



Collibra Platform

Edge Infrastructure

Collibra Platform - Edge Infrastructure

Release date: June 08, 2025

Revision date: June 05, 2025

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

Contents	ii
Steps Overview: Setup and use a site	xii
About Edge sites	1
Edge Responsibilities	1
Edge components	3
Integration steps	3
Edge Command Line Interface (CLI)	5
How you can use the Edge CLI commands on k3S	5
Access help	15
How to use Edge CLI commands on a Managed Kubernetes cluster	16
Access help	40
Edge security	42
Communication between Edge and Colibra	43
Communication between Edge and other services	47
Authentication to data sources	50
Security scanning	51
What's next?	51
How to pull Colibra Edge docker images	52
Steps	52
Storing connection credentials	55
Customer Credentials	57
Credentials storage	57
Secret encryption	57

Credential encryption	57
Credentials transfer	58
Collibra platform credentials	58
Data samples in Edge	60
Edge Cache	61
Edge service repository	62
Monitoring and logging	63
Additional information	63
Host hardening on K3S-based integration	64
Prerequisites	64
Enable host hardening	64
Disable host hardening	65
About private registries with Edge	67
Supported private container image registries	67
Configure an Edge site with an Amazon Elastic Container Registry	69
Configure an Edge site with an Azure Container Registry	70
Access token	70
Service Principal ID with associated secret	71
Access token	71
Service Principal ID with associated secret	72
Azure IAM based authentication for AKS	72
Configure an Edge site with a Google Artifact Registry	74
Service Account Key	74
Service Account Key	75
Workload Identity Federation for GKE	75
Configure an Edge site with a private registry using user/pass authentication	77

User/Pass authentication	77
Edge Vaults	79
About Edge Vaults	80
What's next?	81
Integrate your Edge site with your vault	82
Edit vault integration configuration via Edge CLI	83
Steps	83
Steps	86
Steps	89
Steps	94
Steps	99
Steps	102
Steps	105
Steps	108
Steps	113
Steps	118
Retrieve your vault integration information via the Edge CLI	122
Prerequisites	122
Retrieve information on all vault integrations	122
Retrieve specific vault details	123
Retrieve information on all vault integrations	124
Retrieve specific vault details	124
How to access help for Vaults	126
Delete an Edge site vault integration	127
Prerequisites	127
Steps	127

Steps	128
Installing an Edge site	129
About an Edge site installation	130
Properties	131
Statuses	131
Installation directories on K3S	132
System requirements of an Edge site	133
Software requirements	133
Hardware requirements	134
Network requirements	136
Commercial	136
FedRAMP	138
Whats next	139
EKS requirements	139
Software requirements	140
Hardware requirements	143
Network requirements	143
Commercial	144
FedRAMP	145
GKE requirements	145
Software requirements	146
Hardware requirements	149
Network requirements	149
Commercial	150
FedRAMP	151
AWS Fargate using EKS requirements	151

Software requirements	152
Network requirements	156
Commercial	156
OpenShift requirements	157
Software requirements	157
Hardware requirements	160
Network requirements	161
Commercial	161
FedRAMP	162
AKS requirements	163
Software requirements	163
Hardware requirements	166
Network requirements	167
Commercial	167
FedRAMP	168
Whats next	169
Create an Edge site	170
Prerequisites	170
Steps	170
What's next?	171
Install an Edge site	172
Prerequisites	172
Steps	172
Prerequisites	177
Steps	178
Configure a forward proxy	217

Steps	217
What's next?	221
Supported forward proxy configurations for Edge	222
Explicit proxy	222
Transparent proxy	223
Reinstall an Edge site	224
Steps	224
Steps	229
Upgrade the operating system for k3s Edge sites	251
Steps	251
Troubleshooting	251
Upgrading an Edge site	253
Edge site upgrade methods	254
Automatic upgrade	254
Manual upgrade	254
What's next?	257
How to manually upgrade your Edge site	258
What's next?	258
Enable Automatic upgrade for Edge sites	260
New Edge sites	260
Existing Edge sites	261
What's next?	261
Enable Manual upgrade for Edge sites	262
New Edge sites	262
Existing Edge sites	263
What's next?	263

Maintaining Edge sites	265
Edit an Edge site	266
Prerequisites	266
Steps	266
Update Edge user's username or password	267
Update the outbound proxy configuration	269
Steps	269
Help file of the script	269
Back up an Edge site	271
What's Next?	272
Delete an Edge site	273
Prerequisites	273
Steps	273
Disaster recovery for managed Kubernetes Edge sites	276
Use Case Scenario	276
Migrating to Edge from Jobserver	277
Why migrate to Edge?	278
Migration to Edge overview	280
[[[Undefined variable CollibraGeneral.additional-resources]]]	281
Troubleshooting Edge	282
Edge FAQ	283
About Collibra Cloud sites	290
What is included with a Collibra Cloud site?	290
Limitations	292
What's next?	293
Request a Collibra Cloud site	294

Steps	294
What's next?	294
Edge and Collibra Cloud site site connections	295
About Edge and Collibra Cloud site connections	296
Edit an Edge and Collibra Cloud site site connection	303
Available vaults	303
Prerequisites	303
Steps	303
Delete an Edge or Collibra Cloud sitesite connection	307
Prerequisites	307
Steps	307
JDBC connections	309
Create a JDBC connection	309
Prerequisites	309
Steps	310
What's next?	319
Customizing the database name for database-less data sources	320
Edit a JDBC connection	321
Prerequisites	322
Steps	322
Delete a JDBC connection	332
Prerequisites	332
Steps	332
Use keys to access a database	333
Edge and Collibra Cloud site capabilities	334
About Edge and Collibra Cloud site capabilities	335

Capability templates	335
Capability template structure	344
About preparing an Edge or Collibra Cloud site for data sources	345
Steps	345
What's next?	346
Add a capability to an Edge or Collibra Cloud site site	347
Prerequisites	347
Steps	347
More information	347
Edit an Edge or Collibra Cloud site site capability	349
Prerequisites	349
Steps	349
Delete a capability from an Edge or Collibra Cloud site site	350
Prerequisites	350
Steps	350
Jobs dashboard	351
View Edge site jobs	352
Additional resources	352
Review an Edge or Collibra Cloud site site job details	353
Where do I find the Edge or Collibra Cloud site Site ID and Job ID?	353
Where do I find the Edge or Collibra Cloud site Job ID?	353
Download job output files	355
Prerequisites	355
Steps	355
Cancel jobs	357
Prerequisites	357

Steps 357

Steps Overview: Setup and use a site

Creating and setting up a site allows you to connect your data sources with your Collibra Platform.

#	Step	Description
1	Create an Edge site in the UI or request a Collibra Cloud site site .	<p>A site is where you will manage your connections and capabilities.</p> <ul style="list-style-type: none">• Edge sites: When you create an Edge site, you can choose 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. Once you create your Edge site in the UI, you must install it in your organization's environment.• Collibra Cloud site sites: Collibra creates and manages the Collibra Cloud site site for you. You must submit a request to Collibra, you can't create the site yourself. Collibra Cloud site sites upgrade automatically, so they're always on the latest version.
2	Give the required permissions	<p>You need at least one user who is able to install your sites and manage your connections and capabilities in your site. They typically require the following permission:</p> <p>You have a global role that has the Manage Edge sites global permission.</p>
3	Install an Edge site Note If you have a Collibra Cloud site site, skip this step.	<p>Your Edge site is installed in your organization's environment. You typically need to work with your administrators, IT team, and Kubernetes experts to install your Edge sites and meet your organization's system and security requirements.</p>

#	Step	Description
4	Prepare your site for data sources	<p>Once you have a site, you need to create connections from the site to your data sources, and define capabilities that you want to run on these connections.</p> <p>For example, you create a connection to Databricks Unity Catalog and add a capability that allows you to integrate the metadata from Databricks Unity Catalog in Collibra.</p> <p>If you are installing your Edge site on a managed Kubernetes cluster, you must use the Edge Command Line Interface (CLI). The Edge CLI is the primary Edge utility tool that allows you to set up and manage certain aspects of your Edge site and is included when you download the Edge site installer.</p>
4a	Create a connection	<p>A connection links your site with your data source. Once a connection is available, capability jobs can run through this connection to send information to Collibra.</p> <p>For more information, go to our list of available connections.</p>
4b	Create a capability	<p>A capability calls your data source via the connection to get requested information from the data source and sends the information to Collibra. The results can be assets, such as Schema or Table assets, or extra information about the assets.</p> <p>For more information, go to our list of available capabilities.</p>
6	Upgrade your Edge sites	<p>Occasionally, Edge launches a new functionality or Kubernetes support that requires you to upgrade or reinstall your Edge sites.</p> <p>Collibra communicates about this in the release notes and the Edge compatibility table.</p>

Note

- If you have automatic upgrades enabled for your Edge sites, you don't need to do anything when an upgrade is required. However, if a reinstallation is required, you may still need to perform this action.
- If you have a Collibra Cloud site, your site is automatically upgraded.

#	Step	Description
7	Maintain your Edge site	<p>You can use the Edge CLI to maintain and update certain aspects of your Edge site.</p> <p>For example, set up a forward proxy or update vault credentials.</p> <div>Note If you have a Collibra Cloud site site, you can't use the Edge CLI.</div>

About Edge sites

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.

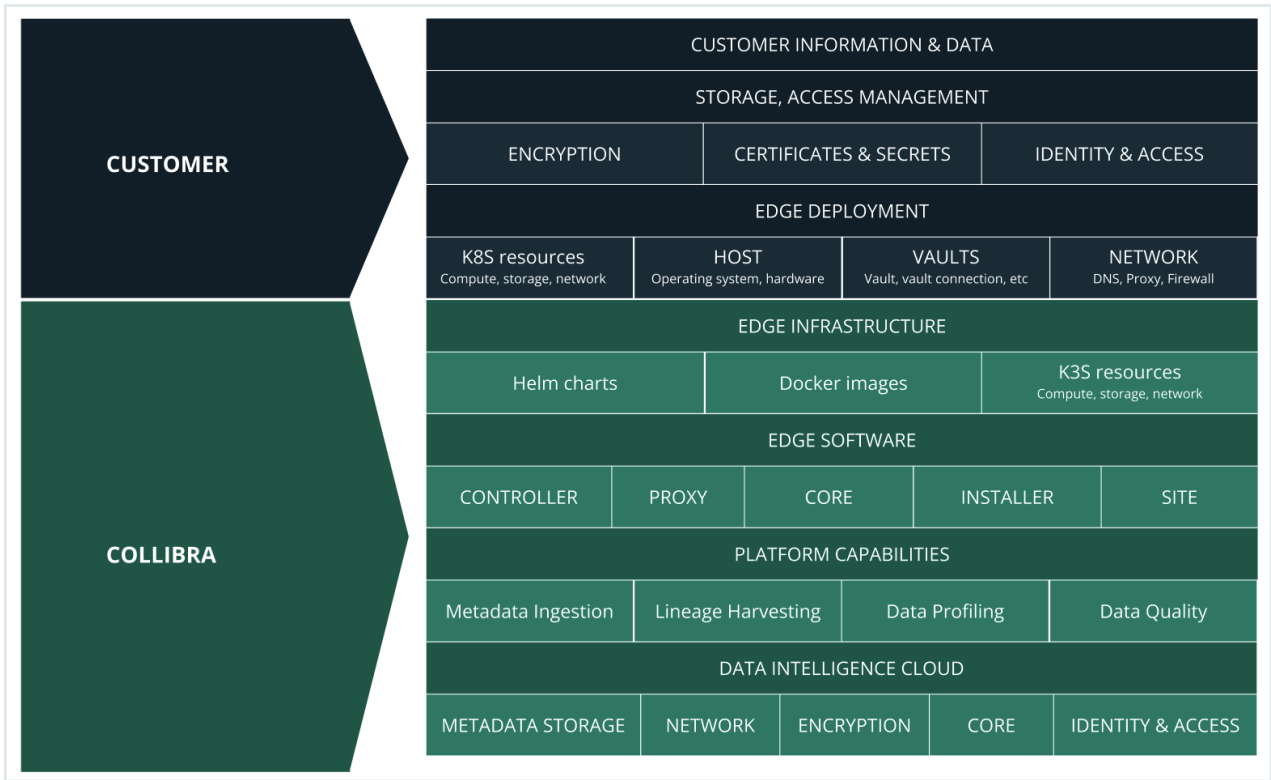
Edge enables Collibra Platform to [safely](#) connect to your data sources hosted in an on-premise or cloud environment. It processes the data source information on the Edge site and sends the process results to Collibra Platform.

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.

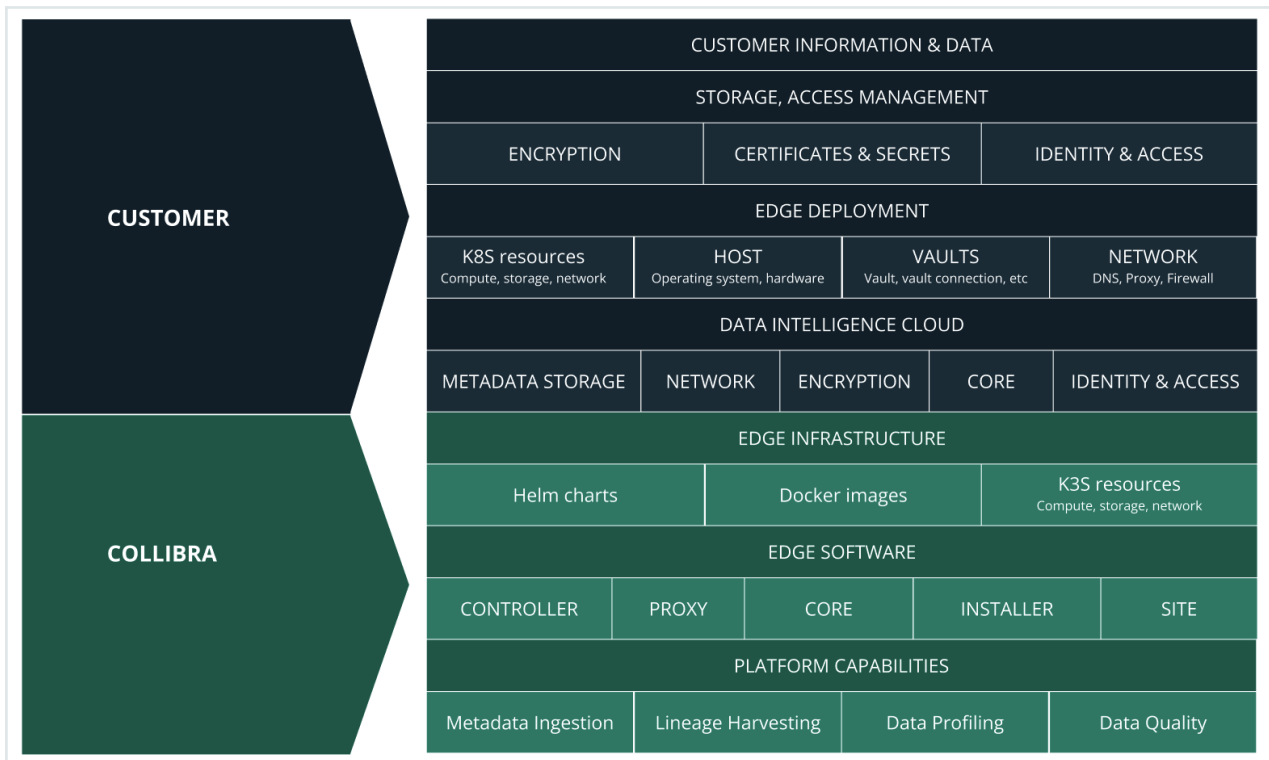
Edge for commercial customers or Collibra Platform for Government are Collibra solutions that allows your Collibra Platform to safely connect to your data sources hosted in an on-

premise or cloud environment.



Colibra Platform Self-Hosted (CPSH) is a Colibra solution that allows you to install your Colibra Platform on an infrastructure of your choice. For Edge, this means that you are hosting both your Colibra platform and your Edge site. For more information about CPSH,

go to our [CPSH documentation](#).



Edge components

Edge consists of three main components:

- An Edge configuration page in Collibra Platform 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. The Edge site installer includes the Edge CLI tool which allows you to install sites on managed Kubernetes clusters and manage configurations such as vaults and Shared Storage connections.

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 Platform Settings.	You have a global role with the Manage Edge sites global permission in Collibra Platform.
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 Platform.
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 Platform.

Edge Command Line Interface (CLI)

Note Edge CLI is only available on Linux for x86_64 architecture.

The Edge Command Line Interface (CLI) is a tool that allows you to set up and manage aspects of your Edge site.

As of 2024.05, the Edge CLI is included in the [Edge site installer](#), and can be found in the extracted Edge site installer directory. If you need to download or update only the Edge CLI, see the commands in the table below based on where your Edge site is installed.

How you can use the Edge CLI commands on k3S

Note This is not an exhaustive list of commands. If you want to see the full list of commands available via the Edge CLI, run the [CLI help command](#).

Action	Definition
Download a new Edge CLI	<p>You may need to download an Edge CLI because you do not have it available locally yet or you need a newer version of the Edge CLI.</p> <ul style="list-style-type: none"> If you do not have the Edge CLI available locally, run the following command from wherever your Edge site is installed: <pre># Get the name of the pod from which we will copy edgecli , run: EDGE_CD_POD=\$(sudo /usr/local/bin/kubectl get pod -n collibra-edge -l app.kubernetes.io/name=edge-cd -o jsonpath='{.items[0].metadata.name}') # Copy edgecli to provided path sudo /usr/local/bin/kubectl cp --retries=-1 -n collibra-edge \${EDGE_CD_POD}:edgecli /usr/local/bin/edgecli # Make the downloaded binary executable sudo sudo chmod +x /usr/local/bin/edgecli</pre> <ul style="list-style-type: none"> If you want to update your Edge CLI to the latest version, run the following command from the Edge CLI: <pre>sudo ./edgecli cli upgrade -d <edgecli_dir></pre>
Get Edge site diagnostics	<p>Use this command from this command and provide the results to Collibra Support when troubleshooting issues on your Edge site:</p> <pre>sudo ./edgecli diagnostics -d <diagfile.tgz></pre>
Create an Edge site backup	<p>Use this command to create a backup of your Edge site. This is required when you want to reinstall your Edge site.</p> <pre>sudo ./edgecli recovery backup --path <backup_path></pre>

Action	Definition
Update the Edge user credentials in Collibra Platform	Use this command to update the Edge user's username or password in Collibra Platform: <pre>sudo ./edgecli config dgc --pass <password> --url <dgc url> --user <username></pre>
Restart Edge components	Use this command if you need to restart an Edge component, but not restart the virtual machine: <pre>sudo ./edgecli restart</pre>
Update forward proxy settings	Use this command to update forward proxy settings for your Edge site: <pre>sudo ./edgecli config proxy --path <path to proxy config></pre>

Action	Definition
Pull list of custom certificates	<p>Use this command to pull a list of all custom certificates configured on your Edge site.</p> <pre>sudo ./edgecli config ca list</pre> <p>Additional parameters:</p> <ul style="list-style-type: none">• <code>--namespace</code>: If your Edge site has a custom namespace, you must add this parameter and the namespace name to the command.• <code>--raw</code>: Pulls the raw data of the certificates. Without this, only the basic certificate information is returned.• You want to see the raw data for all of the custom certificates configured on your Edge site: <pre>sudo ./edgecli config ca list --raw</pre> <ul style="list-style-type: none">• You want to see the custom certificates configured on an Edge site with a specific namespace: <pre>sudo ./edgecli config ca list --raw --namespace <my-namespace></pre>

Action	Definition
Add custom certificates to the Edge site truststore	<p>Use this command to add custom certificates to the Edge truststore after an Edge site has been installed:</p> <pre>sudo ./edgecli config ca merge --path certificate.pem</pre> <p>Additional parameter:</p> <ul style="list-style-type: none">• <code>--namespace</code>: If your Edge site has a custom namespace, you must add this parameter and the namespace name to the command.• You want to add a custom certificate to your Edge site truststore: <pre>sudo ./edgecli config ca merge --path certs.pem</pre> <ul style="list-style-type: none">• You want to add the custom certificates that are configured on an Edge site with a specific namespace: <pre>sudo ./edgecli config ca merge --path certificate.pem --namespace <my-namespace></pre>

Action	Definition
Replace all existing custom certificates in the Edge site truststore	<p>Use this command to replace all existing custom certificates in the Edge site truststore:</p> <pre>sudo ./edgecli config ca replace --path certificate.pem</pre> <p>Important</p> <ul style="list-style-type: none">• This command only replaces custom certificates. System certificates are not impacted by this command. <p>Additional parameter:</p> <ul style="list-style-type: none">• <code>--namespace</code>: If your Edge site has a custom namespace, you must add this parameter and the namespace name to the command.• You want to replace all custom certificates that are configured on an Edge site: <pre>sudo ./edgecli config ca replace --path certs.pem</pre> <ul style="list-style-type: none">• You want to replace all custom certificates that are configured on an Edge site with a specific namespace: <pre>sudo ./edgecli config ca replace --path certificate.pem --namespace <my-namespace></pre>

Action	Definition
Setup and use Vaults with your Edge site	<p>This feature is available only in the latest UI.</p> <p>Vault commands are dependent on which vault and authentication method you use. Visit the dedicated pages to learn what the Edge CLI commands are to:</p> <ul style="list-style-type: none">• Add a vault.• Edit a vault.• Retrieve a vault.• Delete a vault.

Action	Definition
Setup and manage shared files for the Shared Storage connection	<p>The Shared Storage connection for technical lineage allows you to access data source files from a shared folder.</p> <p>To setup and manage the folder and files used for the Shared Storage connection, use the following commands in the Edge CLI:</p>

- Upload a shared folder which contains multiple data source files:

```
sudo ./edgecli objects folder-upload
--source <source-string>
--target <target-string>
--ttlSeconds <time>
```

Key	Definition
<source-string>	The data source file you want to upload for your Shared Storage connection.
<target-string>	The folder in your Edge site where you want to store the shared folder and files.
<time> (optional)	The number of seconds that the uploaded files will be available before being evicted, the default is 15552000 seconds (180 days).

Note This folder does not have to already exist in your Edge site. If it does not, a folder with the name entered here will be created in your Edge site, containing the folder and files you have uploaded.

- Upload a single shared data source file:

```
sudo ./edgecli objects file-upload
--source <source-string>
--target <target-string>
--key <key-string>
--ttlSeconds <time>
```

Action	Definition
--------	------------

Key	Definition
<source-string>	The data source file you want to upload for your Shared Storage connection.
<target-string>	The folder in your Edge site where you want to store the shared file.
	<p>Note This folder does not have to already exist in your Edge site. If it does not, a folder with the name entered here will be created in your Edge site, containing the file you have uploaded.</p>
<key-string> (optional)	The path of a specific file within a folder or nested within multiple folders. For example, you only want to upload the myFile.txt file, which is in the myFolders folder. If you do not specify this property, it will default to the file name.
<time> (optional)	The number of seconds that the uploaded file will be available before being evicted, the default is 15552000 seconds (180 days).
<ul style="list-style-type: none"> Upload multiple folders to their own shared folder: <pre>sudo ./edgecli objects multi-folder-upload --parallelism <parallel-uploads> --source <source-string> --ttlSeconds <time></pre>	
Key	Definition
<parallel-uploads> (optional)	<p>The number of files that are uploaded at the same time.</p> <p>For example, you want to upload 30 files that are in 3 folders and set the parallelism property to 10. Your files are uploaded to the shared storage connection 10 files at a time until all 30 files have been uploaded.</p>

Action	Definition
--------	------------

Key	Definition
-----	------------

<source-string>	The data source folders you want to upload to your Shared Storage connection.
<time> (optional)	The number of seconds that the uploaded file will be available before being evicted, the default is 15552000 seconds (180 days).

- Update or add a new file to an existing folder in Edge:

```

sudo ./edgecli objects file-upload --source
<source-string>
--target <target-string> \
--key <key-string> \

```


Key	Definition
-----	------------

<source-string>	The data source file you want to upload to replace an existing file or add to an existing folder in your Shared Storage connection.
<target-string>	The folder in your Edge site where you want to store the shared file.
<key-string> (optional)	The path of a specific file within a folder or nested within multiple folders. For example, you only want to upload the myFile.txt file, which is in the myFolders folder. If you do not specify this property, it will default to the file name.

- Pull a list of all folders and files that have been uploaded to your Edge site:

```

sudo ./edgecli objects folder-list

```
- Delete a shared folder or file from your Edge site:

Action	Definition
	<pre>sudo ./edgecli objects folder-delete --target <target-string></pre>
Key	Definition
<target-string>	The name of the Edge shared folder you want to delete from your Edge site.

Access help

If you need help when using the Edge CLI tool, you can run the help command to:

- View a full list of supported Edge CLI commands:

```
sudo ./edgecli -h
```

- View the specific parameters and usage of a specific Edge CLI command. For example, running the following command will return help information for the delete shared folder command for the Shared Storage connection:

```
sudo ./edgecli objects folder-delete -h
Deletes shared folder that was uploaded, if it exists

Usage:
  edgecli objects folder-delete [flags]

Flags:
  --target string    The folder name specified in
edge (DGC)

Global Flags:
  -h, --help
```

How to use Edge CLI commands on a Managed Kubernetes cluster

Note This is not an exhaustive list of commands. If you want to see the full list of commands available via the Edge CLI, run the [CLI help command](#).

Actions	Definitions
---------	-------------

Install an Edge site	<p>You can install your Edge site on a managed Kubernetes cluster by following the steps outlined in Install an Edge site, in addition to running one of the following commands:</p> <div data-bbox="430 508 502 542" data-label="Section-Header">Note</div> <div data-bbox="438 542 1351 754" data-label="List-Group"> <ul style="list-style-type: none"> You can install your Edge site with either terminal logging or terminal and file logging. Both options log the output of your Edge site installation. <ul style="list-style-type: none"> Terminal logging only saves the output to the Edge terminal. Terminal and file logging saves the output both to the terminal and a separate file. This file will be saved in the current directory with the naming format: <code>edge-installer-\$(date +"%Y-%m-%d_%H-%M-%S").log</code> </div> <div data-bbox="461 848 783 887" data-label="Text"> <pre>./edgecli install</pre> </div> <div data-bbox="419 952 1386 1030" data-label="Text"> <p>Add additional flags to the install command as needed. For example, if you have a custom namespace or want to use a private docker registry:</p> </div> <div data-bbox="435 1068 788 1108" data-label="Section-Header"> <table border="1"> <thead> <tr> <th>Flag</th> <th>Description</th> </tr> </thead> </table> </div> <div data-bbox="435 1149 1260 1281" data-label="Text"> <table border="1"> <tbody> <tr> <td> -n <my-namespace> e> </td> <td> If you created a custom namespace, add -n <my-namespace> to the command. For example: </td> </tr> </tbody> </table> </div> <div data-bbox="684 1303 1158 1373" data-label="Text"> <pre>./edgecli install -n <my-namespace></pre> </div> <div data-bbox="1468 2101 1516 2139" data-label="Page-Footer">17</div>	Flag	Description	-n <my-namespace> e>	If you created a custom namespace, add -n <my-namespace> to the command. For example:
Flag	Description				
-n <my-namespace> e>	If you created a custom namespace, add -n <my-namespace> to the command. For example:				

Actions Definitions

Flag	Description
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy  
temp/proxy/proxyproperties
```

Actions

Definitions

Flag	Description
<code>--ca</code>	If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.

Note If your custom certificate are not in the default **ca.pem** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--ca temp/certs
```

You can also use this flag to add a custom certificate for data sources.

Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.

As you may not have a list of all required certificates at the time of installation, we recommend the `./edgecli config ca merge --path` command shown in the [Edge CLI](#) topic.

The process functions as follows:

- Edge and the data source connect using the data source certificate.
- Edge communicates the data source metadata to your Collibra Platform using a Collibra certified certificate.

Actions Definitions

Flag	Description
<code>--is-openshift</code>	If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url <registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user <registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>

Actions Definitions

Flag	Description
<code>--registry-pass <registry-pass></code>	Your registry account password. Add this flag if you use a private docker registry with authentication.
<code>--user-id <user-id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.
<code>--group-id <group-id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div> Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures. </div> <p>If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).</p>

Actions

Definitions

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> ◦ <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ◦ <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ◦ <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre>--set global.priorityClassName.platform= critical-priority --set global.priorityClassName.applicati on=high-priority --set</pre> </div>

Actions Definitions

Flag Description

```
global.priorityClassName.job=low-
priority
```

```
./edgecli install
--registry-url https://private-
docker.registry.com
--registry-user user1
--registry-pass pass12
```

```
./edgecli install 2>&1 | tee "edge-installer-
$(date +"%Y-%m-%d_%H-%M-%S").log"
```

Add additional flags to the install command as needed. For example, if you have a custom namespace or want to use a [private docker registry](#):

Flag Description

`-n <my-namespace>` If you created a custom namespace, add `-n <my-namespace>` to the command. For example:

```
./edgecli install -n <my-
namespace>
```

Actions Definitions

Flag	Description
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy  
temp/proxy/proxyproperties
```

Actions Definitions

Flag	Description
<code>--ca</code>	If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.

Note If your custom certificate are not in the default **ca.pem** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--ca temp/certs
```

You can also use this flag to add a custom certificate for data sources.

Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.

As you may not have a list of all required certificates at the time of installation, we recommend the `./edgecli config ca merge --path` command shown in the [Edge CLI](#) topic.

The process functions as follows:

- Edge and the data source connect using the data source certificate.
- Edge communicates the data source metadata to your Collibra Platform using a Collibra certified certificate.

Actions Definitions

Flag	Description
<code>--is-openshift</code>	If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url <registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user <registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>

Actions Definitions

Flag	Description
<code>--registry-pass <registry-pass></code>	Your registry account password. Add this flag if you use a private docker registry with authentication.
<code>--user-id <user-id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.
<code>--group-id <group-id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div> Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures. </div>

If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).

Actions

Definitions

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> ◦ <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ◦ <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ◦ <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre>--set global.priorityClassName.platform= critical-priority --set global.priorityClassName.applicati on=high-priority --set</pre> </div>

Actions	Definitions
---------	-------------

Flag	Description
------	-------------

```
global.priorityClassName.job=low-
priority
```

```
./edgecli install
--registry-url https://private-
docker.registry.com
--registry-user user1
--registry-pass pass12
2>&1 | tee "edge-installer-$(date
+"%Y-%m-%d_%H-%M-%S").log"
```

Uninstall an Edge site If you have installed your Edge site using the [Edge CLI method](#), you can uninstall your Edge site by using one of the following commands:

- With terminal logging

```
./edgecli uninstall
```

- With terminal and file logging:

```
./edgecli uninstall 2>&1 | tee "edge-installer-
$(date +"%Y-%m-%d_%H-%M-%S").log"
```

Note

- If your Edge site was installed using the old method, and not the Edge CLI method, use the uninstall command via the Edge tool.
- If your Edge site is installed via the Edge CLI method and it has a custom namespace, you must add `-n <my-namespace>` to the command.

Actions	Definitions
Download a new Edge CLI	<p data-bbox="387 344 1398 416">You may need to download an Edge CLI because you do not have it available locally yet or you need a newer version of the Edge CLI.</p> <ul data-bbox="387 445 1398 551" style="list-style-type: none"> <li data-bbox="387 445 1398 551">• If you do not have the Edge CLI available locally, run the following command from a Linux machine that has access to the Kubernetes cluster where your Edge site is installed: <pre data-bbox="467 595 1358 1025"> # Get the name of the pod from which we will copy edgecli , run: EDGE_CD_POD=\$(kubectl get pod -n collibra-edge - l app.kubernetes.io/name=edge-cd -o jsonpath='{.items[0].metadata.name}') # Copy edgecli to provided path sudo /usr/local/bin/kubectl cp --retries=-1 -n collibra-edge \${EDGE_CD_POD}:edgecli /usr/local/bin/edgecli # Make the downloaded binary executable sudo chmod +x <PATH_TO_EXISTING_EDGECLI>/edgecli </pre> <ul data-bbox="387 1099 1398 1171" style="list-style-type: none"> <li data-bbox="387 1099 1398 1171">• If you want to update your Edge CLI to the latest version, run the following command from the Edge CLI: <pre data-bbox="467 1211 1246 1272"> ./edgecli cli upgrade -n <my-namespace> -d <edgecli_dir> </pre> <p data-bbox="438 1375 1302 1487">Note If your Edge site is installed on a dedicated cluster via the Edge CLI method and it does not have a custom namespace, you can remove <code>-n <my-namespace></code> from the commands.</p>

Actions	Definitions
Get Edge site diagnostics	<p>Use this command from this command and provide the results to Collibra Support when troubleshooting issues on your Edge site:</p> <pre>./edgecli diagnostics -n <my-namespace> -d <diagfile.tgz></pre> <p>Note If your Edge site is installed on a dedicated cluster via the Edge CLI method and it does not have a custom namespace, you can remove <code>-n <my-namespace></code> from the command.</p>
Create an Edge site backup	<p>Use this command to create a backup of your Edge site. This is required when you want to reinstall your Edge site.</p> <pre>./edgecli recovery backup --path <backup_path></pre>
Update the Edge user credentials in Collibra Platform	<p>Use this command to update the Edge user's username or password in Collibra Platform:</p> <pre>./edgecli config dgc -n <my-namespace> --pass <password> --url <dgc url> --user <username></pre> <p>Note If your Edge site is installed on a dedicated cluster via the Edge CLI method and it does not have a custom namespace, you can remove <code>-n <my-namespace></code> from the command.</p>
Restart Edge components	<p>Use this command if you need to restart an Edge component, but not restart the virtual machine:</p> <pre>./edgecli restart</pre>

Actions	Definitions
Update forward proxy settings	<p>Use this command to update forward proxy settings for your Edge site:</p> <pre>./edgecli config proxy --path <path to proxy config></pre>
Pull list of custom certificates	<p>Use this command to pull a list of all custom certificates configured on your Edge site.</p> <pre>./edgecli config ca list</pre> <p>Additional parameters:</p> <ul style="list-style-type: none">• <code>--namespace</code>: If your Edge site has a custom namespace, you must add this parameter and the namespace name to the command.• <code>--raw</code>: Pull the raw data of the certificates. Without this, only the basic certificate information is returned.• You want to see the raw data for all of the custom certificates configured on your Edge site:<pre>./edgecli config ca list --raw</pre>• You want to see the custom certificates configured on an Edge site with a specific namespace:<pre>./edgecli config ca list --namespace <my-namespace> --raw</pre>

Actions	Definitions
Add custom certificates to the Edge site truststore	<p>Use this command to add custom certificates to the Edge truststore after an Edge site has been installed:</p> <pre>./edgecli config ca merge --path certificate.pem</pre> <p>Additional parameter:</p> <ul style="list-style-type: none">• <code>--namespace</code>: If your Edge site has a custom namespace, you must add this parameter and the namespace name to the command.• You want to add a custom certificate to your Edge site truststore: <pre>./edgecli config ca merge --path certs.pem</pre> <ul style="list-style-type: none">• You want to add the custom certificates that are configured on an Edge site with a specific namespace: <pre>./edgecli config ca merge --namespace <my-namespace> --path certificate.pem</pre>

Actions	Definitions
Replace all existing custom certificates in the Edge site truststore	<p>Use this command to replace all existing custom certificates in the Edge site truststore:</p> <pre>./edgecli config ca replace --path certificate.pem</pre> <p>Important</p> <ul style="list-style-type: none"> This command only replaces custom certificates. System certificates are not impacted by this command. <p>Additional parameter:</p> <ul style="list-style-type: none"> <code>--namespace</code>: If your Edge site has a custom namespace, you must add this parameter and the namespace name to the command. You want to replace all custom certificates that are configured on an Edge site: <pre>./edgecli config ca replace --path certs.pem</pre> <ul style="list-style-type: none"> You want to replace all custom certificates that are configured on an Edge site with a specific namespace: <pre>./edgecli config ca replace --namespace <my-namespace> --path certificate.pem</pre>
Setup and use Vaults with your Edge site	<p>This feature is available only in the latest UI.</p> <p>Vault commands are dependent on which vault and authentication method you use. Visit the dedicated pages to learn what the Edge CLI commands are to:</p> <ul style="list-style-type: none"> Add a vault . Edit a vault. Retrieve a vault. Delete a vault.

Actions	Definitions
Setup and manage shared files for the Shared Storage connection	<p>The Shared Storage connection for technical lineage allows you to access data source files from a shared folder.</p> <p>To setup and manage the folder and files used for the Shared Storage connection, use the following commands in the Edge CLI:</p> <ul style="list-style-type: none">• Upload a shared folder which contains multiple data source files: <pre>./edgecli objects folder-upload --source <source-string> --target <target-string> --ttlSeconds <time></pre>
Key	Definition
<source-string>	The data source file you want to upload for your Shared Storage connection.

Actions

Definitions

Key

Definition

`<target-string>`

The folder in your Edge site where you want to store the shared folder and files.

Note This folder does not have to already exist in your Edge site. If it does not, a folder with the name entered here will be created in your Edge site, containing the folder and files you have uploaded.

`<time>` (optional)

The number of seconds that the uploaded files will be available before being evicted, the default is 15552000 seconds (180 days).

- Upload a single shared data source file:

Actions

Definitions

```
./edgecli objects file-upload  
--source <source-string>  
--target <target-string>  
--key <key-string>  
--ttlSeconds <time>
```

Key

Definition

<source-string>

The data source file you want to upload for your Shared Storage connection.

<target-string>

The folder in your Edge site where you want to store the shared file.

Note This folder does not have to already exist in your Edge site. If it does not, a folder with the name entered here will be created in your Edge site, containing the file you have uploaded.

Actions Definitions

Key	Definition
<key-string> (optional)	<p>The path of a specific file within a folder or nested within multiple folders. For example, you only want to upload the myFile.txt file, which is in the myFolders folder.</p> <p>If you do not specify this property, it will default to the file name.</p>
<time> (optional)	<p>The number of seconds that the uploaded file will be available before being evicted, the default is 15552000 seconds (180 days).</p>

- Upload multiple folders to their own shared folder:

```
./edgecli objects multi-folder-upload
--parallelism <parallel-uploads>
--source <source-string>
--ttlSeconds <time>
```

Actions Definitions

Key	Definition
<parallel-uploads> (optional)	The number of files that are uploaded at the same time. For example, you want to upload 30 files that are in 3 folders and set the parallelism property to 10. Your files are uploaded to the shared storage connection 10 files at a time until all 30 files have been uploaded.
<source-string>	The data source folders you want to upload to your Shared Storage connection.
<time> (optional)	The number of seconds that the uploaded file will be available before being evicted, the default is 15552000 seconds (180 days).

- Update or add a new file to an existing folder in Edge:

```
./edgecli objects file-upload --source <source-string>
--target <target-string> \
--key <key-string> \
```

Key	Definition
<source-string>	The data source file you want to upload to replace an existing file or add to an existing folder in your Shared Storage connection.
<target-string>	The folder in your Edge site where you want to store the shared file.
<key-string> (optional)	The path of a specific file within a folder or nested within multiple folders. For example, you only want to upload the myFile.txt file, which is in the myFolders folder. If you do not specify this property, it will default to the file name.

- Pull a list of all folders and files that have been uploaded to your Edge site:

Actions Definitions

```
./edgecli objects folder-list
```

- Delete a shared folder or file from your Edge site:

```
./edgecli objects folder-delete
--target <target-string>
```

Key	Definition
<target-string>	The name of the Edge shared folder you want to delete from your Edge site.

Access help

If you need help when using the Edge CLI tool, you can run the help command to:

- View a full list of supported Edge CLI commands:

```
./edgecli -h
```

- View the specific parameters and usage of a specific Edge CLI command. For example, running the following command will return help information for the delete shared folder command for the Shared Storage connection:

```
./edgecli objects folder-delete -h
Deletes shared folder that was uploaded, if it exists

Usage:
  edgecli objects folder-delete [flags]

Flags:
  --target string    The folder name specified in
edge (DGC)
```

```
Global Flags:  
-h, --help
```


Edge security

Edge is built with security first approach. All communication channels are secured by TLS 1.3 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 Platform.

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.

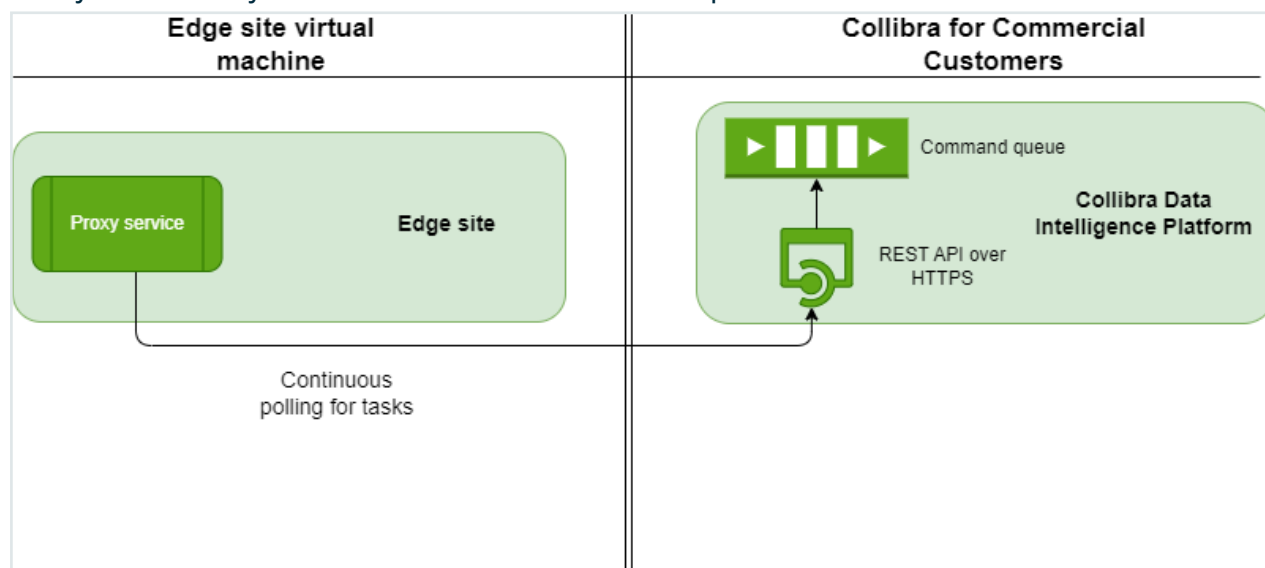


Communication between Edge and Colibra

Edge operates over an outbound-only model — it executes tasks as commands polled from your Colibra platform. All data is encrypted in [transit between your Edge site and the Colibra Platform](#) via certificates issued by a Colibra-chosen Certificate Authority (CA) over TLS 1.3 and basic authentication. However, if there is a forward proxy server between the Edge site and Colibra, you have to use the [proxy server's CA](#).

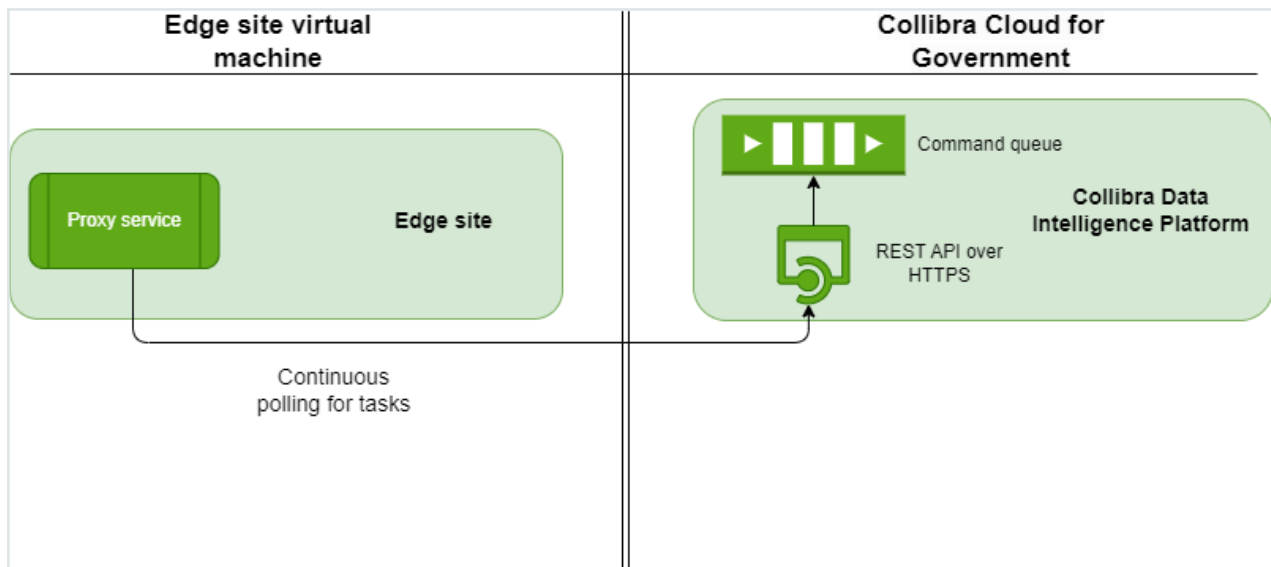
A user account is generated for communicating to Colibra each time the Edge site installer is downloaded. This user account is unique to each Edge site. It is possible to change the password of this user account by following the steps outlined in our [Update Edge user password](#) article.

Edge for commercial customers is a Colibra solutions that allows your Colibra Platform to safely connect to your data sources hosted in an on-premise or cloud environment.



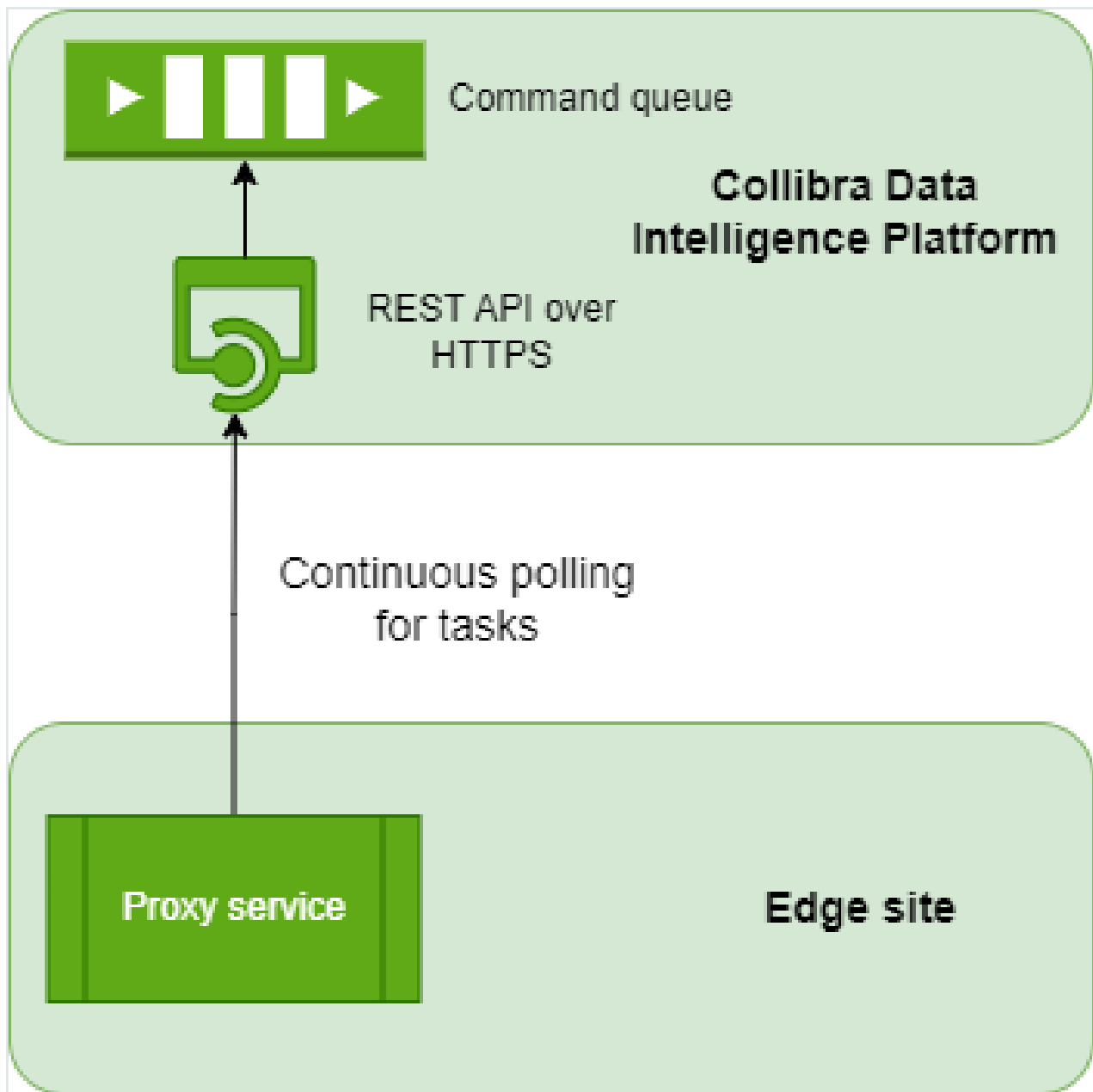
Edge for Colibra Platform for Government is a Colibra solutions that allows your Colibra Platform to safely connect to your data sources hosted in an on-premise or cloud

environment.



Colibra Platform Self-Hosted (CPSH) is a Colibra solution that allows you to install your Colibra Platform on an infrastructure of your choice. For Edge, this means that you are hosting both your Colibra platform and your Edge site. For more information about CPSH,

go to our [CPSH documentation](#).



- Edge sites always use REST API endpoints to establish connections.
- Edge requires access to a Colibra 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 Colibra Platform and data source credentials. This has the following consequences:

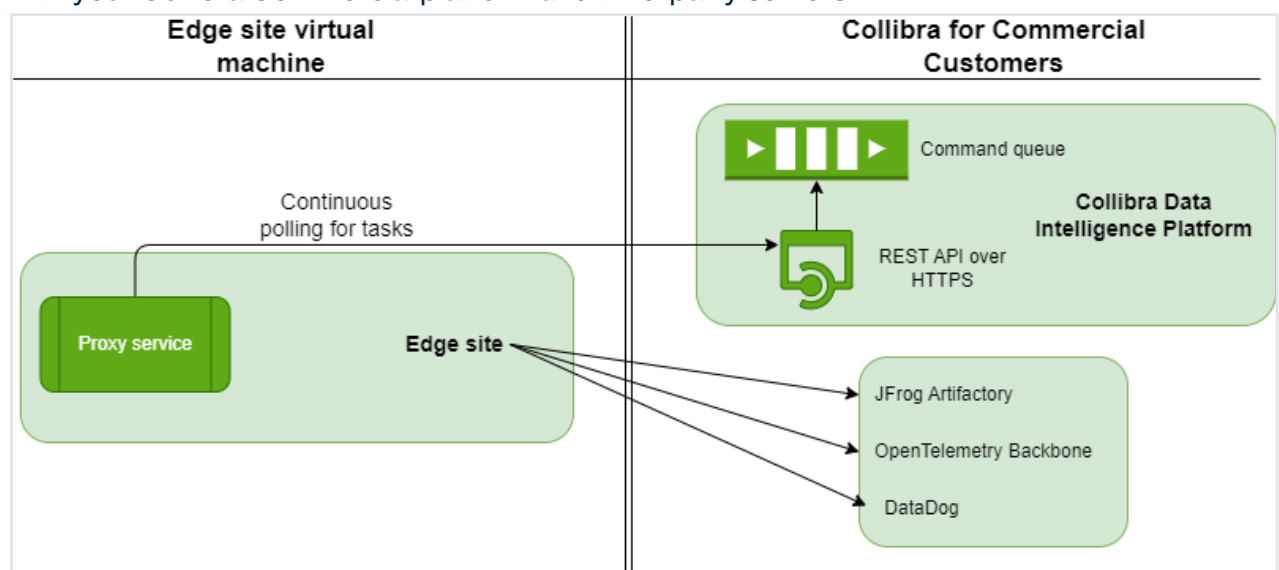
- Credentials are not accessible outside of Edge.
- Credentials used on an Edge site are encrypted with a key that is [secured](#) in Collibra.
- Credentials of data sources and Collibra can be updated if necessary.
- All configuration parameters, files or strings marked as secret, are stored on the 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.

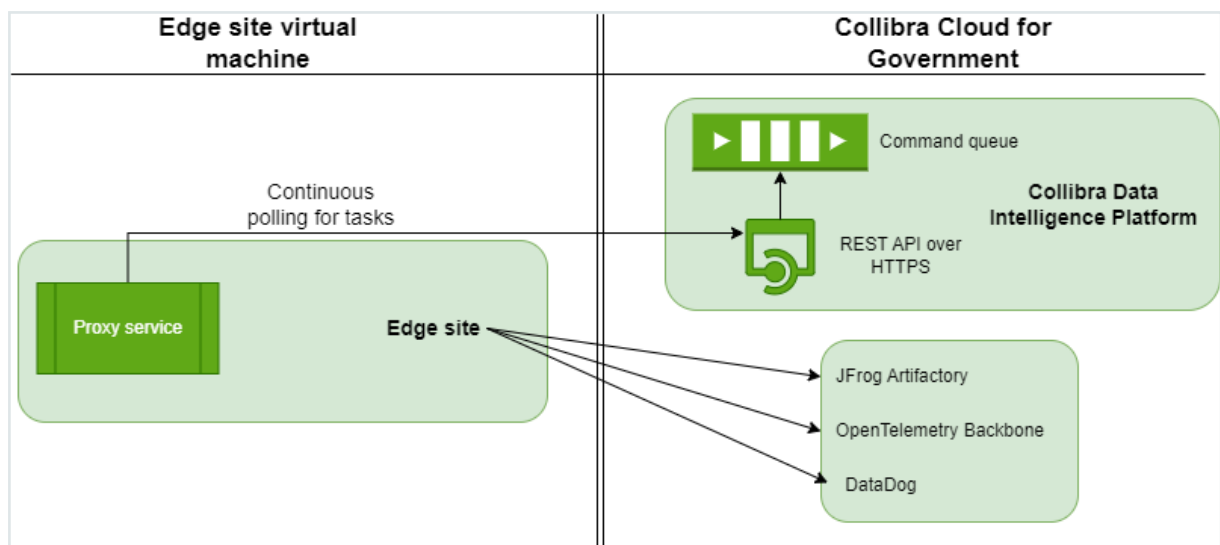
Edge for commercial customers is a Collibra solution that allows your Collibra Platform to safely connect to your data sources hosted in an on-premise or cloud environment.

The diagram below shows how your Edge site installed on your virtual machine connects with your Collibra Commercial platform and third-party servers.



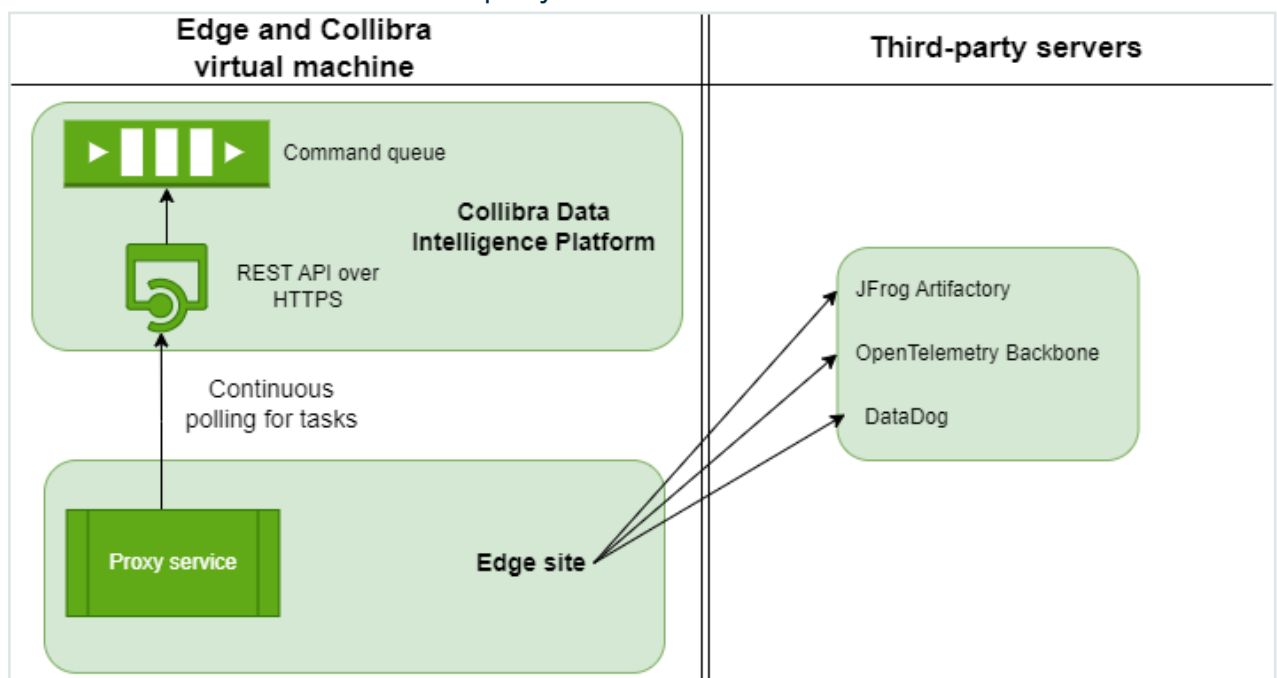
Edge for Collibra Platform for Government is a Collibra solution that allows your Collibra Platform to safely connect to your data sources hosted in an on-premise or cloud environment.

The diagram below shows how your Edge site installed on your virtual machine connects with your Collibra Platform for Government platform and third-party servers.



Collibra Platform Self-Hosted (CPSH) is a Collibra solution that allows you to install your Collibra Platform on an infrastructure of your choice. For Edge, this means that you are hosting both your Collibra platform and your Edge site. For more information about CPSH, go to our [CPSH documentation](#).

The diagram below shows how your Edge site and Collibra platform installed on your virtual machine connects with third-party servers.



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.	HTTPS.
DataDog	<p>This is needed in order to upload logs from all Edge components:</p> <ul style="list-style-type: none">• 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 Colibra composes an Edge installation package, Snyk scans all images consumed by Edge for all planned weekly releases to identify and mitigate vulnerabilities. Additional daily scans on repositories are also performed as well as a quarterly 3rd party penetration test to ensure that Edge remains secure.

You can also run your own security scans. We recommend that you run the following command for Edge sites installed on k3s 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.

For more information about security scanning, go to [Colibra's vulnerability and scanning policy](#).

What's next?

[Pull images from the Colibra 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 → **Settings**.
 - » The [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. `dip_user`: The username you use to log into Collibra Platform.
- b. `dip_pass`: The password you use to log into Collibra.
- c. `dip_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 <dip_user> -p <dgc_pass> <dip_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.

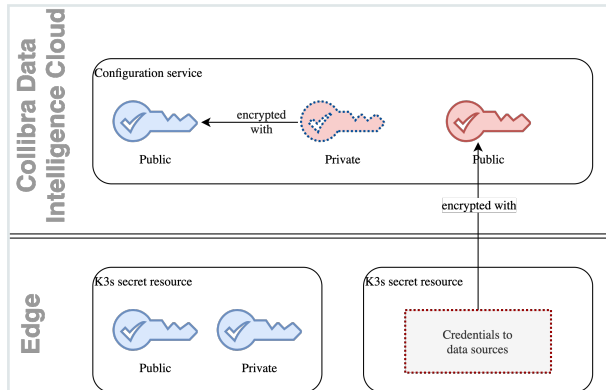
The Edge native encryption mechanism is based on two RSA key pairs. They are stored in the following places:

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 Platform 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`

Additionally, if you don't use vaults, k3s and k8s add another level of encryption using AES 256 to any data that is at rest and not currently communicating from one node to another. For more information, go to the Kubernetes documentation [Encrypting Confidential Data at Rest](#).

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 1.3 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 Collibra server 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. Additionally, Edge purges the oldest data from the cache every 24 hours or when the cache reaches 1 GB of data, whichever occurs first.

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 Platform 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.

You can edit **registries.yaml** and access the registry independently, and download images for Edge to scan them.

For more information about security scanning, go to [Collibra's vulnerability and scanning policy](#).

Monitoring and logging

We monitor and log all interaction between an Edge site and Collibra Platform, 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 must install `iptables-services` package and enable `iptables`.

```
yum install iptables-services
```

- 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

The following configuration file prevents access to the ports mentioned in the table and, with iptables, provides persistence in the event of Edge upgrades and reboots. Please check with your security team for compliance and for the tools used to filter the traffic before applying these commands.

```
*filter
:INPUT ACCEPT [0:0]
:edge hardening - [0:0]
-A INPUT -j edge hardening
-A edge hardening -m state --state RELATED,ESTABLISHED -j
ACCEPT
-A edge hardening -p tcp -m state --state NEW -m tcp --dport
22 -j ACCEPT
-A edge hardening -j ACCEPT -i lo -p tcp -m multiport --dports
6443,10250
-A edge hardening -j ACCEPT -i cni0 -p tcp -m multiport --
dports 6443,10250
-A edge hardening -j DROP -p tcp -m multiport --dports
6443,10250
COMMIT
```

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
```

6. Comment out any undesired restrictions in iptables.
7. Restart iptables.

About private registries with Edge

A private container image registry allows you to use your own infrastructure to perform security scans and audit container images consumed by your Edge site. Before you configure a private container registry, keep the following in mind:

- Switching to a private container image registry is only possible during installation. If you want to add a private container image registry for an existing Edge site, you need to reinstall your [Edge site](#) with your registry.
- All Edge site container images must live in the same registry.
- When copying images to your private container image registry, make sure only the domain name is updated when tagging the new images.

Note

- Custom Helm registries are not supported.
- 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 container image registry.
- Security scan reports are only accepted for supported Edge 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 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 and Colibra Data Intelligence Cloud](#).

Supported private container image registries

- [Amazon Elastic Container Registry](#)
- [Azure Container Registry](#)
- [Google Container Registry](#)
- [JFrog](#)
- [Nexus](#)

Note The above list shows the tested and supported private container image registries. If your private container image registry is not listed but uses user/pass authentication, you can attempt to install your Edge site with your registry. However, Colibra Support can't provide assistance for untested registries.

Configure an Edge site with an Amazon Elastic Container Registry

This topic explores how to configure your Edge site with an Amazon Elastic Container Registry with IAM Role based authentication for EKS.

This registry is only supported for Edge sites installed on the following Amazon managed Kubernetes clusters:

- EKS
- AWS Fargate using EKS

To [install your Edge site](#) on an Amazon managed Kubernetes cluster with an Amazon Elastic Container Registry using an IAM Role based authentication for EKS method, add the following flag to the installation command:

```
--registry-url <url_for_registry>
```

```
./edgecli install --registry-url 812518457384.dkr.ecr.eu-west-1.amazonaws.com
```

For more more information, go to [Amazon's ECR Images with Amazon EKS documentation](#).

Configure an Edge site with an Azure Container Registry

This topic explores how to configure your Edge site with a private Azure Container Registry.

We support the following [Azure Container Registry authentication methods](#):

- Access token
- IAM based authentication for AKS
 - This method is only available for Edge sites installed on an Azure managed Kubernetes cluster.
- Service Principal ID with associated secret

Access token

To [install your Edge site](#) on k3s with an Azure Container Registry using the Access token authentication method, add the following flags to the installation command:

```
-r registries.yaml
--registry-url <url_for_registry>
--registry-user <token_name>
--registry-pass <token_password>
```

```
sudo sh install-master.sh properties.yaml -r registries.yaml
--registry-url edge.azurecr.io
--registry-user azureEdge
--registry-pass azureEdge12
```

For more information, go to the [Azure Container Registry access token](#) documentation.

Service Principal ID with associated secret

To [install your Edge site](#) on k3s with an Azure Container Registry using the Service Principal ID with associated secret authentication method:

- Make sure the service principal has pull permissions from the Azure Container Registry.
- Add the following flags to the installation command:

```
-r registries.yaml
--registry-url <url_for_registry>
--registry-user <service_principal_id>
--registry-pass <service_principal_secret>
```

```
sudo sh install-master.sh properties.yaml -r registries.yaml
--registry-url edge.azurecr.io
--registry-user azureEdge
--registry-pass azureEdge12
```

Access token

To [install your Edge site](#) on managed Kubernetes cluster with an Azure Container Registry using the Access authentication method, add the following flags to the installation command:

```
--registry-url <url_for_registry>
--registry-user <token_name>
--registry-pass <token_password>
```

```
./edgecli install
--registry-url edge.azurecr.io
--registry-user azureEdge
--registry-pass azureEdge12
```

For more information, go to the [Azure Container Registry access token](#) documentation.

Service Principal ID with associated secret

To [install your Edge site](#) a managed Kubernetes cluster with an Azure Container Registry using the Service Principal ID with associated secret authentication method:

- Make sure the service principal has pull permissions from the Azure Container Registry.
- Add the following flags to the installation command:

```
-r registries.yaml
--registry-url <url_for_registry>
--registry-user <service_principal_id>
--registry-pass <service_principal_secret>
```

```
./edgecli install
--registry-url edge.azurecr.io
--registry-user azureEdge
--registry-pass azureEdge12
```

Azure IAM based authentication for AKS

Azure Container Registries that use the Azure IAM based authentication for AKS authentication method are only supported for Edge sites installed on an Azure managed Kubernetes cluster.

To [install your Edge site](#) on a Azure managed Kubernetes cluster with an Azure Container Registry using the Azure IAM based authentication for AKS authentication method, add the following flag to the installation command:

```
--registry-url <url_for_registry>
```

```
./edgecli install  
--registry-url edge.azurecr.io
```

For more information about Amazon's IAM role based authentication, go to the [Azure Container Registry](#) documentation.

Configure an Edge site with a Google Artifact Registry

This topic explores how to configure an Edge site with Google Artifact Registry.

We support the following Google Artifact Registry authentication methods:

- Service Account Key
- Workload Identity Federation for GKE
 - This method is only available for Edge sites installed on a GKE managed Kubernetes cluster.

Service Account Key

To [install your Edge site on k3s](#) with a Google Artifact Registry using the Service Account Key authentication method, add the following flags to the installation command:

```
-r registries.yaml
--registry-url <url_for_registry>
--registry-user _json_key
--registry-pass <path_to_json_key_file>
```

```
sudo sh install-master.sh properties.yaml -r registries.yaml
--registry-url https://europe-west1-
docker.pkg.dev/path/to/registry
--registry-user _json_key
--registry-pass /path/to/json_key_file.json
```

For more information, go to the [Google Artifact Registry service account](#) documentation.

Service Account Key

To [install your Edge site on managed Kubernetes cluster](#) with a Google Artifact Registry using the Service Account Key authentication method, add the following flags to the installation command:

```
--registry-url <url_for_registry>
--registry-user _json_key
# as a single lined version of what's in the json_key json
# file wrapped in single quotes OR you can simply pass the
# path to the json_key json file
--registry-pass <path_to_json_key_file> OR <json_key_in_
format_above>
```

```
./edgecli install
--registry-url https://europe-west1-
docker.pkg.dev/path/to/registry
--registry-user _json_key
--registry-pass /path/to/json_key_file.json
```

For more information, go to the [Google Artifact Registry service account](#) documentation.

Workload Identity Federation for GKE

Google Artifact Registries that use the Workload Identity Federation for GKE authentication method are only supported for Edge sites installed on a GKE managed Kubernetes cluster. We recommend using a service account with GAR access on the GKE node level, however, you can use IAM authentication that doesn't require a specific service account on the cluster to pull container images.

To [install your Edge site on a GKE managed Kubernetes cluster](#) with a Google Artifact Registry using the Workload Identity Federation for GKE authentication method, add the following flag to the installation command:

```
--registry-url <url_for_registry>
```

```
./edgecli install  
--registry-url https://europe-west1-  
docker.pkg.dev/path/to/registry
```

For more information about the Workload Identity Federation for GKE authentication method, go to Google's documentation:

- [How Workload Identity Federation for GKE works](#)
- [About Workload Identity Federation for GKE](#)
- [Service accounts for GKE](#)

Configure an Edge site with a private registry using user/pass authentication

This topic explores how to configure your Edge site with a supported private container image registry using the user/pass authentication method.

Edge supports the following private container registries using the user/pass authentication method:

- JFrog
- Nexus

Note The above list shows the tested and supported private container image registries. If your private container image registry is not listed but uses user/password authentication, you can attempt to install your Edge site with your registry. However, Collibra Support can't provide assistance for untested registries.

User/Pass authentication

To [install your Edge site](#) with a private container image registry using the user/pass authentication method, add the following flags to the installation command:

Note All user/pass authenticated container image registries must use HTTPS for all communication.

If you're installing your Edge site on k3s, you must store any certificates required to communicate to the registry in `/etc/ssl/certs/cert.crt` on the k3s virtual machine, and then restart k3s by running the following command:

```
sudo systemctl restart k3s
```

```
--registry-url <url_for_registry>  
--registry-user <username>  
--registry-pass <password>
```

```
--registry-url nexus.company.com/edge-registry  
--registry-user NexusEdge  
--registry-pass NexusEdge12
```

Edge Vaults

This feature is **available only** in the [latest UI](#).

Edge Vaults is an integration between your Edge site and your vault provider, which allows you to increase the security of your Edge site and data source information.

With the Edge Vaults integration:

- 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 Colibra. Managing your data source credentials only needs to be performed in your organization's vault.

About Edge Vaults

This feature is **available only** in the [latest UI](#).

Note The Vault integration is not available for Collibra Cloud sites.

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.




A vault provider is a third-party secret management service, which should already be implemented by your organization. Your vault provider will store your data source information behind different types of Vault Keys, such as queries or names. Each vault may have requirements or restrictions surrounding what and how this information is stored. We recommend you review your vault provider's documentation for any of these requirements.

Once you integrate your Edge site with your vault provider, you can create [Edge connections](#) which call to your vault to retrieve the data source information. You must enter the vault specific Vault Key for each data source property you need or want to pull into your connection. For example, if you want to pull a data source password into your Edge connection, and in your vault this data source password is stored by the secret name my-secret, then you would enter *my-secret* as the Vault Key for the password field.

Important If your data source connection requires a file to establish a secure connection, then the sensitive contents of the file must be encoded into Base64 and stored as a secret in your vault.

Edge supports the following vault integrations:

-  **CyberArk Vault**
 - Supported version:
 - CyberArk Central Credential Provider (CCP) : 8.0.0
-  **HashiCorp Vault**
 - Supported version:
 - HashiCorp Vault 1.19.x

- Support secret engines:
 - [Key Value V2 Secret Engine](#)
 - [Database Secret Engine](#)
-  [Azure Key Vault](#)
-  [AWS Secrets Manager](#)
-  [Google Secret Manager](#)

What's next?

- [Integrate your Edge site](#) to your vault provider.
- Learn how to set up an [Edge connection](#) with your vault.

Integrate your Edge site with your vault

This feature is **available only** in the [latest UI](#).

You can integrate an [Edge site](#) with your existing vault to more easily and securely manage your data source information and set up your [Edge site connections](#). In this topic, we review how to set up the integration between your Edge site and your existing vault.

Note You must already have an existing vault with a [supported vault provider](#) to use this feature. This topic does not review how to create or setup a vault.

For steps on how to integrate your Edgesite with your vault, see the [online version of this guide](#).

Edit vault integration configuration via Edge CLI

This feature is **available only** in the [latest UI](#).

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](#).

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the mTLSAllow-list authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update cyber tls <name> \  
  --desc <description> \  
  --appId <appID> \  
  --url <url>  
  --caPath <caPath> \  
  --certPath <certPath> \  
  --keyPath <keyPath>
```

```
./edgecli vault update cyber tls <name> \  
  --desc <description> \  
  --appId <appID> \  
  --url <url>  
  --caPath <caPath> \  
  --certPath <certPath> \  
  --keyPath <keyPath>
```

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 .
	<p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<appId> (required)	The application ID configured on the CyberArk server.
<url> (required)	The URL of your CyberArk Vault .
<caPath> (required)	The file containing the Certificate Authority. If you use a --caPath, it must be in the X.509 format (PEM encoded).
<certPath> (required)	The file containing the Client Certificate. If you use a --certPath, it must be in the X.509 format (PEM encoded).
<keyPath> (required)	The file containing the Client Private Key. If you use a --keyPath, it must be in the PKCS#8 format (PEM encoded).

Note When using the mTLS authentication method, you must always include the following three variables, even if you are only updating one variable, such as the name of the vault integration:

- caPath
- certPath
- keyPath

```
sudo ./edgecli vault update cyber tls "Edge CyberArk Vault
mTLS" \
  --appId "edge" \
  --caPath "./certs/ca.crt" \
  --certPath "./certs/aimws.crt" \
  --keyPath "./certs/aimws-pkcs8.key"
```

```
sudo ./edgecli vault update cyber allow-list <name> \
  --desc <description> \
  --appId <appId> \
  --url <url> \
  --caPath <caPath>
```

```
./edgecli vault update cyber allow-list <name> \
  --desc <description> \
  --appId <appId> \
  --url <url> \
  --caPath <caPath>
```

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 .
	<div> Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?. </div>
<description> (optional)	The description of the vault instance.
<appId> (required)	The application ID configured on the CyberArk server.
<url> (required)	The URL of your CyberArk Vault.
<caPath> (required)	The file containing the Certificate Authority. If you use a --caPath, it must be in the X.509 format (PEM encoded).

Note When using the allow-list authentication method, you only need to include the vault integration variable that you want to update.

```
sudo ./edgecli vault update cyber allow-list "Edge CyberArk
allowlist" \
```

```
--appId "edge" \
--caPath "./certs/ca.crt"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the Username and password TLS authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update hashicorp user-pass <name>/
--desc <description> \
--user <username> \
--pass <password> \
--caPath <caPath> \
--url <url>
```

```
./edgecli vault update hashicorp user-pass <name>/
--desc <description> \
--user <username> \
--pass <password> \
--caPath <caPath> \
--url <url>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.

Command	Description
<user> (required)	The username for your vault.
<pass> (required)	The password for your vault.
<caPath> (optional)	The file containing the Certificate Authority. If you use <code>--caPath</code> it must be in the PKCS#8 format .
<p>Note A <code>--caPath</code> file is optionally included for the creation of the authentication endpoint. It will not be required if the HTTP endpoint is used for the Username/Password authentication. The HTTP endpoint is used for the Username/Password authentication.</p>	
<url> (required)	The URL of the HashiCorp Vault.
<vaultNamespace> (optional)	A unique namespace in your vault.

```
sudo ./edgecli vault update hashicorp user-pass "Hasicorp vault
user-pass AuthN"/
  --user "my-edge-site" \
  --pass "EdgePass123" \
  --url "https://hashicorp-vault.edge.collibra.dev:8210/"
```

```
sudo ./edgecli vault update hashicorp tls <name>/
  --authName <authName>
  --desc <description> \
  --caPath <caPath> \
  --certPath <certPath> \
  --keyPath <keyPath> \
  --url <url>
```

```
./edgecli vault update hashicorp tls <name>/
--authName <authName>
--desc <description> \
--caPath <caPath> \
--certPath <certPath> \
--keyPath <keyPath> \
--url <url>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<user> (required)	The username for your vault.
<pass> (required)	The password for your vault.
<caPath> (optional)	<p>The file containing the Certificate Authority.</p> <p>If you use <code>--caPath</code> it must be in the PKCS#8 format.</p> <p>Note A <code>--caPath</code> file is optionally included for the creation of the authentication endpoint. It will not be required if the HTTP endpoint is used for the Username/Password authentication. The HTTP endpoint is used for the Username/Password authentication.</p>
<url> (required)	The URL of the HashiCorp Vault.
<vaultNamespace> (optional)	A unique namespace in your vault.

```
sudo ./edgecli vault update hashicorp tls "tls-vault-auth"/
--authName "tls-vault-auth" \
--certPath "~/hashicorp/vault/edge-site.crt" \
--keyPath "~/hashicorp/vault/edge-site.key" \
--url "https://hashicorp-vault.edge.collibra/"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the Managed Identity assigned to Azure VMService Principal SecretService Principal with PEM certificateService Principal with PFX certificate authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update azure managed-identity <name> \
--desc <description> \
--dnsSuffix <dnsSuffix>
```

```
./edgecli vault update azure managed-identity <name> \
--desc <description> \
--dnsSuffix <dnsSuffix>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.

Command	Description
<code><dnsSuffix></code>	<p>The data-plane endpoint for your vault.</p> <p>Note <code><dnsSuffix></code> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: <code>vault.usgovcloudapi.net</code>.</p> <p>Otherwise, Edge uses the default value: <code>.vault.azure.net</code>.</p>
<pre>sudo ./edgecli vault update azure managed-identity "Azure- managed-identity" \ --dnsSuffix "Azure-managed-identity.azure.net"</pre>	
<pre>sudo ./edgecli vault update azure sp-secret <name> \ --desc <description> \ --dnsSuffix <dnsSuffix> \ --tenantId <tenantId> \ --clientId <clientId> \ --clientSecret <clientSecret></pre>	
<pre>./edgecli vault update azure sp-secret <name> \ --desc <description> \ --dnsSuffix <dnsSuffix> \ --tenantId <tenantId> \ --clientId <clientId> \ --clientSecret <clientSecret></pre>	
Command	Description
<code><name> (required)</code>	<p>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 <code><name></code> parameter, go to Labels and Selectors.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>

Command	Description
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<dnsSuffix>	<p>The data-plane endpoint for your vault.</p> <p>Note <dnsSuffix> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: <code>vault.usgovcloudapi.net</code>.</p> <p>Otherwise, Edge uses the default value: <code>.vault.azure.net</code>.</p>
<tenantId> (required)	The unique identifier of the Azure AD instance that the Azure Key Vault belongs to.
<clientId> (required)	The identifier of the service principal client.
<clientSecret> (required)	The secret of the service principal client.

```
sudo ./edgecli vault update azure sp-secret "Azure-service-
principal" \
  --tenantId "165" \
  --clientId "AZ_22" \
  --clientSecret "Secret123"
```

```
sudo ./edgecli vault update azure sp-pem <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --certPath <certPath>
```

```
./edgecli vault update azure sp-pem <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --certPath <certPath>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<dnsSuffix>	<p>The data-plane endpoint for your vault.</p> <p>Note <dnsSuffix> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: <code>vault.usgovcloudapi.net</code>.</p> <p>Otherwise, Edge uses the default value: <code>.vault.azure.net</code>.</p>
<tenantId> (required)	The unique identifier of the Azure AD instance that the Azure Key Vault belongs to.
<clientId> (required)	The identifier of the service principal client.
<certPath> (required)	The file containing the Client Certificate.

```
sudo ./edgecli vault update azure sp-secret "Azure-service-
principal-PEM" \
```

```
--tenantId "165" \
--clientId "AZ_22" \
--certPath "~/azurekey/vault/edge-site.crt"
```

```
sudo ./edgecli vault update azure sp-pfx <name> \
--desc <description> \
--dnsSuffix <dnsSuffix> \
--tenantId <tenantId> \
--clientId <clientId> \
--certPath <certPath> \
--certPassword <certPassword>
```

```
./edgecli vault update azure sp-pfx <name> \
--desc <description> \
--dnsSuffix <dnsSuffix> \
--tenantId <tenantId> \
--clientId <clientId> \
--certPath <certPath> \
--certPassword <certPassword>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.

Command	Description
<code><dnsSuffix></code>	<p>The data-plane endpoint for your vault.</p> <div> <p>Note <code><dnsSuffix></code> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: <code>vault.usgovcloudapi.net</code>.</p> <p>Otherwise, Edge uses the default value: <code>.vault.azure.net</code>.</p> </div>
<code><tenantId></code> (required)	The unique identifier of the Azure AD instance that the Azure Key Vault belongs to.
<code><clientId></code> (required)	The identifier of the service principal client.
<code><certPath></code> (required)	The file containing the Client Certificate.
<code><certPassword></code> (required)	The password used to protect the PFX certificate.

```
sudo ./edgecli vault update azure sp-secret "Azure-service-
principal-PFX" \
  --tenantId "165" \
  --clientId "AZ_22" \
  --certPath "~/azurekey/vault/edge-site.crt" \
  --certPassword "AZ_PFX_password1"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the IAM Access KeyInstance ProfileAssume Role authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update aws key-secret <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride> \
  --accessKeyId <accessKeyId> \
  --accessKey <accessKey>
```

```
./edgecli vault update aws key-secret <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride> \
  --accessKeyId <accessKeyId> \
  --accessKey <accessKey>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	<p>The description of the vault instance. The maximum character length is 150.</p>
<region>	<p>The region of the AWS Secrets Manager you want to use.</p> <p>Note Region is optional if your Edge site and AWS Secrets Manager are both located in the same AWS region.</p> <p>Region is required if:</p> <ul style="list-style-type: none"> • You are using <endpointOverride>. • Your Edge site is on k3s and running in AWS. • Your Edge site is running in a different region than the AWS Secrets Manager you want to connect to.

Command	Description
<code><endpointOverride></code>	<p>The URL of the entry point for your AWS Secrets Manager vault.</p> <p>Note <code><endpointOverride></code> is required if you use a private version of AWS for security purposes. For example, for FIPS, you would need to specify the FIPS endpoint: <code><protocol>"://:/"<service-code>"-fips."<region>".amazonaws.com.</code></p> <p>Otherwise, Edge will use the default regional value: <code>"<protocol>"://"<service-code>".<region-code>".amazonaws.com</code></p>
<code><accessKeyId> (required)</code>	The ID of the IAM key you want to authenticate with.
<code><accessKey> (required)</code>	The IAM key you want to authenticate with.

```
sudo ./edgecli vault update aws key-secret "AWS-IAM" \
  --accessKeyId "1234" \
  --accessKey "abcd"
```

```
sudo ./edgecli vault update aws key-secret "AWS-IAM" \
  --accessKeyId "1234" \
  --accessKey "abcd"
```

```
sudo ./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride>
```

```
./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride>
```

Command	Description
<code><name></code> (required)	<p>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 <code><name></code> parameter, go to Labels and Selectors.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<code><description></code> (optional)	<p>The description of the vault instance. The maximum character length is 150.</p>
<code><region></code>	<p>The region of the AWS Secrets Manager you want to use.</p> <p>Note Region is optional if your Edge site and AWS Secrets Manager are both located in the same AWS region.</p> <p>Region is required if:</p> <ul style="list-style-type: none"> • You are using <code><endpointOverride></code>. • Your Edge site is on k3s and running in AWS. • Your Edge site is running in a different region than the AWS Secrets Manager you want to connect to.
<code><endpointOverride></code>	<p>The URL of the entry point for your AWS Secrets Manager.</p> <p>Note <code><endpointOverride></code> is required if you use a private version of AWS for security purposes. For example, for FIPS, you would need to specify the FIPS endpoint:</p> <pre><protocol>"://://"<service-code>"-fips."<region>".amazonaws.com.</pre> <p>Otherwise, Edge will use the default regional value:</p> <pre>"<protocol>":///"<service-code>". "<region-code>".amazonaws.com</pre>

```
sudo ./edgecli vault update aws instance-profile "AWS-Instance" \
  --desc "AWS vault with Instant Profile authentication" \
  --region "eu-west-1" \
  --endpointOverride "http://my-secret-vault.aws.com"
```



```
sudo ./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --roleArn <roleArn>"
  --roleSessionName <roleSessionName>
  --region <region>"\
  --endpointOverride <endpointOverride>
```

```
./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --roleArn <roleArn>"
  --roleSessionName <roleSessionName>
  --region <region>"\
  --endpointOverride <endpointOverride>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	<p>The description of the vault instance. The maximum character length is 150.</p>
<roleArn> (required)	<p>The Amazon Resource name of the role you want your Edge site to assume when accessing the AWS Secrets Manager secrets.</p> <p>Note In order for your Edge site to successfully assume this specified role, the Instance Profile role that is attached to the EKS cluster must be trusted by the target role.</p>
<roleSessionName> (optional)	<p>The name of the session you want this role to appear as in AWS security logs.</p>

Command	Description
<code><region></code>	<p>The region of the AWS Secrets Manager you want to use.</p> <p>Note Region is optional if your Edge site and AWS Secrets Manager are both located in the same AWS region.</p> <p>Region is required if:</p> <ul style="list-style-type: none"> • You are using <code><endpointOverride></code>. • Your Edge site is on k3s and running in AWS. • Your Edge site is running in a different region than the AWS Secrets Manager you want to connect to.
<code><endpointOverride></code>	<p>The URL of the entry point for your AWS Secrets Manager.</p> <p>Note <code><endpointOverride></code> is required if you use a private version of AWS for security purposes. For example, for FIPS, you would need to specify the FIPS endpoint:</p> <pre><protocol>"://:/"<service-code>"-fips."<region>".amazonaws.com.</pre> <p>Otherwise, Edge will use the default regional value:</p> <pre>"<protocol>"://"<service-code>". "<region-code>".amazonaws.com</pre>

```
sudo ./edgecli vault update aws instance-profile "AWS-Assume" \
  --roleArn "edge-session"
  --roleSessionName "edge-session"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the IAM Role assigned to the Google Cloud Engine VMService Account JSON KeyService Account P12 Key authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update gcp iam-role <name> \
  --desc <description> \
  --projectId <projectId>
```

```
./edgecli vault update gcp iam-role <name> \
  --desc <description> \
  --projectId <projectId>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<projectId> (required)	The ID of the Google Account project which contains the Google Secret Manager.

```
sudo ./edgecli vault update gcp iam-role "GCP-IAM" \
  --projectId="IAM_145" \
```

```
sudo ./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath>
```

```
./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath>
```

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 . <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<projectId> (required)	The ID of the Google Account project which contains the Google Secret Manager.
<keyPath> (required)	The file containing the Client Private Key.

```
sudo ./edgecli vault update gcp sa-jsone "GCP-JSON" \
  --projectId="JSON_145" \
  --keyPath="/GCP/vault/edge-site.json"
```

```
sudo ./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath> \
  --keyPassword <keyPassword> \
  --emailAddress <emailAddress>
```

```
./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath> \
  --keyPassword <keyPassword> \
  --emailAddress <emailAddress>
```

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 . <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<projectId> (required)	The ID of the Google Account project which contains the Google Secret Manager.
<keyPath> (required)	The file containing the Client Private Key.
<keyPassword>	The P12 password.
<emailAddress>	The Google Service Account email address.

```
sudo ./edgecli vault update aws instance-profile "GCP-P12" \
  --projectId "P12_145" \
  --keyPath "/GCP/vault/edge-site.p12" \
  --keyPassword "GCP_edge_vault_password" \
  --emailAddress "GCPedgeVault@gmail.com"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the mTLSAllow-list authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update cyber tls <name> \
  --desc <description> \
  --appId <appId> \
  --url <url> \
  --caPath <caPath> \
  --certPath <certPath> \
  --keyPath <keyPath>
```

```
./edgecli vault update cyber tls <name> \
  --desc <description> \
  --appId <appId> \
  --url <url> \
  --caPath <caPath> \
  --certPath <certPath> \
  --keyPath <keyPath>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<appId> (required)	The application ID configured on the CyberArk server.
<url> (required)	The URL of your CyberArk Vault.
<caPath> (required)	<p>The file containing the Certificate Authority.</p> <p>If you use a --caPath, it must be in the X.509 format (PEM encoded).</p>
<certPath> (required)	<p>The file containing the Client Certificate.</p> <p>If you use a --certPath, it must be in the X.509 format (PEM encoded).</p>
<keyPath> (required)	<p>The file containing the Client Private Key.</p> <p>If you use a --keyPath, it must be in the PKCS#8 format (PEM encoded).</p>

Note When using the mTLS authentication method, you must always include the following three variables, even if you are only updating one variable, such as the name of the vault integration:

- caPath
- certPath
- keyPath

```
sudo ./edgecli vault update cyber tls "Edge CyberArk Vault
mTLS" \
  --appId "edge" \
  --caPath "./certs/ca.crt" \
  --certPath "./certs/aimws.crt" \
  --keyPath "./certs/aimws-pkcs8.key"
```

```
sudo ./edgecli vault update cyber allow-list <name> \
  --desc <description> \
  --appId <appId> \
  --url <url> \
  --caPath <caPath>
```

```
./edgecli vault update cyber allow-list <name> \
  --desc <description> \
  --appId <appId> \
  --url <url> \
  --caPath <caPath>
```

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 The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.

Command	Description
<description> (optional)	The description of the vault instance.
<appId> (required)	The application ID configured on the CyberArk server.
<url> (required)	The URL of your CyberArk Vault .
<caPath> (required)	The file containing the Certificate Authority. If you use a <code>--caPath</code> , it must be in the X.509 format (PEM encoded).

Note When using the allow-list authentication method, you only need to include the vault integration variable that you want to update.

```
sudo ./edgecli vault update cyber allow-list "Edge CyberArk
allowlist" \
--appId "edge" \
--caPath "./certs/ca.crt"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the Username and password/TLS authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update hashicorp user-pass <name>/
--desc <description> \
--user <username> \
--pass <password> \
--caPath <caPath> \
--url <url>
```



```
./edgecli vault update hashicorp user-pass <name>/
--desc <description> \
--user <username> \
--pass <password> \
--caPath <caPath> \
--url <url>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<user> (required)	The username for your vault.
<pass> (required)	The password for your vault.
<caPath> (optional)	<p>The file containing the Certificate Authority.</p> <p>If you use <code>--caPath</code> it must be in the PKCS#8 format.</p> <p>Note A <code>--caPath</code> file is optionally included for the creation of the authentication endpoint. It will not be required if the HTTP endpoint is used for the Username/Password authentication. The HTTP endpoint is used for the Username/Password authentication.</p>
<url> (required)	The URL of the HashiCorp Vault.
<vaultNamespace> (optional)	A unique namespace in your vault.

```
sudo ./edgecli vault update hashicorp user-pass "Hasicorp vault
user-pass AuthN"/
  --user "my-edge-site" \
  --pass "EdgePass123" \
  --url "https://hashicorp-vault.edge.collibra.dev:8210/"
```

```
sudo ./edgecli vault update hashicorp tls <name>/
  --authName <authName>
  --desc <description> \
  --caPath <caPath> \
  --certPath <certPath> \
  --keyPath <keyPath> \
  --url <url>
```

```
./edgecli vault update hashicorp tls <name>/
  --authName <authName>
  --desc <description> \
  --caPath <caPath> \
  --certPath <certPath> \
  --keyPath <keyPath> \
  --url <url>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<user> (required)	The username for your vault.
<pass> (required)	The password for your vault.

Command	Description
<caPath> (optional)	<p>The file containing the Certificate Authority.</p> <p>If you use <code>--caPath</code> it must be in the PKCS#8 format.</p> <div> <p>Note A <code>--caPath</code> file is optionally included for the creation of the authentication endpoint. It will not be required if the HTTP endpoint is used for the Username/Password authentication. The HTTP endpoint is used for the Username/Password authentication.</p> </div>
<url> (required)	The URL of the HashiCorp Vault.
<vaultNamespace> (optional)	A unique namespace in your vault.

```
sudo ./edgecli vault update hashicorp tls "tls-vault-auth"/
--authName "tls-vault-auth" \
--certPath "~/hashicorp/vault/edge-site.crt" \
--keyPath "~/hashicorp/vault/edge-site.key" \
--url "https://hashicorp-vault.edge.collibra/"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the Managed Identity assigned to Azure VMService Principal SecretService Principal with PEM certificateService Principal with PFX certificate authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update azure managed-identity <name> \
--desc <description> \
--dnsSuffix <dnsSuffix>
```

```
./edgecli vault update azure managed-identity <name> \
--desc <description> \
--dnsSuffix <dnsSuffix>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<dnsSuffix>	<p>The data-plane endpoint for your vault.</p> <p>Note <dnsSuffix> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: <code>vault.usgovcloudapi.net</code>.</p> <p>Otherwise, Edge uses the default value: <code>.vault.azure.net</code>.</p>

```
sudo ./edgecli vault update azure managed-identity "Azure-
managed-identity" \
  --dnsSuffix "Azure-managed-identity.azure.net"
```

```
sudo ./edgecli vault update azure sp-secret <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --clientSecret <clientSecret>
```

```
./edgecli vault update azure sp-secret <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --clientSecret <clientSecret>
```

Command	Description
<code><name></code> (required)	<p>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 <code><name></code> parameter, go to Labels and Selectors.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<code><description></code> (optional)	The description of the vault instance. The maximum character length is 150.
<code><dnsSuffix></code>	<p>The data-plane endpoint for your vault.</p> <p>Note <code><dnsSuffix></code> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: <code>vault.usgovcloudapi.net</code>.</p> <p>Otherwise, Edge uses the default value: <code>.vault.azure.net</code>.</p>
<code><tenantId></code> (required)	The unique identifier of the Azure AD instance that the Azure Key Vault belongs to.
<code><clientId></code> (required)	The identifier of the service principal client.
<code><clientSecret></code> (required)	The secret of the service principal client.

```
sudo ./edgecli vault update azure sp-secret "Azure-service-
principal" \
  --tenantId "165" \
  --clientId "AZ_22" \
  --clientSecret "Secret123"
```

```
sudo ./edgecli vault update azure sp-pem <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --certPath <certPath>
```

```
./edgecli vault update azure sp-pem <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --certPath <certPath>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<dnsSuffix>	<p>The data-plane endpoint for your vault.</p> <p>Note <dnsSuffix> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: vault.usgovcloudapi.net.</p> <p>Otherwise, Edge uses the default value: .vault.azure.net.</p>
<tenantId> (required)	The unique identifier of the Azure AD instance that the Azure Key Vault belongs to.

Command	Description
<clientId> (required)	The identifier of the service principal client.
<certPath> (required)	The file containing the Client Certificate.

```
sudo ./edgecli vault update azure sp-secret "Azure-service-
principal-PEM" \
  --tenantId "165" \
  --clientId "AZ_22" \
  --certPath "~/azurekey/vault/edge-site.crt"
```

```
sudo ./edgecli vault update azure sp-pfx <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --certPath <certPath> \
  --certPassword <certPassword>
```

```
./edgecli vault update azure sp-pfx <name> \
  --desc <description> \
  --dnsSuffix <dnsSuffix> \
  --tenantId <tenantId> \
  --clientId <clientId> \
  --certPath <certPath> \
  --certPassword <certPassword>
```

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 The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.

Command	Description
<code><description></code> (optional)	The description of the vault instance. The maximum character length is 150.
<code><dnsSuffix></code>	The data-plane endpoint for your vault. <div> <p>Note <code><dnsSuffix></code> is required if you use a private version of Azure for security purposes. For example, for FedRAMP, you would need to specify the Azure US Government DNS suffix: <code>vault.usgovcloudapi.net</code>.</p> <p>Otherwise, Edge uses the default value: <code>.vault.azure.net</code>.</p> </div>
<code><tenantId></code> (required)	The unique identifier of the Azure AD instance that the Azure Key Vault belongs to.
<code><clientId></code> (required)	The identifier of the service principal client.
<code><certPath></code> (required)	The file containing the Client Certificate.
<code><certPassword></code> (required)	The password used to protect the PFX certificate.

```
sudo ./edgecli vault update azure sp-secret "Azure-service-
principal-PFX" \
  --tenantId "165" \
  --clientId "AZ_22" \
  --certPath "~/azurekey/vault/edge-site.crt" \
  --certPassword "AZ_PFX_password1"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the IAM Access KeyInstance ProfileAssume Role authentication method to inspect or update all or any of the vault configuration settings.


```
sudo ./edgecli vault update aws key-secret <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride> \
  --accessKeyId <accessKeyId> \
  --accessKey <accessKey>
```

```
./edgecli vault update aws key-secret <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride> \
  --accessKeyId <accessKeyId> \
  --accessKey <accessKey>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	<p>The description of the vault instance. The maximum character length is 150.</p>
<region>	<p>The region of the AWS Secrets Manager you want to use.</p> <p>Note Region is optional if your Edge site and AWS Secrets Manager are both located in the same AWS region.</p> <p>Region is required if:</p> <ul style="list-style-type: none"> • You are using <endpointOverride>. • Your Edge site is on k3s and running in AWS. • Your Edge site is running in a different region than the AWS Secrets Manager you want to connect to.

Command	Description
<endpointOverride>	<p>The URL of the entry point for your AWS Secrets Manager vault.</p> <div> <p>Note <endpointOverride> is required if you use a private version of AWS for security purposes. For example, for FIPS, you would need to specify the FIPS endpoint:</p> <pre><protocol>"://:/"<service-code>"-fips."<region>".amazonaws.com.</pre> <p>Otherwise, Edge will use the default regional value:</p> <pre>"<protocol>"://"<service-code>".<region-code>".amazonaws.com</pre> </div>
<accessKeyId> (required)	The ID of the IAM key you want to authenticate with.
<accessKey> (required)	The IAM key you want to authenticate with.

```
sudo ./edgecli vault update aws key-secret "AWS-IAM" \
  --accessKeyId "1234" \
  --accessKey "abcd"
```

```
sudo ./edgecli vault update aws key-secret "AWS-IAM" \
  --accessKeyId "1234" \
  --accessKey "abcd"
```

```
sudo ./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride>
```

```
./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --region <region> \
  --endpointOverride <endpointOverride>
```

Command	Description
<code><name></code> (required)	<p>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 <code><name></code> parameter, go to Labels and Selectors.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<code><description></code> (optional)	The description of the vault instance. The maximum character length is 150.
<code><region></code>	<p>The region of the AWS Secrets Manager you want to use.</p> <p>Note Region is optional if your Edge site and AWS Secrets Manager are both located in the same AWS region.</p> <p>Region is required if:</p> <ul style="list-style-type: none"> • You are using <code><endpointOverride></code>. • Your Edge site is on k3s and running in AWS. • Your Edge site is running in a different region than the AWS Secrets Manager you want to connect to.
<code><endpointOverride></code>	<p>The URL of the entry point for your AWS Secrets Manager.</p> <p>Note <code><endpointOverride></code> is required if you use a private version of AWS for security purposes. For example, for FIPS, you would need to specify the FIPS endpoint:</p> <pre><protocol>"://://"<service-code>"-fips."<region>".amazonaws.com.</pre> <p>Otherwise, Edge will use the default regional value:</p> <pre>"<protocol>":///"<service-code>".<region-code>".amazonaws.com</pre>

```
sudo ./edgecli vault update aws instance-profile "AWS-Instance" \
  --desc "AWS vault with Instant Profile authentication" \
  --region "eu-west-1" \
  --endpointOverride "http://my-secret-vault.aws.com"
```

```
sudo ./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --roleArn <roleArn>"
  --roleSessionName <roleSessionName>
  --region <region>"\
  --endpointOverride <endpointOverride>
```

```
./edgecli vault update aws instance-profile <name> \
  --desc <description> \
  --roleArn <roleArn>"
  --roleSessionName <roleSessionName>
  --region <region>"\
  --endpointOverride <endpointOverride>
```

Command	Description
<name> (required)	<p>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.</p> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p>
<description> (optional)	<p>The description of the vault instance. The maximum character length is 150.</p>
<roleArn> (required)	<p>The Amazon Resource name of the role you want your Edge site to assume when accessing the AWS Secrets Manager secrets.</p> <p>Note In order for your Edge site to successfully assume this specified role, the Instance Profile role that is attached to the EKS cluster must be trusted by the target role.</p>
<roleSessionName> (optional)	<p>The name of the session you want this role to appear as in AWS security logs.</p>

Command	Description
<code><region></code>	<p>The region of the AWS Secrets Manager you want to use.</p> <p>Note Region is optional if your Edge site and AWS Secrets Manager are both located in the same AWS region.</p> <p>Region is required if:</p> <ul style="list-style-type: none"> • You are using <code><endpointOverride></code>. • Your Edge site is on k3s and running in AWS. • Your Edge site is running in a different region than the AWS Secrets Manager you want to connect to.
<code><endpointOverride></code>	<p>The URL of the entry point for your AWS Secrets Manager.</p> <p>Note <code><endpointOverride></code> is required if you use a private version of AWS for security purposes. For example, for FIPS, you would need to specify the FIPS endpoint: <code><protocol>"://:/"<service-code>"-fips."<region>".amazonaws.com.</code></p> <p>Otherwise, Edge will use the default regional value: <code>"<protocol>"://"<service-code>". "<region-code>".amazonaws.com</code></p>

```
sudo ./edgecli vault update aws instance-profile "AWS-Assume" \
  --roleArn "edge-session"
  --roleSessionName "edge-session"
```

Steps

In the cluster where your Edge site is installed, use the Edge CLI tool to run the command for the IAM Role assigned to the Google Cloud Engine VMService Account JSON KeyService Account P12 Key authentication method to inspect or update all or any of the vault configuration settings.

```
sudo ./edgecli vault update gcp iam-role <name> \
  --desc <description> \
  --projectId <projectId>
```

```
./edgecli vault update gcp iam-role <name> \
  --desc <description> \
  --projectId <projectId>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<projectId> (required)	The ID of the Google Account project which contains the Google Secret Manager.

```
sudo ./edgecli vault update gcp iam-role "GCP-IAM" \
  --projectId="IAM_145" \
```

```
sudo ./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath>
```

```
./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath>
```

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 . <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<projectId> (required)	The ID of the Google Account project which contains the Google Secret Manager.
<keyPath> (required)	The file containing the Client Private Key.

```
sudo ./edgecli vault update gcp sa-jsone "GCP-JSON" \
  --projectId="JSON_145" \
  --keyPath=~/.GCP/vault/edge-site.json"
```

```
sudo ./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath> \
  --keyPassword <keyPassword> \
  --emailAddress <emailAddress>
```

```
./edgecli vault update gcp sa-json <name> \
  --desc <description> \
  --projectId <projectId> \
  --keyPath <keyPath> \
  --keyPassword <keyPassword> \
  --emailAddress <emailAddress>
```

Command	Description
<name> (required)	<p>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.</p> <div> <p>Note The name can only contain alphanumeric, dash (-), underscore (_), or period (.) characters. The name cannot include white spaces or special characters such as /, !, ?.</p> </div>
<description> (optional)	The description of the vault instance. The maximum character length is 150.
<projectId> (required)	The ID of the Google Account project which contains the Google Secret Manager.
<keyPath> (required)	The file containing the Client Private Key.
<keyPassword>	The P12 password.
<emailAddress>	The Google Service Account email address.

```
sudo ./edgecli vault update aws instance-profile "GCP-P12" \
  --projectId "P12_145" \
  --keyPath "/GCP/vault/edge-site.p12" \
  --keyPassword "GCP_edge_vault_password" \
  --emailAddress "GCPedgeVault@gmail.com"
```


Retrieve your vault integration information via the Edge CLI

This feature is **available only** in the [latest UI](#).

You can review the details of your [vault integrations](#) from the [Edge CLI tool](#).

Prerequisites

- Ensure that your environment uses the [latest user interface](#).
- 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:

```
sudo ./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:

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

Properties	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

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>
```

Properties	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 . <div> <p>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.</p> </div>

This command provides the following vault integration information:

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

Note Address is returned only for CyberArk Vault integrations.

- AuthType

How to access help for Vaults

This feature is **available only** in the [latest UI](#).

If you need any help with the vault command parameters, run one of the following commands in the Edge CLI, based on the Kubernetes cluster where your Edge site is installed:

- Bundled k3s installations:

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

- Managed Kubernetes installations:

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

Properties	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](#).

Delete an Edge site vault integration

This feature is **available only** in the [latest UI](#).

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.

Prerequisites

- Ensure that your environment uses the [latest user interface](#).
- 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](#):

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

Properties	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](#).

Steps

Run the following code in the [Edge CLI](#):

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

Properties	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 Platform, you have to install the Edge software on either the bundled k3s in the Edge site installer or one of the following managed Kubernetes clusters:

- Azure Kubernetes Service (AKS)
- AWS Fargate using EKS
- Amazon EKS (EKS)
- Google Kubernetes Engine (GKE)
- OpenShift

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.

Edge sites installed on a managed Kubernetes cluster after the 2024.05 release can be installed in one of the following ways:

- [Edge Command Line Interface](#) - our recommended installation method.
- Helm chart - a manual installation method. You need to deploy the Edge helm chart and all prerequisite resources required by Edge, such as configmaps and secrets. You may want to use this method if you are automating the installation process in your own CI/CD environment.

Important You should only use the Helm Chart installation method if you are familiar with helm and Kubernetes. As such, Collibra Support is limited and they cannot assist with custom helm and Kubernetes configurations.

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 run-time on the Kubernetes cluster, that executes capabilities close to your data but that is configurable from the Collibra Platform 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	<p>The name of the Edge site. Use a meaningful name, for example NetherlandsDataCentre1. Do not use spaces or special characters.</p> <p>This field is mandatory and the name must be globally unique.</p>
Status	<p>The status of the Edge site.</p> <p>The status is automatically shown when you create an Edge site.</p>
ID	<p>The unique ID of the Edge site, which is generated automatically when you create the Edge site.</p>
Description	<p>The description of the Edge site. We recommend to put at least basic location information of the Edge site.</p> <p>This field is mandatory.</p>
Installer and property files	<p>A section where you can download the installer and property files to install an Edge site on a server.</p> <p>This section is only visible when the Edge site has the status To be installed.</p>

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.

Note The default Edge CLI method is an easier solution for installing your Edge site via the Edge CLI. Edge creates the cluster level objects, such as namespaces and priority classes for you. This method can be used for both dedicated and shared clusters.

Note The restrictive Edge CLI method allows you or your company to create the cluster level objects, such as namespaces and priority classes, for your Edge site. This method may be required if your company has security requirements or process that do not allow Edge sites to create the cluster level objects for you. This method can be used for both dedicated and shared clusters.

Warning Collibra Support will not assist with custom Helm or Kubernetes configurations. The following steps are an example, and any assistance for configurations or issues outside of these steps is unsupported. We recommend using the Edge CLI method for managed Kubernetes installations.

A common example of custom Helm configurations is, but not limited to, using an unsupported private repository. At this time, we only support a [JFrog repository](#).

Software requirements

- You must be able to install the Edge software on one of the following supported versions of Red Hat Enterprise Linux (RHEL):
 - RHEL 8.8 or later (8.x).
 - RHEL 9.2 or later (9.x).

Note

- We recommend not installing Edge on [end-of-Life versions of RHEL](#).
- We recommend ensuring the [k3s version installed on your Edge site](#) can be run on the version of [RHEL](#) 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`).
- If you want SE Linux enabled, install the following policy packages before installing Edge:
 - `yum install -y container-selinux selinux-policy-base`
 - If you use RHEL 8:

```
yum install -y https://github.com/k3s-io/k3s-selinux/releases/download/v1.6.latest.1/k3s-selinux-1.6-1.el8.noarch.rpm
```

- If you use RHEL 9:

```
yum install -y https://github.com/k3s-io/k3s-selinux/releases/download/v1.6.latest.1/k3s-selinux-1.6-1.el9.noarch.rpm
```

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

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

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 50 GB of free storage on the partition that contains `/var/lib/rancher/k3s`. This partition is used for:

- K3s cluster configuration data.
- Docker images that are used by the k3s container runtime on the Edge site.

```
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.
- The partition mountpoint should not have the **noexec** option.

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.

- At least 5 GB of free storage on the partition that contains **/var/log**. This partition is used to:
 - Write k3s audit logs. Edge uses up to 1.1 GB of space to write and store these logs. Each log file can be up to 100 MB, and only the last 10 files within a 30-day period are retained.
 - Write pod logs. Edge uses up to 60 MB per container to write and store these logs. The number of containers depends on the workload.
- At least 200 GB of free storage on the partition that holds **/var/lib/kubelet**.
 - We recommend dedicating this storage on the **/var** partition if it exists. If it doesn't exist, you can dedicate this storage on the **/(root)** partition.
 - This partition is used by k3s to write ephemeral data related to kubernetes, using the hardcoded path **/var/lib/kubelet/pods/<containerId>/volumes/kubernetes.io~empty-dir/**.

Note If you have technical lineage capabilities, each concurrent execution of these capabilities requires 15 GB of space on `/var/lib/kubelet`. The number of technical lineage capabilities you can run concurrently depends on the available space on `/var/lib/kubelet`. If you need to run more technical lineage capabilities concurrently than you have space for, you can use the auto-scaling mechanism within the managed k8s platforms.

- 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 Platform 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.colibra.io`
- `https://edge-docker-delivery.repository.colibra.io`
- `https://mirror-docker.repository.colibra.io`

- `https://otlp-http.observability.colibra.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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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 Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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
```

Note For the network requirements specific to [creating a technical lineage](#), go to [Lineage harvester system requirements](#).

Whats next

- Create an Edge site in Collibra Platform.
- [Install an Edge site](#) and learn more about which [upgrade method](#) you should select for your Edge site.
- Optionally, you can [configure your own private docker registry](#).
- Optionally, you can set up a [Vault integration](#).
- [Create an Edge site connection](#).
- [Create an Edge site capability](#).

EKS requirements

You can install the Edge software on managed Kubernetes clusters.

- AWS EKS 1.27, 1.28, 1.29, 1.30, 1.31, and 1.32 are supported for new and existing Edge sites.
- EKS cluster has [IRSA enabled](#).

- Set up security groups to ensure that worker nodes can communicate with each other on non-privileged ports.

Software requirements

- A Linux server for x86_64 architecture where bash is available. This is the server from which you install the Edge software on EKS.

Tip This server will also contain the Edge tools.

- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS AKSAWS Fargate using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
 - A user/service account with a role scoped to the collibra-edge namespace.
 - The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
```

```

metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
    namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role

```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster.
 - The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
 - You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- The kubeconfig environment variable must be set to a valid kubeconfig file that contains the following:

- A user/service account with a role scoped to the collibra-edge namespace.
- The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
    namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role
```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS AKSAWS Fargate using EKSOpenShiftGKE cluster.
- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- [Helm](#) (v3).
- You must have [jq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS AKSAWS Fargate using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- 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 that is running a Linux-based operating system. 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, logs, and ephemeral cluster data.
- We recommend you have at least 2 worker nodes in the EKS cluster.

Note For more information about Linux OS for EKS clusters, go to the Amazon documentation about [Amazon EKS optimized AMIs](#). As Edge sites are only compatible with Linux OS, disregard the Windows AMI option in this resource.

Network requirements

Commercial

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

FedRAMP

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

GKE requirements

You can install the Edge software on managed Kubernetes clusters.

- GKE 1.27, 1.28, 1.29, 1.30, 1.31, and 1.32 are supported for new Edge sites.

Note You can migrate an existing k3s or EKS Edge site to a new managed Kubernetes cluster by following the [Managed Kubernetes reinstallation steps](#) using the Edge CLI method. You can't migrate from an existing Edge site to a new cluster using the Helm chart method.

Software requirements

- A Linux server for x86_64 architecture where bash is available.. This is the server from which you install the Edge software on GKE.

Tip This server will also contain the Edge tools.

- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
 - A user/service account with a role scoped to the collibra-edge namespace.
 - The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
```

```

- apiGroups: ["*"]
  resources: ["*"]
  verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
- kind: User
  name: username> # The user that will perform the
  namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role

```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster.
 - The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
 - You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- The kubeconfig environment variable must be set to a valid kubeconfig file that contains the following:

- A user/service account with a role scoped to the collibra-edge namespace.
- The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
    namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role
```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster.
- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- [Helm](#) (v3).
- You must have [jq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- Ensure your Kubectl client is compatible with the relevant GKE version.

Hardware requirements

You need an operational GKE 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, 2 worker nodes each with 8 core CPU and 32 GB or 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, logs, and ephemeral cluster data.
- We recommend you have at least 2 worker nodes in the GKE cluster.

Note At this time, Edge site installations on GKE clusters are only compatible with nodes running Linux-based operating systems. For more information about the currently supported Linux OS for GKE clusters, go to the Google documentation about [Node images](#).

Network requirements

Commercial

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

FedRAMP

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

AWS Fargate using EKS requirements

You can install the Edge software on managed Kubernetes clusters.

- AWS Fargate using EKS on Kubernetes 1.27, 1.28, 1.29, 1.30, 1.31, and 1.32 are supported for new Edge sites.

Note You can migrate an existing k3s or EKS Edge site to a new managed Kubernetes cluster by following the [Managed Kubernetes reinstallation steps](#) using the Edge CLI method. You can't migrate from an existing Edge site to a new cluster using the Helm chart method.

- EKS cluster has [IRSA enabled](#)
- You must create an [AWS Fargate profile](#) for your cluster with the following namespace selectors:
 - kube-system
 - default
 - collibra-*
 - edge-kube-installer
- EKS cluster has CoreDNS enabled and running on a Fargate Node(s).

Software requirements

- A Linux server for x86_64 architecture where bash is available. This is the server from which you install the Edge software on AWS Fargate using EKS.

Tip This server will also contain the Edge tools.

- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/AKS/AWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- A user/service account with a role scoped to the collibra-edge namespace.
- The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
    namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role
```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargate using EKSOpenShiftGKE cluster.
- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargate using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
 - A user/service account with a role scoped to the collibra-edge namespace.
 - The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
```

```

name: username> # The user that will perform the
installation
namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role

```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in

- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
 - You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster.
 - The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
 - [Helm](#) (v3).
 - You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
 - You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge
- Note** The only thing that should be running inside of the dedicated namespace in
- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
 - Ensure your Kubectl client is compatible with the relevant EKS version.

Network requirements

Commercial

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

OpenShift requirements

You can install the Edge software on managed Kubernetes clusters.

- OpenShift 4.14, 4.15, 4.16, 4.17, and 4.18 are supported for new Edge sites.

Note You can migrate an existing k3s or EKS Edge site to a new managed Kubernetes cluster by following the [Managed Kubernetes reinstallation steps](#) using the Edge CLI method. You can't migrate from an existing Edge site to a new cluster using the Helm chart method.

Software requirements

- A Linux server for x86_64 architecture where bash is available. This is the server from which you install the Edge software on OpenShift.

Tip This server will also contain the Edge tools.

- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
 - A user/service account with a role scoped to the collibra-edge namespace.
 - The rules within the role must at minimum be set to "*".

Note You need to set each rule's value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
  namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role

```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster.
 - The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
 - You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- The kubeconfig environment variable must be set to a valid kubeconfig file that contains the following:
 - A user/service account with a role scoped to the collibra-edge namespace.
 - The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
    namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role
```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to

create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster.
- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- Ensure your Kubectl client is compatible with the relevant OpenShift version.

Hardware requirements

You need an operational OpenShift 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, logs, and ephemeral cluster data.
- We recommend you have at least 2 worker nodes in the OpenShift cluster.

Note At this time, Edge site installations on OpenShift clusters are only compatible with nodes running Linux-based operating systems. For more information about the currently supported Linux OS for OpenShift clusters, go to the OpenShift documentation.

Network requirements

Commercial

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

FedRAMP

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

AKS requirements

You can install the Edge software on managed Kubernetes clusters.

- AKS 1.27, 1.28, 1.29, 1.30, 1.31, and 1.32 are supported for new Edge sites.

Note You can migrate an existing k3s or EKS Edge site to a new managed Kubernetes cluster by following the [Managed Kubernetes reinstallation steps](#) using the Edge CLI method. You can't migrate from an existing Edge site to a new cluster using the Helm chart method.

Software requirements

- A Linux server for x86_64 architecture where bash is available. This is the server from which you install the Edge software on AKS.

Tip This server will also contain the Edge tools.

- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
 - A user/service account with a role scoped to the collibra-edge namespace.
 - The rules within the role must at minimum be set to "*".

Note You need to set each rule's value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```

apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
  namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role

```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster.
 - The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
 - You must have a kubeconfig file with plain cluster_admin kubectl access to the EKSAKSAWS Fargae using EKSOpenShiftGKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- The kubeconfig environment variable must be set to a valid kubeconfig file that contains the following:
 - A user/service account with a role scoped to the collibra-edge namespace.
 - The rules within the role must at minimum be set to "*".

Note You need to set each rules' value to "*" because the apiVersions and resources rules can change or be deprecated at any point within Kubernetes. Setting these values to "*" ensures that your Edge site remains compatible with the latest versions of Kubernetes. If the role has stricter permissions, your site may experience breaking changes that will require reinstallation.

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-namespace-role
  namespace: collibra-edge
rules:
  - apiGroups: ["*"]
    resources: ["*"]
    verbs: ["*"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-namespace-rb
  namespace: collibra-edge
subjects:
  - kind: User
    name: username> # The user that will perform the
installation
    namespace: collibra-edge
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: edge-namespace-role
```

- [Helm](#) (v3).
- You must have [yq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKSOpenShiftGKE cluster. This kubeconfig file is used to

create the Custom Resource Definitions (CRDs) and namespace required for the Edge

Note The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.

- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster.
- The kubeconfig environment variable must be set to a kubeconfig that has plain cluster_admin kubectl access to the cluster.
- [Helm](#) (v3).
- You must have [jq](#) version 4.18.1 or later, and [jq](#) installed on your Linux machine.
- You must have a kubeconfig file with plain cluster_admin kubectl access to the EKS/SAWS Fargate using EKS/OpenShift/GKE cluster. This kubeconfig file is used to create the Custom Resource Definitions (CRDs) and namespace required for the Edge

- **Note** The only thing that should be running inside of the dedicated namespace in the shared cluster is the Edge site. We do not support running third-party components, such as service mesh, inside of the Edge site's dedicated namespace.
- Ensure your Kubectl client is compatible with the relevant AKS version.

Hardware requirements

You need an operational AKS 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, 2 worker nodes each with 8 core CPU and 32 GB or 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, logs, and ephemeral cluster data.
- We recommend you have at least 2 worker nodes in the AKS cluster.

Note At this time, Edge site installations on AKS clusters are only compatible with nodes running Linux-based operating systems. For more information about the currently supported Linux OS for AKS clusters, go to the Azure documentation about [Azure Kubernetes core concepts](#).

Network requirements

Commercial

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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.

FedRAMP

- An Edge site needs outbound connections to all of the following:
 - The URL of your Collibra Platform 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.
- Set the Linux system value for IP forwarding to 1: `net.ipv4.ip_forward=1`

Note If IP forwarding is turned off (`net.ipv4.ip_forward=0`), your Edge site may become unhealthy. Follow the steps in this [Support article](#) to turn IP forwarding on.

- 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 truststore. 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 For the network requirements specific to [creating a technical lineage](#), go to [Lineage harvester system requirements](#).

Whats next

- Create an Edge site in Collibra Platform.
- [Install an Edge site](#) and learn more about which [upgrade method](#) you should select for your Edge site.
- Optionally, you can [configure your own private docker registry](#).
- Optionally, you can set up a [Vault integration](#).
- [Create an Edge site connection](#).
- [Create an Edge site capability](#).

Create an Edge site

As jobs are run on an Edge site, rather than on the Colibra 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 **Manage Edge sites** [global permission](#).
- Your server meets all [system requirements](#).
- You have [enabled](#) database registration via Edge in Colibra Console.

Note You must restart the Colibra Platform service when you have enabled this option.

Steps

1. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
2. Click **Edge**.
 - » The Edge sites overview opens.
3. Above the table, to the right, click **Create Site**.
 - » The **Create Edge site** wizard starts.
4. Enter the required information.

Field	Description
Site 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 Site**.
 - » 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 Platform, 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 [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](#).
- You will install your Edge site on a [supported Kubernetes cluster](#).

Steps

1. Download the installer:
 - a. Open a site.
 - i. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - ii. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - iii. In the site overview, click the name of a site.
 - » The site page appears.

- b. Click **Download Installer**.

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. From inside the extracted TGZ archive directory, run the k3s installer script, including any additional flags you may need to configure. For example, if you want to configure a forward proxy or use a private docker registry for your Edge site.

Important

- If the Edge site has to connect via a forward HTTP proxy, then first [configure the forward proxy](#) before executing the installation.

```
sudo sh install-master.sh properties.yaml -r  
registries.yaml
```

Flag	Description
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script. <div><p>Note If your proxy properties are not in the default proxy.properties file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p><pre>--proxy temp/proxy/proxyproperties</pre></div>

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div data-bbox="748 508 1339 651"> <p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p> </div> <div data-bbox="798 719 1085 757"> <pre>--ca temp/certs</pre> </div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>sudo ./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none"> Edge and the data source connect using the data source certificate. Edge communicates the data source metadata to your Collibra Platform using a Collibra certified certificate.

Flag	Description
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker- delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>edge-docker-delivery.my- registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id <user_id></code>	<p>If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.</p>
<code>--group-id <group_id></code>	<p>If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.</p>

» 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
```

Note The default Edge CLI method is an easier solution for installing your Edge site via the Edge CLI. Edge creates the cluster level objects, such as namespaces and priority classes for you. This method can be used for both dedicated and shared clusters.

Note The restrictive Edge CLI method allows you or your company to create the cluster level objects, such as namespaces and priority classes, for your Edge site. This method may be required if your company has security requirements or process that do not allow Edge sites to create the cluster level objects for you. This method can be used for both dedicated and shared clusters.

Warning Collibra Support will not assist with custom Helm or Kubernetes configurations. The following steps are an example, and any assistance for configurations or issues outside of these steps is unsupported. We recommend using the Edge CLI method for managed Kubernetes installations.

A common example of custom Helm configurations is, but not limited to, using an unsupported private repository. At this time, we only support a [JFrog repository](#).

Prerequisites

- You have a [global role](#) with the Install Edge sites and the User Administration [global permission](#), for example Edge site administrator.

- 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](#).
- You will install your Edge site on a [supported Kubernetes cluster](#).
- You must have namespace level access to the Kubernetes cluster where you want to install your Edge site.
- You must have admin level access to your the Kubernetes cluster where you want to install your Edge site.
- You must have admin privileges to create the collibra-edge namespace, priority classes, and CRD's when executing the install script.
- You must run the following commands on a virtual machine where `yq` version 4.18.1 or later, and `jq` can be executed.

Steps

1. Download the installer:
 - a. Open a site.
 - i. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - ii. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - iii. In the site overview, click the name of a site.
 - » The site page appears.
 - b. Click **Download Installer**.

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. Set the `EDGE_INSTALLER_PATH` environment variable to the path of the root of the extracted installer.
 - a. Go to the extracted installer and run `pwd`. The result should look similar to this:

```
/path/to/installer/installer-111e8a59-b842-4f57-970c-32aa72000598
```

- b. Set the environment variable to the result:

```
export EDGE_INSTALLER_PATH=/path/to/installer/installer-111e8a59-b842-4f57-970c-32aa72000598
```

4. Run the following command to confirm that the [Kubeconfig](#) environment variable has been set to a valid kubeconfig:

```
echo $KUBECONFIG
```

5. Deploy cluster level objects:

- Create the namespace for collibra-edge.

Note Clusters that have more than one Edge site installed must have unique namespaces.

- i. Copy the following command, replacing `<my-namespace>` with a unique name for the namespace:

```
apiVersion: v1
kind: Namespace
metadata:
  labels:
    pod-security.kubernetes.io/enforce: baseline
    pod-security.kubernetes.io/enforce-version:
v1.27
name: <my-namespace>
```

- ii. Store this copied yaml into a new file called `collibra-edge-ns.yaml`.
- iii. Create the namespace using `kubectl`:

```
kubectl apply -f collibra-edge-ns.yaml <my-namespace>
```

Note Throughout the remaining installation steps, replace `<my-namespace>` in the provided commands with this new namespace name. Example commands will have `edge-namespace` as an example namespace name.

- If you are using an Openshift cluster, deploy Security Context Constraints (SCC) which provide Edge service accounts with the required permissions.
 - i. Create the SCC file, for example, `edge-scc-minimal.yaml`, and paste the following information into it:

```

apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: edge-scc-documented
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: []
allowedUnsafeSysctls: []
defaultAddCapabilities: []
fsGroup:
  type: MustRunAs
priority: null
readOnlyRootFilesystem: true
requiredDropCapabilities: []
runAsUser:
  type: MustRunAs
  uid: 1000
seLinuxContext:
  type: MustRunAs
  seLinuxOptions:
    level: "s0"
    role: "system_r"
    type: "container_t"
    user: "system_u"
seccompProfiles:
  - 'runtime/default'
supplementalGroups:
  type: MustRunAs
# who can use it
users: []
groups: [system:authenticated]

```

- ii. Create the SCC-role file, for example, `edge-scc-role-minimal.yaml`, and paste the following information into it:

```
# File: edge-scc-role-minimal.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-scc-minimal
rules:
- apiGroups:
  - security.openshift.io
  resourceNames:
  - edge-scc-minimal
  resources:
  - securitycontextconstraints
  verbs:
  - use
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-scc-minimal
subjects:
- kind: Group
  name: system:serviceaccounts
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: edge-scc-minimal
  apiGroup: rbac.authorization.k8s.io
```

iii. Deploy both the SCC and SCC-role files:

```
kubectl apply -f edge-scc-minimal.yaml
kubectl apply -f edge-scc-role-minimal.yaml -n
<my-namespace>
```

- For all cluster types, deploy priority classes:

```
kubectl apply -f resources/custom/priorityclass.yaml
```

6. Run one of the following installation commands on the machine that has the Kubernetes cluster connection:

Note

- You can install your Edge site with either terminal logging or terminal and file logging. Both options log the output of your Edge site installation.
 - Terminal logging only saves the output to the Edge terminal.
 - Terminal and file logging saves the output both to the terminal and a separate file. This file will be saved in the current directory with the naming format: **edge-installer-\$(date +%Y-%m-%d_%H-%M-%S).log**

```
./edgecli install -n <my-namespace>
```

Additional installation flags:

Flag	Description
<code>-n <my-namespace></code>	If you created a custom namespace, add <code>-n <my-namespace></code> to the command. For example:
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy
temp/proxy/proxyproperties
```

o

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div> <p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p> <pre>--ca temp/certs</pre> </div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none"> Edge and the data source connect using the data source certificate. Edge communicates the data source metadata to your Collibra Platform using a Collibra certified certificate.
<code>--is-openshift</code>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.

Flag	Description
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</div> <p>If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> ■ <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre> --set global.priorityClassName.platform=critical-priority --set global.priorityClassName.application=high-priority --set global.priorityClassName.job=low-priority </pre> </div>

```
./edgecli install -n edge-namespace
--is-openshift
--registry-url https://private-docker.registry.com
--registry-user user1
--registry-pass pass12
```

```
./edgecli install -n <my-namespace> 2>&1 | tee
"edge-installer-$(date +"%Y-%m-%d_%H-%M-%S").log"
```

Additional installation flags:

Flag	Description
-n <my-namespace>	If you created a custom namespace, add -n <my-namespace> to the command. For example:
--proxy	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy
temp/proxy/proxyproperties
```

o

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div> <p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p> <pre>--ca temp/certs</pre> </div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none"> Edge and the data source connect using the data source certificate. Edge communicates the data source metadata to your Colibra Platform using a Colibra certified certificate.
<code>--is-openshift</code>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.

Flag	Description
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</div> <p>If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> ■ <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre> --set global.priorityClassName.platform=critical-priority --set global.priorityClassName.application=high-priority --set global.priorityClassName.job=low-priority </pre> </div>


```
./edgecli install -n edge-namespace
--is-openshift
--registry-url https://private-docker.registry.com
--registry-user user1
--registry-pass pass12
2>&1 | tee "edge-installer-$(date
+ "%Y-%m-%d_%H-%M-%S").log"
```

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

7. Run the following command to verify the status of the installation.

```
kubectl get pods -n <my-namespace>
```

1. Download the installer:

- a. Open a site.

- i. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
- ii. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
- iii. In the site overview, click the name of a site.
 - » The site page appears.

- b. Click **Download Installer**.

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 -xzf <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. Set the `EDGE_INSTALLER_PATH` environment variable to the path of the root of the extracted installer.
 - a. Go to the extracted installer and run `pwd`. The result should look similar to this:

```
/path/to/installer/installer-111e8a59-b842-4f57-970c-32aa72000598
```

- b. Set the environment variable to the result:

```
export EDGE_INSTALLER_PATH=/path/to/installer/installer-111e8a59-b842-4f57-970c-32aa72000598
```

4. Run the following command to confirm that the [Kubeconfig environment variable](#) has been set to a valid kubeconfig:

```
echo $KUBECONFIG
```

5. If you intend to have multiple Edge sites in your Kubernetes cluster, you must give each Edge site a unique namespace.

- a. Copy the following command, replacing `<my-namespace>` with a unique name for the namespace:

```
apiVersion: v1
kind: Namespace
metadata:
  labels:
    pod-security.kubernetes.io/enforce: baseline
    pod-security.kubernetes.io/enforce-version:
v1.27
name: <my-namespace>
```

- b. Store this copied yaml into a new file called `collibra-edge-ns.yaml`.
- c. Create the namespace using `kubectl`:

```
kubectl apply -f collibra-edge-ns.yaml <my-namespace>
```

Note Throughout the remaining installation steps, add this new namespace to the provided commands.

6. If you are using an Openshift cluster, deploy Security Context Constraints (SCC) which provide Edge service accounts with the required permissions.
 - a. Create the SCC file, for example, `edge-scc-minimal.yaml`, and paste the following information into it:

```

apiVersion: security.openshift.io/v1
kind: SecurityContextConstraints
metadata:
  name: edge-scc-documented
allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: false
allowPrivilegedContainer: false
allowedCapabilities: []
allowedUnsafeSysctls: []
defaultAddCapabilities: []
fsGroup:
  type: MustRunAs
priority: null
readOnlyRootFilesystem: true
requiredDropCapabilities: []
runAsUser:
  type: MustRunAs
  uid: 1000
seLinuxContext:
  type: MustRunAs
  seLinuxOptions:
    level: "s0"
    role: "system_r"
    type: "container_t"
    user: "system_u"
seccompProfiles:
  - 'runtime/default'
supplementalGroups:
  type: MustRunAs
# who can use it
users: []
groups: [system:authenticated]

```

- b. Create the SCC-role file, for example, `edge-scc-role-minimal.yaml`, and paste the following information into it:

```
# File: edge-scc-role-minimal.yaml
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: edge-scc-minimal
rules:
- apiGroups:
  - security.openshift.io
  resourceNames:
  - edge-scc-minimal
  resources:
  - securitycontextconstraints
  verbs:
  - use
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: edge-scc-minimal
subjects:
- kind: Group
  name: system:serviceaccounts
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: edge-scc-minimal
  apiGroup: rbac.authorization.k8s.io
```

c. Deploy both the SCC and SCC-role files:

```
kubectl apply -f edge-scc-minimal.yaml
kubectl apply -f edge-scc-role-minimal.yaml -n <my-namespace>
```

7. Run one of the following installation commands on the machine that has the Kubernetes cluster connection:

Note

- You can install your Edge site with either terminal logging or terminal and file logging. Both options log the output of your Edge site installation.
 - Terminal logging only saves the output to the Edge terminal.
 - Terminal and file logging saves the output both to the terminal and a separate file. This file will be saved in the current directory with the naming format: **edge-installer-\$(date +%Y-%m-%d_%H-%M-%S).log**

```
./edgecli install
```

Flag	Description
<code>-n <my-namespace></code>	If you created a custom namespace, add <code>-n <my-namespace></code> to the command. For example:
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy
temp/proxy/proxyproperties
```

◦

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div> <p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p> <pre>--ca temp/certs</pre> </div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none"> Edge and the data source connect using the data source certificate. Edge communicates the data source metadata to your Colibra Platform using a Colibra certified certificate.
<code>--is-openshift</code>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.

Flag	Description
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</div> <p>If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> ■ <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre> --set global.priorityClassName.platform=critical-priority --set global.priorityClassName.application=high-priority --set global.priorityClassName.job=low-priority </pre> </div>

```
./edgecli install
--is-openshift
--registry-url https://private-docker.registry.com
--registry-user user1
--registry-pass pass12
```

```
./edgecli install 2>&1 | tee "edge-installer-$(date
+"%Y-%m-%d_%H-%M-%S").log"
```

Flag	Description
<code>-n <my-namespace></code>	If you created a custom namespace, add <code>-n <my-namespace></code> to the command. For example:
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

```
./edgecli install -n <my-namespace>
```

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path to the installation prerequisite script. For example:

```
--proxy
temp/proxy/proxyproperties
```

o

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div> <p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p> <pre>--ca temp/certs</pre> </div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none"> Edge and the data source connect using the data source certificate. Edge communicates the data source metadata to your Collibra Platform using a Collibra certified certificate.
<code>--is-openshift</code>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.

Flag	Description
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</div> <p>If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> ■ <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ■ <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre> --set global.priorityClassName.platform=critical-priority --set global.priorityClassName.application=high-priority --set global.priorityClassName.job=low-priority </pre> </div>

```
./edgecli install
--is-openshift
--registry-url https://private-docker.registry.com
--registry-user user1
--registry-pass pass12
2>&1 | tee "edge-installer-$(date
+ "%Y-%m-%d_%H-%M-%S").log"
```

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

8. Run the following command to verify the status of the installation.

```
kubectl get pods -n collibra-edge
```

1. Download the installer:

- a. Open a site.

- i. On the main toolbar, click → **Settings**.

» The [Settings page](#) opens.

- ii. In the tab pane, click **Edge**.

» The **Sites** tab opens and shows a table with an overview of your sites.

- iii. In the site overview, click the name of a site.

» The site page appears.

- b. Click **Download Installer**.

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 directory 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. From inside the extracted TGZ archive directory, run the helm installer prerequisite script, including any additional helm install script flags you may need to configure. For example, if you want to use a custom namespace or install your Edge site on an OpenShift cluster.

Collibra-edge is the single helm chart containing the Edge site. The following prerequisites are handled in 1 execution step via a bash script:

- Cluster scoped resourced, such as namespace, priority classes, and, if you have an OpenShift cluster, SCC, must be installed on the managed Kubernetes cluster.
- Secrets, such as repository access for Collibra and Datadog, must be preprocessed and installed.
- Forward proxy and custom ca information must be preprocessed in order for **proxy.properties** and **ca.pem** to be installed in the managed Kubernetes cluster.
- Many helm chart values are generated in the **site-values.yaml** file based on the optional flags added to the install prerequisite script. This means you don't manually have to specify this information in the installation script.

```
sh collibra-edge-helm-chart/helm-install-prerequisites.sh
--namespace <my-namespace> --installer .
```

Note When you run the installation command, a list of all of these flags are listed. When you run the full command, every command and property run is listed.

Flag	Description
<pre>--installer <path_to_ extracted_ installer></pre>	<p>The path to the downloaded, extracted Edge installer.</p>
<pre>--namespace <my- namespace></pre>	<p>The identifier of the Edge site.</p> <ul style="list-style-type: none"> ◦ If you are installing multiple Edge sites in the same Kubernetes cluster, each Edge site namespace must be unique. For example, <code>--namespace edge-namespace</code>. ◦ If you do not specify a namespace, the default <code>collibra-edge</code> namespace is used.
<pre>--is- openshift</pre>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Note This property is required for all Edge site installations

Note Throughout the remaining installation steps, replace `<my-namespace>` in the provided scripts with this new namespace name. Example scripts will have `edge-namespace` as an example namespace name.

Flag	Description
<code>--proxy</code>	<p>If you are using a forward proxy, add this flag to the install prerequisite script.</p> <div><p>Note If your proxy properties are not in the default proxy.properties file in the root of the Edge installer, you must:</p><ol style="list-style-type: none">Add your proxy properties to a folder relative to the Edge installer.Add the file path of the proxy properties file, relative to the Edge installer, to the install prerequisite script. For example, if you added the proxy properties file to a folder called proxy within the Edge installer folder, add the following to the script:</div> <div><pre>--proxy proxy/myproxy.properties</pre></div>
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the install prerequisite script.</p> <div><p>Note If your custom certificate is not in the default ca.pem file in the root of the Edge installer, you must:</p><ol style="list-style-type: none">Add your certificate file to a folder relative to the Edge installer.Add the file path of the custom certificate file, relative to the Edge installer, to the installation prerequisite script. For example, if you added the custom certificate file to a folder called mycerts within the Edge installer folder, add the following to the script:</div> <div><pre>--ca mycerts/certs.pem</pre></div>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the install prerequisite script to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the install prerequisite script.
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the install prerequisite script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.

Flag	Description
<code>--no-priority-class-install</code>	<div>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</div> <p>If you need to skip installing priority classes, add this flag to the install script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following priority class name flags, to the install script:</p> <ul style="list-style-type: none"> ◦ <code>--global.priorityClassName.platform</code> <code><priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ◦ <code>--global.priorityClassName.application</code> <code><priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ◦ <code>--global.priorityClassName.job</code> <code><priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre>--use-custom-priority-class --global.priorityClassName.platform critical-priority --global.priorityClassName.application high-priority --global.priorityClassName.job low- priority</pre> </div>

4. Install your Edge site using the Helm installer script:

```
helm install collibra-edge collibra-edge-helm-
chart/collibra-edge -n <my-namespace> --values site-
values.yaml
```

Note

- Replace `<my-namespace>` with your Edge site namespace.
- If you need to review the default collibra-edge chart values included when you run the Helm installer script, you can either inspect the **values.yaml** or **README.md** files in **./collibra-edge-helm-chart/collibra-edge**. If you need to override, manually add the value to the command using the `--set` flag.

Configure a forward proxy

You can configure a forward HTTP proxy when you install your Edge site. We support 2 kinds of forward proxy configuration:

- [Explicit proxy](#)
- [Transparent Proxy](#)

For more information, go to [Supported forward proxy configurations for Edge](#).

Note

- For direct forward proxy configurations, complete steps 1-3 to update **proxy.properties** before installing the Edge site.
- For traffic intercepting configurations, such as a Man-in-the-Middle (MITM) proxy, complete all 4 steps to configure the forward proxy and enable the MITM proxy.

Steps

1. Download the Edge site installer:
 - a. Open a site.
 - i. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - ii. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - iii. In the site overview, click the name of a site.
 - » The site page appears.
 - b. Click **Download Installer**.
 - 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. 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. Open the **proxy.properties** file.
4. Remove the "#" symbol from the following lines to uncomment and update the outbound-proxy properties:

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

Setting	Value
noProxy	<p>A comma-separated list of IP or DNS addresses that can bypass the proxy server.</p> <p>This list must include at least the Kubernetes cluster's internal IP addresses and the Kubernetes nodes' IP and DNS addresses.</p> <p>The list may not contain spaces.</p> <div> <p>Tip To get the values for this setting, you can use the <code>edge-get-noproxy.sh</code> script, which you can find in the extracted installer directory under <code>/resources/tools</code>. However, make sure that your network administrator reviews these values.</p> </div> <p>where:</p> <ul style="list-style-type: none"> ◦ <code><host-ip-addresses></code>: for example <code>172.20.0.0/16</code>. ◦ <code><host-dns-names></code>: for example <code>*.compute.internal</code>. ◦ <code><k8s-svc-ip-addresses></code>: is by default <code>10.43.0.0/16</code>, but this can differ for other k8s flavors or configurations. ◦ <code><k8s-pod-ip-addresses></code>: is by default <code>10.42.0.0/16</code>, but this can differ for other k8s flavors or configurations. ◦ <code><others></code>: other IP addresses that don't need to be proxied. Add at least <code>169.254.169.254</code> for AWS. <div> <p>Example</p> <pre>noProxy=172.20.0.0/16,*.compute.internal,10.43.0.0/16,10.42.0.0/16,169.254.169.254</pre> </div>
proxyHost	<p>The IP or DNS address of the proxy server.</p> <div> <p>Example <code>proxyHost=site4-proxy.shared.edge.collibra.dev</code></p> </div>
proxyPort	<p>The TCP port of the proxy server. This value must be a quoted string and not an integer value.</p> <div> <p>Example <code>"proxyPort=3128"</code></p> </div>

Setting	Value
proxyUser- name	<p>The username to authenticate at the proxy server.</p> <div> Example <code>proxyUsername=edge</code> </div> <div> 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\\. </div>
proxyPas- sword	<p>The password to authenticate at the proxy server.</p> <div> Example <code>proxyPassword=la;fs90jpo4j3rR%</code> </div> <div> 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\'. </div>

Important If you are installing your Edge site on a managed Kubernetes cluster, and you add a new node to a cluster, you should 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.

5. Optional, if you want 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](#).

6. Run the [install command for your Edge site](#), beginning from step 3.

What's next?

- If this is a new 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 site installation, you must [reinstall your Edge site](#).
- If you want to update the forward proxy afterwards, you can use the [update script](#).

Supported forward proxy configurations for Edge

For security reasons, you may need your [Edge site](#) to connect to cloud services via a forward HTTP proxy. You can configure this forward proxy during the [Edge site installation](#) process.

We support the following forward proxy configurations:

1. [Explicit proxy](#)
2. [Transparent Proxy](#)

For either type of forward proxy, you can have one of the following configurations:

- A direct, end-to-end encrypted communication between Edge and your Collibra Platform, and Edge and your data sources. This communication is encrypted using standard TLS encryption protocol. By default, Edge only trusts certificates signed by a Public Certificate Authority.
- A traffic intercepting configuration, such as a Man-in-the-Middle (MITM) proxy, which allows your proxy to inspect the communication between Edge and your Collibra Platform, and Edge and your data sources. With this configuration, your proxy needs to be able to decrypt and re-encrypt the communication. In order to do this, you must add private certificates signed by a Private Certificate Authority to your [Edge site truststore](#).

Explicit proxy

There are two options when you configure an explicit forward proxy for Edge:

- A direct explicit proxy. This is a proxy in your network that requires you to configure a specific proxy argument and forwards data from your Edge site to your Collibra Platform. If you want to use a direct explicit proxy, you must add the `--proxy` flag to the Edge site installation script.
- A man-in-the-middle (MITM) explicit proxy. This is a proxy server that stops all incoming, internal traffic based on your specific proxy argument and decrypts it, before forwarding it. An example of this type of proxy is a Squid proxy with SSLBump. If you want to use a MITM explicit proxy, you must add the `--proxy` and `--ca` flags to the Edge site installation script.

Transparent proxy

There are two options when you configure a transparent forward proxy for Edge:

- A direct transparent proxy. This is a proxy server in your network that forwards data from your Edge site to your Collibra Platform. You don't need to configure anything for this type of forward proxy.
- A man-in-the-middle (MITM) transparent proxy. This is a proxy that stops all incoming, internal traffic and decrypts it, before forwarding it. An example of this type of proxy is an AWS TLS Inspection. If you want to use a MITM transparent proxy, you must add the `--ca` flag to the Edge site installation script.

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.

Steps

1. Back up your current Edge site.

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

```
sudo ./edgecli recovery backup --path <backup_path>
```

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

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 --remove-local-data
```

- b. Recreate the Linux disk mount for the `/var/lib/rancher/k3s` directory.
 - i. Create `/var/lib/rancher/k3s` with `mkdir -p /var/lib/rancher/k3s`
 - ii. Mount the disk with "mount -a"
 - iii. Delete the contents with `rm -rf /var/lib/rancher/k3s/*`

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 environment.
 - b. Click **Site Actions**.
 - c. Click **Redownload Installer**.
 - d. Review and check the required acknowledgment checkbox.
 - e. Click **Download Installer**.
 - f. 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 downloaded installer to an empty folder.

```
tar -xf installer-<edge-site-id>.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. Reinstall using the new installer with the backup option, including any additional installation scripts:

```
sudo sh install-master.sh properties.yaml -r
registries.yaml -b <backup_path>
```

Flag	Description
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy
temp/proxy/proxyproperties
```

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div><p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p><pre>--ca temp/certs</pre></div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>sudo ./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none">Edge and the data source connect using the data source certificate.Edge communicates the data source metadata to your Collibra Platform using a Collibra certified certificate.

Flag	Description
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker- delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>edge-docker-delivery.my- registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id <user_id></code>	<p>If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.</p>
<code>--group-id <group_id></code>	<p>If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.</p>

```
sudo sh install-master.sh properties.yaml -r
registries.yaml -b backup.yaml
--registry-url https://private-docker.registry.com
--registry-user user1
--registry-pass pass12
```

Note The default Edge CLI method is an easier solution for installing your Edge site via the Edge CLI. Edge creates the cluster level objects, such as namespaces and priority classes for you. This method can be used for both dedicated and shared clusters.

Note The restrictive Edge CLI method allows you or your company to create the cluster level objects, such as namespaces and priority classes, for your Edge site. This method may be required if your company has security requirements or process that do not allow Edge sites to create the cluster level objects for you. This method can be used for both dedicated and shared clusters.

Warning Collibra Support will not assist with custom Helm or Kubernetes configurations. The following steps are an example, and any assistance for configurations or issues outside of these steps is unsupported. We recommend using the Edge CLI method for managed Kubernetes installations.

A common example of custom Helm configurations is, but not limited to, using an unsupported private repository. At this time, we only support a [JFrog repository](#).

You can reinstall your Edge site on a managed Kubernetes cluster by using the [Edge CLI](#) tool.

Steps

1. Back up your current Edge site.

On the server from which you manage your managed Kubernetes cluster, run the following command:

```
./edgecli recovery backup --path <backup_path>
```

- » Edge creates a backup of your Edge site in the defined folder of the last command.
- 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 **Site Actions**.
 - c. Click **Redownload Installer**.
 - d. Review and check the required acknowledgment checkbox.
 - e. Click **Download Installer**.
 - f. 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 downloaded installer to an empty folder.

```
tar -xvf installer-<edge-site-id>.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 uninstall command, depending on your Edge site installation method, to uninstall the old installation.

- If you installed your Edge site using the previous method, follow the path inside the extracted installer and run the following command:

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

- If you installed your Edge site using the [Edge CLI method](#), run one of the following command:

```
./edgecli uninstall
```

- Optional, if you used a custom namespace, you must add `-n` `<my-namespace>` to the command, replacing `<my-namespace>` with your custom Edge site namespace.

Example:

```
./edgecli uninstall -n <my-namespace>
```

```
./edgecli uninstall 2>&1 | tee "edge-installer-  
$(date +%Y-%m-%d_%H-%M-%S)".log"
```

- Optional, if you used a custom namespace, you must add `-n my-namespace>` to the command, replacing `my-namespace>` with your custom Edge site namespace.

Example

```
./edgecli uninstall 2>&1 | tee "edge-  
installer-$(date +%Y-%m-%d_%H-%M-%S)".log"  
-n my-namespace
```

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:

Note

- You can install your Edge site with either terminal logging or terminal and file logging. Both options log the output of your Edge site installation.
 - Terminal logging only saves the output to the Edge terminal.
 - Terminal and file logging saves the output both to the terminal and a separate file. This file will be saved in the current directory with the naming format: **edge-installer-\$(date +%Y-%m-%d_%H-%M-%S).log**

```
./edgecli install -b backup
```

Add additional flags to the install command as needed. For example, if you have a custom namespace or want to use a [private docker registry](#):

Flag	Description
<code>-n <my-namespace></code>	If you created a custom namespace, add <code>-n <my-namespace></code> to the command. For example:
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy
temp/proxy/proxyproperties
```

◦

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div> <p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p> <pre>--ca temp/certs</pre> </div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none"> Edge and the data source connect using the data source certificate. Edge communicates the data source metadata to your Colibra Platform using a Colibra certified certificate.
<code>--is-openshift</code>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.

Flag	Description
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</div> <p>If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre> --set global.priorityClassName.platform=critical-priority --set global.priorityClassName.application=high-priority --set global.priorityClassName.job=low-priority </pre> </div>

```
./edgecli install -b backup
--registry-url https://private-
docker.registry.com
--registry-user user1
--registry-pass pass12
```

```
./edgecli install -b backup 2>&1 | tee "edge-
installer-$(date +"%Y-%m-%d_%H-%M-%S").log"
```

Add additional flags to the install command as needed. For example, if you have a custom namespace or want to use a [private docker registry](#):

Flag	Description
<code>-n <my-namespace></code>	If you created a custom namespace, add <code>-n <my-namespace></code> to the command. For example:
<code>--proxy</code>	If you are using a forward proxy, add this flag to the installation prerequisite script.

Note If your proxy properties are not in the default **proxy.properties** file in the root of the installer, you must add the file path the installation prerequisite script. For example:

```
--proxy
temp/proxy/proxyproperties
```

o

Flag	Description
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the installation prerequisite script.</p> <div> <p>Note If your custom certificate are not in the default ca.pem file in the root of the installer, you must add the file path the installation prerequisite script. For example:</p> <pre>--ca temp/certs</pre> </div> <p>You can also use this flag to add a custom certificate for data sources.</p> <p>Your data source may require the injection of a custom certificate in order to connect with your Edge site. This custom certificate is typically signed by a private, untrusted Certificate Authority, and therefore must be added to your Edge site truststore.</p> <p>As you may not have a list of all required certificates at the time of installation, we recommend the <code>./edgecli config ca merge --path</code> command shown in the Edge CLI topic.</p> <p>The process functions as follows:</p> <ol style="list-style-type: none"> Edge and the data source connect using the data source certificate. Edge communicates the data source metadata to your Colibra Platform using a Colibra certified certificate.
<code>--is-openshift</code>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the installation command to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-host</code>	<p>Where your private docker is hosted. If you do not specify this parameter, it is automatically derived from <code>--registry-url</code></p> <p>For example:</p> <pre>--registry-host edge-docker-delivery.my-registry.docker.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the installation script.

Flag	Description
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the installation script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.
<code>--no-priority-class-install</code>	<div>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</div> <p>If you need to skip installing priority classes, add this flag to the installation script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<div> <p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following <code>--set</code> flags, to the installation script:</p> <ul style="list-style-type: none"> <code>--set</code> <code>global.priorityClassName.platform=<priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. <code>--set</code> <code>global.priorityClassName.application=<priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. <code>--set</code> <code>global.priorityClassName.job=<priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre> --set global.priorityClassName.platform=critical-priority --set global.priorityClassName.application=high-priority --set global.priorityClassName.job=low-priority </pre> </div>


```
./edgecli install -b backup
--registry-url https://private-
docker.registry.com
--registry-user user1
--registry-pass pass12
2>&1 | tee "edge-installer-$(date
+ "%Y-%m-%d_%H-%M-%S").log"
```

Warning Do not exclude `-b backup.yaml` from this command. If you exclude `-b backup.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.

```
kubectl get -n <my-namespace> secrets -l
edge.collibra.com/backup -o yaml > <PATH_OF_BACKUP_FILE>
```

Property	Description
<my-namespace>	<p>The Edge site namespace.</p> <ul style="list-style-type: none"> ◦ If your Edge site has a custom namespace, add it here. ◦ If your Edge site does not have a custom namespace, add the default namespace, <code>collibra-edge</code>.
<PATH_OF_BACKUP_FILE>	<p>The name of the output yaml file containing your Edge site backup. For example, myBackupFile.yaml.</p>

2. Redownload the installer and save it on your Linux server that has kubectl access to the k8s cluster.
- Go to the Edge site page in your Edge environment.
 - Click **Site Actions**.
 - Click **Redownload Installer**.
 - Review and check the required acknowledgment checkbox.

- e. Click **Download Installer**.
- f. 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 downloaded installer to an empty folder.

```
tar -xf installer-<edge-site-id>.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. From the extracted TGZ archive directory, run the uninstall command.

Note For each of the following commands, replace `<my-namespace>` with the name of your Edge site namespace.

- If you installed your Edge site prior to the 2025.06 release and used the `edge-cd` helm chart, use the following command:

```
sh edge-cd-helm-chart/helm-uninstall.sh --namespace  
<my-namespace>
```

- If you installed your Edge site from or after the 2025.06 release, using the `colibra-edge` helm art, use the following command:

```
sh collibra-edge-helm-chart/helm-uninstall.sh --
namespace <my-namespace>
```

5. Run the following command to apply the Edge site backup file:

```
kubectl apply -f <PATH_OF_BACKUP_FILE>
```

6. From inside the extracted TGZ archive directory , run the helm installer prerequisite script, including any additional helm install script flags you may need to configure. For example, if you want to use a custom namespace or install your Edge site on an OpenShift cluster.

Collibra-edge is the single helm chart containing the Edge site. The following prerequisites are handled in 1 execution step via a bash script:

- Cluster scoped resourced, such as namespace, priority classes, and, if you have an OpenShift cluster, SCC, must be installed on the managed Kubernetes cluster.
- Secrets, such as repository access for Collibra and Datadog, must be preprocessed and installed.
- Forward proxy and custom ca information must be preprocessed in order for **proxy.properties** and **ca.pem** to be installed in the managed Kubernetes cluster.
- Many helm chart values are generated in the **site-values.yaml** file based on the optional flags added to the install prerequisite script. This means you don't manually have to specify this information in the installation script.

```
sh collibra-edge-helm-chart/helm-install-prerequisites.sh
--namespace <my-namespace> --installer .
```

Note When you run the installation command, a list of all of these flags are listed. When you run the full command, every command and property run is listed.

Flag	Description
<pre>--installer <path_to_ extracted_ installer></pre>	<p>The path to the downloaded, extracted Edge installer.</p>
<pre>--namespace <my- namespace></pre>	<p>The identifier of the Edge site.</p> <ul style="list-style-type: none"> ◦ If you are installing multiple Edge sites in the same Kubernetes cluster, each Edge site namespace must be unique. For example, <code>--namespace edge-namespace</code>. ◦ If you do not specify a namespace, the default <code>collibra-edge</code> namespace is used.
<pre>--is- openshift</pre>	<p>If you are using an OpenShift cluster, add this flag to deploy Security Context Constraints (SCC) which provide the Edge service accounts with the required permissions.</p>

Note This property is required for all Edge site installations

Note Throughout the remaining installation steps, replace `<my-namespace>` in the provided scripts with this new namespace name. Example scripts will have `edge-namespace` as an example namespace name.

Flag	Description
<code>--proxy</code>	<p>If you are using a forward proxy, add this flag to the install prerequisite script.</p> <div><p>Note If your proxy properties are not in the default proxy.properties file in the root of the Edge installer, you must:</p><ol style="list-style-type: none">Add your proxy properties to a folder relative to the Edge installer.Add the file path of the proxy properties file, relative to the Edge installer, to the install prerequisite script. For example, if you added the proxy properties file to a folder called proxy within the Edge installer folder, add the following to the script:</div> <div><pre>--proxy proxy/myproxy.properties</pre></div>
<code>--ca</code>	<p>If you want to use a custom certificate, for example to configure a forward man-in-the-middle proxy, add this flag to the install prerequisite script.</p> <div><p>Note If your custom certificate is not in the default ca.pem file in the root of the Edge installer, you must:</p><ol style="list-style-type: none">Add your certificate file to a folder relative to the Edge installer.Add the file path of the custom certificate file, relative to the Edge installer, to the installation prerequisite script. For example, if you added the custom certificate file to a folder called mycerts within the Edge installer folder, add the following to the script:</div> <div><pre>--ca mycerts/certs.pem</pre></div>

Flag	Description
<code>--disable-otel</code>	If you don't want to send your metrics and logs to Edge, add this flag to the install prerequisite script to disable OpenTelemetry.
<code>--registry-url</code> <code><registry-url></code>	<p>The URL of your registry.</p> <p>Add this flag if you use a private docker registry either with or without authentication.</p> <p>For example:</p> <pre>--registry-url edge-docker-delivery.repository.collibra.io</pre>
<code>--registry-user</code> <code><registry-user></code>	<p>Your registry account username.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--registry-pass</code> <code><registry-pass></code>	<p>Your registry account password.</p> <p>Add this flag if you use a private docker registry with authentication.</p>
<code>--user-id</code> <code><user_id></code>	If you want to run all of your Edge site pods and containers with a specific user ID (UID), add this flag to the install prerequisite script.
<code>--group-id</code> <code><group_id></code>	If you want to run all of your Edge site pods and containers with a specific group ID (GID), add this flag to the install prerequisite script.
<code>--unset-run-as-ids</code>	If your Edge site is installed on an OpenShift Kubernetes cluster, and you want to run all of your Edge site pods and containers from random UIDs and GIDs, add this flag to the installation script.

Flag	Description
<code>--no-priority-class-install</code>	<p>Warning Don't skip priority class configuration unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you need to skip installing priority classes, add this flag to the install script. Running this flag sets all Edge site pods to the default priority (0).</p>

Flag	Description
<code>--use-custom-priority-class</code>	<p>Warning Don't configure custom priority classes unless you have an experienced Kubernetes engineer in your organization, as doing so may result in Edge site failures.</p> <p>If you want to configure custom priority classes for your Edge site pods, add this flag, along with the following priority class name flags, to the install script:</p> <ul style="list-style-type: none"> ◦ <code>--global.priorityClassName.platform</code> <code><priority></code>: This flag sets the custom priority class name for Edge platform pods. This should be the highest priority class in Edge. ◦ <code>--global.priorityClassName.application</code> <code><priority></code>: This flag sets the custom priority class name for Edge application pods. This should be the second highest priority class in Edge. ◦ <code>--global.priorityClassName.job</code> <code><priority></code>: This flag sets the custom priority class name for Edge job pods. This should be the third highest priority class in Edge. <pre>--use-custom-priority-class --global.priorityClassName.platform critical-priority --global.priorityClassName.application high-priority --global.priorityClassName.job low- priority</pre>

7. Install your Edge site using the Helm installer script, replacing `<my-namespace>` with your Edge site namespace.:


```
helm install collibra-edge collibra-edge-helm-  
chart/collibra-edge -n <my-namespace> --values site-  
values.yaml
```

Note

- If you need to review the default collibra-edge chart values included when you run the Helm installer script, you can either inspect the **values.yaml** or **README.md** files in **./collibra-edge-helm-chart/collibra-edge**. If you need to override, manually add the value to the command using the **--set** flag.

Upgrade the operating system for k3s Edge sites

When you have a running Edge site on bundled k3s, 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 Platform 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 → **Settings**.
 - » The [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 **Site 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 -r registries.yaml  
-r registries.yaml  
-b /<path to backup file>/edge-backup.yaml
```

4. Wait until the Edge site becomes healthy in the Collibra Platform 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 Platform quarterly release. Upgrades may become required sooner due to:
 - Important security updates.
 - Migrations.
 - New feature requirements.

Automatic upgrade

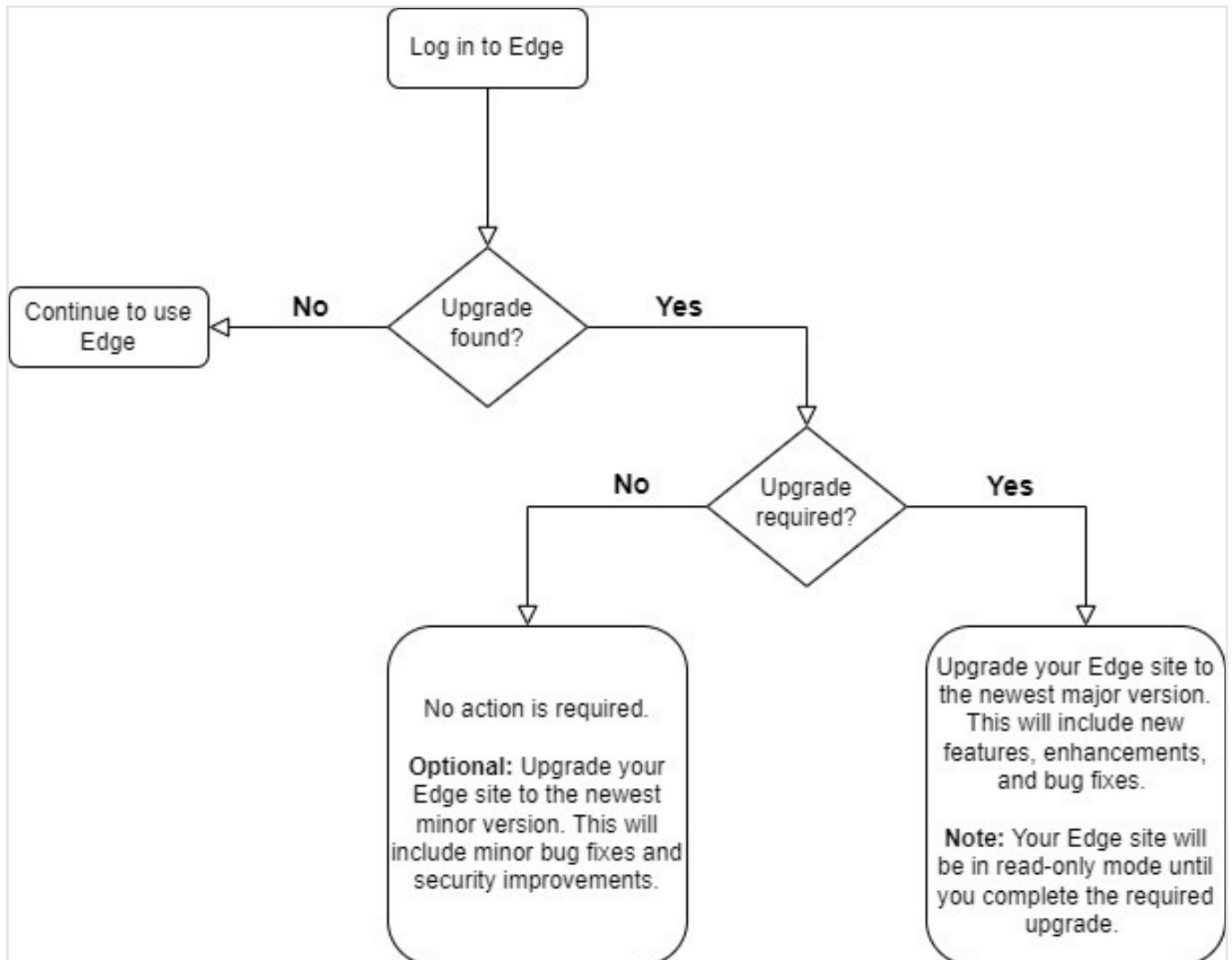
The Automatic mode is the default upgrade mode for Edge sites. This means that when a new Edge site version is released, you do not need to initiate the upgrade, as it will automatically be applied to your Edge site. You will only need to take action if the new version includes new software requirements 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 which Edge site versions may require action, such as a reinstall for sites installed on k3s or a Kubernetes upgrade for sites installed on managed Kubernetes. 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 Platform 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 Platform](#). 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

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.

Note Security scans report are only accepted for the most recent generally available (GA) release. For more information, go to one of the following resources:

- [Vulnerability and scanning policy](#)
- [Edge security scanning](#)
- [Compatibility between Edge sites and Collibra Platform](#)

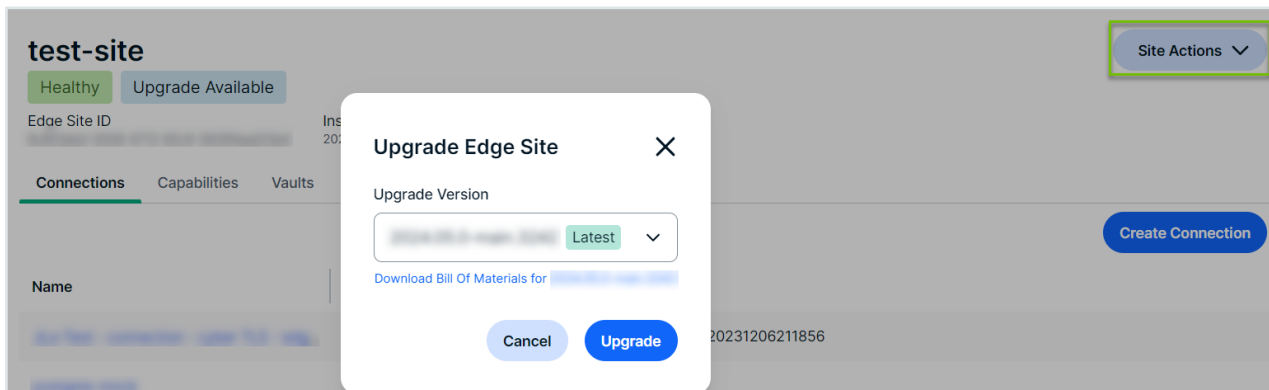
What's next?

- 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 a site.
 - a. On the main toolbar, click → **Settings**.
 - » The **Settings** page opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the top right corner, click **Site Actions** → **Upgrade Site**.
 - » 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**.



What's next?

- Review the [Compatibility between Edge sites and Colibra 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 Edge supported version of your managed Kubernetes.

- 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.

Create Edge Site

Site name

Please provide a unique name.

Description

Please provide a meaningful description of the site including i.e. data center name, location, necessary connection and capabilities.

Upgrade mode

☒ Automatic

☐ Manual

You can choose to update this site automatically (default) or manually (you will need to perform updates yourself).

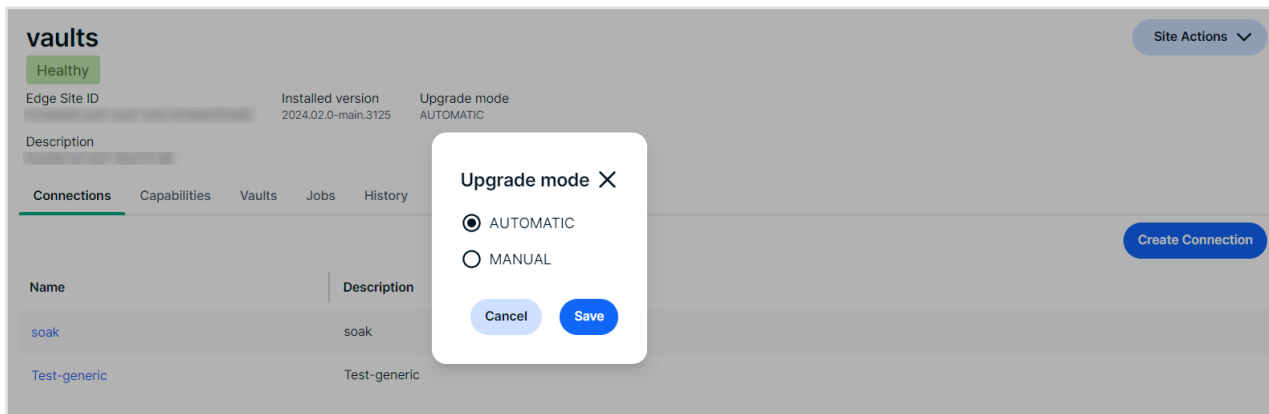
Cancel

Create site

Existing Edge sites

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

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The **Settings** page opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the top right corner, click **Site Actions** → **Change 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 Edge supported version of your managed Kubernetes.

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**.

Create Edge Site

Site name

Please provide a unique name.

Description

Please provide a meaningful description of the site including i.e. data center name, location, necessary connection and capabilities.

Upgrade mode

☐ Automatic

☒ Manual

You can choose to update this site automatically (default) or manually (you will need to perform updates yourself).

Edge Version

2024.02.0-main.3125

The Edge version you wish to install

[Download Bill Of Materials for 2024.02.0-main.3125](#)

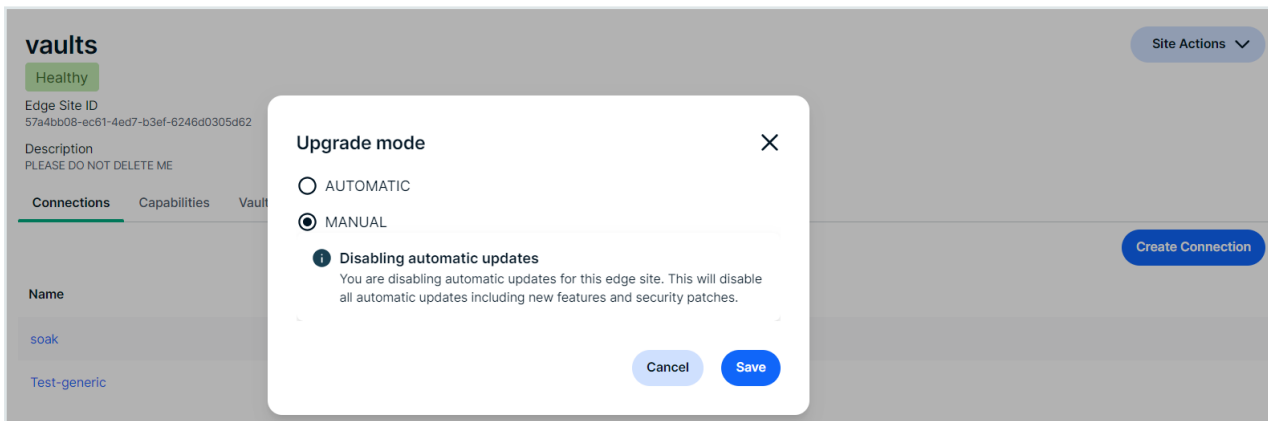
Cancel

Create site

Existing Edge sites

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

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The **Settings** page opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the top right corner, click **Site Actions** → **Change 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 Colibra Platform](#) to know when you need to either [reinstall your Edge site](#) for an upgraded version of k3 or upgrade to the latest Edge supported version of your managed Kubernetes.

- 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.

Maintaining Edge sites

In this section, you will learn how you can maintain your Edge site installations, such as performing backups or updating credentials.



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 **Manage Edge sites** [global permission](#).

Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the top right corner, click **Site Actions** → **Edit Site**.
 - » The Edit Edge site wizard starts.
3. Enter the required information.

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

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

Update Edge user's username or password

When you [download the Edge site installer](#), a dedicated user account is created in Collibra Platform. 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.

Run the following command:

Note

- The commands below provide both password and username. If you want to only update one, you can remove the other when running the command. For example:

```
./edgecli config dgc --pass <Password123!>
```

- When resetting the password, follow the steps in our [Set or reset a user password](#) article. The only non-alphanumeric characters accepted for passwords are: !, \$, %, &, (,), *, +, ,, -, ., /, :, ;, <, =, >, ?, @, [,], ^, _, {, |, }, ~. For more information about the default password requirements, go to the [Password settings](#).

- For Edge sites installed on a bundled k3s cluster:

```
sudo ./edgecli config dgc --pass <password> --url <dgc url>
--user <username>
```

- For Edge sites installed on a managed Kubernetes cluster:

```
./edgecli config dgc --pass <password> --url <dgc url> --
user <username> -n <my-namespace>
```

Note If your Edge site is installed on a dedicated cluster via the Edge CLI method and it does not have a custom namespace, you can remove `-n <my-namespace>` from the command.

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,
 - If your Edge site is installed on a bundled k3s cluster, run the following command wherever your Edge site is installed:

```
sudo ./edgecli config proxy --path <path to proxy config>
```

- If your Edge site is installed on a managed Kubernetes cluster, run the following command from a Linux machine that has access to the Kubernetes cluster where your Edge site is installed:

```
./edgecli config proxy --path <path to proxy config>
```

Help file of the script

```
~/edgecli config proxy --help
Usage:
  edgecli config proxy [flags]
```

```
Flags:
  --path string

Global Flags:
  -h, --help
```

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:

```
sudo ./edgecli recovery backup --path <backup_path>
```

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

On the server from which you manage your managed Kubernetes cluster, run the following command:

```
./edgecli recovery backup --path <backup_path>
```

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

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

```
kubect1 get -n <my-namespace> secrets -l  
edge.collibra.com/backup -o yaml > <PATH_OF_BACKUP_FILE>
```

Property	Description
<code><my-namespace></code>	The Edge site namespace. <ul style="list-style-type: none">• If your Edge site has a custom namespace, add it here.• If your Edge site does not have a custom namespace, add the default namespace, <code>collibra-edge</code>.
<code><PATH_OF_BACKUP_FILE></code>	The name of the output yaml file containing your Edge site backup. For example, myBackupFile.yaml .

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

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.

Note Deleting an Edge site does not delete the data ingested in Collibra Platform. The ingested data must be [deleted manually](#).

Prerequisites

- You have [created](#) an Edge site.
- You have a [global role](#) that has the **Manage Edge sites** [global permission](#).
- Ensure that your environment uses the [latest user interface](#).

Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the top right corner, click **Site Actions** → **Delete Site**.
 - » The Delete Edge site wizard starts.
3. Click **Delete**.
 - » 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
<pre>/usr/local/bin/uninstall-edge.sh</pre>	<p>Delete Edge site, but keep its data.</p> <p>The data consists of drivers, required files for capabilities, and data that was saved by Edge capabilities</p>
<pre>/usr/local/bin/uninstall-edge.sh --remove-local-data</pre>	Delete Edge site and its data.
<pre>/usr/local/bin/uninstall-edge.sh --remove-local-data --force</pre>	<p>Delete Edge site without confirmation request, for example if you want to delete the site via a script.</p> <p>You can use this in combination with removing the site data.</p>

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The **Settings** page opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the top right corner, click **Site Actions** → **Delete Site**.
 - » The Delete Edge site wizard starts.
3. Click **Delete**.
 - » The Edge sites overview appears, without the deleted Edge site.
4. On the server from which you manage your cluster, run one of the following command:

- With terminal logging

```
./edgecli uninstall
```

- With terminal and file logging:

```
./edgecli uninstall 2>&1 | tee "edge-installer-  
$(date +"%Y-%m-%d_%H-%M-%S").log"
```

Note

If your Edge site has a custom namespace, add `-n <my-namespace>` to the command.

5. From the extracted TGZ archive, run the following command:

```
sh edge-cd-helm-chart/helm-uninstall.sh
```

Note If your Edge site had a custom namespace, add `--namespace <my-namespace>`, replacing `<my-namespace>` with the name of your Edge site namespace.

Important If you do not plan on reinstalling your Edge site on the same Kubernetes cluster as it was originally installed on, then you must run the following command after you reinstall on the new cluster to delete unmanaged resources:

```
kubectl delete priorityclass job application platform  
kubectl delete crd clusterworkflowtemplates.argoproj.io  
\  
cronworkflows.argoproj.io  
workflowartifactgctasks.argoproj.io  
workfloweventbindings.argoproj.io workflows.argoproj.io  
workflowtaskresults.argoproj.io \  
workflowtasksets.argoproj.io  
workflowtemplates.argoproj.io
```

Disaster recovery for managed Kubernetes Edge sites

Edge's disaster recovery procedure allows you to recover your data in instances where your Kubernetes cluster fails while Edge is running. This procedure is only applicable when your Edge site is installed on a [managed Kubernetes cluster](#).

Use Case Scenario

Below is an example scenario which demonstrates when this procedure would be used and how to initiate it.

In this scenario, an Edge site has already been installed on Cluster A. We recommend that you set up a periodic backup for Cluster A to avoid losing your Edge site configurations. In the event of a failure, you have to uninstall Cluster A, reinstall Edge on a new cluster (Cluster B), and then finally restore the Cluster A Edge site backup on Cluster B.

Note You can follow the steps and use the commands in [Reinstall an Edge site](#) per your preferred installation method. However, in the case of a disaster, the Edge site would be installed in a new cluster, as shown below.

The process of recovering from a disaster requires the following steps:

1. Set up a periodic backup of Edge on Cluster A.
2. Uninstall the Edge site running on Cluster A.
3. Install the new Edge site and restore a backup on Cluster B.
 - If you use the Helm chart method, the restore backup command must be run before you reinstall the Edge site.

Tip All steps require that the `kubectl` utility can connect to the cluster.

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	<ul style="list-style-type: none"> • Edge-certified JDBC connectors • JDBC metadata ingestion on Database level • JDBC data profiling and classification • JDBC data sampling • Data Notebooks • Protect • Multiple integrations, such as Amazon S3, Databricks, ADLS 	<ul style="list-style-type: none"> • Jobserver-certified JDBC connectors • JDBC metadata ingestion: Schema level • Data profiling • Data classification • Data sampling • Some integrations
Security	<ul style="list-style-type: none"> • 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. • You can 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. • 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). 	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • Can ingest and profile in parallel. • No limit on the table size you can profile. 	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> ◦ 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. 	<ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • Catalog JDBC ingestion • JDBC Profiling • To classify data via Unified Data Classification, also enable the Unified Data Classification method.
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.

Note Once all data sources have been migrated, you can decommission Jobserver. For all details, go to the [Support center](#).

[[[Undefined variable CollibraGeneral.additional-resources]]]

- Collibra University courses
 - [The Value of Edge](#)
 - [Preparing Edge for Migration](#)
 - [Migrating schemas from Jobserver to Edge](#)
- [Schedule a coaching session](#) with a Collibra expert who can provide best practices, answer migration questions, and help you get started with the migration process.
- [Schedule a migration session](#) with an Edge expert who can discuss your migration needs and ensure a successful migration.
- Speak with the Collibra Account Team about the Edge Migration Accelerator Program to work with a Collibra expert who can help you seamlessly migrate from Jobserver to Edge.

Troubleshooting Edge

For a list of all Edge troubleshooting topics, go to [the Support Portal](#).

Most popular resources:

- [Common Edge troubleshooting topics](#)
- [Create an Edge diagnostics file](#)

Edge FAQ

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

Question	Answer
Who benefits from using Edge?	<p>All customers who want to ingest data into Collibra Platform benefit from Edge.</p> <p>Some of the benefits for using Edge are:</p> <ul style="list-style-type: none">• Data is processed in the customer's secure environment and only the process results are sent to Collibra Platform.• Edge can automatically anonymize sensitive profiling data before sending it to Collibra Platform.• Edge can automatically classify the metadata and send the classification results together with the profiling results to Collibra Platform.• 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 Colibra Platform.
- 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 Colibra 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 Colibra Platform 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)?

Yes, you can install Edge on the following managed Kubernetes clusters:

- Azure Kubernetes Service (AKS)
- AWS Fargate using EKS
- Amazon EKS
- Google Kubernetes Engine (GKE)
- OpenShift

Currently, we only support basic integrations with these Cloud services. Please contact your Account Team if you have any questions.

Note Alternatively, if you install Edge on a Cloud environment, the Edge site installer includes the k3s Kubernetes version.

Can you use Autopilot mode if your Edge site is installed on a Google Kubernetes Engine (GKE) cluster?

Yes, but we cannot support troubleshooting your Edge site installed on a GKE cluster if Autopilot mode is enabled.

Can Edge be installed on Windows servers?

If you use Microsoft technologies, you can install your Edge site on a managed Azure Kubernetes Service (AKS) cluster.

We prioritize your experience on Linux-based operating systems, and as such, because Microsoft does not currently provide seamless support for k8s clusters and container technology, we do not provide support for Edge installations on any other Microsoft technologies at this time.

Can Edge be installed on a cluster with existing resources?

From the 2024.05 Edge release, Edge sites can be installed on shared Kubernetes clusters. To learn more, go to the [system requirements](#) for installing Edge on a managed, shared Kubernetes cluster.

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?	Yes, you can integrate your Edge site with the following vault providers : <ul style="list-style-type: none"> • CyberArk Vault • HashiCorp Vault • Azure Key Vault • AWS Secrets Manager • Google Secret Manager
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 Platform?	An Edge site is installed in the customer's environment, close to the data source. The Edge site communicates to Collibra Platform using an outbound HTTPS connection via port 443.
Does deleting an Edge site delete the data from capabilities already ingested in Collibra Platform?	No. When you delete an Edge site, only the site and its configurations are deleted. Data that has already been ingested in Collibra Platform must be deleted manually .
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 Platform 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 Platform environment.

Question

Answer

Can Edge use customer-provided certificates to connect to Collibra Platform?

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 Platform 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.

Does Edge support mTLS when connecting to Collibra Platform?

Currently, we do not support this.

Is Edge horizontally scalable?

Yes, Edge sites installed on a [managed, shared Kubernetes cluster](#) are horizontally scalable.

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.

Question

Answer

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](#).

For more information about security scanning, go to [Collibra's vulnerability and scanning policy](#).

How are secrets stored on an Edge site?

You can find the details of how Edge stores secrets in this [Storing secrets overview](#).

About Collibra Cloud sites

A Collibra Cloud site is hosted by Collibra, which allows you to integrate with cloud-native data sources out-of-the-box. A Collibra Cloud site is set up and managed by Collibra, allowing you to focus on your business needs. While this solution simplifies implementation and maintenance, it offers slightly fewer features than customer-managed Edge sites and connects to data sources over the internet. Collibra Cloud sites upgrade automatically, meaning they are always on the latest, most secure version.

All Collibra Cloud sites are named Collibra Cloud site, as shown below: Edge

Sites	Name ▼	Description
Jobs	Collibra Cloud	
	Collibra Cloud Site ☁	Collibra Cloud Site

What is included with a Collibra Cloud site?

You can create and manage your Collibra Cloud site connections and capabilities in a similar way to how these are managed in a customer-managed Edge site. However, because a Collibra Cloud site connects to data sources over the internet and is maintained by Collibra, some connections and capabilities are not available.

The following list shows the supported data sources per capability:

- Metadata ingestion and synchronization
 - Amazon Redshift (JDBC)
 - Athena (JDBC)
 - AWS Glue (JDBC)
 - Azure Data Lake Storage
 - Azure Synapse Analytics
 - Databricks Unity Catalog
 - Databricks (JDBC)
 - Google BigQuery (JDBC)
 - Google Cloud Storage
 - Google Dataplex

- Salesforce (JDBC)
- SAP Datasphere Catalog
- SAP HANA
- Snowflake (JDBC)
- S3
- **Classification and Profiling**
 - Amazon Redshift (JDBC)
 - Athena (JDBC)
 - AWS Glue (JDBC)
 - Azure SQL server
 - Azure Synapse Analytics
 - Databricks (JDBC)
 - Databricks Unity Catalog
 - Google BigQuery (JDBC)
 - Salesforce (JDBC)
 - SAP HANA Cloud
 - Snowflake (JDBC)
- **Technical lineage**
 - Amazon Redshift (JDBC)
 - Azure SQL Data Warehouse
 - Azure SQL server
 - Azure Synapse Analytics
 - Databricks Unity Catalog
 - Google BigQuery (JDBC)
 - Google Dataplex
 - Power BI
 - SAP HANA Cloud/Advanced
 - Snowflake (JDBC)
 - Tableau
- **Protect**
 - AWS Lake Formation
 - Databricks (JDBC)
 - Google BigQuery
 - Snowflake (JDBC)
- **AI Governance**
 - AWS Bedrock AI
 - AWS SageMaker AI

- Azure AI Foundry
- Azure ML
- Databricks Unity Catalog
- Google Vertex AI
- MLflow
- SAP AI Core
- Unified Data Quality
 - Amazon Redshift (JDBC)
 - Athena (JDBC)
 - Databricks (JDBC)
 - Google BigQuery (JDBC)
 - SAP HANA Cloud/Advanced
 - Snowflake (JDBC)

For more information, go to our [connections](#) and [capabilities](#) documentation.

Additionally, you can register the following data sources for [Data Notebook](#):

- Amazon Redshift (JDBC)
- Athena (JDBC)
- Databricks Unity Catalog
- Google BigQuery
- Google BigQuery (JDBC)
- Snowflake (JDBC)

Note If a data source you want to integrate with is not listed below, contact your Account Executive for more options.

Limitations

As Collibra Cloud sites are managed by Collibra, some customer-managed Edge functionalities are not supported on Collibra Cloud sites:

- [Edge CLI](#)

Note

As the Edge CLI is unsupported for Collibra Cloud sites,

- You can't backup or restore your Collibra Cloud site.
- The following Lineage integrations are not available:
 - IBM InfoSphere DataStage
 - Informatica PowerCenter
 - SQL Server Integration Services (SSIS)
 - dbt Core
 - Custom Lineage
 - JDBC Lineage via Shared Storage connection
 - Open Lineage

- [Sampling](#)
- [Data Notebook's Postgres Database storage capability](#)
- [Control of managed Kubernetes clusters](#)
- [Manual upgrades](#)
- [Customer hosted Vault integrations](#)
- [Forward proxies](#)
- [Custom repositories](#)
- [FedRAMP authorization](#)
- [Security Scanning](#)

What's next?

- [Request a Collibra Cloud site](#)
- [Available connections](#)
- [Available capabilities](#)
- [Jobs dashboard](#)

Request a Collibra Cloud site

You can request 1 Collibra Cloud site per environment you have. For example, 1 Collibra Cloud site for your development environment and 1 Collibra Cloud site for your production environment. You can submit a request to Collibra to set up your Collibra Cloud site by following the steps in this topic.

Note If you've already requested a Collibra Cloud site in an environment, you aren't able to request another Collibra Cloud site.

Steps

1. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
2. Click **Edge**.
 - » The Edge sites overview opens.
3. Click **Create Edge site**.
 - » The Edge site creation wizard opens.
4. Click **Select Collibra Cloud** in the Collibra Cloud site section.
5. Review and confirm the Collibra Cloud site information with your internal team.
6. Click **Request site**.
 - » A request for a Collibra Cloud site is submitted to Collibra. Your Collibra Cloud site is in the **Read only** status until it is approved.
7. When your Collibra Cloud site is approved, you will receive a confirmation email, including any IP addresses you may need to include in your allowlist. Once your site is approved and setup, you can begin adding connections and capabilities.

What's next?

- [Available connections](#)
- [Available capabilities](#)
- [Jobs dashboard](#)

Edge and Collibra Cloud site site connections

Connections define how a [capability](#) communicates with a data source in order to collect and send metadata to Collibra.

About Edge and Collibra Cloud site connections

To collect metadata from a data source and add it into Collibra via an Edge or Collibra Cloud site, your site needs to be able to communicate with the data source. This is managed via a connection. Once a connection is established, it 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 connection types and the associated capabilities.

Connection type	Description	Supported for Edge sites?	Supported for Collibra Cloud site?
AWS (Amazon Web Services)	<p>Used for the integration and protection of Amazon S3 data sources and Amazon SageMaker AI models.</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> • S3 synchronization • Protect for AWS Lake Formation • AWS Bedrock AI • AWS SageMaker AI 	✓ Yes	✓ Yes

Connection type	Description	Supported for Edge sites?	Supported for Collibra Cloud site?
Azure	<p>Used for the integration of Azure Data Lake Storage (ADLS) data sources, Azure AI Foundry models and agents, and Microsoft Azure AI models.</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> • ADLS synchronization • Azure AI Foundry • Azure ML 	✓ Yes	✓ Yes
Databricks	<p>Used for the integration of Databricks Unity Catalog.</p> <p>Show associated capability</p> <ul style="list-style-type: none"> • Data Unity Catalog 	✓ Yes	✓ Yes
dbt	<p>Used for the integration of</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> • Technical Lineage for dbt Cloud 	✓ Yes	✗ No
GCP (Google Cloud Platform)	<p>Used for the integration and protection of Google Cloud Storage and Dataplex data sources.</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> • GCS synchronization • Protect for Google BigQuery • technical lineage for Google Dataplex 	✓ Yes	✓ Yes
HTTP <ul style="list-style-type: none"> • Basic Auth • No Auth • OAuth 2.0 	<p>Allows workflows to communicate with other HTTP systems, such as external REST APIs.</p>	✓ Yes	✓ Yes

Connection type	Description	Supported for Edge sites?	Supported for Collibra Cloud site?
Informatica Intelligent Cloud Services	<p>Used to connect to Informatica Intelligent Cloud Services.</p> <p>Show associated capability</p> <ul style="list-style-type: none"> • Technical lineage for Informatica Intelligent Cloud Services (IICS) 	✓ Yes	✗ No

Connection type	Description	Supported for Edge sites?	Supported for Collibra Cloud site?
JDBC <ul style="list-style-type: none"> Generic JDBC connection (recommended) Username/Password JDBC connection 	<p>Used to connect to JDBC data sources, for example, Snowflake, Salesforce, and PostgreSQL.</p> <p>In most cases, you need to create a connection for each database you want to register. Some data sources, however, allow you to use a single connection to register multiple databases. You can find this information in the Supports registration of multiple databases? in Overview of connectors.</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> Catalog JDBC ingestion Catalog JDBC Sampling JDBC Profiling Catalog Data Classification Protect for Snowflake Technical lineage capabilities for data sources that use the JDBC connection <ul style="list-style-type: none"> 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 	✓ Yes	✓ Yes, limited depending on the capability. For more information, go to available capabilities .

Connection type	Description	Supported for Edge sites?	Supported for Collibra Cloud site?
	<ul style="list-style-type: none"> ◦ 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 		
Looker	<p>Used to connect to Looker.</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> • Technical Lineage for Looker 	✓ Yes	✗ No
Matillion	<p>Used to connect to Matillion.</p> <p>Show associated capability</p> <ul style="list-style-type: none"> • Technical Lineage for Matillion 	✓ Yes	✗ No
Microsoft SSRS-PBRS	<p>Used to connect to SSRS-PBRS.</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> • Technical Lineage for SSRS-PBRS 	✓ Yes	✗ No
MicroStrategy	<p>Used to connect to MicroStrategy.</p> <p>Show associated capabilities</p> <ul style="list-style-type: none"> • Technical Lineage for MicroStrategy 	✓ Yes	✗ No

Connection type	Description	Supported for Edge sites?	Supported for Collibra Cloud site?
MLflow	Used to connect to MLflow. Show associated capabilities <ul style="list-style-type: none"> • MLflow AI 	✓ Yes	✓ Yes
Power BI	Used to connect to Power BI. Show associated capabilities <ul style="list-style-type: none"> • Technical Lineage for Power BI 	✓ Yes	✓ Yes
SAP AI Core	Used to connect to SAP AI Core. Show associated capabilities <ul style="list-style-type: none"> • SAP AI Core capability 	✓ Yes	✓ Yes
SAP Datasphere Catalog	Used to connect to SAP Analytics Cloud. Show associated capabilities <ul style="list-style-type: none"> • SAP Datasphere Catalog synchronization 	✓ Yes	✓ Yes
Shared Storage connection	Used to access files from a shared folder. Show associated capability <ul style="list-style-type: none"> • Technical Lineage for SqlDirectory • Technical Lineage for Custom Technical Lineage • Technical Lineage for DataStage • Technical Lineage for dbt • Technical Lineage for Informatica PowerCenter • Technical Lineage for SQL Server Integration Services (SSIS) 	✓ Yes	✗ No

Connection type	Description	Supported for Edge sites?	Supported for Collibra Cloud site?
Tableau	Used to connect to Tableau Server or Tableau Online. Show associated capability <ul style="list-style-type: none"> • Technical Lineage for Tableau 	✓ Yes	✓ Yes
Technical Lineage Admin	Used to connect to the Collibra Data Lineage service instances, to run any of the following technical lineage admin options: <ul style="list-style-type: none"> • List sources • Ignore sources • Analyze files • Sync 	✓ Yes	✓ Yes

Edit an Edge and Collibra Cloud site 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.

Available vaults

Prerequisites

- You have a [global role](#) that has the **Product Rights > System administration** [global permission](#).
- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#).
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud site](#).
- You have [added a vault to your 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 a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.

- c. In the site overview, click the name of a site.
 - » The site page appears.
- 2. Locate and click the connection you want to edit.
- 3. At the bottom of the page, click **Edit**.
- 4. Edit the connection or vault information.

How to use your vault...

To use your vault, do the following:

- a. In the **Value Type** field, select **Vault Key**.
- b. Enter the query value to identify the secret in your vault.

Example

Username *

Vault Key ▾

Query

CyberArk Credential Provider [Query*]
The username to use to establish a connection

Password *

Vault Key ▾

Query

CyberArk Credential Provider [Query*]
The password to use to establish a connection

To use your vault, do the following:

- a. In the **Value Type** field, select **Vault Key**.
- b. Enter the required information:

Name	Description
Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database
Engine Path	The engine path to your vault where the value is stored.
Secret Path	The secret path to your vault where the value is stored.
Field	The name of the field to your vault where the value is stored.

Note Only available if you selected **Key Value** in the **Secret Engine Type** field.


Name	Description
Role	The role specified in the Database engine.

Note Only available if you selected **Database** in the **Secret Engine Type** field.

Example

Username *

Vault Key ▾


Key Value ▾Engine PathSecret PathField

HashiCorp Vault [Engine Path*] [Secret Path*] [Field*]

The username to use to establish a connection

Password *

Vault Key ▾

Database ▾Engine PathRoleusername ▾

HashiCorp Vault [Engine Path*] [Role*] [Field*]

The password to use to establish a connection


- To use your vault, do the following:
- a. In the **Value Type** field, select **Vault Key**.
 - b. Enter the required information:

Name	Description
Vault Name	The name of your Azure Key Vault in your Azure Key Vault service where the value is stored.
Secret Name	The name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▾


Vault NameSecret Name

Azure Key Vault [Vault Name*] [Secret Name*]

The username to use to establish a connection

Password *

Vault Key ▾

Vault NameSecret Name

Azure Key Vault [Vault Name*] [Secret Name*]

The password to use to establish a connection

To use your vault, do the following:

- a. In the **Value Type** field, select **Vault Key**.
- b. Enter the required information:


Name	Description
Secret Name	The name of the secret in your vault where the value is stored.
Field	If the secret stored in your AWS Secrets Manager is a JSON value, for example { "pass1": "my-password", "pass2": "my-password2" }, then you need to specify the Field to point to the exact JSON value that should be used. For example, Secret Name: edge-db-customer; Field: pass.

Note If the secret stored in your AWS Secrets Manager is a plain string value, for example my-password, then you do not need to specify the **Field**.

Example

Username *

Vault Key ▾

 Secret Name


Field

AWS Secrets Manager [Secret Name*] [Field*]

The username to use to establish a connection

Password *

Vault Key ▾

 Secret Name

Field

AWS Secrets Manager [Secret Name*] [Field*]

The password to use to establish a connection


To use your vault, do the following:

- a. In the **Value Type** field, select **Vault Key**.
- b. Enter the name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▾


 Secret Name

GCP Secret Manager [Secret Name*]

The username to use to establish a connection

Password *

Vault Key ▾

 Secret Name

GCP Secret Manager [Secret Name*]

The password to use to establish a connection

5. Click **Save**.

Delete an Edge or Collibra Cloud site connection

You can delete a connection from a [Edge or Collibra Cloud 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 **Product Rights > System administration** [global permission](#).
- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#).
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud site](#).

Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. Locate and select the connection you want to delete.
3. At the bottom of the page, click **Delete**.
 - » The **Confirm Delete of Connection** dialog box appears.
4. Click **Delete**.

Warning When you delete a JDBC connection that Data Quality & Observability Classic uses from a 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 or Collibra Cloud site to a data source. You can then [register the data source via Edge](#).

Note If you're using a Collibra Cloud site, go the [Collibra Cloud site documentation](#) to check if your data source is supported.

Note If you're using a Collibra Cloud site, go the [Collibra Cloud site documentation](#) to check if your data source is supported.

Prerequisites

- You have a [global role](#) that has the **Product Rights > System administration** [global permission](#).

- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#).
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud site](#).
- You have [added a vault to your Edge site](#).
- If your data source connection requires a file from your vault, the file must be encoded into Base64 and stored as a regular secret in your vault.

Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the JDBC Connections section, click **Create Connection**.
 - » The **Create Connection** dialog box appears.
3. Click the **Generic JDBC connection** connection type.

Important

- If the authentication method you use includes the `username` and `password` properties, you can use either the Generic or Username/Password JDBC connection type. If you use the Generic JDBC connection type, add the `username` and `password` properties manually.
- After creating a Generic JDBC Connection, you can update your connection to use a different authentication method if needed. However, if you initially use the Username/Password JDBC connection type and want to change to another authentication method, you must create a new connection by using the Generic JDBC Connection.

4. Enter the required information.

Field	Description
Connection settings	This section contains the settings to connect to your data source.

Field	Description
Name	<p>The name of the JDBC connection.</p> <div> <p>Note We recommend not to use the special character > in the JDBC connection name. This character is part of the full name of assets created via Edge. If you use this character, features such as sampling or classification can be impacted.</p> </div>
Description	The description of the JDBC connection. This field is also visible when you register content.
Vault	<p>The vault whose secrets you want to use to fill out fields.</p> <p>This field is only available if one or more vaults have been configured for your Edge site.</p>
Connection parameters	This section contains general settings to connect to your data source.
Driver class name	The driver class name of the connection.
Driver jar	<p>The JAR file contains the JDBC driver.</p> <p>Click Upload to upload a JAR file.</p>
Additional classpath files	Any additional classpath files that you want to upload. Use this field if you want to upload more than one driver file.

Field	Description
Connection string	<p>The JDBC connection string.</p> <p>How to use your vault...</p> <p>To use your vault, do the following:</p> <ol style="list-style-type: none">In the Value Type field, select Vault Key.Enter the query value to identify the secret in your vault. <div><div><h3>Example</h3><div><div>Username *</div><div><div>Vault Key ▾</div><div><div>Query</div></div></div><div>CyberArk Credential Provider [Query*] The username to use to establish a connection</div><div>Password *</div><div><div>Vault Key ▾</div><div><div>Query</div></div></div><div>CyberArk Credential Provider [Query*] The password to use to establish a connection</div></div></div></div>


- To use your vault, do the following:
- In the **Value Type** field, select **Vault Key**.

Field	Description										
b. Enter the required information:											
	<table><tr><th>Name</th><th>Description</th></tr><tr><td>Secret Engine Type</td><td>Select one of the following:<ul style="list-style-type: none">▪ Key Value▪ Database</td></tr><tr><td>Engine Path</td><td>The engine path to your vault where the value is stored.</td></tr><tr><td>Secret Path</td><td>The secret path to your vault where the value is stored.</td></tr><tr><td>Field</td><td>The name of the field to your vault where the value is stored.</td></tr></table>	Name	Description	Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database	Engine Path	The engine path to your vault where the value is stored.	Secret Path	The secret path to your vault where the value is stored.	Field	The name of the field to your vault where the value is stored.
Name	Description										
Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database										
Engine Path	The engine path to your vault where the value is stored.										
Secret Path	The secret path to your vault where the value is stored.										
Field	The name of the field to your vault where the value is stored.										
	<div>Note Only available if you selected Key Value in the Secret Engine Type field.</div>										
Role	The role specified in the Database engine.										
	<div>Note Only available if you selected Database in the Secret Engine Type field.</div>										

Example

Username *

Vault Key

 Key Value

Engine Path

Secret Path


Field

HashiCorp Vault (Engine Path*) (Secret Path*) (Field*)

The username to use to establish a connection

Password *

Vault Key

 Database

Engine Path

Role

username

HashiCorp Vault (Engine Path*) (Role*) (Field*)

The password to use to establish a connection

- To use your vault, do the following:
- In the **Value Type** field, select **Vault Key**.

Field	Description
b. Enter the required information:	
Name	Description
Vault Name	The name of your Azure Key Vault in your Azure Key Vault service where the value is stored.
Secret Name	The name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The username to use to establish a connection

Password *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The password to use to establish a connection

- To use your vault, do the following:
- a. In the **Value Type** field, select **Vault Key**.
 - b. Enter the required information:

Name	Description
Secret Name	The name of the secret in your vault where the value is stored.
Field	If the secret stored in your AWS Secrets Manager is a JSON value, for example <code>{"pass1": "my-password", "pass2": "my-password2"}</code> , then you need to specify the Field to point to the exact JSON value that should be used. For example, Secret Name: edge-db-customer; Field: pass.


Note If the secret stored in your AWS Secrets Manager is a plain string value, for example my-password, then you do not need to specify the **Field**.

Field

Description


Example

Username *

Vault Key  Secret Name Field

AWS Secrets Manager [Secret Name*] [Field*]
The username to use to establish a connection

Password *

Vault Key  Secret Name Field


AWS Secrets Manager [Secret Name*] [Field*]
The password to use to establish a connection

To use your vault, do the following:

- In the **Value Type** field, select **Vault Key**.
- Enter the name of the secret in your vault where the value is stored.


Example

Username *

Vault Key  Secret Name

GCP Secret Manager [Secret Name*]
The username to use to establish a connection

Password *

Vault Key  Secret Name

GCP Secret Manager [Secret Name*]
The password to use to establish a connection

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.

Field	Description
Property	<p>This section contains the connection properties.</p> <p>How to use your vault...</p> <p>To use your vault, do the following:</p> <ol style="list-style-type: none">In the Value Type field, select Vault Key.Enter the query value to identify the secret in your vault. <div><div><h3>Example</h3><div><div>Username *</div><div><div>Vault Key ▾</div><div><div>Query</div></div></div><div>CyberArk Credential Provider [Query*] The username to use to establish a connection</div><div>Password *</div><div><div>Vault Key ▾</div><div><div>Query</div></div></div><div>CyberArk Credential Provider [Query*] The password to use to establish a connection</div></div></div></div>


- To use your vault, do the following:
- In the **Value Type** field, select **Vault Key**.

Field	Description										
b. Enter the required information:											
	<table><tr><th>Name</th><th>Description</th></tr><tr><td>Secret Engine Type</td><td>Select one of the following:<ul style="list-style-type: none">▪ Key Value▪ Database</td></tr><tr><td>Engine Path</td><td>The engine path to your vault where the value is stored.</td></tr><tr><td>Secret Path</td><td>The secret path to your vault where the value is stored.</td></tr><tr><td>Field</td><td>The name of the field to your vault where the value is stored.</td></tr></table>	Name	Description	Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database	Engine Path	The engine path to your vault where the value is stored.	Secret Path	The secret path to your vault where the value is stored.	Field	The name of the field to your vault where the value is stored.
Name	Description										
Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database										
Engine Path	The engine path to your vault where the value is stored.										
Secret Path	The secret path to your vault where the value is stored.										
Field	The name of the field to your vault where the value is stored.										
	<div>Note Only available if you selected Key Value in the Secret Engine Type field.</div>										
Role	The role specified in the Database engine.										
	<div>Note Only available if you selected Database in the Secret Engine Type field.</div>										

Example

Username *

Vault Key



Key Value

Engine Path

Secret Path


Field

HashiCorp Vault (Engine Path*) (Secret Path*) (Field*)

The username to use to establish a connection

Password *

Vault Key



Database

Engine Path

Role

username

HashiCorp Vault (Engine Path*) (Role*) (Field*)

The password to use to establish a connection

- To use your vault, do the following:
- In the **Value Type** field, select **Vault Key**.

Field	Description
b. Enter the required information:	
Name	Description
Vault Name	The name of your Azure Key Vault in your Azure Key Vault service where the value is stored.
Secret Name	The name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The username to use to establish a connection

Password *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The password to use to establish a connection

- To use your vault, do the following:
- a. In the **Value Type** field, select **Vault Key**.
 - b. Enter the required information:

Name	Description
Secret Name	The name of the secret in your vault where the value is stored.
Field	If the secret stored in your AWS Secrets Manager is a JSON value, for example <code>{"pass1": "my-password", "pass2": "my-password2"}</code> , then you need to specify the Field to point to the exact JSON value that should be used. For example, Secret Name: <code>edge-db-customer</code> ; Field: <code>pass</code> .

Note If the secret stored in your AWS Secrets Manager is a plain string value, for example `my-password`, then you do not need to specify the **Field**.

Field	Description
-------	-------------

Example

Username *

Vault Key ▾

Secret Name Field

AWS Secrets Manager [Secret Name*] [Field*]
The username to use to establish a connection

Password *

Vault Key ▾

Secret Name Field

AWS Secrets Manager [Secret Name*] [Field*]
The password to use to establish a connection

- To use your vault, do the following:
- a. In the **Value Type** field, select **Vault Key**.
 - b. Enter the name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▾

Secret Name

GCP Secret Manager [Secret Name*]
The username to use to establish a connection

Password *

Vault Key ▾

Secret Name

GCP Secret Manager [Secret Name*]
The password to use to establish a connection

5. 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** drop-down list on the **Register a Data Source** dialog box.

You can use the `Other` connection property and set the value to `CustomizedDefaultCatalogName=<custom database name>` to customize the database name when you connect to your data source. Collibra then uses this customized database name when you register the data source via Edge. To use this property and value, you must use a Collibra-provided driver that is newer than version 23.0.8409. For details about specifying the `CustomizedDefaultCatalogName=<custom database name>` value in the `Other` connection property for each data source, go to [Overview of Catalog connectors](#).

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=<custom database name>` value in the `Other` 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=<custom database name>` value in the `Other` connection property to the JDBC connection after the database was listed for the first time in the **Database** drop-down list on the **Register a Data Source** dialog box, both the new database name and `CData` will appear in the **Database** drop-down list. 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:

- 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.

If you created a Generic JDBC Connection, you can edit your connection to use a different authentication method. However, if you initially use the Username/Password JDBC connection type and want to change to another authentication method, you must [create a new connection](#) by using the Generic JDBC Connection.

You can then [register the data source via Edge](#).

Note If you're using a Collibra Cloud site, go the [Collibra Cloud site documentation](#) to check if your data source is supported.

Prerequisites

- If required, you have created a [JDBC connection](#).
- You have a [global role](#) that has the **Product Rights > System administration** [global permission](#).
- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#).
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud site](#).

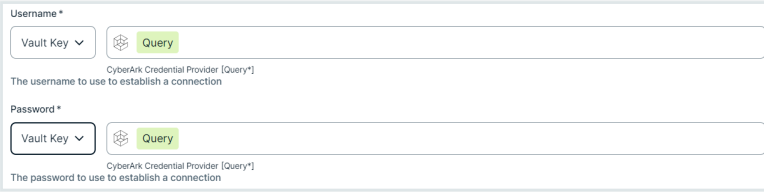
Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the JDBC Connections section, click the name of a JDBC connection.
 - » The **Edit Connection** dialog box appears.
3. At the bottom of the dialog box, 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	<p>The name of the JDBC connection.</p> <div> <p>Note We recommend not to use the special character > in the JDBC connection name. This character is part of the full name of assets created via Edge. If you use this character, features such as sampling or classification can be impacted.</p> </div>
Description	The description of the JDBC connection. This field is also visible when you register content.
Vault	<p>The vault whose secrets you want to use to fill out fields.</p> <p>This field is only available if one or more vaults have been configured for your Edge site.</p>
Connection parameters	This section contains general settings to connect to your data source.
Driver class name	The driver class name of the connection.
Driver jar	<p>The JAR file contains the JDBC driver.</p> <p>Click Upload to upload a JAR file.</p>
Additional classpath files	Any additional classpath files that you want to upload. Use this field if you want to upload more than one driver file.

Field	Description
Connection string	<p>The JDBC connection string.</p> <p>How to use your vault...</p> <p>To use your vault, do the following:</p> <ol style="list-style-type: none">In the Value Type field, select Vault Key.Enter the query value to identify the secret in your vault. <div><p>Example</p></div>


- To use your vault, do the following:
- In the **Value Type** field, select **Vault Key**.

Field	Description										
b. Enter the required information:											
	<table><tr><th>Name</th><th>Description</th></tr><tr><td>Secret Engine Type</td><td>Select one of the following:<ul style="list-style-type: none">▪ Key Value▪ Database</td></tr><tr><td>Engine Path</td><td>The engine path to your vault where the value is stored.</td></tr><tr><td>Secret Path</td><td>The secret path to your vault where the value is stored.</td></tr><tr><td>Field</td><td>The name of the field to your vault where the value is stored.</td></tr></table>	Name	Description	Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database	Engine Path	The engine path to your vault where the value is stored.	Secret Path	The secret path to your vault where the value is stored.	Field	The name of the field to your vault where the value is stored.
Name	Description										
Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database										
Engine Path	The engine path to your vault where the value is stored.										
Secret Path	The secret path to your vault where the value is stored.										
Field	The name of the field to your vault where the value is stored.										
	<div>Note Only available if you selected Key Value in the Secret Engine Type field.</div>										
Role	The role specified in the Database engine.										
	<div>Note Only available if you selected Database in the Secret Engine Type field.</div>										

Example

Username *

Vault Key



Key Value

Engine Path

Secret Path


Field

HashiCorp Vault (Engine Path*) (Secret Path*) (Field*)

The username to use to establish a connection

Password *

Vault Key



Database

Engine Path

Role

username

HashiCorp Vault (Engine Path*) (Role*) (Field*)

The password to use to establish a connection

- To use your vault, do the following:
- In the **Value Type** field, select **Vault Key**.

Field	Description
b. Enter the required information:	
Name	Description
Vault Name	The name of your Azure Key Vault in your Azure Key Vault service where the value is stored.
Secret Name	The name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The username to use to establish a connection

Password *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The password to use to establish a connection

- To use your vault, do the following:
- a. In the **Value Type** field, select **Vault Key**.
 - b. Enter the required information:

Name	Description
Secret Name	The name of the secret in your vault where the value is stored.
Field	If the secret stored in your AWS Secrets Manager is a JSON value, for example <code>{"pass1": "my-password", "pass2": "my-password2"}</code> , then you need to specify the Field to point to the exact JSON value that should be used. For example, Secret Name: edge-db-customer; Field: pass.


Note If the secret stored in your AWS Secrets Manager is a plain string value, for example my-password, then you do not need to specify the **Field**.

Field

Description


Example

Username *

Vault Key  Secret Name Field

AWS Secrets Manager [Secret Name*] [Field*]
The username to use to establish a connection

Password *

Vault Key  Secret Name Field


AWS Secrets Manager [Secret Name*] [Field*]
The password to use to establish a connection

To use your vault, do the following:

- In the **Value Type** field, select **Vault Key**.
- Enter the name of the secret in your vault where the value is stored.


Example

Username *

Vault Key  Secret Name

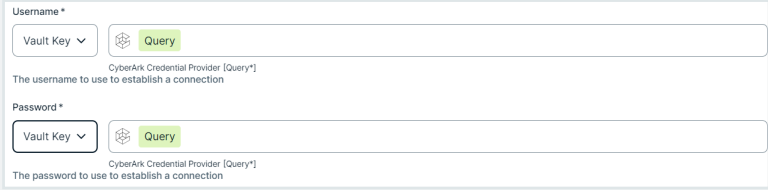
GCP Secret Manager [Secret Name*]
The username to use to establish a connection

Password *

Vault Key  Secret Name

GCP Secret Manager [Secret Name*]
The password to use to establish a connection

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.

Field	Description
Property	<p>This section contains the connection properties.</p> <p>How to use your vault...</p> <p>To use your vault, do the following:</p> <ol style="list-style-type: none">In the Value Type field, select Vault Key.Enter the query value to identify the secret in your vault. <div><p>Example</p></div>

To use your vault, do the following:


- In the **Value Type** field, select **Vault Key**.

Field	Description										
b. Enter the required information:											
	<table><tr><th>Name</th><th>Description</th></tr><tr><td>Secret Engine Type</td><td>Select one of the following:<ul style="list-style-type: none">▪ Key Value▪ Database</td></tr><tr><td>Engine Path</td><td>The engine path to your vault where the value is stored.</td></tr><tr><td>Secret Path</td><td>The secret path to your vault where the value is stored.</td></tr><tr><td>Field</td><td>The name of the field to your vault where the value is stored.</td></tr></table>	Name	Description	Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database	Engine Path	The engine path to your vault where the value is stored.	Secret Path	The secret path to your vault where the value is stored.	Field	The name of the field to your vault where the value is stored.
Name	Description										
Secret Engine Type	Select one of the following: <ul style="list-style-type: none">▪ Key Value▪ Database										
Engine Path	The engine path to your vault where the value is stored.										
Secret Path	The secret path to your vault where the value is stored.										
Field	The name of the field to your vault where the value is stored.										
	<div>Note Only available if you selected Key Value in the Secret Engine Type field.</div>										
Role	The role specified in the Database engine.										
	<div>Note Only available if you selected Database in the Secret Engine Type field.</div>										

Example

Username *

Vault Key



Key Value

Engine Path

Secret Path


Field

HashiCorp Vault (Engine Path*) (Secret Path*) (Field*)

The username to use to establish a connection

Password *

Vault Key



Database

Engine Path

Role

username

HashiCorp Vault (Engine Path*) (Role*) (Field*)

The password to use to establish a connection

- To use your vault, do the following:
- In the **Value Type** field, select **Vault Key**.

Field	Description
b. Enter the required information:	
Name	Description
Vault Name	The name of your Azure Key Vault in your Azure Key Vault service where the value is stored.
Secret Name	The name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The username to use to establish a connection

Password *

Vault Key ▼ Vault Name Secret Name

Azure Key Vault [Vault Name*] [Secret Name*]
The password to use to establish a connection

- To use your vault, do the following:
- a. In the **Value Type** field, select **Vault Key**.
 - b. Enter the required information:

Name	Description
Secret Name	The name of the secret in your vault where the value is stored.
Field	If the secret stored in your AWS Secrets Manager is a JSON value, for example <code>{"pass1": "my-password", "pass2": "my-password2"}</code> , then you need to specify the Field to point to the exact JSON value that should be used. For example, Secret Name: edge-db-customer; Field: pass.

Note If the secret stored in your AWS Secrets Manager is a plain string value, for example my-password, then you do not need to specify the **Field**.

Field	Description
-------	-------------

Example

Username *

Vault Key ▼ [Secret Name] Field

AWS Secrets Manager [Secret Name*] [Field*]
The username to use to establish a connection

Password *

Vault Key ▼ [Secret Name] Field

AWS Secrets Manager [Secret Name*] [Field*]
The password to use to establish a connection

To use your vault, do the following:

- In the **Value Type** field, select **Vault Key**.
- Enter the name of the secret in your vault where the value is stored.

Example

Username *

Vault Key ▼ [Secret Name]

GCP Secret Manager [Secret Name*]
The username to use to establish a connection

Password *

Vault Key ▼ [Secret Name]

GCP Secret Manager [Secret Name*]
The password to use to establish a connection

- Click **Save**.
- If required, you can now test the connection.
 - At the bottom of the page, click **Test connection**.
 - » The **Connection test** dialog box appears.
 - 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 or Collibra Cloud site to a data source if you no longer need it.

Prerequisites

- You have a [global role](#) that has the **Product Rights > System administration** [global permission](#).
- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#).
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud site](#).
- You have created a [JDBC connection](#).

Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the JDBC Connections section, click the name of a JDBC connection.
 - » The **Edit Connection** dialog box appears.
3. At the bottom of the dialog box, click **Delete**.
 - » The **Confirm Deletion** dialog box appears.
4. Click **Delete**.

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.

The screenshot shows a 'Create Connection' dialog box with the following sections:

- Additional classpath files:** A dashed box with the text 'Upload a file or drop your file(s) here' and a note 'The optional list of additional classpath files'.
- Connection string *:** A text input field containing 'jdbc:snowflake://<url>' with a note 'The jdbc connection string' below it.
- Property:** A table-like structure for defining connection properties.

Type *	Value Type *	Name *	Value *
File ▾	Plaintext ▾	private_key_file	Upload a file or drop your file(s) here snowflake.pk
Text ▾	Secret ▾	private_key_file_pwd

At the bottom of the 'Property' section is a '+ Add Property' button. At the bottom of the dialog are '<' and 'Cancel' buttons, and a 'Create' button.

Edge and Collibra Cloud site capabilities

A capability is an application that runs on an Edge or Collibra Cloud site to extract and process data. It delivers the results to Collibra Platform.



About Edge and Collibra Cloud site capabilities

A capability, like Sampling or S3 synchronization, is an application that can run on an Edge or Collibra Cloud 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. A capability for a specific data source runs as a job and delivers the output to Collibra Platform in a secure and reliable way.

A 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 capability.

Note If there is an integration you want that is not listed below, contact your Account Executive for more options.

Capability template	Description	Supported for Edge sites?	Supported for Collibra Cloud sites?
ADLS synchronization	Used to connect to Azure Data Lake Storage (ADLS)	✓ Yes	✓ Yes
AWS Bedrock AI	Used to integrate with Amazon Bedrock . This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes
AWS SageMaker AI	Used to integrate with Amazon SageMaker . This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes

Capability template	Description	Supported for Edge sites?	Supported for Colibra Cloud sites?
Azure AI Foundry	Used to integrate with Azure AI Foundry . This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes
Azure ML	Used to integrate with Microsoft Azure AI . This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes
Catalog Data Classification	Used to classify data from a registered JDBC data source in the site. This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes, but limited. <ul style="list-style-type: none"> • Amazon Redshift (JDBC) • Athena (JDBC) • Databricks (JDBC) • Databricks Unity Catalog • Google BigQuery (JDBC) • Salesforce (JDBC) • Snowflake (JDBC) • S3 (JDBC)

Capability template	Description	Supported for Edge sites?	Supported for Col- libra Cloud sites?
Catalog JDBC ingestion	<p>Used to register a data source and synchronize schemas from a data source via a JDBC connection.</p> <p>This capability can't be added to an Edge site that uses a MITM proxy.</p>	✓ Yes	<p>✓ Yes, but limited.</p> <ul style="list-style-type: none"> • Amazon Redshift (JDBC) • Athena (JDBC) • Azure Data Lake Storage • Azure Synapse Analytics • Databricks Unity Catalog • Databricks (JDBC) • Google BigQuery (JDBC) • Google Cloud Storage • Google Dataplex • Salesforce (JDBC) • SAP Datasphere Catalog • SAP HANA Cloud/Advanced • Snowflake (JDBC) • S3 • S3 (JDBC)
Catalog JDBC Sampling	<p>Used to collect and cache sample data from a data source in the site via a JDBC connection.</p> <p>Ensure that you meet the additional Catalog JDBC Sampling hardware requirements, in addition to the Edge site requirements.</p> <p>This capability can't be added to an Edge site that uses a MITM proxy.</p>	✓ Yes	✗ No

Capability template	Description	Supported for Edge sites?	Supported for Colibra Cloud sites?
Colibra 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 .	✓ Yes	✓ Yes
Colibra Protect for Databricks	Used to set up Protect for Databricks. This capability appears only if the following parameter is added to the JVM configuration in Colibra Console: – <code>Dfeature.protect.databricks=true</code> This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes
Colibra 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 .	✓ Yes	✓ Yes
Colibra 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 .	✓ Yes	✓ Yes
DQ Pushdown Capability	Used to run Data Quality & Observability Pushdown jobs on data sources via Edge. This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes, but limited. <ul style="list-style-type: none"> • Amazon Redshift (JDBC) • Athena (JDBC) • Databricks (JDBC) • Google BigQuery (JDBC) • SAP HANA Cloud/Advanced • Snowflake (JDBC)

Capability template	Description	Supported for Edge sites?	Supported for Colibra Cloud sites?
GCS synchronization	Used to connect to Google Cloud Storage .	✓ Yes	✓ Yes
Google Dataplex Catalog synchronization	Used to connect to Google Dataplex Catalog. The way to complete this capability depends on the Dataplex integration type you want to use: Dataplex ingestion or Dataplex Catalog ingestion .	✓ Yes	✓ Yes
Google Vertex AI	Used to integrate Google Vertex AI. This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes
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 .	✓ Yes	✓ Yes, but limited. <ul style="list-style-type: none"> • Amazon Redshift (JDBC) • Athena (JDBC) • Databricks (JDBC) • Databricks Unity Catalog • Google BigQuery (JDBC) • Salesforce (JDBC) • Snowflake (JDBC) • S3 (JDBC)
MLflow AI	Used to integrate MLflow. This capability can't be added to an Edge site that uses a MITM proxy .	✓ Yes	✓ Yes
SAP AI Core	Used to integrate with SAP AI Core .	✓ Yes	✓ Yes

Capability template	Description	Supported for Edge sites?	Supported for Colibra Cloud sites?
S3 synchronization	Used to connect to Amazon S3 .	✓ Yes	✓ Yes
Databricks Unity Catalog synchronization	Used to connect to Databricks Unity Catalog .	✓ Yes	✓ Yes
Technical Lineage Admin	<p>Used to run any of the following technical lineage admin options:</p> <ul style="list-style-type: none"> • List sources • Ignore sources • Analyze files • Sync 	✓ Yes	✓ Yes

Capability template	Description	Supported for Edge sites?	Supported for Colibra Cloud sites?
Technical lineage capabilities	<p>Used to create technical lineage for different data sources. For details, go to: Add a technical lineage capability to an Edge site.</p> <p>Ensure that you meet the additional Technical Lineage minimum network requirements, in addition to the Edge site requirements.</p> <p>Technical Lineage requirements...</p> <p>Firewall rules so that the lineage harvester can connect to:</p> <ul style="list-style-type: none"> The host names of all data sources in your lineage harvester configuration file. 	✓ Yes	<p>✓ Yes, but limited.</p> <ul style="list-style-type: none"> Amazon Redshift (JDBC) Azure SQL server Databricks Unity Catalog Google BigQuery (JDBC) Google Dataplex Power BI SAP HANA Cloud/Advanced Snowflake (JDBC) Tableau

Capability template	Description	Supported for Edge sites?	Supported for Colibra Cloud sites?																								
	<ul style="list-style-type: none">All Colibra Data Lineage service instances in your geographic location: <table><tr><th>Region</th><th>DNS name</th></tr><tr><td>aws-ca</td><td>techlin-ca-central-1.colibra.com</td></tr><tr><td>aws-eu</td><td>techlin-eu-central-1.colibra.com</td></tr><tr><td>aws-me</td><td>techlin-me-central-1.colibra.com</td></tr><tr><td>aws-sg</td><td>techlin-ap-southeast-1.colibra.com</td></tr><tr><td>aws-us</td><td>techlin-us-east-1.colibra.com</td></tr><tr><td>gcp-au</td><td>techlin-australia-southeast1.colibra.com</td></tr><tr><td>gcp-ca</td><td>techlin-northamerica-northeast1.colibra.com</td></tr><tr><td>gcp-eu</td><td>techlin-europe-west-1.colibra.com</td></tr><tr><td>gcp-sg</td><td>techlin-asia-southeast1.colibra.com</td></tr><tr><td>gcp-uk</td><td>techlin-europe-west-2.colibra.com</td></tr><tr><td>gcp-us</td><td>techlin-us-east-1.colibra.com</td></tr></table> <p>We recommend that you only use DNS names in your network configurations, as the IP addresses are subject to</p>	Region	DNS name	aws-ca	techlin-ca-central-1.colibra.com	aws-eu	techlin-eu-central-1.colibra.com	aws-me	techlin-me-central-1.colibra.com	aws-sg	techlin-ap-southeast-1.colibra.com	aws-us	techlin-us-east-1.colibra.com	gcp-au	techlin-australia-southeast1.colibra.com	gcp-ca	techlin-northamerica-northeast1.colibra.com	gcp-eu	techlin-europe-west-1.colibra.com	gcp-sg	techlin-asia-southeast1.colibra.com	gcp-uk	techlin-europe-west-2.colibra.com	gcp-us	techlin-us-east-1.colibra.com		
Region	DNS name																										
aws-ca	techlin-ca-central-1.colibra.com																										
aws-eu	techlin-eu-central-1.colibra.com																										
aws-me	techlin-me-central-1.colibra.com																										
aws-sg	techlin-ap-southeast-1.colibra.com																										
aws-us	techlin-us-east-1.colibra.com																										
gcp-au	techlin-australia-southeast1.colibra.com																										
gcp-ca	techlin-northamerica-northeast1.colibra.com																										
gcp-eu	techlin-europe-west-1.colibra.com																										
gcp-sg	techlin-asia-southeast1.colibra.com																										
gcp-uk	techlin-europe-west-2.colibra.com																										
gcp-us	techlin-us-east-1.colibra.com																										

Capability template	Description	Supported for Edge sites?	Supported for Colibra Cloud sites?
	<p>change periodically. If you need to use IP addresses in your network configuration, we recommend using a command line utility like nslookup to query the DNS and obtain the mapping between domain name and IP address.</p> <div> <p>Note Edge connects to different Colibra 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 Colibra 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 Colibra Data Lineage service instances.</p> </div> <p>You can use a man-in-the-middle (MITM) proxy between your Edge site and the Colibra 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.</p>		

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

Capability template structure

Each 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 preparing an Edge or Collibra Cloud site for data sources

After you create an Edge or request a Collibra Cloud site, you can start creating connections to your data sources. You can then add capabilities that use these connections to get information from the data sources to Collibra.

Typically, you create a connection for a data source and add capabilities for this connection. It is important to have a connection set up with the correct information and add the correct capabilities based on the data source and your needs. Each connection and capability may have slightly different steps or requirements, so be sure to review the data source specific information.

For example, you set up a PostgreSQL data source connection. This is a [JDBC connection](#). You want to integrate the metadata in Collibra, profile the data, and get samples. For the connection to be able to do this, you need to add the [Catalog JDBC ingestion](#) capability, [JDBC Profiling](#) capability, and [Catalog JDBC Sampling](#) capability for the connection.

JDBC (Java Database Connectivity) integrations allow you to connect directly to your data source from your Edge or Collibra Cloud site. When you create a JDBC connection, you will enter your login credentials, which will then be stored for authentication. This means that you don't 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 or Collibra Cloud site. The connection information is defined and stored as a Connection instance. The connection properties are shown on an Edge or Collibra Cloud site's **Connections** tab.

Steps

See a general overview of the Edge and Collibra Cloud site integration process below:

- 1 Create a connection. A connection links your Edge or Collibra Cloud site with your data source, whether that be a database, file share, or REST service. The subsequent capability jobs that are run through this connection send information back to your Collibra Platform.

For more information, go to our list of [available Edge and Collibra Cloud site connections](#).
- 2 Create a capability. A capability calls to your data source, and sends the metadata back to your Collibra Platform. The end results are your assets, schemas, tables, and so on.

For more information, go to our list of [available Edge and Collibra Cloud site capabilities](#).

What's next?

Once your Edge or Collibra Cloud site is prepared, you can use the capabilities. In most cases, you need to first make sure metadata is available in Collibra. The way to do this differs depending on your data source.

- For JDBC connections:
 - When you create a [JDBC connection](#) to a data source, you must first [register](#) the data source in your Collibra Platform. This creates a Database asset that you then need to [synchronize](#). The synchronization process ingests metadata from the data source into Collibra. This results in assets with information, such as Schema assets, Tables assets, and so on. Collibra does not include the actual data from the data source, only the data about the data. This full flow is called register a data source. For more information, go to [About registering a data source](#).
- For non-JDBC connections:
 - When you create any other kind of connection, you only need to synchronize the data source in your Collibra Platform. The synchronization process ingests metadata from the data source into Collibra. This results in assets with information, such as Schema assets, Tables assets, and so on. And creates a structure of the assets that represents the structure in the data source. For more information about synchronizing non-JDBC integrations, go to the [data source specific documentation](#).

Add a capability to an Edge or Collibra Cloud site

After you have created and installed an [Edge site](#) or [requested a Collibra Cloud site](#), you can add an [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 capability.

Prerequisites

- You have a [global role](#) that has the **Product Rights > System administration** [global permission](#).
- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#), for example, **Edge integration engineer**.
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud 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](#).

Note If you're using a Collibra Cloud site, go the [Collibra Cloud site documentation](#) to check if your data source is supported.

More information

[ADLS integration](#)

[Catalog Data Classification](#)

[Catalog JDBC ingestion](#)

[JDBC Profiling](#)

Catalog JDBC Sampling

S3 synchronization

GCS synchronization

Databricks Unity Catalog integration

DQ Connector

Technical lineage via Edge

Protect for AWS Lake Formation

Protect for BigQuery

Protect for Databricks

Protect for Snowflake

Azure AI Foundry

Microsoft Azure AI

AWS SageMaker AI

AWS Bedrock AI

MLflow AI

SAP AI Core

SAP Datasphere integration

Google Dataplex integration

Google Dataplex Catalog integration

Edit an Edge or Collibra Cloud site site capability

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

Prerequisites

- You have a [global role](#) that has the **Product Rights > System administration** [global permission](#).
- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#), for example, **Edge integration engineer**.
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud site](#).
- You have added a [capability](#) to the site.

Steps

Tip For information about the various capabilities, go to the [online version of the documentation](#).

Note If you're using a Collibra Cloud site, go the [Collibra Cloud site documentation](#) to check if your data source is supported.

Delete a capability from an Edge or Collibra Cloud site

You can remove an [capability](#) from an Edge or Collibra Cloud site [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 **Product Rights > System administration** [global permission](#).
- You have a [global role](#) that has the **Manage connections and capabilities** [global permission](#), for example, **Edge integration engineer**.
- You either created and installed an [Edge site](#) or were granted a [Collibra Cloud site](#).
- You have added a [capability](#) to the site.

Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the site overview, click the name of a site.
 - » The site page appears.
2. In the **Capabilities** section, click the name of a capability.
 - » The **Capability** page appears and shows a read-only overview of the capability.
3. Click **Delete**.
4. Click **Delete**.
 - » The capability is deleted from the site.

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 (in preview) in Colibra Console, the Edge Jobs dashboard becomes available in the Colibra Platform settings.

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

Important This is a [preview feature](#).

Edge

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

[Sites](#) | [Jobs](#)

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.

Colibra

Settings

General

Operating Model

Roles and Permissions

Users and Groups

Services Configuration

Migration

Logs

Edge

Sites

Jobs

This feature is being actively developed and may not meet all necessary criteria for enterprise-grade functionality. In the future, the beta label will be removed and this feature will be considered Generally Available.

Output Files

Cancel Job

Job ID	Capability	Edge Site	Status	Started ↓	Completed	Duration
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>			
<input type="checkbox"/> 5784e8ed-6ef5-4c7e-9fc0-f2795fe4d620	Connection_Techlin	site01	Cancelled	-	14/12/2023, 13:03:59	-
<input type="checkbox"/> 19db6762-88bc-457c-81c0-0153ed873ca0	Connection_Techlin	site01	Succee...	14/12/2023, 13:01:18	14/12/2023, 13:01:32	0m 13s
<input type="checkbox"/> 70391614-cc8c-4c1e-8ad0-0d3ba323c17b	ing capa	test-site	Succee...	14/12/2023, 10:11:53	14/12/2023, 10:12:11	0m 17s
<input type="checkbox"/> 0898d0a1-37b6-4d07-8ba3-dee7ab67c68c	Connection_Techlin	site01	Succee...	-	12/12/2023, 12:50:15	-
<input type="checkbox"/> fbcaa55b-65a0-445c-a37f-73415bb32b21	Connection_Techlin	site01	Cancelled	-	12/12/2023, 12:49:15	-
<input type="checkbox"/> f9a07168-80fb-4018-966a-974f9e597f2a	Connection_Techlin	site01	Succee...	12/12/2023, 12:47:51	12/12/2023, 12:47:49	-1m -3s
<input type="checkbox"/> 69f87c58-24ae-4b78-95e6-24047dcff01a	ing capa	test-site	Succee...	07/12/2023, 18:06:42	07/12/2023, 18:07:03	0m 20s
<input type="checkbox"/> a774b7b6-9cb7-41e7-8386-ef77550dbe25	Capability_ingestion_1	site01	Succee...	-	07/12/2023, 10:42:05	-
<input type="checkbox"/> 52744215-b291-4592-b564-4004057e26fd	Capability_ingestion_1	site01	Cancelled	07/12/2023, 10:40:46	07/12/2023, 10:41:05	0m 19s
<input type="checkbox"/> fa6f7891-6a30-46c4-bc52-0d8f8e82467c	Capability_ingestion_1	site01	Succee...	-	07/12/2023, 10:40:34	-

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.

The screenshot shows the Collibra Edge site configuration page for 'test-site'. The 'Jobs' tab is selected, displaying a table of jobs. A sidebar on the left shows 'Sites' and 'Jobs' tabs. The main content area includes a 'Healthy' status indicator, 'Edge Site ID', 'Installed version', and 'Upgrade mode'. Below this is a tab menu with 'Connections', 'Capabilities', 'Vaults', 'Jobs', and 'History'. A warning message states: 'This feature is being actively developed and may not meet all necessary criteria for enterprise-grade functionality. In the future, the beta label will be removed and this feature will be considered Generally Available.' Below the warning are 'Output Files' and 'Cancel Job' buttons. The table below lists jobs with columns for Job ID, Capability, Status, Started, Completed, and Duration.

Job ID	Capability	Status	Started ↓	Completed	Duration
<input type="checkbox"/> 70391614-cc8c-4c1e-8ad0-0d3ba323c17b	ing capa	Succee...	14/12/2023, 10:11:53	14/12/2023, 10:12:11	0m 17s
<input type="checkbox"/> 69f87c58-24ae-4b78-95e6-24047dcff01a	ing capa	Succee...	07/12/2023, 18:06:42	07/12/2023, 18:07:03	0m 20s

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.

Review an Edge or Collibra Cloud site job details

When you run a site capability job, you may need to review key details for reporting or troubleshooting with support. The following information helps you identify the site jobs:

- Site ID: The identification number of the site that ran the job.
- Job ID: The identification number of the job.

Where do I find the Edge or Collibra Cloud site Site ID and Job ID?

To retrieve the Site ID:

1. Go to **Settings**.
2. In the **Edge** section, click **Sites**.
3. Click the name of the site.
4. The Site ID is available in the **ID** field.

Where do I find the Edge or Collibra Cloud site Job ID?

There are 3 locations where you can find the Job ID:

- The **Jobs dashboard** on Edge.

This feature is being actively developed and may not meet all necessary criteria for enterprise-grade functionality. In the future, the beta label will be removed and this feature will be considered Generally Available.

Output Files Cancel Job

Job ID	Capability	Edge Site	Status	Started ↓	Completed	Duration
<input type="checkbox"/> 4754345a-7740-4aa3-b1d3-06ab455856a2	excel-test-sampling	av-weekly-0306	Succeeded	10/03/2025, 20:36:43	10/03/2025, 20:36:54	0m 10s
<input type="checkbox"/> e996fad3-dbcd-454c-b335-3641018bcaa8	excel-test-ingestion	av-weekly-0306	Succeeded	10/03/2025, 20:32:43	10/03/2025, 20:32:58	0m 14s
<input type="checkbox"/> eecd8a8f-d452-4a02-a050-1a31d26da1f3	5f571da8-1941-4124-92ed-4423364dda9e	av-weekly-0306	Failed	-	10/03/2025, 20:27:42	-

- The **Job** tab of the site where the capability was run.

Sites

Jobs

Healthy Upgrade Available

Edge Site ID
6411ee47-35db-4b2b-9867-1d21c94d2b78

Description
[Redacted]

Installed version
[Redacted]

Upgrade mode
MANUAL

Site Actions

Connections Capabilities Vaults Jobs History

This feature is being actively developed and may not meet all necessary criteria for enterprise-grade functionality. In the future, the beta label will be removed and this feature will be considered Generally Available.

Output Files Cancel Job

☐ Job ID

☐ 60ad3328-9aa4-4575-a369-016e4df348ff

☐ d5fedc47-7ee7-4a9d-b27f-0bdf2ea3ac6d

Capability	Status	Started	Completed	Duration
sparkWorkerTest	Succeeded	06/03/2025, 19:05:27	06/03/2025, 19:06:45	1m 18s
sparkWorkerTest	Failed	-	06/03/2025, 19:07:25	-

- The **Synchronization Results** page in your Collibra Platform.

Download job output files

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

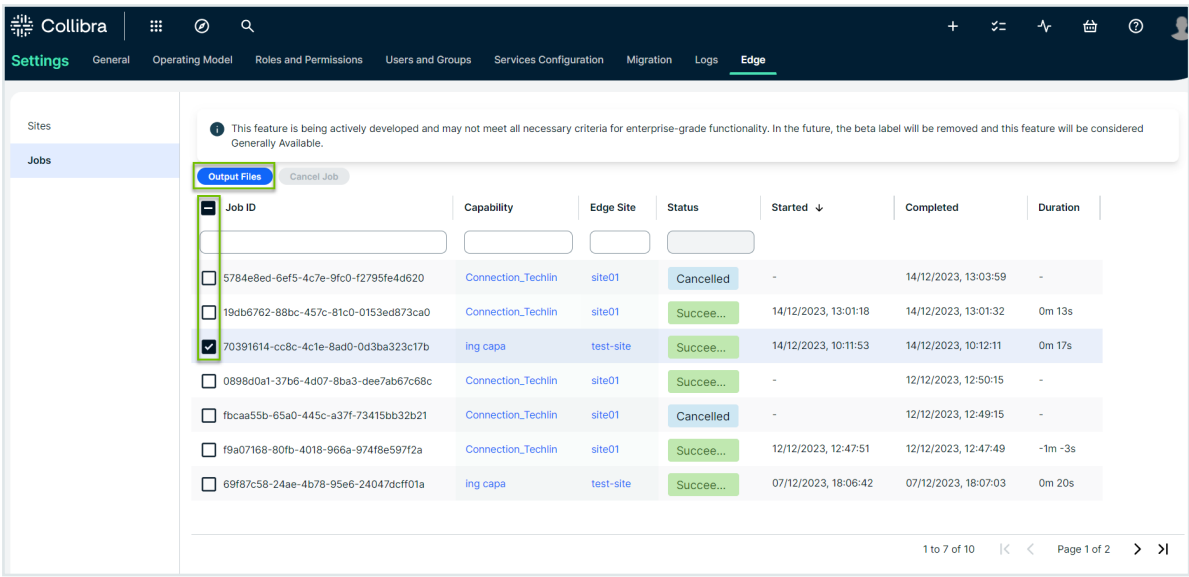
Prerequisites

- You have [Edge View Log permission](#).
- You have commercial JDBC jobs which have been completed.

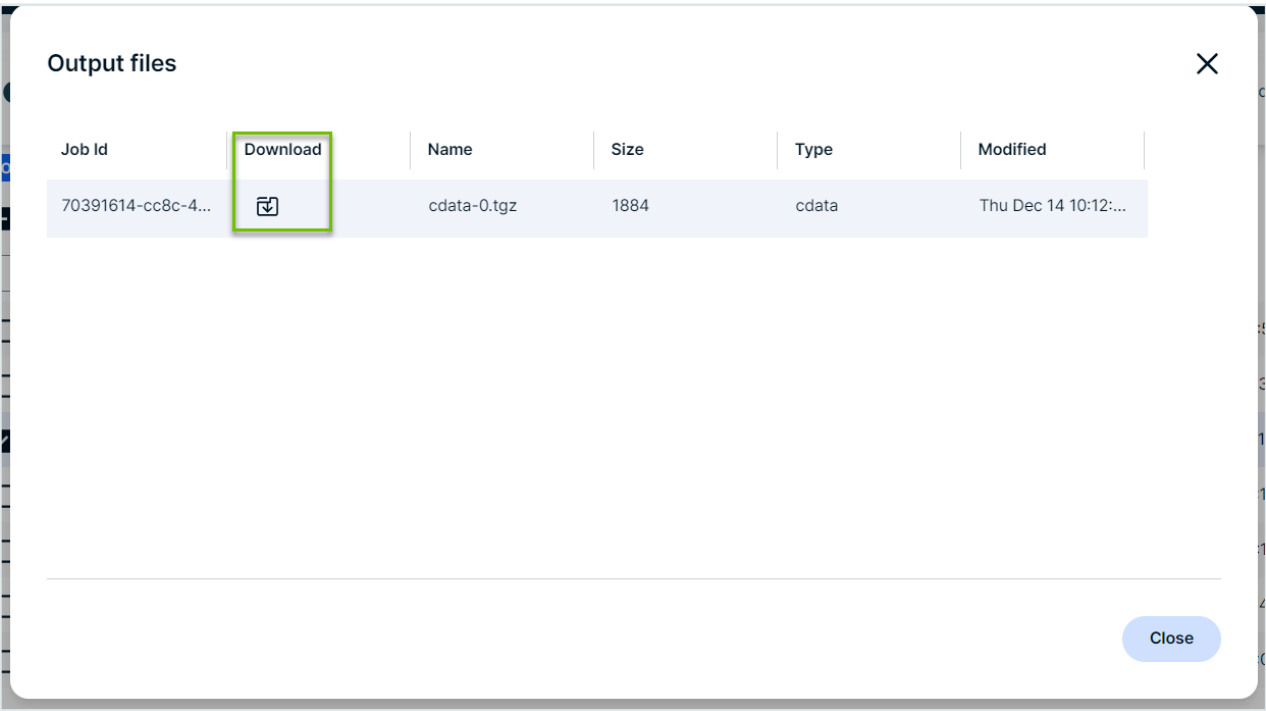
Steps

1. Open a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the table, click the name of the site whose status is **Healthy**.
 - » The 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 or Collibra Cloud 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 a site.
 - a. On the main toolbar, click → **Settings**.
 - » The [Settings page](#) opens.
 - b. In the tab pane, click **Edge**.
 - » The **Sites** tab opens and shows a table with an overview of your sites.
 - c. In the table, click the name of the site whose status is **Healthy**.
 - » The 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.

This feature is being actively developed and may not meet all necessary criteria for enterprise-grade functionality. In the future, the beta label will be removed and this feature will be considered Generally Available.

Output Files Cancel Job

Job ID	Capability	Edge Site	Status	Started ↓	Completed	Duration
<input checked="" type="checkbox"/> 9e486163-3072-473d-b59d-ab4f888296fc	capability	vaults	Running	20/12/2023, 14:26:36	-	0m 19s
<input type="checkbox"/> 155dc2fe-3450-4d12-af15-be846d015ac5	capability	vaults	Running	20/12/2023, 14:26:36	-	0m 19s
<input type="checkbox"/> 7d6cf7ba-0ca6-4496-b04c-413b6b01ea03	capability	vaults	Running	20/12/2023, 14:26:36	-	0m 19s
<input type="checkbox"/> 6440e9ab-3478-4426-937a-f2a8936e576f	capability	vaults	Cancelled	18/12/2023, 17:07:59	18/12/2023, 17:08:15	0m 15s
<input type="checkbox"/> 0cf240a3-34f6-48b3-bea2-8f92a924eff2	capability	vaults	Failed	18/12/2023, 17:07:59	18/12/2023, 17:16:00	8m 1s
<input type="checkbox"/> 8bd54c85-6298-400b-bbc9-55723b1df9a2	capability	vaults	Succes...	18/12/2023, 17:07:59	18/12/2023, 17:16:00	8m 1s
<input type="checkbox"/> 20cce2a4-7ba2-4e95-a6ee-7e02de16d25a	capability	vaults	Failed	18/12/2023, 17:07:59	18/12/2023, 17:08:59	1m 0s

1 to 7 of 19 | Page 1 of 3

4. In the action toolbar, click **Cancel Job**.
 - » The job is canceled, and the status of this job is **CANCELED**.