, network
HOST
Operating system, hardware
VAULTS
Vault, vault connection, etc
NETWORK
DNS, Proxy, Firewall
EDGE INFRASTRUCTURE
Helm charts
Docker images
K3S resources
Compute, storage, network
EDGE SOFTWARE
CONTROLLER
PROXY
CORE
INSTALLER
SITE
PLATFORM CAPABILITIES
Metadata Ingestion
Lineage Harvesting
Data Profiling
Data Quality
DATA INTELLIGENCE CLOUD
METADATA STORAGE
NETWORK
ENCRYPTION
CORE
IDENTITY & ACCESS



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	Create 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.
2	Install the Edge site close to the data source you want to access. You can only install an Edge site on a Linux system that meets the necessary system requirements.	You have a global role with the Install Edge sites global permission in Collibra Data Intelligence Cloud.
3	Update the credentials of the Edge site user.	You have a global role with the Connect Edge sites to Collibra global permission in Collibra Data Intelligence Cloud.

Migrating to Edge from Jobserver

You can migrate Jobserver to Collibra's Edge for enhanced security, improved performance and even more functionality!

Why migrate to Edge?

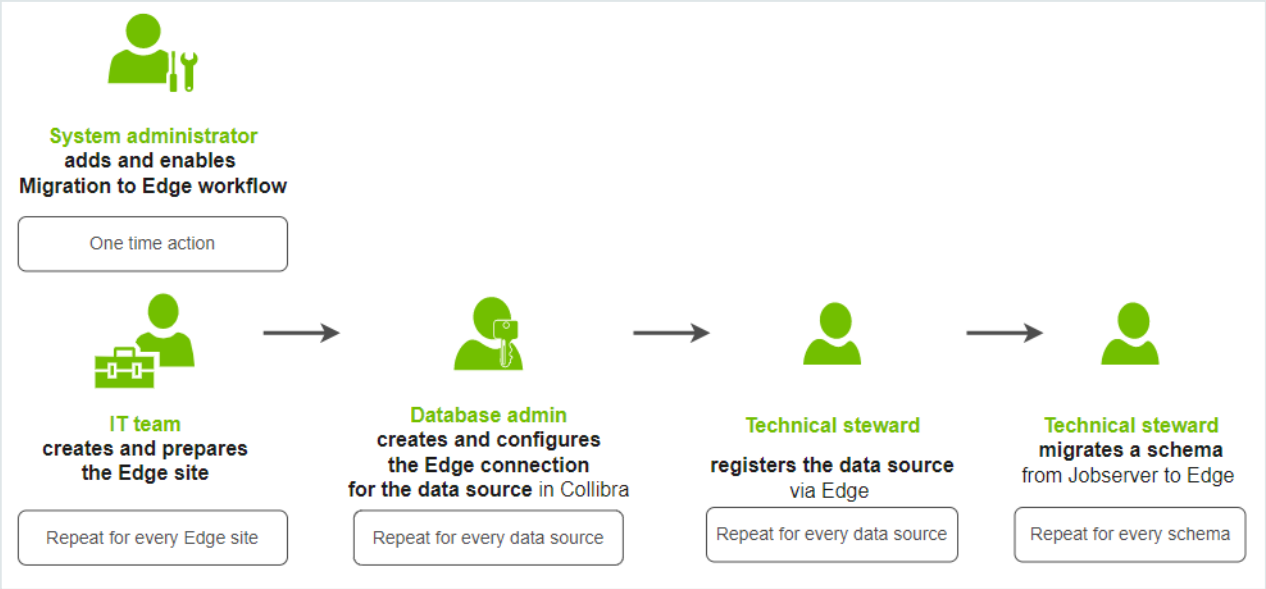
Edge provides our customers with all of the capabilities provided with Jobserver, but with better security controls and added capabilities. Edge provides seamless native integrations and on-site data processing solutions that prioritize security and proximity to the data, while keeping the processing of your data within your own environment.

	Edge	Jobserver
Capabilities	• Edge-certified JDBC connectors • JDBC metadata ingestion on Database level • JDBC data profiling and classification • JDBC data sampling • DQ connector • Multiple integrations, such as Amazon S3, Databricks, ADLS	• Jobserver-certified JDBC connectors • JDBC metadata ingestion: Schema level • Data profiling • Data classification • Data sampling • Some integrations
Security	• Edge provides the ability to mirror images in your private docker registry which allows you to scan containers per your security policy. This provides you with information on vulnerabilities around Edge containers and opens the dialogue among your security stakeholders regarding risk, tolerance, and remediation requirements. • Data source secrets / credentials are stored on the Edge site. • Classification is local to the Edge site, which means that none of your data leaves your premises. • Sample data will be requested from the data source live, and cached on the Edge site for a certain time frame (24 to 48 hours).	• Data source secrets / credentials are stored in the Collibra Cloud. • To classify data, client sample data is sent to the Cloud Classification platform. • Sample data is stored in the Collibra Cloud.
Performance	• Can ingest and profile in parallel. • No limit on the table size you can profile.	• All capabilities executed sequentially. • Limit on table size you can profile.

	Edge	Jobserver
Extensibility	Edge is a run time platform to host all capabilities, and this includes any new capabilities that will be developed and deployed in the future.	Jobserver requires several separate components for new capabilities, such as Separate Jobserver for Tableau ingestion or separate command line application for lineage harvester.
Maintenance	• Installer bundled with minimal binaries and artifacts. The installation is "live" in that the installer pulls images from the repository over the internet at install time. • Edge provides two upgrade modes. Edge can be upgraded as soon as there is a release available. It can also be updated when the customer prefers and is typically to review security vulnerabilities in a release, a feature in Smart Upgrade. ◦ Automatic: your Edge site is upgraded as soon as a new release version is available. ◦ Manual: you control when your Edge site is upgraded, allowing you to review security vulnerabilities in a release version before upgrading your Edge site.	• Console application on the Jobserver offers a user interface for some configuration and access to logs. Some configuration changes must be made directly in configuration files. • Jobserver only has one upgrade mode: Manual.

Migration to Edge overview

The following image illustrates the high-level steps of each user and the frequency these steps need to be performed to migrate data sources from Jobserver to Edge.



To migrate a data source, following steps are needed:

Step	Description
1	Enable the Migrate Schema to Edge workflow in your environment.
2	Install an Edge site.
3	Create an Edge connection for the data source and add the following capabilities for those connections: • Catalog JDBC ingestion • JDBC Profiling
4	Register the data source via Edge.
5	For each schema that you want to migrate from Jobserver to Edge for the data source, migrate the existing Schema asset. Once a schema is migrated, this schema can now be synchronized, profiled, classified, and so on via Edge.

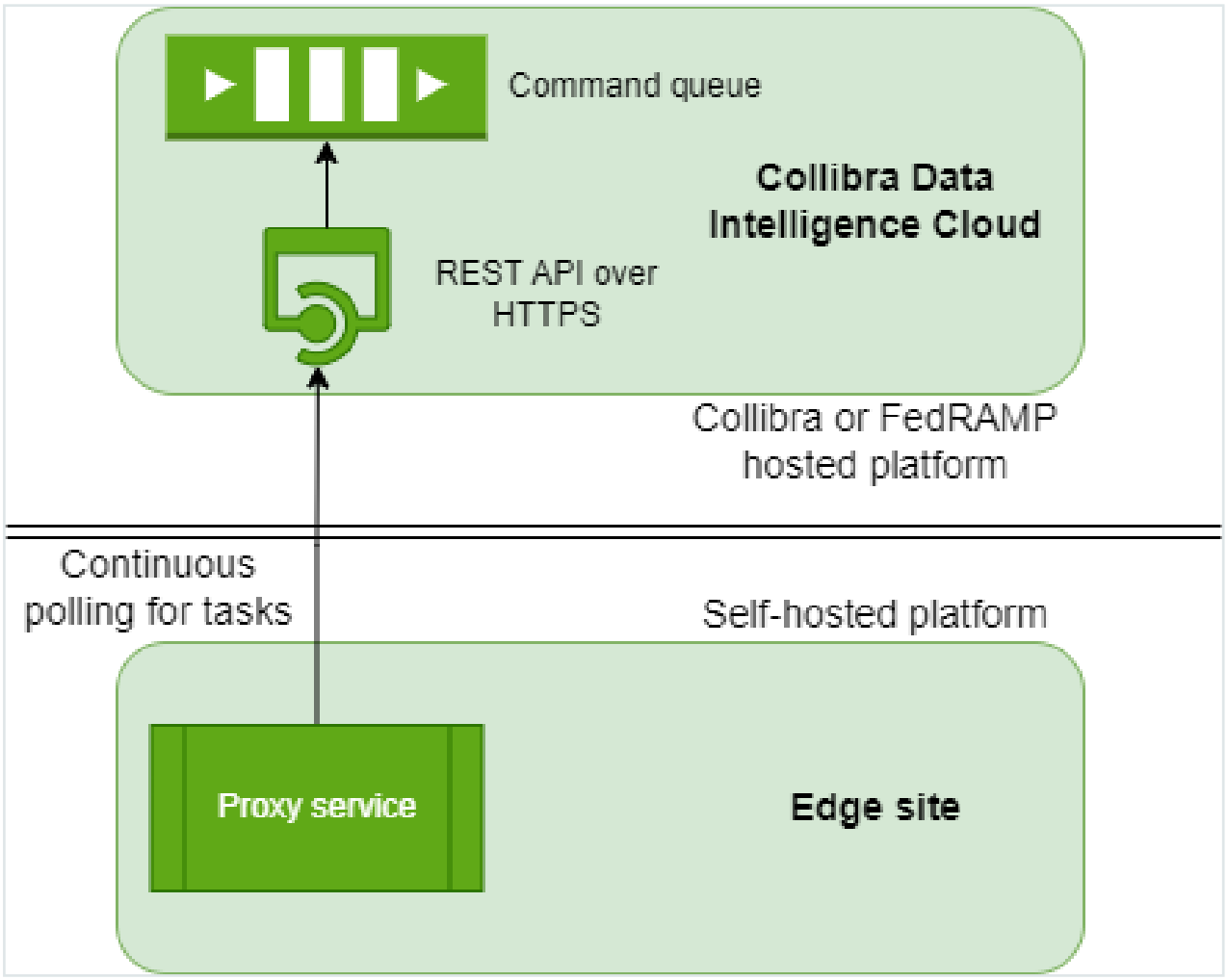
Edge security

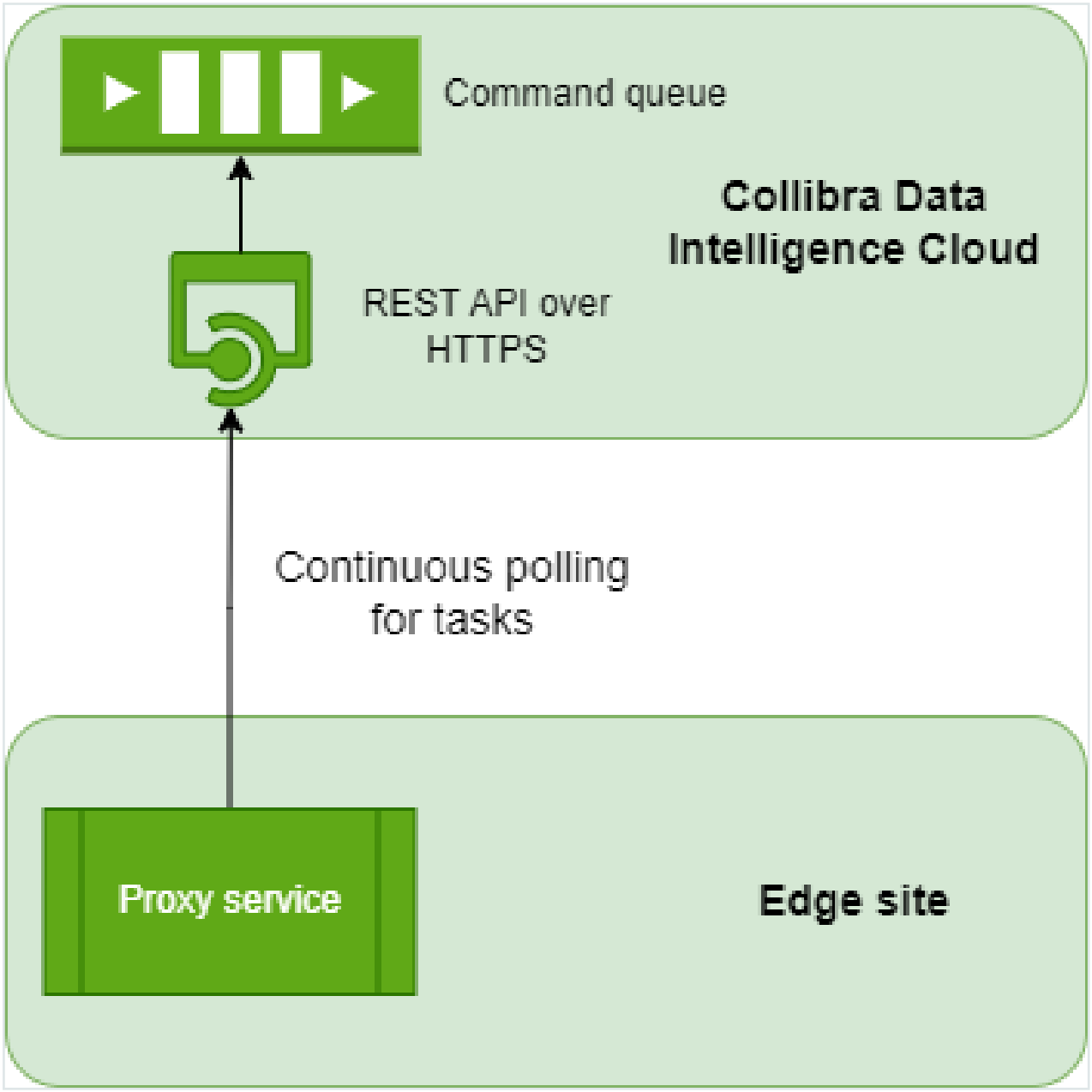
Edge is built with security first approach. All communication channels are secured by TLS 1.2 and all endpoints outside Edge are accessible only via authentication. Edge does not send or store any customer data, its purpose is to host capabilities that process the data in its own environment and to send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

> Note If you have any questions about data privacy and what information is sent via third part components, such as Datadog, please reach out to your Collibra representative.

Communication between Edge and Collibra

Edge operates over an outbound-only model – it executes tasks as commands polled from your Collibra platform. Communication to Collibra uses basic authentication over TLS 1.2. A user account is generated for communicating to Collibra each time the Edge site installer is downloaded. This user account is unique to each Edge site. It is possible to change the password of this user account by following the steps outlined in our Update Edge user password article.



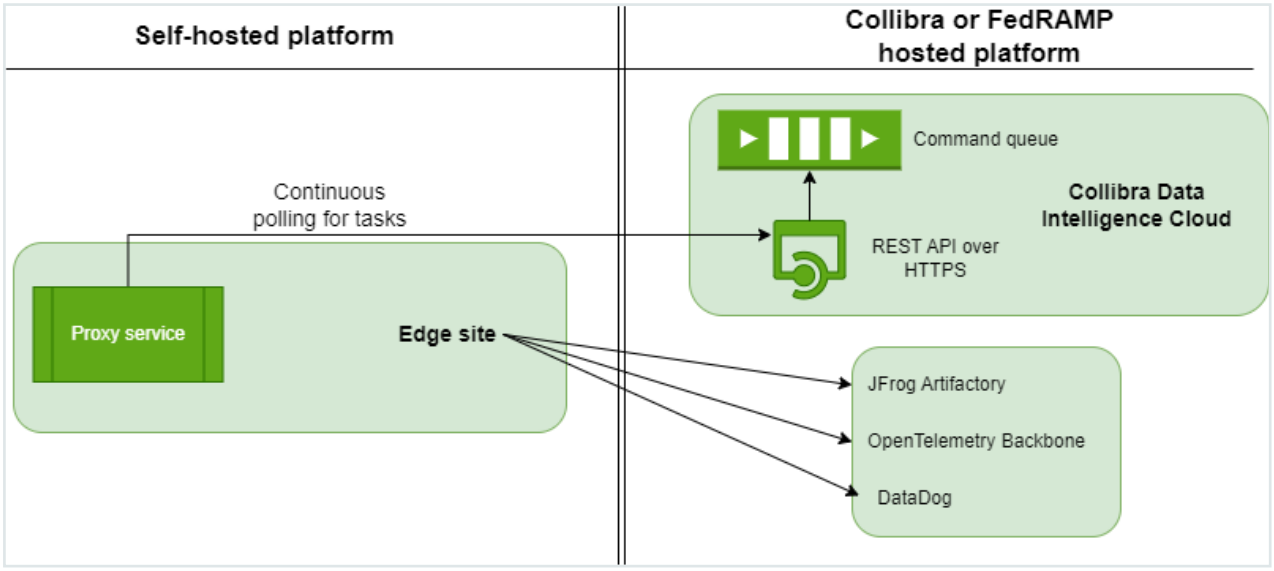


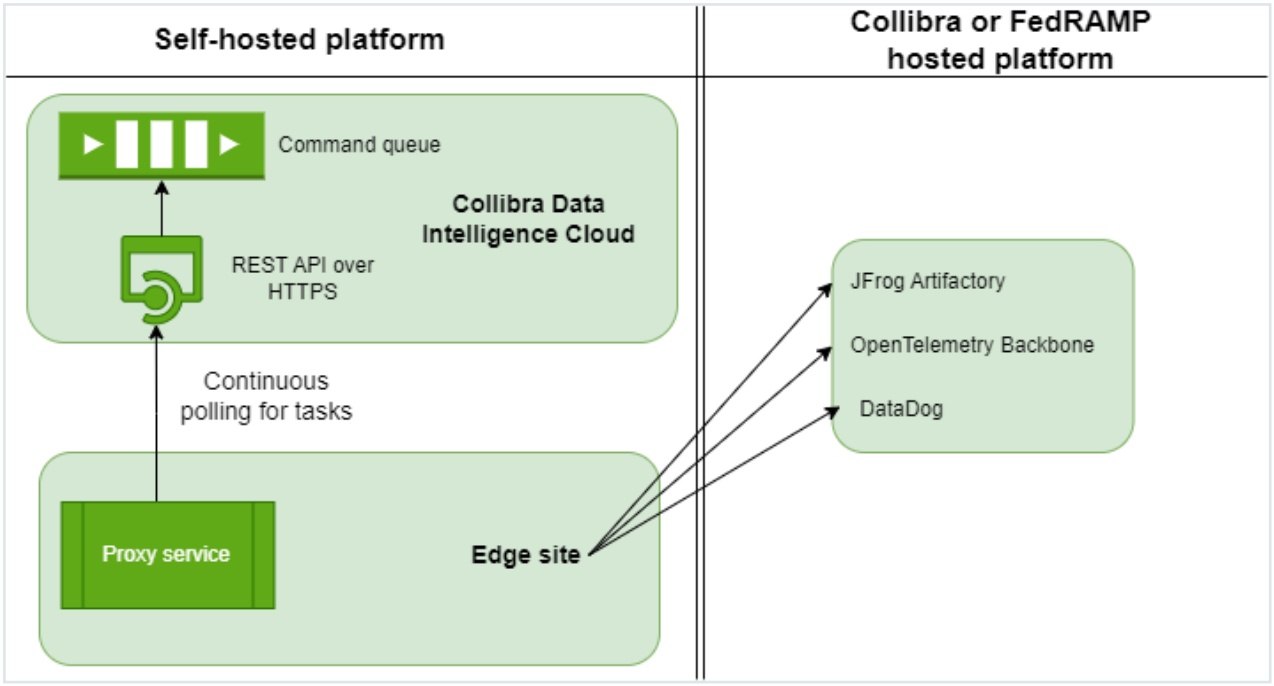
- Edge sites always use REST API endpoints to establish connections.
- Edge requires access to a Collibra server. It is needed for:
  - Reading a request queue, which is a queue with jobs that need to be run on Edge.
  - Returning the metadata results of Edge jobs.

- 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 over a secure channel 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.

Communication between Edge and other services

Edge communicates with other servers, such as JFrog, for maintenance purposes.





Edge requires access to the following servers:

Server	Communication	Authentication
JFrog	This is needed in order to download Helm Charts and Docker Images that are running on Edge.	API Key Pair over HTTPS.
OpenTelemetry Backbone	This is needed in order to upload various Edge related metrics.	Basic Authentication.
DataDog	This is needed in order to upload logs from all Edge components: • Core edge components • Edge capabilities , for example, ingestion, profiling, lineage, classification, quality.	API Key Pair over HTTPS.

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.

Security scanning

Before Collibra composes an Edge installation package, XRay scans are performed on all images consumed by Edge to identify and mitigate vulnerabilities. Contrast scanning is performed post installation for runtime vulnerability detection. This strategy ensures that Edge remains secure.

You can also run your own security scans. We recommend that you run the following command in order to remove old containers and images from an Edge host before running your own scans:

```
sudo /usr/local/bin/k3s crictl rmi --prune.
```

This prune command is a native docker command to clean unused docker objects such as images, containers, volumes and networks. Running this command will avoid false positive vulnerabilities when performing scans as Kubernetes, which is responsible for the garbage control of old Edge images and containers, is not guaranteed to have cleaned up the files before the scan is run.

What's next?

Pull images from the Collibra Edge docker registry with each new version to perform security scans and audits.

How to pull Collibra Edge docker images

You can pull docker images used by Edge to perform security scans and audits. With this process, you use docker CLI to authenticate to the Collibra Edge docker registry in order to get a list of images used by each Edge site version.

> Note This method of pulling images is only supported for security scanning of supported Edge versions, and not for new installations of an Edge site. If you want to use a private docker registry for new Edge site installations, use the method outlined in the Configure a private docker registry documentation. For more information on which versions of Edge are supported with the latest release, go to our Compatibility matrix.

Steps

1. Authenticate with Collibra Edge docker registry.
   a. In your Edge site installer, find the registries.yaml file, which contains credentials to download an installer.
   b. Run the following command with the username and password from the registries.yaml file:

```
docker login edge-docker-
delivery.repository.collibra.io -u username -p
<password>
```

> Note Docker credentials are read-only

2. Define the version of Edge you want to scan.
   - We recommend using the latest Edge site version.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.

c. Above the table, to the right, click **Create Edge site**.
   » The **Create Edge site** wizard starts.
d. Select **Manual** as the **Upgrade Mode**.
e. Copy the Edge site version you want to use. The latest version should be preselected.

3. Obtain a list of images that need to be mirrored.

> Note Parameter key:
> a. dic_user: The username you use to log into Collibra Data Intelligence Cloud.
> b. dic_pass: The password you use to log into Collibra.
> c. dic_url: The Collibra URL.
> d. edge_version: The version of Edge you want to upgrade to found in step 2. For example, <2023.11>.

- Use the CURL command to get the list of images.

> Note You must install the jq command to use the CURL command.

```
curl -u <dic_user> -p <dgc_pass>  <dic_
url>/edge/api/rest/v2/releaseinfo/<edge_version> |
\
jq '.images[].image' -r
```

> Important If you use SSO, instead of username and password, you need to open **/resources/manifests/sc-dgc-secret.yaml** to obtain the username and password listed in the file. Enter the username for `<dic_user>` and enter the password for `<dgc_pass>`.

4. Pull the images.
   Perform the following command for each image mentioned in the list obtained in step 3.

```
docker pull <image>
```

Example

```
docker pull edge-docker-
delivery.repository.collibra.io/capabilities/edgeharv
ester:1.5.0
```

Storing connection credentials

> Note You can manage your data source secrets and credentials by using Vaults.

Connections and capabilities credentials are stored solely on the Edge site. While at rest, credentials use envelope encryption where the credentials are encrypted by a key, which on its turn is encrypted by another key.

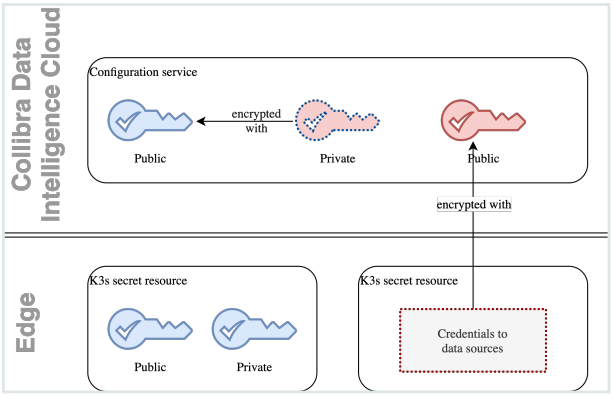
The Edge native encryption mechanism is based on two RSA key pairs. They are stored in the following places:

Keys	DIC server	Edge server	Purpose	When is it generated?	Where is it stored?
Red public key	Yes	No	Used to encrypt connection credentials.	After the Edge site is successfully installed.	In the Collibra Cloud.
Red private key	Yes (encrypted using public blue key)	No	Used to decrypt connection credentials.	After the Edge site is successfully installed.	Encrypted by the Blue public key in the Collibra Cloud
Blue public key	Yes	Yes	Used to encrypt red private keys.	During the installation or re-installation of the Edge site is	Encrypted on the Edge site.
Blue private key	No	Yes	Used to decrypt red private key.	During the installation or re-installation of the Edge site is	Encrypted on the Edge site.

The blue key pair is stored as a Kubernetes credential on the Edge server 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 credential on Edge is encrypted with the red public key, which is sent to the Edge site for each capability execution, encrypted with the blue public key. Once on the Edge site, Edge be decrypted

with the red private key, and credentials that are needed to execute a connection or a capability are decrypted and injected into the capability container.



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

Customer Credentials

> Note You can manage your data source secrets and credentials by using Vaults.

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 (k3a 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 128 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. The text is placed in a command queue for your Edge site to execute, as it does for other commands such as run job or cancel job. The text is picked up by the Edge site's polling mechanism for execution, which in this case, stores the Edge site credentials as a Kubernetes secret.
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 Edge secret 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

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.

Collibra 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.

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

Edge Cache

Any metadata, logs or metrics stored in the Edge cache are encrypted by default to improve the security of your data and the platform. You are not required to make any changes to this security policy, and there is no impact on the functionality of your Edge sites.

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.

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 are troubleshooting an issue, you have to extract these logs as soon as possible after the completion or failure of the capability, as these files will be removed after a day, and send them to Collibra Support via a support ticket.

Additional information

For more information, go to the following resources:

- Edge logging
- Create Metadata connector log file

Host hardening on K3S-based integration

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 following lines to the **k3s.service.env** file:
   - *K3S_KUBECONFIG_OUTPUT=/dev/null*.
   - *K3S_KUBECONFIG_MODE=666*

   > 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 following lines from the **k3s.service.env** file:
   - *K3S_KUBECONFIG_OUTPUT=/dev/null.*
   - *K3S_KUBECONFIG_MODE=666*
4. Restart the K3S service: `systemctl restart k3s`
5. Check if the KUBECONFIG file is empty: `cat /etc/rancher/k3s/k3s.yaml`

Configure a private docker registry

You can set up a private docker registry using JFrog's Artifactory. A private docker registry allows you to use your own infrastructure to perform your own security scans and audit docker images consumed by Edge. It also reduces the risk of a compromised Collibra registry running compromised code on your network.

Your private docker registry, or the location where you pull your images from, must be behind a secure firewall that is not accessible via the internet. Otherwise, unintended users may gain access to proprietary information, which will trigger a security incident within your organization and Collibra.

> Note
> - Other forms of security scanning, such as penetration tests, can be performed either independently or as a part of the security flow that includes a private docker registry.
> - Security scan reports are only be accepted for supported Edge site versions. This is because security fixes are not applied to old, out-dated versions of Edge. For example, from November 19 2023 to February 24, 2024, security scans are only accepted for Edge site version 2023.11 and subsequent weekly updates (2023.11.x). For information on which Edge versions are supported with the latest release, go to the Compatibility between Edge sites and Collibra Data Intelligence Cloud.

Before you begin

- Switching to a private docker registry is only possible during installation.

Requirements and permissions

- You must have admin access to JFrog Artifactory.
- You have the Edge site administrator global role.
- The registry user should have read/pull permissions for the docker registry. This should be validated manually prior to installing Edge.

We use JFrog's Artifactory, which is a repository manager that allows for dynamic mirroring of docker registries, to manage our repository. If your company has their own JFrog Artifactory, you can configure it to automatically mirror images from Collibra's Artifactory.

This method is easy to set up and supports both manual and automatic upgrade modes. However, it has limited options for security scanning.

Steps

1. Mirror Collibra's registry.

   > Note When configuring a private Artifactory docker registry, the registry must follow the "subdomain first" method for pulling images.
   > - Correct example: `<repository-key>.artifactory.my.org`
   > - Incorrect example: `artifactory.my.org/<repository-key>`

   a. From the **Administration** pane in JFrog, in the **Repositories** menu, click **Repositories**.
   b. Click the **Remote** tab.
   c. Click **New Remote Repository**.
   d. In the **Select Package Type** section, select **Docker** .
   e. Name the registry using the repository key of your choice.
   f. Set the URL to: *edge-docker-delivery.repository.collibra.io*
   g. Enable remote authentication.

      > Note The username and password can be found in the unzipped installer file **registries.yaml**.

   h. Click **Save and finish**.

2. Install the new or reinstall the existing Edge site using the full private registry URL. This full private registry URL is the private Artifactory hostname and repository key from sub-step 5, in the previous step.

   > Note We recommend you download or redownload a new installer before performing this step.

a. Create a new Edge site.
b. Select the Edge site upgrade mode.
c. Select the latest version of Edge.
d. Download the installer.

a. Upgrade to the latest Edge site version.
b. Create a backup, and reinstall your Edge site.

a. The installer supports the following installation parameters:
   - `--registry-url`, example: `https://edge-docker-delivery.repository.collibra.io`
   - `--registry-host`, this is optional and is automatically derived from `--registry-url, i.e., edge-docker-delivery.my-registry.-docker.io`
   - `--registry-pass`, not required if registry is public.
   - `--registry-user`, not required if registry is public.

3. Add the following setup parameters to install Edge with support for a private docker registry:
   - For Bare metal installations: `./install-master.sh (...usual parameters...) --registry-url https://private-docker-er.registry.com --registry-user user --registry-pass pass`
   - For EKS installations: `./run-installer-job.sh (...usual parameters...) --registry-url https://private-docker-er.registry.com --registry-user user --registry-pass pass`

Edge Vaults

This feature is **available only** in the latest user interface (beta).

The Edge Vault feature allows you to integrate your Edge site with your existing vault provider and implement your organization's credential management policies for any data source to which Edge connects. Unlike Edge secrets, with Edge Vaults:

- You can pull your sensitive information from your vault application or service, rather than manually entering your information into Edge where it is encrypted and stored as Kubernetes secrets.
- Edge does not keep any sensitive information in the Edge site, it relies on the vault integration to establish a secure connection to your data sources.
- It is easier to rotate your secrets, because you do not have to manually rotate them in Collibra. Managing your data source credentials only needs to be performed in your organization's vault.

The following vault integrations are supported:

- CyberArk Vault
  - Supported version:
    - CyberArk Credential Provider: 8.0.0
- HashiCorp Vault
  - Supported version:
    - HashiCorp Vault 1.15.x
  - Support secret engines:
    - Key Value V2 Secret Engine
    - Database Secret Engine
- Azure Key Vault
- AWS Secrets Manager
- Google Secret Manager

Edge Command Line Interface

This feature is **available only** in the latest user interface (beta).

The Edge Command Line Interface (CLI) is a tool that allows you to set up and manage your vault integrations and secrets, and included with your Edge site installer. With the CLI, you can:

- Add a Vault integration.
- Edit a Vault integration or rotate authentication credentials.
- Delete a Vault integration.

You can downland the Edge CLI by running the following command in the Edge tool:

```
edge download-edgecli
```

> Note The Edge CLI tool is only available on Linux.

Access help

If you need any help with the command parameters, run the following command using the Edge CLI, where `<authMethod>` is the type of authentication method you are using for your vault:

```
./edgecli vault create "<vault>" "<authMethod>" -h
```

Command	Description
`<vault>`	The type of vault application you use, for example, CyberArk or HashiCorp.
`<authMethod>`	The authentication method you use to connect to your vault.

For specific vault help examples, see the online version of this guide.

Integrate your Edge site with your vault

This feature is **available only** in the latest user interface (beta).

You can integrate Edge site with your vault provider to more easily manage your data source information and set up your Edge site connections.

For steps on how to integrate your Edgesite with your vault, see the online version of this guide.

Retrieve your vault integration information via the Edge CLI

This feature is **available only** in the latest user interface (beta).

You can review the details of your vault integrations from the Edge CLI tool.

Before you begin

- Ensure that your environment uses the latest user interface (beta).
- You have integrated your Edge site with your vault.
- You have installed and configured the Edge CLI tool.

Retrieve information on all vault integrations

You can retrieve a list of all of your configured vault integrations on an Edge site by running the following command in the Edge CLI:

```
./edgecli vault list
```

This command provides the following vault integration information:

- ID
- Name
- Type
- Description
- Address

> Note Address is returned only for CyberArk Vault integrations.

> Note This command will not return any authentication information of your vault integrations. If you want to retrieve authentication information about your vaults, you need to retrieve the details of a specific vault.

Retrieve specific vault details

Run the following command in the Edge CLI to compile all of the details of a specific vault integration, such as ID, description, and authentication type:

```
./edgecli vault get "<name>"
```

Command	Description
`<name>` (required)	The name of the vault instance. It is required and it must be unique within an Edge site. For Kubernetes guidelines on the required naming conventions of the <name> parameter, go to Labels and Selectors. > Note You can choose to provide `<vaultId>` instead of `<name>`. If you don't have the vault ID, you can get it by retrieving all vault integrations.

This command provides the following vault integration information:

- ID
- Name
- AppID
- Type
- Description
- Address

  > Note Address is returned only for CyberArk Vault integrations.

- AuthType

Edit vault integration configuration via Edge CLI

This feature is **available only** in the latest user interface (beta).

You can inspect and update the configuration of your vault integration and rotate the vault credentials using the Edge CLI tool.

For steps on how to edit your Edge site vault configuration, see the online version of this guide.

Delete an Edge site vault integration

This feature is **available only** in the latest user interface (beta).

If you no longer need a vault integration on your Edge site, you can use the Edge CLI tool to delete the vault integration.

> Warning Deleting a vault integration will impact any connection on the same Edge site that uses this vault integration. Before deleting a vault integration, ensure that no Edge site connections use that vault integration.

Before you begin

- Ensure that your environment uses the latest user interface (beta).
- You have added a vault in your Edge site.
- You have installed and configured the Edge CLI tool.

Steps

Run the following code in the Edge CLI:

```
./edgecli vault delete "<name>"
```

Command	Description
`<name>` (required)	The name of the vault instance. It is required and it must be unique within an Edge site. For Kubernetes guidelines on the required naming conventions of the <name> parameter, go to Labels and Selectors. Note You can to choose provide `<vaultId>` instead of `<name>` . If you don't have the vault ID, you can get it by retrieving all vault integrations.

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 your 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 their requirements, for example the number of networks and secure environments, as well as 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.
ID	The unique ID of the Edge site, which is generated automatically when you create the Edge site.

Property	Description
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 install an Edge site on a server. This section is only visible when the Edge site has the status **To be installed**.

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 installed 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

To use Edge, you must ensure that the following system requirements are met.

Software requirements

- You must be able to install the Edge software on the latest version of RedHat Enterprise Linux 8.x.

  > Note
  > - We recommend not installing Edge on end-of-Life versions of RedHat Enterprise Linux.
  > - We recommend ensuring the k3s version installed on your Edge site can be run on the version of RedHat Enterprise Linux you have.
  > - For more information on installing Edge on a Linux server, go to How to prepare a Linux server for running and installing Edge on the Collibra Support Portal.

- Your Edge site installer must use an Edge supported k3s version.
- The **sudo** package is installed on the Linux host.
- The user who installs Edge has full sudo access (`ALL=(ALL) ALL`).
- Optionally, if you want SE Linux enabled, install the following policy packages before installing Edge:
  Packages[1]

> Tip If you are an early adopter or you use Edge for beta testing purposes, we highly recommend that you disable SELinux.

1
- yum install -y container-selinux selinux-policy-base
- yum install -y https://rpm.rancher.io/k3s/stable/common/centos/7/noarch/k3s-selinux-0.2-1.el7_8.noarch.rpm

These packages are not hosted by Collibra. If you have any questions, contact your internal teams.

Hardware requirements

> Note When installing on k3s, the Virtual Machine (VM) must be dedicated to a single Edge site installer.

You need the following minimum hardware requirements:

- 64 GB memory.
- 16-core CPU with x86_64 architecture.
- At least 60 GB of free storage for Edge application storage requirements:
  - You have at least 50 GB of free storage on the partition that contains **/var/lib/rancher/k3s**. The partition mountpoint should not have the noexec option.

```
mkdir -p /var/lib/rancher/k3s
mkfs.xfs /dev/<block-device-name>
mount /dev/<block-device-name> /var/lib/rancher/k3s
echo '/dev/<block-device-name> /var/lib/rancher/k3s
xfs defaults 0 0' >> /etc/fstab
```

> Note This is the default install path. If it is not created as a separate mount point after following the steps above, the install will use 50 GB of disk space from either /var, or if not present, the root level of the drive.

> Warning Any data in this location is fully managed by the Edge site. Do not save any other data in this location as the data can be removed by Edge without notification.

  - You have at least an additional 5 GB of space in /var/log for Edge components. Edge uses hardcoded /var/log to write logs:

        - Up to 1.1 GB of space for writing K3S audit logs.
        - Maximum of 60 MB per container for pod logs. The number of containers depends on the workload.
    - You have at least an additional 5 GB of space on the partition that holds /var/lib/kubelet. Edge uses hardcoded /var/lib/kubelet/pods/*/volumes/kubernetes.io~empty-dir/* to write ephemeral data related to kubernetes.
- At least 500 GB of dedicated storage for Edge data storage requirements:
    - 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 must not be shared with other services because Edge can delete and overwrite files on this location without notice. Therefore, do not use **/home/<username>** or **/var**.

- If you run the Linux server on AWS, Azure, 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

Commercial

- An Edge site needs outbound connections to all of the following:
  - The URL of your Collibra Data Intelligence Cloud environment.
  - **https://http-intake.logs.datadoghq.com**: This URL is used to collect some of the logs from Edge for issue diagnosis. We do not send JDBC driver logs from Edge to Datadog.
  - **https://*.repository.collibra.io**: This URL serves as the primary source for downloading the latest Edge docker images from Collibra's docker registry and helm-chart repository.

    > Note If the allowlist does not accept wildcards:
    > - https://repository.collibra.io
    > - https://edge-docker-delivery.repository.collibra.io
    > - https://mirror-docker.repository.collibra.io

  - **https://otlp-http.observability.collibra.dev/**: This URL is used to ingest metrics and traces for monitoring the health and usage of Edge sites.
- Access to all data sources you need to connect to your Edge sites.
- Your Edge site has to be able to connect to port 443.
- If you intend to use a man-in-the-middle (MITM) proxy, you need to add the specific truststores customization to the ca.pem, because Edge does not use the host TLS trustsore. For more information, go to Configure a forward proxy.

- The resolve configuration file of your Linux host has maximum three search domains and two name servers.

Note
- Ensure that the network connectivity between the internal cluster and the service CIDRs use by k3s (which are by default 10.42.0.0/16 and 10.43.0.0/16) is not blocked.
- In case `firewalld` 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
```

FedRAMP

- An Edge site needs outbound connections to all of the following:
  - The URL of your Collibra Data Intelligence Cloud environment.
  - **https://http-intake.logs.ddog-gov.com**
  - https://*.artifactory-gov2prod.collibra.com/

    Note If the allowlist does not accept wildcards:
    - https://artifactory-gov2prod.collibra.com
    - https://edge-docker-delivery.artifactory-gov2prod.collibra.com
- Access to all data sources you need to connect to your Edge sites.

- Your Edge site has to be able to connect to port 443.
- If you intend to use a man-in-the-middle (MITM) proxy, you need to add the specific truststores customization to the ca.pem, because Edge does not use the host TLS trustsore. For more information, go to Configure a forward proxy.
- The resolve configuration file of your Linux host has maximum three search domains and two name servers.

Note
- Ensure that the network connectivity between the internal cluster and the service CIDRs use by k3s (which are by default 10.42.0.0/16 and 10.43.0.0/16) is not blocked.
- In case `firewalld` 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 to a single Edge site installer, do not use the cluster for other purposes.

- AWS EKS 1.27 is supported for existing Edge sites.

  > Note
  > - EKS versions 1.26 and older are no longer supported for new Edge site installations.
  > - You must upgrade to the latest supported EKS version, EKS 1.27, by the 2024.02 release. If you are EKS 1.24 or lower, you must follow the instructions outlined in our How to upgrade Edge sites on EKS from EKS 1.24 and lower to EKS 1.27 on 2023.11 to ensure you can upgrade successfully.

  - We support EBS-CSI driver for 1.23.
- AWS EKS worker nodes use the EKS optimized Amazon Linux 2 AMI
- EKS cluster has IRSA enabled
- Set up security groups to ensure that worker nodes can communicate with each other on non-privileged ports.

```
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.
- Ensure your Kubectl client is compatible with the relevant EKS version.

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 each with 4 core CPU and 16 GB.
- Each worker node needs at least 100 GB free disk space to store Docker

images.
- We recommend you have at least 2 worker nodes in the EKS cluster.

> Note The number of jobs that can run at the same time is limited by the available EBS volumes attached to the specific AWS EC2 instance type. On average, an AWS EC2 instance type has 20 EBS volumes available and Edge jobs consume 3 EBS volumes. In order to run more jobs at the same time, you need to increase the number of nodes in the cluster, which increases your available EBS volumes. If you have any questions, contact Collibra support.

Network requirements

Commercial

- An Edge site needs outbound connections to all of the following:
  - The URL of your Collibra Data Intelligence Cloud environment.
  - **https://http-intake.logs.datadoghq.com**: This URL is used to collect some of the logs from Edge for issue diagnosis. We do not send JDBC driver logs from Edge to Datadog.
  - **https://*.repository.collibra.io**: This URL serves as the primary source for downloading the latest Edge docker images from Collibra's docker registry and helm-chart repository.

    > Note If the allowlist does not accept wildcards:
    > - https://repository.collibra.io
    > - https://edge-docker-delivery.repository.collibra.io
    > - https://mirror-docker.repository.collibra.io

  - **https://otlp-http.observability.collibra.dev/**: This URL is used to ingest metrics and traces for monitoring the health and usage of Edge sites.
- Access to all data sources you need to connect to your Edge sites.
- Your Edge site has to be able to connect to port 443.

- If you intend to use a man-in-the-middle (MITM) proxy, you need to add the specific truststores customization to the ca.pem, because Edge does not use the host TLS trustsore. For more information, go to Configure a forward proxy.
- 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
> ```

FedRAMP

- An Edge site needs outbound connections to all of the following:
  - The URL of your Collibra Data Intelligence Cloud environment.
  - **https://http-intake.logs.ddog-gov.com**
  - https://*.artifactory-gov2prod.collibra.com/

    > Note If the allowlist does not accept wildcards:
    > - https://artifactory-gov2prod.collibra.com
    > - https://edge-docker-delivery.artifactory-gov2prod.collibra.com
- Access to all data sources you need to connect to your Edge sites.
- Your Edge site has to be able to connect to port 443.

- If you intend to use a man-in-the-middle (MITM) proxy, you need to add the specific truststores customization to the ca.pem, because Edge does not use the host TLS trustsore. For more information, go to Configure a forward proxy.
- 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

As jobs are run on an Edge site, rather than on the Collibra platform, creating an Edge site allows you to have a processing runtime at your own 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.
- Your server meets all system requirements.
-
  > Note You must restart the Data Governance Center service when you have enabled this option.

Steps

1. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
   » The Collibra settings page opens.
2. Click **Edge**.
   » The Edge sites overview opens.
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	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.

5. Select the **Upgrade Mode** for this Edge site.

6. 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, or if necessary, first configure a forward proxy.

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.

Steps

1. Download the installer:
    a. Open an Edge site.
        a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
            » The Collibra settings page opens.
        b. Click **Edge**.
            » The Edge sites overview opens.
        c. Click the name of an Edge site in the Edge site overview.
            » The Edge site page opens.

b. Click **Download** in the **Installer and properties files** section.

> 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 want to install the Edge site software.

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

> Note
> - Keep the installer or the contents of the extracted installer in a secure location on your server. These contents contain various tools that you may need later, for example to troubleshoot issues.
> - If you want to run a script or executable file from the extracted directory, ensure that the directory is not mounted as `noexec`. If a directory is mounted as `noexec`, scripts and executable files will be prevented from being run within the directory.

3. Run 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.

a. Installation with profiling and classification enabled:

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

for example:

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

b. Installation with profiling and classification disabled:

```
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
```

» 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 have already installed the Edge site and you want to enable classification afterwards, see this article.

1. Download the installer:
   a. Open an Edge site.
      a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
         » The Collibra settings page opens.
      b. Click **Edge**.
         » The Edge sites overview opens.
      c. Click the name of an Edge site in the Edge site overview.
         » The Edge site page opens.
   b. Click **Download** in the **Installer and properties files** section.

      > 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 want to install the Edge site.

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

> Note
> - Keep the installer or the contents of the extracted installer in a secure location on your server. These contents contain various tools that you may need later, for example to troubleshoot issues.
> - If you want to run a script or executable file from the extracted directory, ensure that the directory is not mounted as `noexec`. If a directory is mounted as `noexec`, scripts and executable files will be prevented from being run within the directory.

3. Run the installation on the machine that has the Kubernetes connection.
   a. Clean installation:

```
./run-installer-job.sh properties.yaml
```

   b. Installation with classification enabled:

```
./run-installer-job.sh properties.yaml --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 have already installed the Edge site and you want to enable classification afterwards, see this article.

Configure a forward proxy

For security reasons, it is possible that an Edge site has to connect cloud services via a forward HTTP proxy. Complete steps 1-3 to update **proxy.properties** before installing the Edge site.

If you use a forward proxy that decrypts TLS traffic, a so-called man-in-the-middle (MITM) proxy, complete all 4 steps to configure the forward proxy and enable the MITM proxy.

> Warning MITM is not supported by all capabilities. Review the capability documentation to confirm if MITM is supported for your capability.

Steps

1. Download the Edge site installer:
    a. Open an Edge site.
        a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
            » The Collibra settings page opens.
        b. Click **Edge**.
            » The Edge sites overview opens.
        c. Click the name of an Edge site in the Edge site overview.
            » The Edge site page opens.
    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	A comma-separated list of IP or DNS addresses that can bypass the proxy server. This list must include at least the Kubernetes cluster's internal IP addresses and the Kubernetes nodes' IP and DNS addresses. The list may not contain spaces. Tip To get the values for this setting, you can use the **edge-get-noproxy.sh** script, which you can find in the extracted installer directory under **/resources/tools**. However, make sure that your network administrator reviews these values. where: ◦ <host-ip-addresses>: for example `172.20.0.0/16`. ◦ <host-dns-names>: for example `*.compute.internal`. ◦ <k8s-svc-ip-addresses>: is by default `10.43.0.0/16`, but this can differ for other k8s flavors or configurations. ◦ <k8s-pod-ip-addresses>: is by default `10.42.0.0/16`, but this can differ for other k8s flavors or configurations. ◦ <others>: other IP addresses that don't need to be proxied. Add at least `169.254.169.254.` for AWS. Example `noProxy=172.20.0.0/16,*.compute.internal,10.43.0.0/16,10.42.0.0/16,169.254.169.254`

Setting	Value
proxyHost	The IP or DNS address of the proxy server. Example `proxyHost=site4-proxy.shared.edge.collibra.dev`
proxyPort	The TCP port of the proxy server. This value must be a quoted string and not an integer value. Example `proxyPort=3128`
proxyUsername	The username to authenticate at the proxy server. Example `proxyUsername=edge` Note Usernames with single quotations ', double quotations ", and backslashes \ need to be escaped using an additional backslash. For example, if the username is ge'smith\, it would need to be entered into proxy.properties as username: ge\'smith\\.
proxyPassword	The password to authenticate at the proxy server. Example `proxyPassword=la;fs90jpo4j3rR%` Note Passwords with single quotations ', double quotations ", and backslashes \ need to be escaped using an additional backslash. For example, if the password is te"st\1234', it would need to be entered into proxy.properties as password: te\"st\\1234\'.

```
#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	A comma-separated list of IP or DNS addresses that can bypass the proxy server. This list must include at least the Kubernetes cluster's internal IP addresses and the Kubernetes nodes' IP and DNS addresses. The list may not contain spaces. Tip To get the values for this setting, you can use the **edge-get-noproxy.sh** script. However, make sure that your network administrator reviews these values. where: ◦ <host-ip-addresses>: for example `172.20.0.0/16`. ◦ <host-dns-names>: for example `*.compute.internal`. ◦ <k8s-svc-ip-addresses>: depends on your EKS installation. Typically this is `10.100.0.0/16` or `172.20.0.0/16`. ◦ <k8s-pod-ip-addresses>: depends on your EKS installation. Typically they are the same subnets as in the VPC, for example `172.20.0.0/16`. ◦ <others>: other IP addresses that don't need to be proxied, for EKS, always add `169.254.169.254`.. Example `noProxy=172.20.0.0/16,*.compute.internal,10.100.0.0/16,169.254.169.254`

Setting	Value
proxyHost	The IP or DNS address of the proxy server. Example `proxyHost=site4-proxy.shared.edge.collibra.dev`
proxyPort	The TCP port of the proxy server. This value must be a quoted string and not an integer value. Example `proxyPort=3128`
proxyUsername	The username to authenticate at the proxy server. Example `proxyUsername=edge` Note Usernames with single quotations ', double quotations ", and backslashes \ need to be escaped using an additional backslash. For example, if my username is ge'smith\, it would need to be entered into proxy.properties as username:ge\'smith\\.
proxyPassword	The password to authenticate at the proxy server. Example `proxyPassword=la;fs90jpo4j3rR%` Note Passwords with single quotations ', double quotations ", and backslashes \ need to be escaped using an additional backslash. For example, if my password is te"st\1234', it would need to be entered into proxy.properties as password: te\"st\\1234\'.

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. To enable Edge via a MITM proxy (a forward proxy that decrypts TLS traffic), follow the steps below:

   > Note On-the-fly TLS certificates that are generated by the MITM proxy must use the subjectAltName (SAN) extension.

   a. Export your proxy server's CA certificate in PEM format.
      - When using your own ca.pem file be sure to only include the certificate or certificate chain of the MITM proxy. A custom ca.pem file cannot exceed 100kb.
   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?

- If this is a new Edge installation, install the Edge site.
- If you use a MITM proxy and the **ca.pem** has changed or was not included in the initial Edge installation, you must reinstall your Edge site.
- If you want to update the forward proxy afterwards, you can use the update script.

Enable or disable classification on an Edge site

Classification on an Edge site is part of a functionality called Profiling and Classification via Edge, which combines both processes into one. You must set up both Profiling and Classification to use this functionality. For more information, go to About profiling and classification via Edge.

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

> Note Enabling or disabling classification can take a few minutes before the changes are in effect.

Enable classification

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

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

sudo /usr/local/bin/kubectl  -n collibra-edge exec -it
${POD_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:

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

```
kubectl  -n collibra-edge exec -it ${POD_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:

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

sudo /usr/local/bin/kubectl  -n collibra-edge exec -it
${POD_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 between disabling classification and enabling classification is that the last argument is false instead of true.

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

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

kubectl  -n collibra-edge exec -it ${POD_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 between disabling classification and enabling classification is that the last argument is false instead of true.

Successful execution of either command returns the following output:

```
INFO[0000] Starting configmap/secret informers
INFO[0000] Configmap/secret informer synced
```

> Note You do not need to restart Edge when you have enabled or disabled classification.

What's next

If you enable classification, the next step is to enable profiling for Edge.

Reinstall an Edge site

You always reinstall an Edge site by restoring a backup of that Edge site. Reinstallation may be necessary to resolve an issue or to upgrade the software included in the Edge site installer.

> Warning If you have any existing connections, we highly recommend backing up your site and reinstalling the site from the backup. If you do not use a backup, you will need to manually re-enter passwords, encrypted text parameters, and any file parameters in each connection to restore full functionality.

> Note This process is certified for restoring an Edge site to the Collibra environment on which the site was originally created, for example, restoring Development to Development or Production to Production. The process is not certified or tested for promoting an Edge site migration from one environment to another, for example, from Development to Production. These types of migrations require the reinstallation of the Edge application each time the migration is promoted.

1. Back up your current Edge site.
   On the server that runs your Edge site, run 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.

   > Important If the Edge command is not available, you will need to download the Edge tool and make it available in `/usr/local/bin/edge`

2. If you are reusing the same server as your old Edge site:
   a. Use the Edge tool command to uninstall the old installation.
      Run the following Edge command from any location on the server Edge is installed on: `uninstall-edge.sh`
   b. Recreate the Linux disk mount for the `/var/lib/rancher/k3s` directory.

```
mkdir -p /var/lib/rancher/k3s
mkfs.xfs /dev/<block-device-name>
mount /dev/<block-device-name> /var/lib/rancher/k3s
echo '/dev/<block-device-name> /var/lib/rancher/k3s xfs
defaults 0 0' >> /etc/fstab
```

> Note This is the default installation path. If it is not created as a separate mount point after following the steps above, the installation will use 50 GB of disk space from either **/var**, or if not present, the root level of the drive.

3. Redownload the installer.
   a. Go to the Edge site page in your Edge enviornment.
   b. Click **Actions**.
   c. Click **Redownload Installer**.
   d. Click **Download**.
   e. Save the new installer to your server where the old installer was saved.

> Note This is a new installer for your Edge site. The previous installer will no longer work.

4. Extract the installer.

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

> Note
> - Keep the installer or the contents of the extracted installer in a secure location on your server. These contents contain various tools that you may need later, for example to troubleshoot issues.
> - If you want to run a script or executable file from the extracted directory, ensure that the directory is not mounted as `noexec`. If a directory is mounted as `noexec`, scripts and executable files will be prevented from being run within the directory.

5. If you use a custom setup, such as **proxy.properties** and **ca.pem** for forward proxies or classification, ensure that it is available or included as it was in the previous setup.
6. Reinstall using the new installer with the backup option:
   - Without classification: `./install-master.sh --storage-path <storagepath> -b backup.yaml`
   - With classification: `./install-master.sh --storage-path <storagepath> -b backup.yaml --set collibra_edge.collibra.classification.enabled=true`

> Warning Do not exclude `-b backup.yaml` from this command. If you exclude `-b bckup.yaml` from the command, your Edge site will be reinstalled without your backup and previous configurations, such as passwords, encrypted text parameters, and any file parameters in each connection. Additionally, you will not be able to use that backup in any future reinstallations.

1. Back up your current Edge site.
   On the server from which you manage your EKS cluster, run 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.

> Important If the Edge command is not available, you will need to download the Edge tool and make it available in `/usr/local/bin/edge`

2. Redownload the installer and save it on your Linux server that has kubectl access to the k8s cluster.
   a. Go to the Edge site page in your Edge environment.
   b. Click **Actions**.
   c. Click **Redownload Installer**.
   d. Click **Download**.
   e. Save the new installer to your server where the old installer was saved.

> Note This is a new installer for your Edge site. The previous installer no longer works.

3. Extract the installer.

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

> Note
> - Keep the installer or the contents of the extracted installer in a secure location on your server. These contents contain various tools that you may need later, for example to troubleshoot issues.
> - If you want to run a script or executable file from the extracted directory, ensure that the directory is not mounted as `noexec`. If a directory is mounted as `noexec`, scripts and executable files will be prevented from being run within the directory.

4. Use the Edge tool command to uninstall the old installation.

   Follow the path inside the extracted installer and run the uninstall command.
   Example: `<extracted installer>/resources/installer-job/tools/uninstall-edge-on-managed-k8s.sh`
5. If you use a custom setup, such as **proxy.properties** and **ca.pem** for forward proxies or classification, ensure that it is available or included as it was in the previous setup.
6. Reinstall using the new installer and backup:
   - Without classification: `./run-installer-job.sh properties.yaml -b backup.yaml`
   - With classification: `-b backup.yaml --set collibra_edge.collibra.classification.enabled=true`

     > Warning Do not exclude `-b backup.yaml` from this command. If you exclude `-b bckup.yaml` from the command, your Edge site will be reinstalled without your backup and previous configurations, such as passwords, encrypted text parameters, and any file parameters in each connection. Additionally, you will not be able to use that backup in any future reinstallations.

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.

   > Note 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. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   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.

Upgrading an Edge site

Edge site upgrades occur on a quarterly basis for major releases, which include new features and enhancements, and on an as-needed weekly basis for minor releases, which include security and minor bug fixes.

You can configure your Edge sites to either upgrade automatically whenever a new version is released, or upgrade manually in order to control when and to which version your sites are upgraded.

Edge site upgrade methods

There are two ways to upgrade your Edge site:

- Automatic: your Edge site automatically upgrades when a new version is available.
- Manual: your Edge site alerts you when a new version is available, and you can review the Software bill of materials and perform security scans before completing the upgrade. If an upgrade is mandatory, your Edge site will be in read-only mode until you upgrade the site. A mandatory upgrade is required within 3 months of the Collibra Data Intelligence Cloud quarterly release.

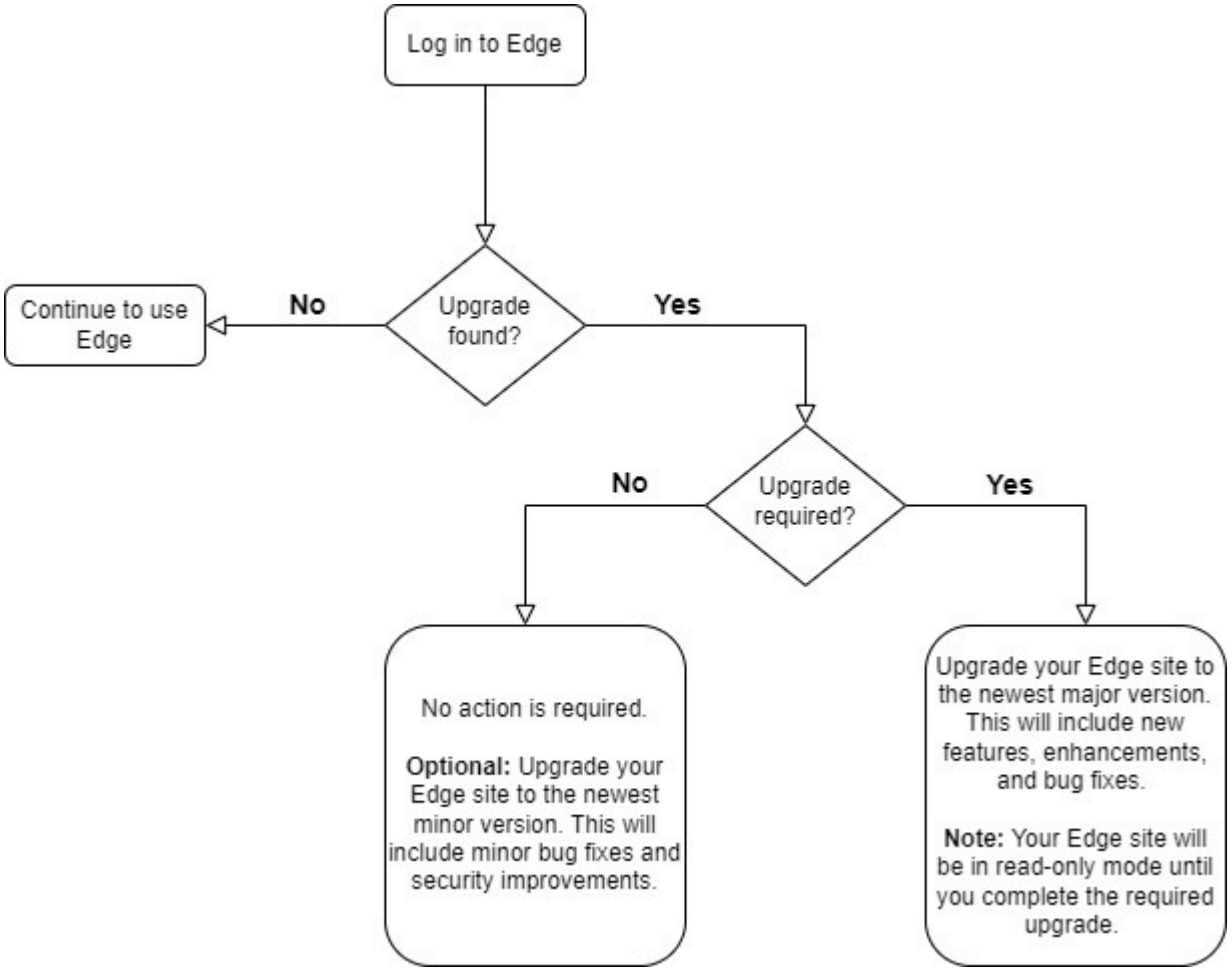
Automatic upgrade

The Automatic mode is the default upgrade mode for Edge sites. This means that when a new Edge site version is released, you do not need to initiate the upgrade, as it will automatically be applied to your Edge site. You will only need to take action if the new version includes new software requirements or your installer becomes out-of-date. This information will be provided to you through release notes, and you can review the compatibility table to see what versions of the Edge site may require a reinstall for k3s or upgrade of EKS. For how to enable automatic upgrade mode for your Edge sites that use the manual upgrade mode, go to Enable Automatic upgrade for Edge sites.

> Note If you created an Edge site prior to the 2023.08 release, your Edge sites have Automatic upgrade enabled.

Manual upgrade

The Manual upgrade mode allows you to choose when, and to which version, you want to upgrade your Edge sites. Whenever an Edge site version becomes available, a banner is displayed at the top of the page with an **Upgrade Now** button. After you select the version to which you want to upgrade your site, you can download the Software Bill of Materials to review and scan before beginning the upgrade.



Upgrade types

There are two types of upgrades:

- Optional: minor updates which occur between quarterly releases, and include security and minor bug fixes. You can choose to wait or upgrade your Edge site.
- Mandatory: major releases which occur on a quarterly basis, and include new features and enhancements. A mandatory upgrade is required within 3 months of the Collibra Data Intelligence Cloud quarterly release. When a mandatory upgrade becomes available and you have manual upgrades enabled for an Edge site, your site will be in read-only mode until you upgrade the site to the mandatory version. For more information, go to the Compatibility between Edge sites and Collibra Data Intelligence Cloud. This is to ensure that all Edge features are appropriately updated and compatible with Collibra.

> Important You cannot start or configure any connections or capabilities if your Edge site is in read-only mode. You must perform the mandatory upgrade or wait until an upgrade has been completed to resume full access to Edge.

Your Edge site lists whether an upgrade is optional or mandatory. For how to enable manual upgrade mode for your Edge sites that use the automatic upgrade mode, go to Enable Manual upgrade for Edge sites.

Software Bill of Materials

> Note
> Security scan reports are only be accepted for supported Edge site versions. This is because security fixes are not applied to old, out-dated versions of Edge. For example, from November 19 2023 to February 24, 2024, security scans are only accepted for Edge site version 2023.11 and subsequent weekly updates (2023.11.x). For information on which Edge versions are supported with the latest release, go to the Compatibility between Edge sites and Collibra Data Intelligence Cloud.

You can download a Software Bill of Materials (SBOM) to review the contents of an Edge site version. A SBOM is a list of images included in an Edge site version that your security team may want to perform security scans and evaluations on before your Edge site is upgraded to a new version.

For more information about Edge security and scanning, go to Security scanning.

You can retrieve the SBOM through one of the following methods:

- A REST API.
  - Location: *<hostname>/edge/api/rest/v2/releaseinfo/<edge version>/bom*
- Selecting an upgrade version in the Edge platform.
  - When you select a version to upgrade your Edge site to, you are provided with a link to download the SBOM, as shown in Enable Manual upgrade mode for Edge sites.

The SBOM is downloaded as a zip file containing JSON files. These are in SPDX and CYCLONEDX formats which you can use as input files for your security scanning tools.

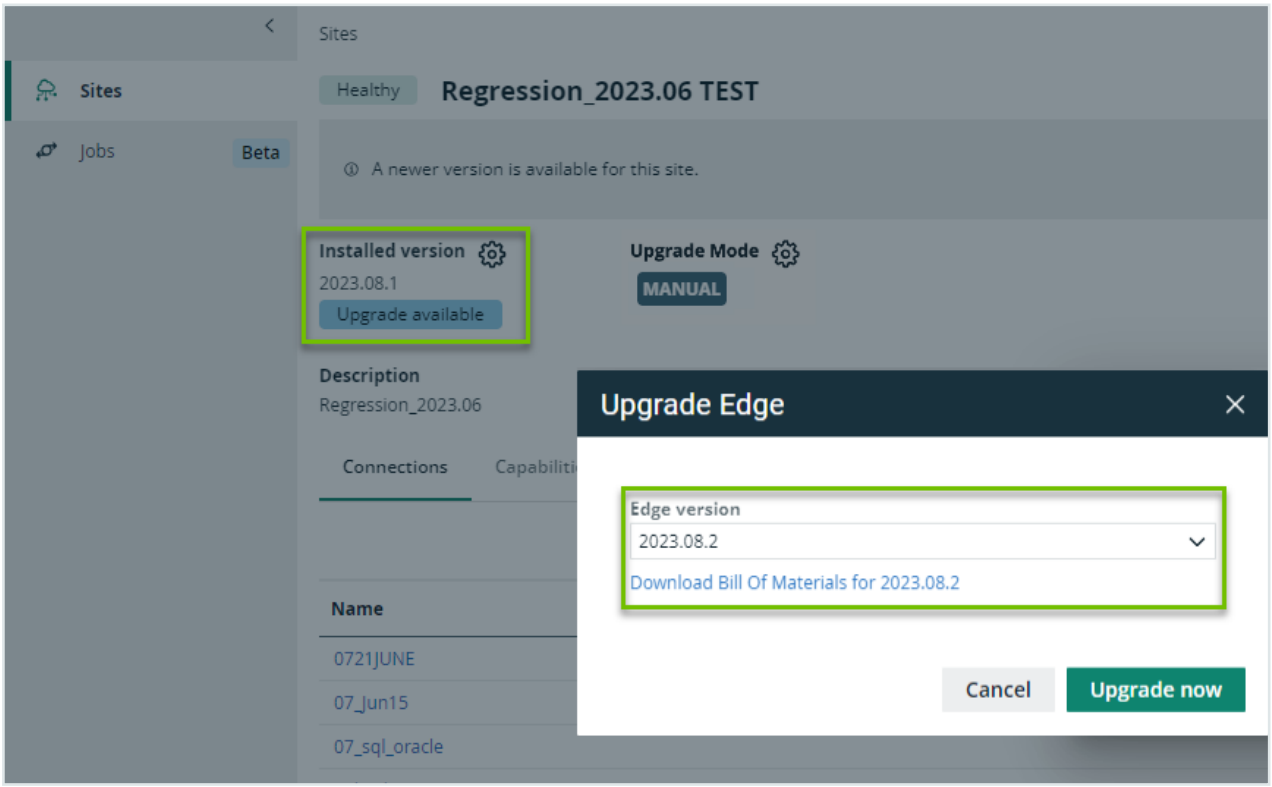
What's next?

- Learn how to enable Manual or Automatic upgrade mode for your Edge sites.
- Learn how to perform your own security scans before upgrading to a new version of Edge if you set up a private docker registry.

How to manually upgrade your Edge site

You can either upgrade to the newest version by clicking **Upgrade now** on the Edge site page or manually select an available version by following the steps below:

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
2. Click ⚙ next to **Installed version**.
   » The **Upgrade Edge Site** dialog box appears.
3. Open the drop-down list to review available Edge site versions.
4. Select the version from the drop-down list you want to review or upgrade to.
5. Optional: Click the hyperlink to download the Software Bill of Materials.
6. Click **Upgrade now**.



What's next?

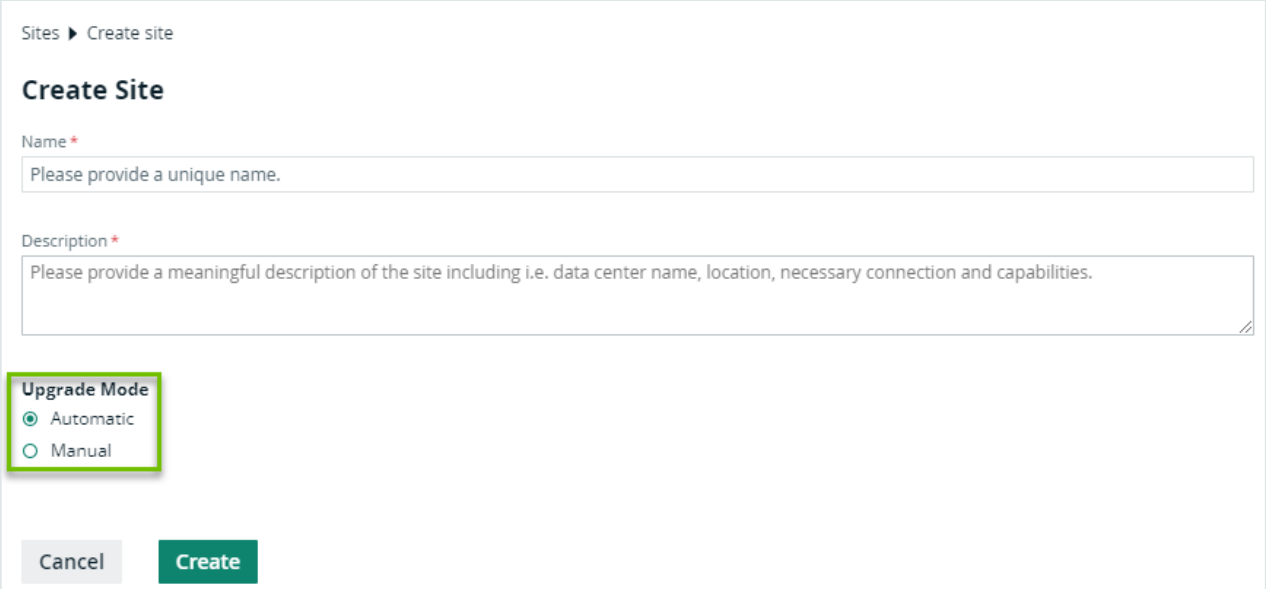
- Review the Compatibility between Edge sites and Collibra Data Intelligence Cloud to know when you need to either reinstall your Edge site for an upgraded version of k3 or upgrade to the latest version of EKS.
- Optionally, set up a private docker registry to easily incorporate Edge into your existing security procedures and perform your own security scans before upgrading to a new version of your Edge site.

Enable Automatic upgrade for Edge sites

You can enable automatic upgrade for new and existing Edge sites that use the manual upgrade mode. This mode automatically upgrades your Edge site whenever a new version has been detected.

New Edge sites

Automatic upgrades are enabled by default for all new Edge sites. When you are creating a new Edge site, ensure Automatic is selected before you click the **Create** button.

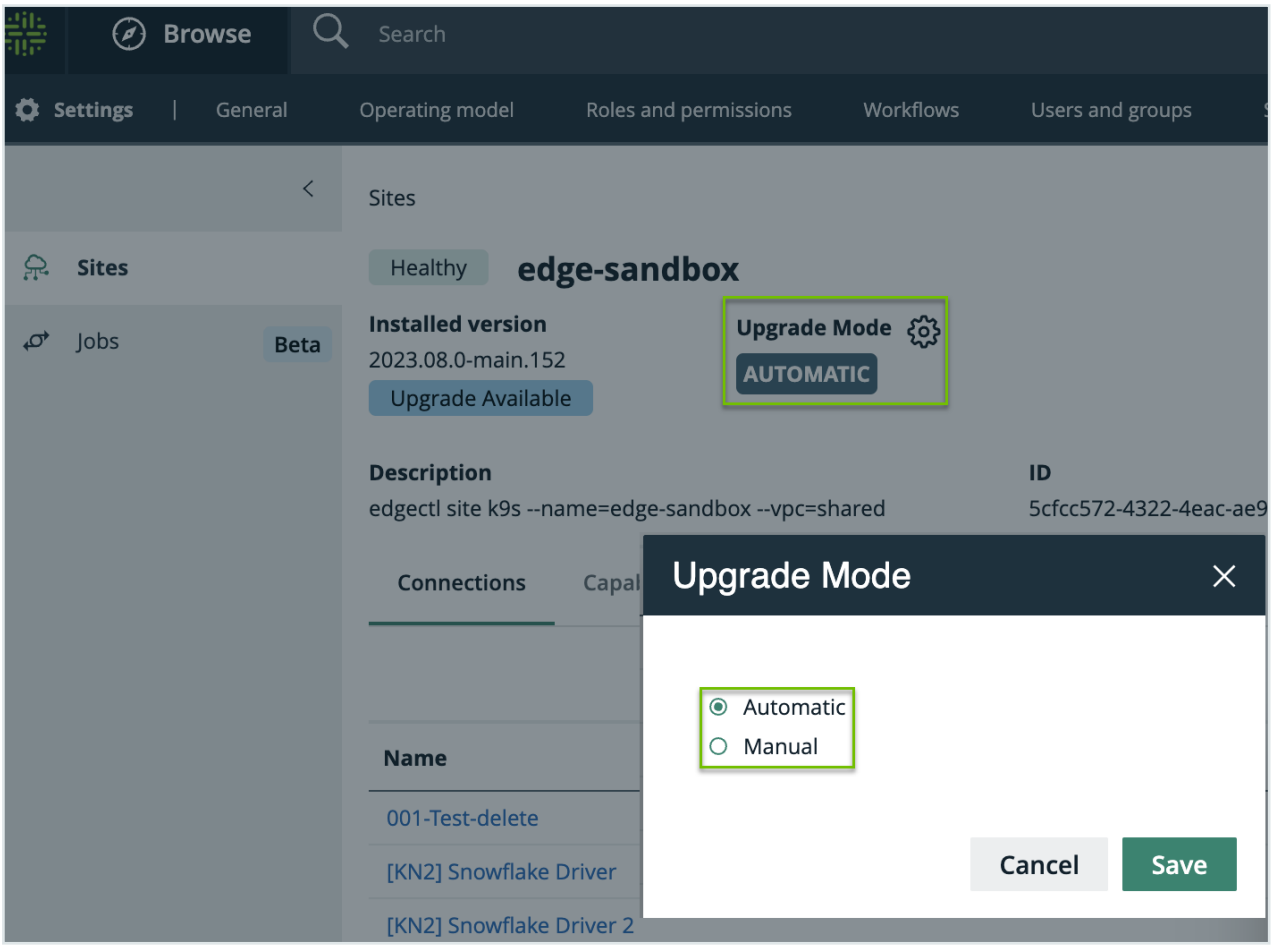


Existing Edge sites

You can change the upgrade mode of existing Edge sites to automatic by following the steps below:

1. Open an Edge site.
    a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
        » The Collibra settings page opens.
    b. Click **Edge**.
        » The Edge sites overview opens.

c. Click the name of an Edge site in the Edge site overview.
   » The Edge site page opens.
2. Click ⚙ next to **Upgrade Mode**.
   » The **Upgrade Mode** dialog box appears.
3. Select **Automatic**.
4. Click **Save**.



What's next?

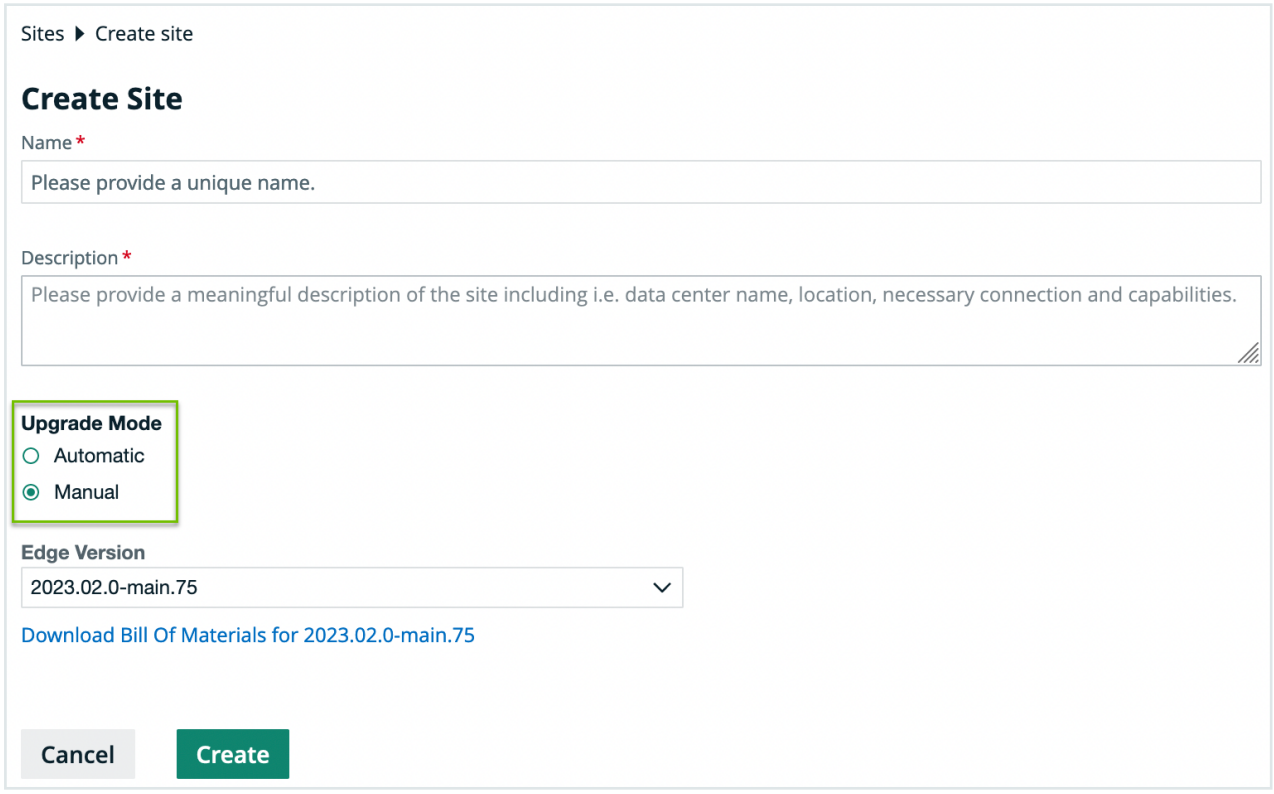
Review the compatibility table to know when you need to either reinstall your Edge site for an upgraded version of k3 or upgrade to the latest version of EKS.

Enable Manual upgrade for Edge sites

You can enable manual upgrade for new Edge sites or change existing sites to manual upgrade mode. This mode allows you to control when, and to which version, you upgrade your Edge sites to. You can also review the Software Bill of Materials, which outlines what is included in the upgrade, before upgrading your Edge sites.

New Edge sites

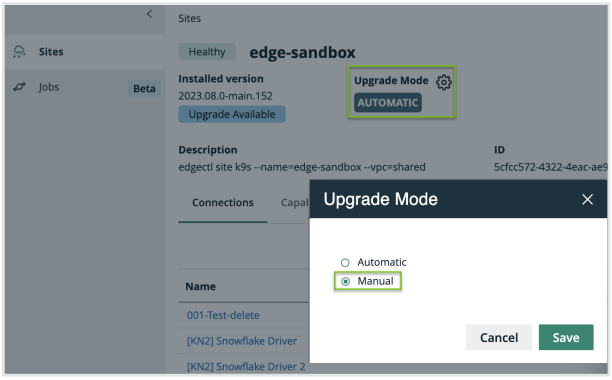
When creating a new Edge site, select **Manual** under the **Upgrade Mode** and click **Create**.



Existing Edge sites

You can change the upgrade method to manual for existing Edge sites by following the steps below:

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
2. Click ⚙ next to **Upgrade Mode**.
   » The **Upgrade Mode** dialog box appears.
3. Select **Manual**.
4. Click **Save**.



Your Edge site will no longer automatically upgrade to the newest available version.

What's next?

- Learn how to manually upgrade your Edge site when a new version becomes available.
- Review the Compatibility between Edge sites and Collibra Data Intelligence Cloud to know when you need to either reinstall your Edge site for an upgraded version of k3 or upgrade to the latest version of EKS.
- Optionally, set up a private docker registry to easily incorporate Edge into your existing security procedures and perform your own security scans before upgrading to a new version of Edge site.

Edge connections

Edge connections define how an Edge capability communicates with a data source in order to collect and send metadata to Collibra.

Available Edge connections

To collect metadata from a data source and add it into Collibra via Edge, Edge needs to be able to communicate with the data source. This is managed via an Edge connection. Once an Edge connection is established, the connection can be used by some of the Collibra capabilities to, for example, register the metadata or collect sample data.

> Example
> You want to add the metadata of a Snowflake data source in Collibra and create technical lineage for it. By defining a JDBC connection between Edge and your Snowflake data source, you establish a secure line of communication between Collibra and your data source. This line of communication is then used to register the metadata and create technical lineage for it in your Collibra platform.

Multiple connection types are available. The connection type that you need to use depends on what you want to achieve. The following table contains the available Edge connection types and the associated capabilities.

Connection type	Description
Azure	Used for the integration Azure Data Lake Storage (ADLS) data sources. Show associated capabilities • ADLS synchronization
AWS (Amazon Web Services)	Used for the integration and protection of Amazon S3 data sources. Show associated capabilities • S3 synchronization • Protect for AWS Lake Formation
Databricks	Used for the integration of Databricks Unity Catalog. Show associated capability • Data Unity Catalog

Connection type	Description
GCP (Google Cloud Platform)	Used for the integration and protection of Google Cloud Storage and Dataplex data sources. Show associated capabilities • GCS synchronization • Protect for Google BigQuery
Informatica Intelligent Cloud Services	Used to connect to Informatica Intelligent Cloud Services. Show associated capability • Technical lineage for Informatica Intelligent Cloud Services (IICS)
JDBC	Used to connect to JDBC data sources, for example, Snowflake, Salesforce, and PostgreSQL. Show associated capabilities • Catalog JDBC ingestion • Catalog JDBC Sampling • JDBC Profiling • Catalog Data Classification(Beta) • Protect for Snowflake • Technical lineage capabilities for data sources that use the JDBC connection ◦ Technical Lineage for Azure ◦ Technical Lineage for BigQuery ◦ Technical Lineage for DataStage ◦ Technical Lineage for Db2 ◦ Technical Lineage for Greenplum ◦ Technical Lineage for SAP HANA ◦ Technical Lineage for Hive ◦ Technical Lineage for Informatica PowerCenter ◦ Technical Lineage for MySQL ◦ Technical Lineage for SQL Server ◦ Technical Lineage for Netezza ◦ Technical Lineage for Oracle ◦ Technical Lineage for PostgreSQL ◦ Technical Lineage for Amazon Redshift ◦ Technical Lineage for Snowflake ◦ Technical Lineage for Spark SQL ◦ Technical Lineage for SQL Server Integration Services (SSIS) ◦ Technical Lineage for Sybase ◦ Technical Lineage for Teradata

Connection type	Description
Matillion	Used to connect to Matillion. Show associated capability • Technical Lineage for Matillion
MicroStrategy	Used to connect to MicroStrategy. Show associated capabilities • Technical Lineage for MicroStrategy
Power BI	Used to connect to Power BI. Show associated capabilities • Technical Lineage for Power BI
Shared Storage connection	Used to access files from a shared folder. Show associated capability • Technical Lineage for SqlDirectory • Technical Lineage for Custom Technical Lineage
Tableau	Used to connect to Tableau Server or Tableau Online. Show associated capability • Technical Lineage for Tableau

Edit a connection

You can update the details of a data source by editing the connection. This topic will discuss how you can generally edit a connection. For more specific information, review the requirements for your data source, such as Technical lineage and Sample data.

> Note Refer to the JDBC connections documentation for how to edit JDBC connections.

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.

> Note It is possible there are extra requirements for your specific data source. Review the requirements and permissions of your data source before making any changes.

Steps

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
2. Locate and click the connection you want to edit.
3. At the bottom of the page, click **Edit**.
4. Edit the connection information.
5. Click **Save**.

Delete a connection

You can delete a connection from an Edge site to a data source if you no longer need it. This topic will discuss how you can generally delete a connection. For more specific information, review the requirements for your data source, such as Technical lineage and Sample data.

> Note Refer to the JDBC connections documentation for how to edit JDBC connections.

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.

Steps

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
2. Locate and select the connection you want to delete.
3. At the bottom of the page, click **Delete**.
   » The **Delete confirmation** dialog box appears.
4. Click **Delete Connection**.

Warning When you delete a JDBC connection that Collibra Data Quality & Observability uses from an Edge site, all associated Collibra DQ metadata for that connection will also be deleted from Collibra. You cannot undo this action.

JDBC connections

JDBC connections define how an Edge capability accesses a data source.

To create a connection to your data source, you need to select a connection type, 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.

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.

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.

Steps

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. In the tab pane, click **Edge**.
      » The **Sites** tab opens and shows a table with an overview of the Edge sites.
   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 **Create Connection**.
   » The **Connection settings** page appears.

3. Enter the required information.

Field	Description
Connection settings	This section contains the settings to connect to your data source.
Name	The name of the JDBC connection.
Description	The description of the JDBC connection. This field is also visible when you register content.
Connection provider	The connection provider, which determines the available connection parameters.
Connection parameters	This section contains general settings to connect to your data source.
Username	The username to access your data source.
Password	The corresponding password to access the data source.
Driver class name	The driver class name of the connection.
Driver jar	The JAR file contain the JDBC driver. Click **Upload** to upload a JAR file.
Additional classpath files	An optional list of additional classpath files.
Connection string	The JDBC connection string. Warning Some connection properties can be added to the URL as name-value pairs separated by semicolons. However, most properties in the URL are ignored. Therefore, we recommend you not to use this mechanism unless we explicitly ask you to. We recommend you to specify all connection properties in the **Connection properties** section.
Connection properties	This section contains the connection properties.

4. Click **Create**.

What's next?

You can now add a capability to ingest or profile a data source.

Customizing the database name for database-less data sources

When you create a JDBC connection for a database-less data source, such as Hive, MongoDB, or Teradata, the default database name is set to `CData`. When you register the data source via Edge, `CData` is listed in the **Database name** drop-down menu on the **Add Database** page.

You can use the `CustomizedDefaultCatalogName` connection property to customize the database name when you connect to your data source. Collibra then uses the value of the `CustomizedDefaultCatalogName` connection property as the database name when you register the data source via Edge. To use this property, you must use a Collibra-provided driver that is newer than version 23.0.8409.

If you customized the database name and want to create technical lineage for the database-less data sources, ensure that you take the following actions:

- If you use technical lineage via Edge, add the customized database name in the **External Database Name** field when you add the technical lineage capability for the data source.
- If you use the lineage harvester, specify the `externalDbName` property in the lineage harvester configuration file.

> Important Don't update the database name after you have registered the data source.
> If you add or change the `CustomizedDefaultCatalogName` connection property after a database was registered, we treat the database as a new one, and you must register the data source again with the new database name. Renaming a database while keeping the existing registered assets is not possible.

> Note If you add the `CustomizedDefaultCatalogName` property to the JDBC connection after the database was listed for the first time in the Database name drop-down menu on the **Add Database** page, both the new database name and `CData` will appear in the Database name drop-down menu. Make sure to select the new database name when you register the data source.

This property is available for the following database-less data sources. For details about specifying the `CustomizedDefaultCatalogName` connection property for each data source, go Overview of Catalog connectors.

- Amazon DynamoDB
- Apache Cassandra
- Apache HBase
- Apache Hive
- Apache Spark SQL
- Avro
- Azure Cosmos DB
- Azure Table Storage
- CSV
- Elasticsearch
- Excel
- Google Sheets
- Greenplum
- IBM Cloudant
- IBM Db2
- Impala
- JSON
- MarkLogic
- MongoDB
- Parquet
- Salesforce
- SAS Data Sets
- Splunk
- Teradata
- XML

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.

Prerequisites

- If required, you have created a JDBC connection.
- 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.

Steps

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. In the tab pane, click **Edge**.
      » The **Sites** tab opens and shows a table with an overview of the Edge sites.
   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 **Edit**.
   » The fields become editable.
4. Enter the required information.

Field	Description
Connection settings	This section contains the settings to connect to your data source.
Name	The name of the JDBC connection.
Description	The description of the JDBC connection. This field is also visible when you register content.
Connection provider	The connection provider, which determines the available connection parameters.
Connection parameters	This section contains general settings to connect to your data source.

Field	Description
Username	The username to access your data source.
Password	The corresponding password to access the data source.
Driver class name	The driver class name of the connection.
Driver jar	The JAR file contain the JDBC driver. Click **Upload** to upload a JAR file.
Additional classpath files	An optional list of additional classpath files.
Connection string	The JDBC connection string. Warning Some connection properties can be added to the URL as name-value pairs separated by semicolons. However, most properties in the URL are ignored. Therefore, we recommend you not to use this mechanism unless we explicitly ask you to. We recommend you to specify all connection properties in the **Connection properties** section.
Connection properties	This section contains the connection properties.

5. Click **Save**.
6. If required, you can now test the connection.
   a. At the bottom of the page, click **Test connection**.
      » The **Connection test** dialog box appears.
   b. When the test is finished,click **OK**.

> Tip If the connection failed, you can click **View Stacktrace** to identify the problem.

Delete a JDBC connection

You can delete a 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.

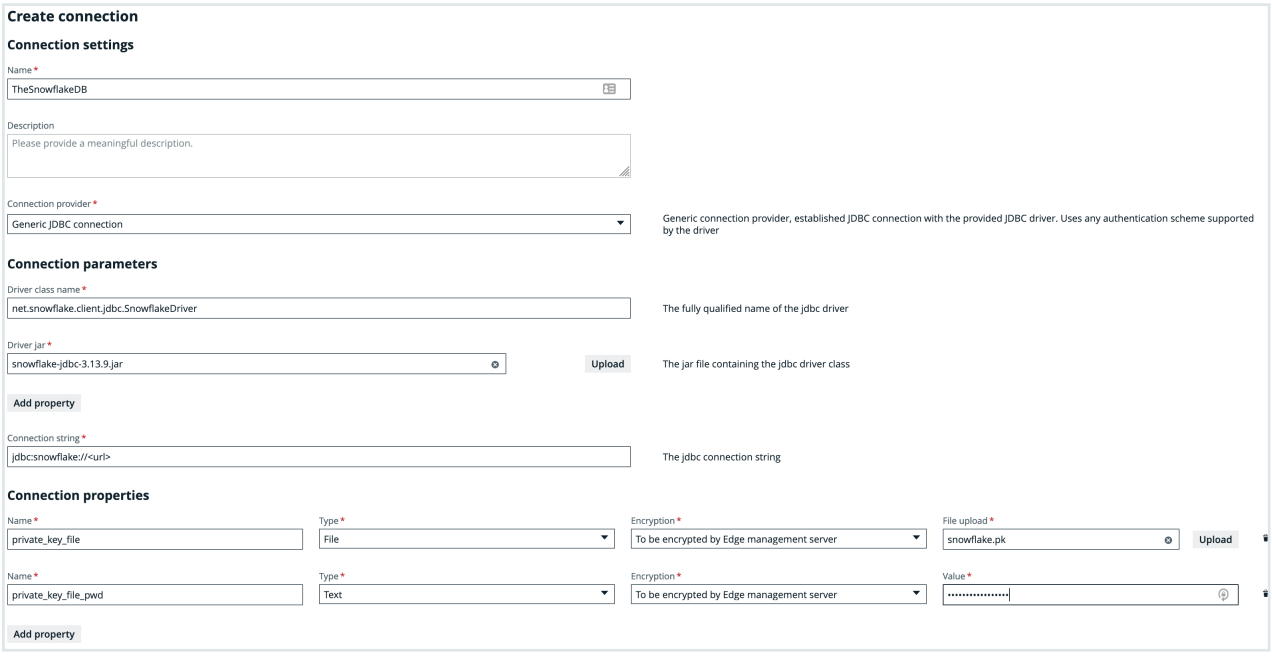
Steps

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
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.



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, like Sampling or S3 synchronization, is an application that can run on an Edge site. It can access a data source to extract and process data as needed. This data can be stored in an encrypted cache to improve the security of your data and platform. An Edge capability for a specific data source runs as a job and 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.

Capability template	Description
ADLS synchronization	Used to connect to Azure Data Lake Storage (ADLS)
Catalog Data Classification(Beta)	Used to classify data from a registered JDBC data source in the Edge site. This capability can't be added to an Edge site that uses a MITM proxy.
Catalog JDBC ingestion	Used to register a data source and synchronize schemas from a data source via a JDBC connection. This capability can't be added to an Edge site that uses a MITM proxy.

Capability template	Description
Catalog JDBC Sampling	Used to collect and cache sample data from a data source in the Edge site via a JDBC connection. Ensure that you meet the additional Catalog JDBC Sampling hardware requirements, in addition to the Edge site requirements. This capability can't be added to an Edge site that uses a MITM proxy.
Collibra Protect for AWS Lake Formation	Used to set up Protect for AWS Lake Formation. This capability can't be added to an Edge site that uses a MITM proxy.
Collibra Protect for Google BigQuery	Used to set up Protect for BigQuery. This capability can't be added to an Edge site that uses a MITM proxy.
Collibra Protect for Snowflake	Used to set up Protect for Snowflake. This capability can't be added to an Edge site that uses a MITM proxy.
DQ Connector	Used to ingest Collibra Data Quality & Observability user-defined rules, metrics, and dimensions into Collibra Data Catalog. This capability can't be added to an Edge site that uses a MITM proxy.
GCS synchronization	Used to connect to Google Cloud Storage.
JDBC Profiling	Used to profile and classify data from a registered data source. This capability can't be added to an Edge site that uses a MITM proxy.
S3 synchronization	Used to connect to Amazon S3.
Databricks Unity Catalog synchronization	Used to connect to Databricks Unity Catalog.

Capability template	Description
Technical lineage capabilities	Used to create technical lineage for different data sources. For details, go to: Add a technical lineage capability to an Edge site. Ensure that you meet the additional Technical Lineage minimum network requirements, in addition to the Edge site requirements. Technical Lineage requirements... Firewall rules so that the lineage harvester can connect to: • The host names of all data sources in your lineage harvesterconfiguration file. • All Collibra Data Lineage service instances in your geographic location: ◦ 15.222.200.199 (techlin-aws-ca.collibra.com) ◦ 18.198.89.106 (techlin-aws-eu.collibra.com) ◦ 13.228.38.245 (techlin-aws-sg.collibra.com) ◦ 54.242.194.190 (techlin-aws-us.collibra.com) ◦ 51.105.241.132 (techlin-azure-eu.collibra.com) ◦ 20.102.44.39 (techlin-azure-us.collibra.com) ◦ 35.197.182.41 (techlin-gcp-au.collibra.com) ◦ 34.152.20.240 (techlin-gcp-ca.collibra.com) ◦ 35.205.146.124 (techlin-gcp-eu.collibra.com) ◦ 34.87.122.60 (techlin-gcp-sg.collibra.com) ◦ 35.234.130.150 (techlin-gcp-uk.collibra.com) ◦ 34.73.33.120 (techlin-gcp-us.collibra.com) Note Edge connects to different Collibra Data Lineage service instances based on your geographic location and cloud provider. If your location or cloud provider changes, Edge rescans all your data sources. You have to allow all Collibra Data Lineage service instances in your geographic location. In addition, we highly recommend that you always allow the techlin-aws-us instance as a backup, in case Edge cannot connect to other Collibra Data Lineage service instances. You can use a man-in-the-middle (MITM) proxy between Edge and the Collibra Data Lineage service instances. For details on which data sources support the use of proxies, go to Create a technical lineage via Edge, select your data source, and see our test results in the Connect to a proxy server section.

> 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:

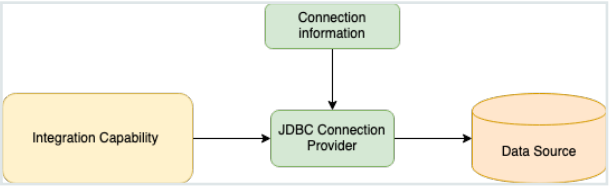
File	Description
A manifest file (YAML)	This file contains the capability metadata and input parameter requirements.
A workflow file (YAML)	This file defines the workflow and binds the parameters to capability containers.
Docker images	One or more Docker images that implement the business logic.

> Note Each type of capability has its own required custom properties. These properties appear after you select a capability template from the dropdown menu.

About Edge capabilities connecting to data sources

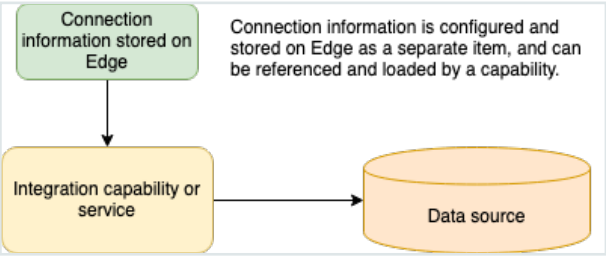
A connection on an Edge site identifies a unique system, whether it's a database, a file share or a REST service.

For JDBC (Java Database Connectivity), you can connect directly from the Edge user interface. When you create a JDBC connection, you will enter your login credentials, which will then be stored for authentication. This means that you will not need to enter these credentials again for any capability that uses this JDBC connection.



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.

Below is an example of a 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 have 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 by 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 that has 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.
- Ensure the max cardinality of the asset attributes is at least 1.

Steps

> Tip For more information about all fields in the capability, go to the online version of the documentation.

More information

ADLS integration

Catalog Data Classification (Beta)

Catalog JDBC ingestion

JDBC Profiling

Catalog JDBC Sampling

S3 synchronization

GCS synchronization

Databricks Unity Catalog integration

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 that has 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

> Tip For information about the various capabilities, go to the online version of the documentation.

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 that has 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. On the main toolbar, click , and then click **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
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 dashboard

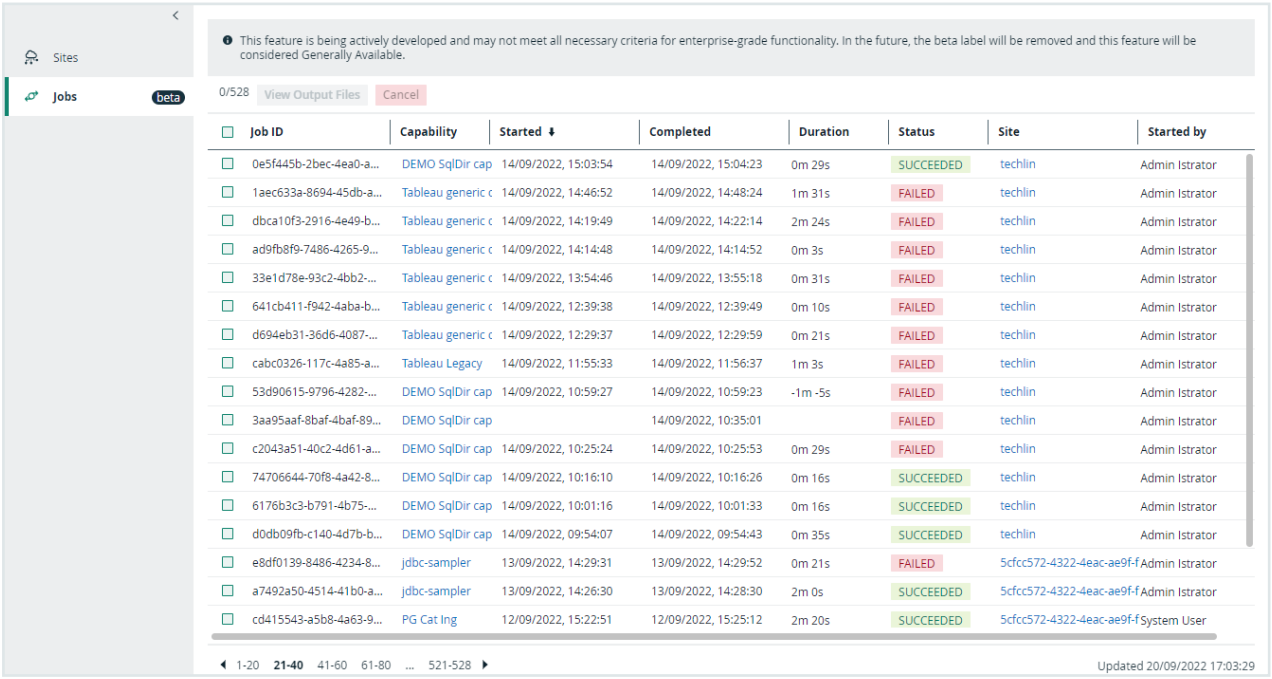
The Edge Jobs dashboard gives you an overview of all jobs that are executed by an Edge site.

When you enable the Edge Jobs feature (beta) in Collibra Console, the Edge Jobs dashboard becomes available in the Collibra Data Intelligence Cloud settings.

> **Note** Only users with the Admin role can enable this feature.

> **Important** This is a beta feature.

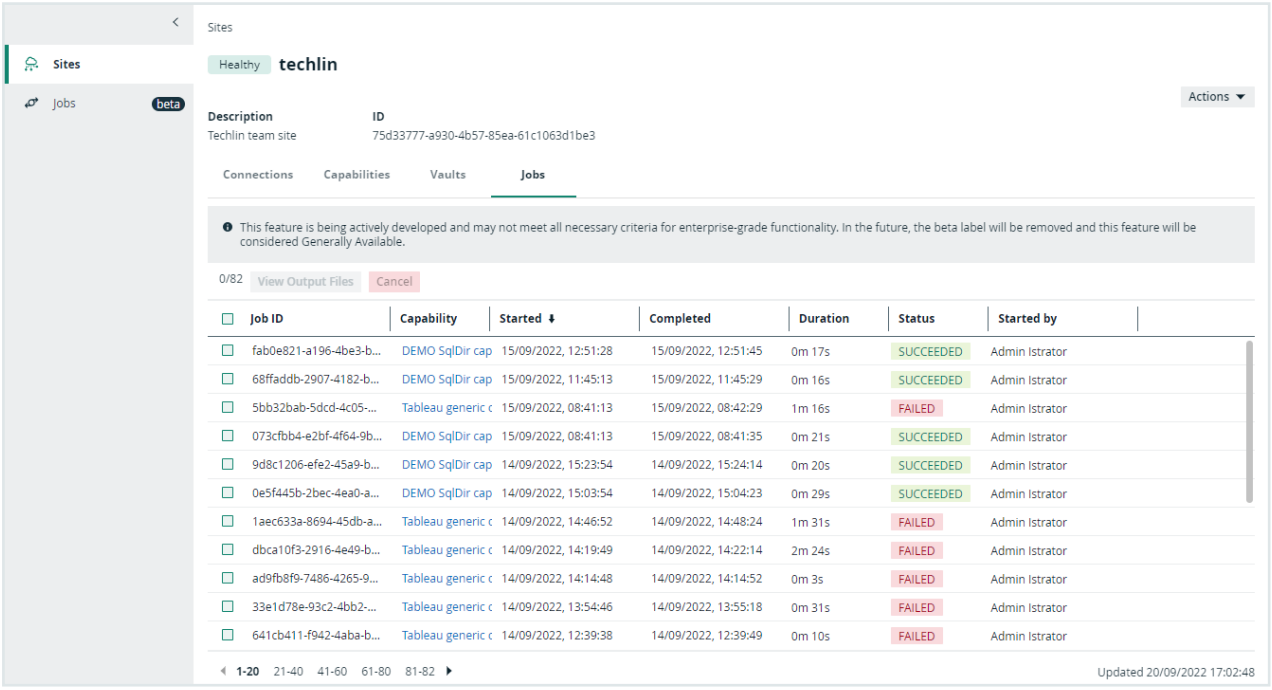
On the Edge Jobs dashboard, you find an overview of all jobs that have either been scheduled or completed in your Edge sites. Each job is a row in the table and contains basic information such as start and completion date, status, Edge site, capability and so on. You can also open the log files of a job and cancel a scheduled job from this dashboard.



View Edge site jobs

You can also view the jobs associated to a specific Edge site by going to the **Jobs** tab of that site.

1. Click **Sites**.
2. Select your site from the list.
3. Click **Jobs** in the tab menu.



Additional resources

You can also download the output file of a JDBC job from the Job dashboard. You can provide this file to our support team if a job fails.

Download job output files

You can download the output file of a JDBC job, which contains logs you can provide to support if a job has failed. Only completed jobs are available for download.

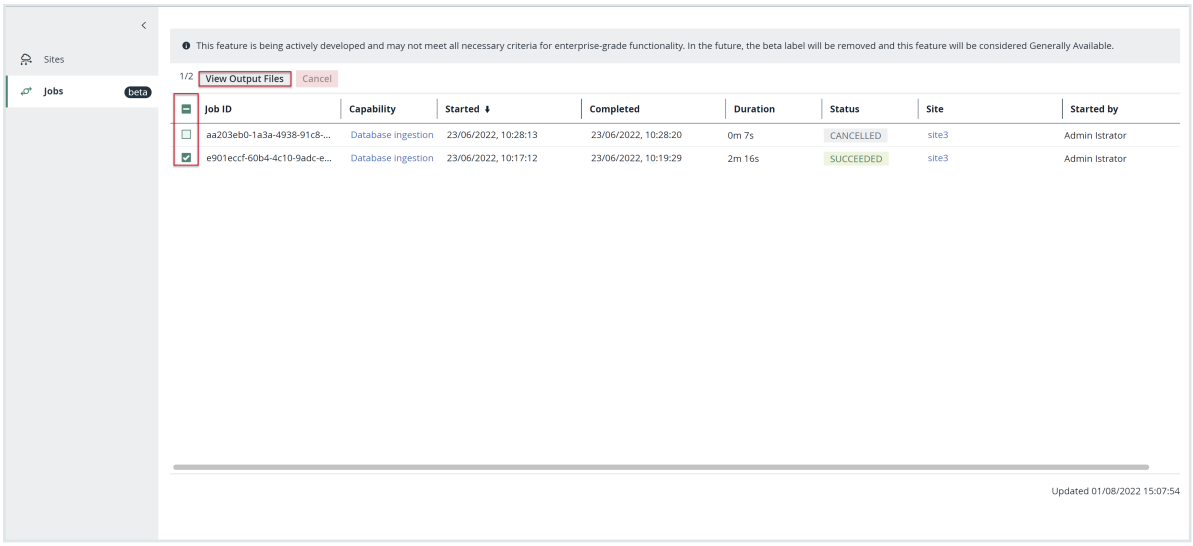
Prerequisites

- You have Edge View Log permission.
- You have jobs which have been completed.

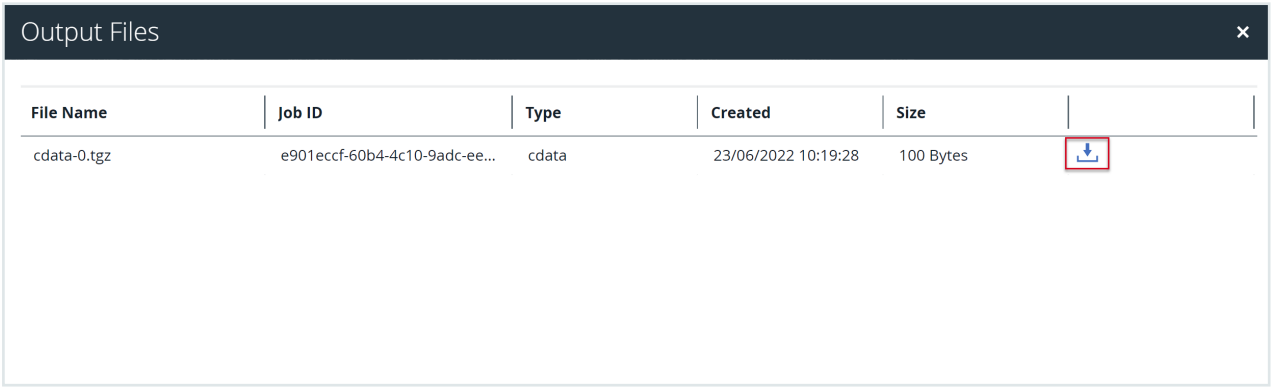
Steps

1. Open an Edge site.
   a. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. In the tab pane, click **Edge**.
      » The **Sites** tab opens and shows a table with an overview of the Edge sites.
   c. In the table, click the name of the Edge site whose status is **Healthy**.
      » The Edge site page opens.
2. Click **Jobs**.
3. Select the checkboxes next to the jobs you want to download the output file for.
4. Click **View Output Files.**
   » The View Output Files window appears.

> Tip If you select the checkbox next to a job which has been canceled or has not been completed, the **View Output Files** window is empty.



5. Click ⬇ to download the job output file.



Your downloaded job output file is now available to review from your local drive.

Cancel jobs

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

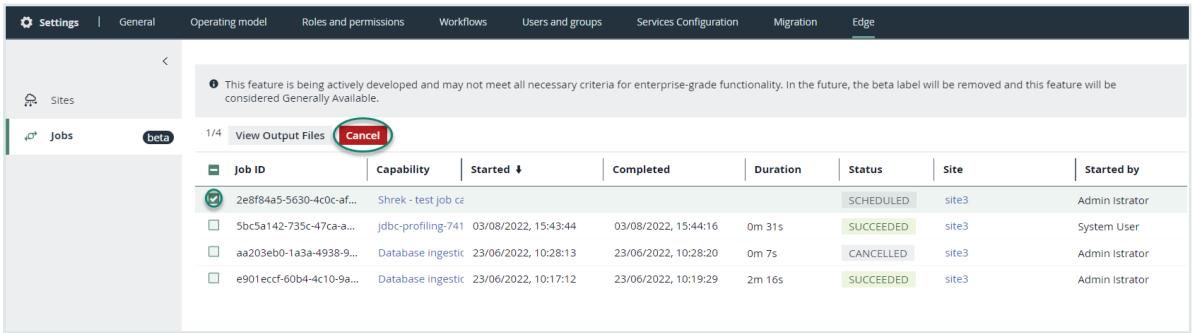
Prerequisites

- 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. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. In the tab pane, click **Edge**.
      » The **Sites** tab opens and shows a table with an overview of the Edge sites.
   c. In the table, click the name of the Edge site whose status is **Healthy**.
      » The Edge site page opens.
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. In the action toolbar, click **Cancel**.
   » The job is canceled, and the status of this job is CANCELED.

Chapter 6

Maintaining Edge sites

In this section, you will learn how you can maintain your Edge site installations, such as performing 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. This 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	`/usr/local/bin/uninstall-edge.sh` Note If you intend to reinstall Edge, you need to recreate the Linux disk mount for the directory `/var/lib/rancher/k3s`

Edge tool	Command for K3S

Create Edge diagnostics file	• Edge site is not yet installed:
	`<extracted installer directory>/resources/tools/edge-diagnostics.sh -d <file name>.tgz`
	• Edge site is up and running:
	`edge diagnostics -d <file name>.tgz`

Create an Edge site backup	`edge backup -o /<path to folder>/<backup-name>.yaml`
Set Edge storage cache ttl	`edge cachettl --ttl <value in days>`
Retrieve logs from a catalog connector	`edge catalog-connector --jobid <Edge job ID> \ --dst <path to destination>`

Update Collibra credentials	• Interactive way:
	`edge update-dgc-creds -i`
	• Explicit update:
	`edge update-dgc-creds <username> <password> <url collibra environment>`

Update forward proxy settings	`edge update-outbound-proxy --update-outbound-proxy /path/to/proxy.properties`

Get help to set up no_proxy configuration	• Edge site is not yet installed:
	`<extracted installer directory>/resources/tools/edge-get-noproxy.sh k3s`
	• Edge site is up and running:
	`edge get-noproxy k3s`

Edge tool	Command for K3S
Edge CLI	`edge download-edgecli` By default, the `edgecli` tool is downloaded into the current working directory. Once the command has been executed, you can use it as expected. For example: `./edgecli <command>` Note The Edge CLI tool is only available in the linux/amd64 binary and can only be used to configure vaults.

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 the Edge tool after installation because we don't want to enforce it in a specific location. Therefore, on your Linux machine, download the Edge tool to a folder of your choice. For example:

```
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	`<extracted installer>/resources/installer-job/tools/uninstall-edge-on-managed-k8s.sh`
Create Edge diagnostics file	• Edge site is not yet installed: `<extracted installer directory>/resources/tools/edge-diagnostics.sh -d <file name>.tgz` • Edge site is up and running: `edge diagnostics -d <file name>.tgz`
Create an Edge site backup	`edge backup -o /<path to folder>/<backup-name>.yaml`
Set Edge storage cache ttl	`edge cachettl <value in seconds>`
Retrieve logs from a catalog connector	`edge catalog-connector --jobid <Edge job ID> \ --dst <path to destination>/<file name>.txt`
Update Collibra credentials	• Interactive way: `edge update-dgc-creds -i` • Explicit update: `edge update-dgc-creds <username> <password> <url collibra environment>`
Update forward proxy settings	`edge update-outbound-proxy --update-outbound-proxy /path/to/proxy.properties`

Edge tool	Command for EKS
Get help to set up no_proxy configuration	• Edge site is not yet installed: `<extracted installer directory>/resources/tools/edge-get-noproxy.sh eks <clustername>` • Edge site is up and running: `edge get-noproxy eks <clustername>`
Edge CLI	`edge download-edgecli` By default, the `edgecli` tool is downloaded into the current working directory. Once the command has been executed, you can use it as expected. For example: `./edgecli <command>` Note The Edge CLI tool is only available in the linux/amd64 binary and can only be used to configure vaults.

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. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
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 account is created in Collibra Data Intelligence Cloud. This user always has "Edge" as the first name and the "Edge's site name" as the 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 account must have the Connect Edge to Collibra global permission.

Steps

1. Reset the password of the Edge user in Collibra by following the steps in our Set or reset a user password article.
   > Note The only non-alphanumeric characters accepted for passwords are: `!`, `$`, `%`, `&`, `(`, `)`, `*`, `+`, `,`, `-`, `.`, `/`, `:`, `;`, `<`, `=`, `>`, `?`, `@`, `[`, `]`, `^`, `_`, `{`, `|`, `}`, `~`.
   > For more information about the default password requirements, go to the Password settings.
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. Depending on your setup, do one of the following:
   - If you use a MITM proxy and the **ca.pem** has changed or was not included in the initial Edge installation, reinstall your Edge site.
   - Otherwise, 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, such as passwords and file parameters in connections, you can back up an Edge site. You can use this backup to reinstall it later, for example, when you want to reinstall an Edge site with a new installer.

The backup contains the following content:

- The public/private key of the site that is used for sending and encrypting secrets.
- The secrets that are used in connections, capabilities and vaults.

> 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, run 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.

> Important If the Edge command is not available, you will need to download the Edge tool and make it available in `/usr/local/bin/edge`

On the server from which you manage your EKS cluster, run 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.

> Important If the Edge command is not available, you will need to download the Edge tool and make it available in `/usr/local/bin/edge`

What's Next?

> Note You can only restore a backup by reinstalling the Edge site using the created backup.

Reinstall your Edge site using the backup you created.

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. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
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:

> Note If you intend to reinstall the Edge site after performing an uninstall command, you need to recreate the Linux disk mount for the directory `/var/lib/rancher/k3s`

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

Warning When you delete an Edge site, the Elastic Block Store (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. On the main toolbar, click ⠿, and then click ⚙ **Settings**.
      » The Collibra settings page opens.
   b. Click **Edge**.
      » The Edge sites overview opens.
   c. Click the name of an Edge site in the Edge site overview.
      » The Edge site page opens.
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/installer-
job/tools/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 may encounter while working with Edge. Select the tab of your installation type, K3S or EKS.

Issue	Proposed solution
You get the following error message: `Out of disk space`	You have to restart the Kubernetes pod in Edge. Run the following command: `sudo /usr/local/bin/kubectl delete pod <pod_name> --namespace <pod_namespace>` Tip For more information about Pods and namespaces, see the Kubernetes documentation.

You get one of the following error message:
• `Cannot allocate memory`
• `Error syncing pod`
	You have to restart Edge.
	Do the following:
	1. Run the following commands to remove all workflows:
	`sudo /usr/local/bin/kubectl delete --all workflows --namespace=collibra-edge`
	`sudo /usr/local/bin/kubectl delete --all workflows --namespace=collibra-fast`
	2. Run the following command to reboot Edge:
	`sudo reboot`

Issue	Proposed solution
You get the following error message: `Out of disk space`	You have to restart the Kubernetes pod in Edge. Run the following command: `kubectl delete pod <pod_name> --namespace <pod_namespace>` Tip For more information about Pods and namespaces, see the Kubernetes documentation.

You get one of the following error message:
 	
• `Cannot allocate memory`
• `Error syncing pod`
	You have to restart Edge.
	
	Do the following:
	
	1. Run the following commands to remove all workflows:
	
	`kubectl delete --all workflows --namespace=collibra-edge`
	
	`kubectl delete --all workflows --namespace=collibra-fast`
	
	2. Run the following command to reboot Edge:
	
	`sudo reboot`

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 and log files in Datadog which provide data about the issue. If you want to report a problem to Collibra Support, you can include these 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 ZIP file with logs and information about the server or EKS environment on which you installed the Edge site. Edge also generates two types of log files that are not included in the diagnostics file:

- Edge infrastructure log files, which are immediately sent to Collibra Data Intelligence Cloud once the file is created.
- Metadata connector log files, which can be stored only 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 and capability information. The logs from Datadog can be used by Collibra

Support to help solve general Edge issues. 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 logs 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 be accessed only by Collibra Support.

> Note By default, **Debug** logging for an Edge capability is set to `False`. We highly recommend enabling only 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 are generated when you:

- Register a database.
- Synchronize a schema.
- Trigger data sampling.
- Run profiling and classification.
- Synchronize technical lineage.
- Synchronize data quality.

> Note Metadata connector log files are not created when "Test Connection" is performed on an Edge.

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 them, review them locally, and then attach them to a Collibra support ticket.

Edge system monitoring

The system monitoring, run 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. This data is available only 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 level.
Low	The Catalog Connector logs contain the following: • All connection query logs • Any errors
Medium	The Catalog Connector logs include the Low logs and the following: • All cache queries • Additional information about the request

Verbosity log level	Description
High	The Catalog Connector logs include the Medium logs and the following: • The body of the request • The response

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

> Note If the Edge site installation fails, a diagnostics file is automatically generated for you. You can find this diagnostics file in the installation log Creating diagnostics file, in the following format: `Create /path/to/diagnostics/<diagnostics_file>.tgz`

You can run the diagnostics script without an Edge site installed to check if your system meets all requirements to install the Edge site.

1. Extract the Edge installer.

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

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.

> Important If the Edge command is not available, you will need to download the Edge tool and make it available in `/usr/local/bin/edge`

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, attach the reviewed Metadata connector log files to your ticket so that Collibra Support can help you with your issue.

Job logs are only kept for 15 days after the job is created. Job logs that are older than 15 days are removed from the platform, however, you can find records of these jobs on the **Jobs** tab of an Edge site or the **Jobs (beta)** dashboard of Edge.

> Tip You can also download the output file of a JDBC job from the Job dashboard.

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, 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.

> Important If the Edge command is not available, you will need to download the Edge tool and make it available in `/usr/local/bin/edge`

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 that has 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. On the main toolbar, click ⠿, and then click ⚙ **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?	All customers who want to ingest data into Collibra Data Intelligence Cloud benefit from Edge. Some of the benefits for using Edge are: • Data is processed in the customer's secure environment and only the process results are sent to Collibra Data Intelligence Cloud. • Edge can automatically anonymize sensitive profiling data before sending it to Collibra Data Intelligence Cloud. • Edge can automatically classify the metadata and send the classification results together with the profiling results to Collibra Data Intelligence Cloud. • Edge enables better profiling performance, because data no longer has to be copied or moved. • Edge can execute capabilities in parallel, considering this is dependent of available resources. Jobserver only executes capability jobs sequentially.

Question	Answer

Why should I migrate from Jobserver to Edge?	Edge provides our customers with all of the capabilities provided with Jobserver, but with better security controls and added capabilities. Edge provides seamless native integrations and on-site data processing solutions that prioritize security and proximity to the data, while keeping the processing of your data within your own environment. For more information, go to Migrate to Edge from Jobserver.
	
	The main differences between Edge and Jobserver are the following:
	
	• Edge is based on Kubernetes, a distributed runtime, which means:
	  ◦ It offers built in resource management.
	  ◦ It has reliable delivery of results to Collibra Data Intelligence Cloud.
	• Edge provides the ability to mirror images in your private docker registry to better fit your security policy.
	• Edge offers two upgrade modes to best suit your needs: Automatic and Manual.
	• Edge is a Collibra service compatible with on-premises as well as cloud environments.
	• Edge offers continuous delivery of capability types and updates will be delivered on a regular basis.
	• Edge updates are included with Collibra Data Intelligence Cloud releases.
	
	New capabilities will not be developed for Jobserver, as it will be made end of life from September 30, 2024. We recommend migrating to Edge before this date.

Can Edge run alongside Jobserver?	Yes, both can technically be run at the same time, however, we strongly recommend that you do not install both Jobserver and Edge on the same server. Edge should be installed on its own dedicated server.
What does the Edge architecture look like?	You can see how Edge interacts with other components in this architecture and components overview.

Question	Answer
Can Edge use Kubernetes provided by a Cloud vendor, for example Google Kubernetes Engine (GKE), Azure Kubernetes Services (AKS) or Amazon Elastic Kubernetes Service (EKS)?	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. You can install Edge on Amazon EKS. 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. Edge on AKS and GKE are a part of the current road map. Please contact your Customer Success Manager if you have any questions.
Can Edge be installed on Windows servers?	No, you cannot install an Edge site on Windows servers. Support for K8S, K3S in particular, and container technology is underserved on Windows without the equivalent of a Linux sub-system. We will continue to prioritize your experience on Linux-based operating systems, and as such, will not support Edge installation on Windows servers until the support is seamless.
Why can Edge only be run on a dedicated cluster? Will we be able to run Edge on a shared cluster in the future?	Edge currently runs on two namespaces. This requires Edge to be run on a dedicated cluster for optimized security. The ability for Edge to run securely on shared clusters for Amazon Kubernetes Service (EKS), Azure Kubernetes Service (AKS, GKE (Google Kubernetes Enginer), AWS Fargate, Openshift Container platform and other services is a part of the current road map. Please contact your Customer Success Manager if you have any questions.
What are the supported data sources on Edge?	You can find the list of supported data sources in the Data sources supported by Edge section.
How does authentication from Edge to the customer's data sources work?	Authentication to data sources depends on the source type that the capability is connecting to. JDBC sources are covered via Edge connection providers. Other sources are accessed in different ways by capabilities themselves.

Question	Answer
Can you connect using a cloud provider key manager such as AWS Secrets Manager, GCP Secret Manager or Azure Key Vault?	Not at this time.
Why do you not support CentOS Linux 8?	CentOS Linux 8 has been made end-of-life. We are committed to using the latest technologies to ensure the best performance of our software, and as such RedHat 8 is required in order to receive support for Edge installations.
How does Edge connect to Collibra Data Intelligence Cloud?	An Edge site is installed in the customer's environment, close to the data source. The Edge site communicates to Collibra Data Intelligence Cloud using an outbound HTTPS connection via port 443.
Is Edge on premises or in the Cloud?	Edge is always close to your data, and therefore can be on your premises or in a private or public Cloud setup.
Who controls Edge?	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. For more information, go to About Edge.
How is Edge updated?	Edge sites can be configured to either upgrade automatically whenever a new version is released, or upgrade manually, in order to control when and to which version your sites are upgraded. For more information, go to Upgrading an Edge site.
Can an Edge site connect to more than one Collibra environment?	No. Every Edge site belongs and authenticates to only one Collibra Data Intelligence Cloud environment.

Question	Answer
Do you need multiple instances of Edge for Data Quality to run?	No, only one Edge site is required for DQ Cloud. However, while you can technically run Collibra Data Quality & Observability and capabilities in the same Edge instance, you will need to ensure resources and space are available if you have a large Edge site. • Unlike DQ Cloud, Edge is not available for on-prem instances of Collibra DQ. • If you have an existing Edge site that runs capabilities without Data Quality, you can update Edge Config to enable/disable any service or configuration during any run time, in order to provide space to run Data Quality. • If you have an existing Edge site and are open to reinstalling, then you can enable the Data Quality flags during the reinstallation process in order to keep one instance of Edge. Note It is not recommended to run Classification and Data Quality in the same Edge instance, as they will compete for resources. Best practice is to have separate Edge sites for Classification and Data Quality.
Can Edge use customer-provided certificates to connect to Collibra Data Intelligence Cloud?	Currently, we do not support this. 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 keys and credentials are generated when you install the Edge site.
When do internal K3S certificates expire?	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 reinstall the Edge site.
Does Edge implement Cross-Site Request Forgery (CSRF) tokens?	Yes, the Edge management user interface can now implement CSRF tokens. Note The CSRF token needs to be unique per user session and should be a large, random value.

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.
Does Edge support High Availability and disaster recovery?	Edge does not support High Availability, but core Edge services can be replicated if Edge is installed on a multi-node cluster, and Edge capabilities can be restarted in the event of a failure. Disaster recovery is supported through regular backups. More information about our disaster recovery process can be found in this overview.
What troubleshooting information is collected and where is it stored?	When Edge is operational and has deployed running capabilities, jobs or services, it can collect information on multiple levels: • Infrastructure logs - default level info is collected, sent to the Cloud and accessible by Collibra. • Edge system monitoring - sent to the Cloud and accessible by Collibra. • Metadata connector logs - off by default and accessible by the customer . • Edge diagnostics - information is collected on demand by the customer on site and sent to Collibra as part of the support ticket.
Edge Sample Data capability: 1. Can everybody see sample data? 2. How is sample data queried from the database? 3. Which user account pulls the sample data from the database?	The Sample Data capability for Edge is a feature and needs to be activated. 1. Only users with the permission will be able to view the sample data. 2. Samples are queried from the data source upon request. 3. The samples will be pulled from the database using the ID of the account specified in the Edge connection.
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.

Question	Answer
What are Edge security considerations?	Edge is designed around security first principles. Several highlights: 1. No inbound connectivity - Edge site is always polling the platform via a REST endpoint. 2. Data is not stored on Edge after a job has finished. 3. Credentials are managed by Edge and not accessible outside of it. 4. Credentials on Edge site are encrypted with the key secured in the Collibra Data Governance Center. 5. Credentials can be updated both for data sources and Collibra Data Governance Center. 6. With the Edge Smart Upgrade feature, you can configure your Edge sites to upgrade manually. Manual upgrade allows you to run security scans on images included in a new release version before upgrading your Edge site version. Furthermore, these security scans can be performed in your own private docker registry. For more information on how your Edge sites can be upgraded, go to Upgrading an Edge site.
How are secrets stored on an Edge site?	You can find the details of how Edge stores secrets in this Storing secrets overview.