opentext™

OpenText™ Cloud Bridge Agent

Installation and Administration Guide

November 2024

Legal Notice

Contents

About This Book

The *Installation and Administration Guide* provides conceptual information about the OpenText™ Cloud Bridge Agent. This book includes conceptual information and step-by-step guidance for common tasks.

Intended Audience

This book provides information for individuals responsible for installing, configuring, and managing one or more OpenText Cloud Bridge Agents in a software-as-a-service (SaaS) environment. A working knowledge of network operations, network security, and cloud SaaS technologies is assumed.

Additional Documentation

For the most recent version of this guide and other OpenText™ Cloud Bridge Agent documentation resources, visit the Identity and Access Management Services Documentation web page (https://www.microfocus.com/documentation/identity-and-access-management/iam-services/).

Contact Information

We want to hear your comments and suggestions about this book and the other documentation included with this product. You can use the **comment on this topic** link at the bottom of each page of the online documentation, or send an email to Documentation-Feedback@microfocus.com.

For specific product issues, contact OpenText Support at https://www.microfocus.com/support-and-services/.

1 Introduction to OpenText Cloud Bridge

In OpenText software-as-a-service (SaaS) or hybrid environments, OpenText Cloud Bridge is a data transfer bridge between applications in the cloud and data sources on premises. The OpenText Cloud Bridge Agent ("the Agent") is the on-premises entity that responds to the collection and fulfillment commands and directs them to the proper data source within the multi-tenant OpenText Cloud Bridge service for execution. An on-premises administrator installs the Agent and configures it with the necessary service parameters, including locally-persisted and protected credentials for on-premises and third-party data sources.

Understanding the Benefits of OpenText Cloud Bridge

OpenText Cloud Bridge provides several important benefits. For more information, see the following topics:

OpenText Cloud Bridge Secures Communication

OpenText Cloud Bridge simplifies and secures communication between SaaS applications, such as OpenText™ Identity Governance or OpenText™ Advanced Authentication, and on-premises identity sources and applications, such as Microsoft® Active Directory™ or OpenText™ Identity Manager. The OpenText Cloud Bridge Agent communicates with the SaaS applications through a secure messaging service outside the corporate firewall. This messaging service is adaptable for various workloads and provides guaranteed delivery of messages. No VPN is needed and all OpenText Cloud Bridge Agent connections are outbound connections to a well-defined port. Data is protected both in transit and at rest.

In a common scenario, you might have both on-premises and SaaS products interacting with OpenText Cloud Bridge. For example, you have both on-premises OpenText Identity Manager and SaaS licenses for OpenText Advanced Authentication and OpenText Identity Governance. Your employees need to log in to their SaaS accounts as well as their on-premises applications. Your employees are authenticated through the OpenText Advanced Authentication SaaS service, which communicates with on-premises identity sources through an OpenText Cloud Bridge messaging layer.

After your OpenText Cloud Bridge Agent is installed and running in your on-premises environment, it begins sending heartbeat messages. The OpenText SaaS operations team sets up the necessary data protection features and monitors the health of your installed Agent.

OpenText Cloud Bridge Manages Credentials

The credential management feature in the OpenText Cloud Bridge Agent ensures that the credentials for a target data source never leave your network. The Agent associates the credentials with the service configuration on demand.

OpenText Cloud Bridge Ensures High Availability

High availability capabilities in OpenText Cloud Bridge also help you meet your organizational goals for operational performance. After you configure your environment to specify your preferred OpenText Cloud Bridge Agent sites and instances, when a planned or unplanned shutdown takes place, failover to the specified Agent site and instance occurs automatically and with minimal service interruption. As part of the failover process, the OpenText Cloud Bridge Client loads active service configurations previously used by the primary Agent instance into a new target Agent instance, enabling consuming applications to quickly resume their collection, provisioning, and other activities.

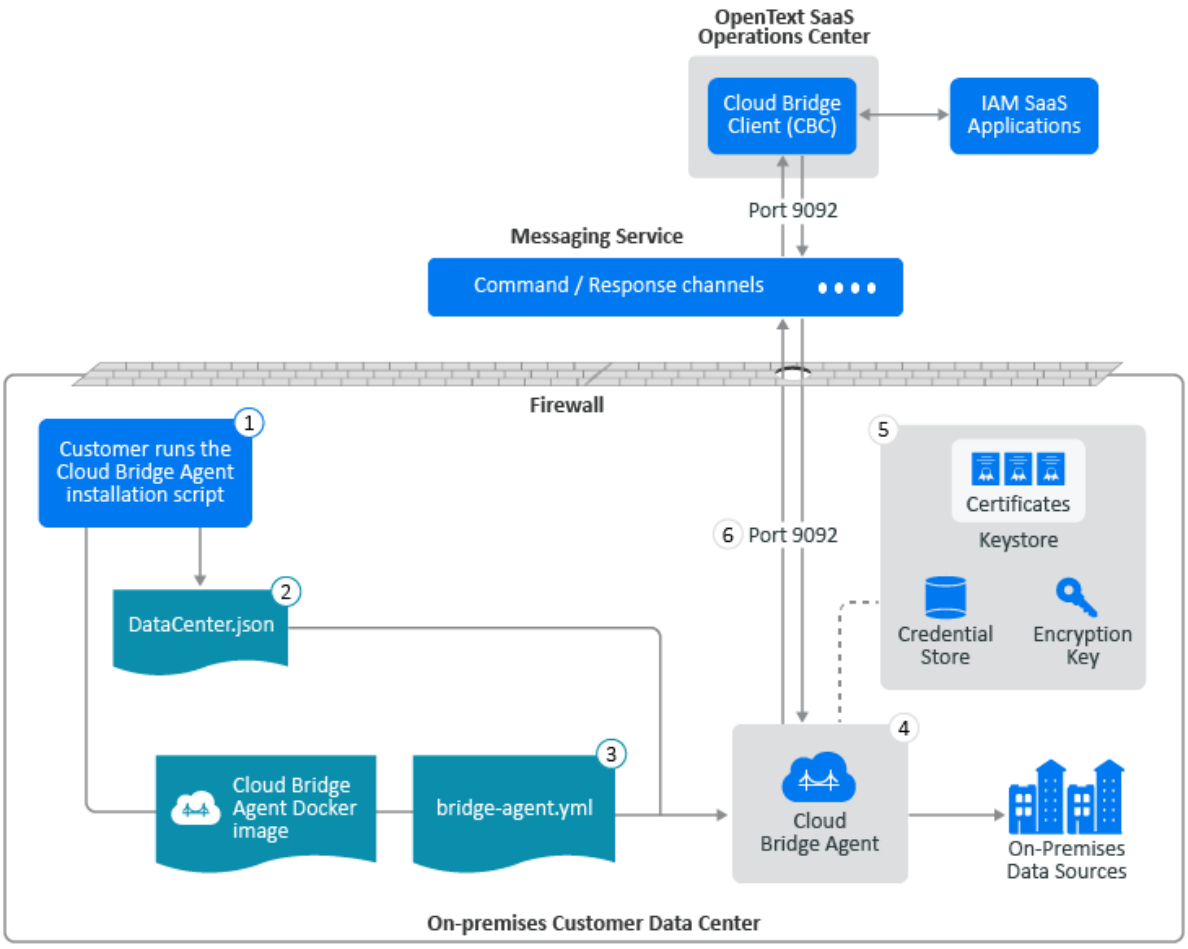
The high availability architecture means that more instances of the Agent simply require you to run additional Agent containers without any orchestration software or databases needed. You can set up as many failover instances as your organization requires, and all Agent instances can be in an active state concurrently. Not only do Agent administrators have the ability to view each Agent's instance configuration and the current target Agent instance, but the SaaS operations team also monitors your Agent instances. For more information about planning for high availability, see "Planning for High Availability" on page 13.

How the OpenText Cloud Bridge Agent Works

Your OpenText Cloud Bridge data center is configured as part of your OpenText SaaS product licensing (for example, OpenText Identity Governance) based on the information you provide in the technical questionnaire. **Data centers** are a conceptual representation of your OpenText Cloud Bridge Agent instance. You install one or more Agents on your local systems, then configure data sources and data source connections as needed to connect to your on-premises data sources. If you want to collect data from multiple data centers, you must install an OpenText Cloud Bridge Agent in each on-premises data center.

The following diagram illustrates the standard OpenText Cloud Bridge Agent configuration in your on-premises environment, regardless of the other components you might have installed.

Figure 1-1 *Overview of OpenText Cloud Bridge Agent Configuration On Premises*



For the numbered components on the above diagram, see the following additional information:

1. Requires access to the AWS™ (Amazon® Web Services™) download site to download the Agent package on port 443. The customer runs a custom curl/bash command. The curl portion downloads the container image and the main part of the installation script and then executes it.
2. `DataCenter.json` file – Contains customer-specific data center information. Defines the connection information between the OpenText Cloud Bridge Client and a single Agent instance.
3. `bridge-agent.yml` file – Contains configuration settings for the OpenText Cloud Bridge Agent.
4. Host server – OpenText Cloud Bridge Agent runs in a Docker® or Podman container on this server.
5. Secure communication components – Configurable on the Agent host server.
6. Port 9092 – Facilitates the secure outbound data transfer from the on-premises Agent to the OpenText Cloud Bridge API in the OpenText AWS™ cloud (`*.aws.confluent.cloud`).

2 Planning Your OpenText Cloud Bridge Agent Environment

Before you can install the OpenText Cloud Bridge Agent (“the Agent”) on premises, the SaaS operations team must have granted you the privilege to use OpenText Cloud Bridge.

IMPORTANT: You should not attempt to install the OpenText Cloud Bridge Agent in a production environment until after you have successfully deployed the Agent in a test environment.

Understanding Requirements

Review the following sections and ensure that your environment meets all requirements before you install the OpenText Cloud Bridge Agent.

Access Prerequisites

You must have administrator privileges on the server where you install the OpenText Cloud Bridge Agent. In addition, you must have the following rights:

- Access to the installer of a supported Linux® operating system. For more information, see “Hardware and Software Requirements” on page 12.
- Access to the relevant repositories to install or upgrade the operating system and the Docker® or Podman container.
- Access to outbound TCP port 9092 to allow outbound TLS communication from the on-premises OpenText Cloud Bridge Agent to the OpenText Cloud Bridge API in the AWS™ (Amazon® Web Services™) OpenText cloud (`*.aws.confluent.cloud`).

  IMPORTANT: You must use DNS filtering because OpenText cannot provide specific hosts.

- Access to the AWS™ download site to download the OpenText Cloud Bridge Agent package on port 443.
- Access rights for each server on which you plan to install the Agent for high availability.

If you are using OpenText Identity Governance with OpenText Cloud Bridge, you must also have:

- Access from your on-premises OpenText Cloud Bridge Agent to the on-premises authentication directory or directories for OpenText Advanced Authentication and other applications from which you expect to collect data as part of the OpenText Identity Governance collection processes
- Internal access to your OpenText Cloud Bridge Agent using a supported browser on port 8080

IMPORTANT: OpenText Advanced Authentication can use the following user lookup attributes by default for authentication purposes to OpenText Advanced Authentication and OpenText Identity Governance:

- Microsoft® Active Directory™: mail, sAMAccountName, and userPrincipalName
- OpenText™ eDirectory: cn and mail

Every OpenText Advanced Authentication repository has a configured Base DN. The value found on the specified Base DN for a specific user must be unique for the authentication service to work. This means that repeated values for a given user trying to authenticate will result in an authentication error.

Hardware and Software Requirements

This section provides the minimum hardware and software requirements for each server on which you install the OpenText Cloud Bridge Agent.

NOTE: OpenText recommends that you host the Agent on a dedicated server. If your organization has a dedicated server with sufficient resources, it is possible for the Agent to co-exist with other containers. However, ensure that the Agent does not have to compete for resources.

If you do install the OpenText Cloud Bridge Agent on the same server with another on-premises product, the installation order is not important.

Table 2-1 *OpenText Cloud Bridge Agent Server Requirements*

Category	Minimum Requirement
Processors	4 CPUs
Memory	16 GB of RAM
Hard disk space	200 GB **NOTE:** If you are using non-standard Linux® partitioning and you are partitioning `/var/tmp`, ensure that at least 16 GB of the required 200 GB are allocated to this partition.
VM environment	(Conditional) If you plan to install the Agent on a virtual machine, VMware® ESXi 6.7 or later.

Category	Minimum Requirement
Operating system	One of the following Linux® operating systems: ◆ Ubuntu® 18.04 LTS Server Edition or later patched version of 18.*x* ◆ Red Hat® Enterprise Linux® Server 8.3 or later patched version of 8.*x* ◆ SUSE™ Linux® Enterprise Server 15.1 or later patched version of 15.*x* **IMPORTANT:** Having FIPS enabled at the operating system level is not currently a supported configuration for the OpenText Cloud Bridge Agent. **NOTE:** Ensure that wget and awk are installed before you attempt to deploy the OpenText Cloud Bridge Agent container.
Container	One of the following: ◆ Docker® 19.03.*x* or later ◆ Podman 1.6.4 or later
Browser	One of the following to access the OpenText Cloud Bridge Agent console: ◆ Google® Chrome™ (latest version) ◆ Mozilla™ Firefox™ (latest version)

NOTE: You must configure a DNS entry for your LDAP server in order to use a secure connection. You will not be able to connect over SSL (port 636) if you are using a host file.

Planning for High Availability

You can ensure that your organization provides reliable services to your employees without downtime by using the high availability features that OpenText Cloud Bridge offers.

Understanding High Availability in OpenText Cloud Bridge

OpenText Cloud Bridge addresses both instance and site failover scenarios as follows:

- **Instance failover:** If the hardware or container hosting the Agent fails, OpenText Cloud Bridge can immediately switch to another Agent instance with the same data center configuration that has access to the same data sources.
- **Site failover:** If an entire data center location becomes inoperative due to a catastrophic event, OpenText Cloud Bridge can immediately switch to an Agent instance at another site that has access to the same data sources.

When you install an Agent, you specify the following properties. These properties are saved to the `bridge-agent.yml` configuration file for the Agent:

- **Instance ID** - Specifies a unique identifier for that instance

- **Site Priority** - Specifies the priority (Primary, Secondary, or Backup) of the site in relation to other data center sites

  In the event of failure, all Agent instances within a Primary site take precedence over Agent instances with a Secondary site configuration, and the Secondary site instances take precedence over Agent instances with a Backup site configuration.

- **Instance Priority** - Specifies the priority (Primary, Secondary, or Backup) of the instance in relation to other Agent instances

  In the event of failure, all Agent instances (within the same site) with a Primary instance weight take precedence over Agent instances with a Secondary instance configuration, and the Secondary instances take precedence over Agent instances with a Backup instance configuration.

OpenText Cloud Bridge calculates the configured site priority and instance priority into a single weight value. When a shutdown takes place, whether planned or unplanned, OpenText Cloud Bridge uses a simple numerical comparison of the weight values from all active Agent instances to determine which Agent instance becomes the target Agent instance. OpenText Cloud Bridge fails over to the highest weighted instance that it finds. You can use any combination of weights you choose.

For a site, the weight values are as follows:

- Primary = 30
- Secondary = 20
- Backup = 10

For an instance, the weight values are as follows:

- Primary = 5
- Secondary = 3
- Backup = 1

So, for example, if you configured nine Agent instances in total, with a single Agent instance at each possible weight, failover would occur in the following order:

Table 2-2 *Configuration Combinations*

Order	Site	Instance	Calculated haWeight Value
1.	Primary (30)	Primary (5)	35
2.	Primary (30)	Secondary (3)	33
3.	Primary (30)	Backup (1)	31
4.	Secondary (20)	Primary (5)	25
5.	Secondary (20)	Secondary (3)	23
6.	Secondary (20)	Backup (1)	21
7.	Backup (10)	Primary (5)	15
8.	Backup (10)	Secondary (3)	13

Order	Site	Instance	Calculated haWeight Value
9.	Backup (10)	Backup (1)	11

NOTE: ◆OpenText Cloud Bridge does not prevent you from configuring multiple Agent instances with the same site and instance priority. For example, if you configure three Agent instances as Primary sites and instances, in a failover scenario the OpenText Cloud Bridge Client will select the *first* instance with that priority that it detects.

- Secondary and Backup sites and instances provide additional layers for failover in larger environments, and you can have multiple sites and instances at each level if needed. However, you do not need to assign both Secondary and Backup site and instance roles if your organization does not need this level of complexity.

Understanding OpenText Cloud Bridge Communication in Failover Scenarios

Each active OpenText™ Cloud Bridge Agent sends a heartbeat message when it starts up and every 30 seconds thereafter. The OpenText Cloud Bridge Client ("the Client") monitors these heartbeats and is able to quickly detect an instance outage.

- In a scenario where an administrator intentionally shuts down an Agent instance – for example, for maintenance or re-hosting – the Agent sends a heartbeat immediately to the OpenText Cloud Bridge Client. The Client immediately performs the necessary steps for failover to a new Agent target.
- In an unplanned shutdown scenario where the target Agent instance is no longer able to communicate its heartbeat to the OpenText Cloud Bridge Client, the Client detects when a target Agent has not communicated within a configured timeframe (with a default of one minute). It then attempts to ping the Agent. If the ping attempt fails, the Client marks the target Agent as unresponsive and initiates the Agent target selection process.

Whether the shutdown was planned or unplanned, the following process takes place:

1. The OpenText™ Cloud Bridge Client selects a new target Agent from the current list of active Agent instances.
2. The Client sends a ping command to all Agent instances to inform them of the new Agent target selection.
3. The Client loads the current data source configurations into the new target Agent.
4. All data collection sessions that were active at the time of the shutdown fail, and new commands and collections are routed to the new target Agent.

If no active and initialized Agent is available, the Client marks the data center as unconnected and uninitialized. All command traffic that is sent to the data center receives an immediate `no agent available` error response.

If a non-target Agent instance is shut down, whether intentionally or not, the Client simply removes it from the list of active Agent instances.

Understanding Encryption IV and Key Values

OpenText Cloud Bridge uses initialization vectors (IVs) and keys to encrypt and decrypt data for protection of customer data and data source connection passwords stored in the Agent instances. For more information about initialization vectors, see TechTarget and Wikipedia.

Each Agent instance in your high availability environment must use the same encryption IV and Key values. Using the same values on all Agent instances ensures that no disruptions will occur during failovers.

If you create new encryption IV and Key values for the Agent configurations, the set of credentials for the Agents (which were encrypted using the older keys) become invalid. At that point you must reset the password values. The Agent console indicates which credential sets are invalid.

For information about how to set the same encryption IV and Key values on secondary and backup Agent instances after you have installed your primary Agent instances, see "Installing Secondary and Backup Agent Instances" on page 23.

Essential Components of Successful Failover

The following are critical elements of successful configuration of the Agent high availability components:

- Each Agent instance must have a unique Instance Id.
- Each instance of the Agent configuration must use the same encryption IV and Key values. For more information, see "Understanding Encryption IV and Key Values" on page 16.
- The set of data source credentials that are stored in the high availability Agent instances must be correct and consistent. After you have set up credentials for your first Agent, you can export and import credentials for the remaining Agents.
- Each Agent instance must be configured with the appropriate instance and site priority so that failover occurs as expected. For more information, see "Understanding High Availability in OpenText Cloud Bridge" on page 13.

Planning for Failover

Before you begin preparing servers to install the OpenText Cloud Bridge Agent, you should determine the following:

- If you have more than one data center, which one will be the primary site, and which ones will serve as secondary and backup sites in the event of catastrophic site failure
- The number of Agent servers you plan to install in each data center
- The priority (primary, secondary, or backup) of each Agent server you plan to install
- The naming convention you will use to identify each Agent instance

As a best practice, create a spreadsheet similar to the following example to record your decisions. Ensure that you keep this document up to date as you make changes in your environment.

Table 2-3 *High Availability Planning Spreadsheet*

Data Center Site Name	Site Priority	Instance Id	Instance Priority
Houston data center	Primary	Hou_Instance_1	Primary
		Hou_Instance_2	Secondary
		Hou_Instance_3	Backup
Provo data center	Secondary	Pro_Instance_1	Primary
		Pro_Instance_2	Secondary
		Pro_Instance_3	Backup
Cambridge data center	Backup	Cam_Instance_1	Primary
		Cam_Instance_2	Secondary
		Cam_Instance_3	Backup

Recommended Installation Order for High Availability

If you plan to install multiple Agent instances for high availability, the most efficient method for you to set up your environment is as follows:

1. Install your first Agent instance, saving your encryption key and IV for reuse.
2. Install all subsequent Agent instances using the encryption key and IV that you copied from your first Agent instance.
3. Configure your first Agent instance with the credentials for all data sources you plan to use, then export those credentials to a file for reuse.

   NOTE: OpenText recommends that you verify all data source credentials are correct before you export them for reuse.

4. Import the data source credentials to each secondary and backup instance one at a time.

For more information, see Chapter 3, “Installing the OpenText Cloud Bridge Agent,” on page 19.

3 Installing the OpenText Cloud Bridge Agent

Before you install and configure your OpenText Cloud Bridge Agent host servers, ensure that you have reviewed the server requirements and planned your environment for high availability. For more information, see the following topics:

- “Understanding Requirements” on page 11
- “Planning for High Availability” on page 13

OpenText Cloud Bridge requires that you install components in a specific order. You must set up the Agent server with a supported Linux® operating system and a supported Docker® or Podman container before you can install the Agent. For more information, see the following topics:

- “Setting Up the Agent Server” on page 19
- “Installing Docker® or Podman” on page 20
- “Installing OpenText Cloud Bridge Agent Instances” on page 21
- “Upgrading Your Agent Environment” on page 24

Setting Up the Agent Server

Before you install the OpenText Cloud Bridge Agent (“the Agent”), you must set up a Linux® server with a supported operating system and a Docker® or Podman environment. You can use either a virtual machine or a physical server. The following steps assume you are performing the most common installation, where the OpenText Cloud Bridge Agent is running on premises as a guest virtual machine.

IMPORTANT: If your current environment requires the OpenText Cloud Bridge Agent to be running in a cloud environment such as Amazon® Web Services™, Microsoft® Azure®, or Google® Cloud Platform™, you do not need to download an installer image, because you can create a new instance directly with those operating systems.

To set up the Agent server:

1. Download an installer image of a supported operating system. For more information, see the documentation for the selected operating system.
2. Create a new virtual machine that meets all specified requirements. For more information, see the documentation for the selected virtual environment.
3. Install your preferred operating system on your newly created virtual machine.

   IMPORTANT: Having FIPS enabled at the operating system level is not currently a supported configuration for the OpenText Cloud Bridge Agent.

   ---

4. Verify that your server has access to the internet as well as to your internal authentication repositories and applications.

5 Depending on your chosen operating system, complete additional steps to install Docker® or Podman as follows:

- “Setting Up Ubuntu® with Docker®” on page 20
- “Setting Up SUSE™ Linux® with Docker®” on page 20
- “Setting Up Red Hat® Enterprise Linux® with Podman” on page 20

Installing Docker® or Podman

After you install a supported operating system on your OpenText Cloud Bridge Agent host server, you must complete additional steps to install a Docker® or Podman environment for the Agent.

Setting Up Ubuntu® with Docker®

Set up Ubuntu® with Docker® using the instructions provided on the Docker® documentation site. If any of the commands fail, it is most likely because of formatting issues. OpenText recommends that you copy the lengthy commands directly from the online Ubuntu® Docker® documentation.

After you have set up your Docker® environment, you can proceed with installing the OpenText Cloud Bridge Agent. For more information, see “Installing OpenText Cloud Bridge Agent Instances” on page 21.

Setting Up SUSE™ Linux® with Docker®

Set up SUSE™ Linux® with Docker® using the instructions provided on the Docker® documentation site. If any of the commands fail, it is most likely because of formatting issues. OpenText recommends that you copy the lengthy commands directly from the Docker®, Docker® Compose, and openSUSE online documentation.

After you have set up your Docker® environment, you can proceed with installing the OpenText Cloud Bridge Agent. For more information, see “Installing OpenText Cloud Bridge Agent Instances” on page 21.

Setting Up Red Hat® Enterprise Linux® with Podman

Complete the following steps to set up Red Hat® Enterprise Linux® with Podman:

1 Ensure that your Red Hat® Enterprise Linux® 8.3 Server is registered with Red Hat®.

2 Check the installed Podman version. Run:

```
podman version
```

3 (Conditional) If you already have Podman 1.6.4 or later, you do not need to install it.

4 (Conditional) If you do not have Podman installed, run:

```
yum install podman
```

5 After you have set up your Podman environment, you can proceed with installing the OpenText Cloud Bridge Agent. For more information, see “Installing OpenText Cloud Bridge Agent Instances” on page 21.

Installing OpenText Cloud Bridge Agent Instances

This section provides instructions for installing the OpenText Cloud Bridge Agent ("the Agent") in a new environment. To upgrade an existing Agent environment, see "Upgrading Your Agent Environment" on page 24.

Before you can install the OpenText Cloud Bridge Agent, you must have already installed a supported Linux® operating system and a Docker® or Podman environment on the host server where you plan to install the Agent. For more information, see the following topics:

- "Setting Up the Agent Server" on page 19
- "Installing Docker® or Podman" on page 20

You can install the OpenText Cloud Bridge Agent anywhere on the Docker® or Podman host server, but OpenText recommends that you use a standard installation location for each Agent instance. Wherever you install the Agent, the installation script installs an `agent` directory. The `agent` directory contains the scripts, the `.env` file containing the encryption Key and IV values, and additional directories as follows:

- *<Agent_install_dir>*`/agent/conf` contains the `bridge-agent.yml` and `DataCenter.json` files
- *<Agent_install_dir>*`/agent/log` holds the rolling log files
- *<Agent_install_dir>*`/agent/bridgelib` holds the connector `.jar` files
- *<Agent_install_dir>*`/agent/update` contains temporary connector `.jar` files and pending update actions
- *<Agent_install_dir>*`/agent/backup` holds `.jar` files for connectors that you have deleted or updated, and `.json` files containing the history of actions you have performed

Installing Your Primary Agent Instance

You can install a single Agent instance in your environment, but OpenText recommends also installing secondary and backup Agent instances to ensure high availability for your users. Depending on the size of your organization, you might need to set up more than one site. For more information, see "Planning for High Availability" on page 13.

IMPORTANT: Ensure that you download the installation script within the time window that the SaaS operations team specified for your installation. After this time, the script will no longer be available for download and you will have to request another script.

---

1. After you receive the Agent download instructions from the SaaS operations team, open a command line and navigate to the folder where you want to install the Agent.
2. Copy and paste the provided `curl` command, then press **Enter**.

   This command downloads and runs the installation script specific to your organization.
3. At the prompt, specify the desired priority for the Agent instance as follows:
   - If you want the instance to be the Primary instance (the default), enter `0` or just press **Enter**.

- Enter 1 for Secondary.
- Enter 2 for Backup.

NOTE: If you need to make any changes to your Agent instance or site settings at a later time, you can rerun the installation script. For more information, see "Updating High Availability and Other Agent Settings" on page 47.

---

4 At the prompt, specify the desired priority for the Agent site as you did for the Agent instance.

The installer displays a generated Instance ID for the Agent consisting of the host name and random letters, but you can change this ID to a more meaningful name.

5 (Optional) Type your desired Instance ID and press **Enter** to save it.

The installer then checks whether Docker® or Podman is installed and displays the version.

6 At the prompt, enter the user name for the Agent administrator (`cbagent`) and set a password.

You will use these credentials to log in to the Agent console and add credentials for your data source.

The script then installs the OpenText Cloud Bridge Agent. When installation is complete, the Agent comes up and sends a heartbeat to the SaaS operations center.

7 Log in to OpenText Advanced Authentication and perform the following steps:

7a Configure an external repository to an on-premises LDAP source. For more information, see "Adding a Cloud Bridge External Repository" (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/t4donma2ncp4.html) in the *OpenText™ Advanced Authentication SaaS Administration Guide*.

IMPORTANT: If you are setting up an OpenText™ Cloud Bridge external repository in OpenText Advanced Authentication for an OpenText Identity Governance tenancy, use the format *TENANT_ID*_AA_ER for the external repository name, where *TENANT_ID* is in uppercase.

7b Copy the data source connection credential ID from the OpenText Advanced Authentication user interface. You will need this credential ID in OpenText Cloud Bridge.

8 (Conditional) If you have OpenText Identity Governance, log in to OpenText Identity Governance, configure a data source connection, then copy the unique ID for that connection. For more information, see "Collecting Data Using Cloud Bridge" (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/user-guide/t4h6p598xpxq.html) in the *OpenText™ Identity Governance User and Administration Guide*.

9 (Conditional) If you have other OpenText SaaS products that you want to use with OpenText Cloud Bridge, log in to those products and configure data source connections as described in the documentation for those products.

10 In a supported browser, go to the OpenText Cloud Bridge Agent URL:

`http://localhost (Agent_IP_address_or_DNS_name):8080`

11 Log in to the OpenText Cloud Bridge Agent console using the bootstrap administrator credentials:

User name: `cbagent`

Password: The password that you set when you ran the installation script

12 (Optional) On the **Dashboard** tab, verify that the site and instance priorities you set for the Agent during installation are correct.

NOTE: The **Instance Id** is the generated identifier or the name that you specified for the *current* Agent instance.

The **Target Id** is the name of the *primary* Agent instance in a high availability environment. It is the instance with which the OpenText Cloud Bridge Client communicates. The Instance Id and the Target Id might be the same if you are currently viewing the primary Agent instance or if you have only one Agent instance in your installation.

---

13 Click the **Data Source Management** tab and add your data source connection credentials. For more information, see "Manage Credentials for Data Source Connections" on page 28.

14 (Optional) In OpenText Advanced Authentication, click the **Test** button to verify that the data source connection works.

After you have installed your primary Agent instance, consider installing additional Agent instances for high availability. For more information, see "Installing Secondary and Backup Agent Instances" on page 23.

For more information about using OpenText Cloud Bridge with other OpenText SaaS products, see the following resources:

- *OpenText™ Identity Governance and Administration Quick Start* (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/quick-start/quick-start.html)
- *OpenText™ Identity Governance User and Administration Guide* (https://www.microfocus.com/documentation/identity-governance-and-administration/igaas/user-guide/front.html)
- *OpenText™ Advanced Authentication Administration Guide* (https://www.netiq.com/documentation/advanced-authentication-64/tenant-administrator-guide/data/bookinfo.html)
- Other product documentation on the Identity and Access Management Services Documentation web page (https://www.microfocus.com/documentation/identity-and-access-management/iam-services/)

Installing Secondary and Backup Agent Instances

After you have installed your primary Agent instance, you can install secondary and backup instances following the same steps that you used for your primary Agent instance. For more information, see "Installing Your Primary Agent Instance" on page 21.

Each instance of the Agent configuration in your high availability environment must use the same encryption IV and key values. Using the same values on all Agent instances ensures that no disruptions will occur during failovers. For more information about encryption keys and IVs, see "Understanding Encryption IV and Key Values" on page 16.

NOTE: If you have a scenario where you need to back up credentials on an existing Agent server and plan to reinstall the Agent, you should make note of the original key and IV from the old Agent before you uninstall it. When you import the credentials into the newly installed Agent, you will need to provide the key and IV from the old Agent.

To reuse the encryption key and IV from the first Agent instance:

1. On your primary Agent instance, locate and open the `.env` file. Copy the encryption key and IV from this file.
2. Install your secondary Agent instance.

   The installation script creates the `/agent` directory on the secondary instance, but the instance has its own encryption key and IV at this point.
3. On the secondary Agent instance:
   - **3a** Enter the following command to stop the instance:

     `sh <Agent_install_dir>/agent/stop.sh`
   - **3b** Open the `agent/.env` file:

     `vi <Agent_install_dir>/agent/.env`
   - **3c** Replace the KEY and IV entries in the `.env` file with the KEY and IV values from the primary Agent instance.
4. Because the second instance created the Docker® container on the installation, it must be replaced for the new key and IV to be used. Enter the following commands to replace the pod definition with the updated key and IV properties:

   `sh <Agent_install_dir>/agent/remove.sh`

   `sh <Agent_install_dir>/agent/create.sh`
5. Start the secondary instance back up with the start script:

   `sh <Agent_install_dir>/agent/start.sh`

   The second instance initializes, and OpenText Cloud Bridge recognizes both Agents as running in a high availability system.

NOTE: If you need to change your Agent instance or site settings at any point, you can rerun the installation script and make updates. For more information, see "Updating High Availability and Other Agent Settings" on page 47.

Upgrading Your Agent Environment

You can upgrade an existing OpenText Cloud Bridge Agent instance by running the installation script on the host server. The installation script detects that an older version is already installed and prompts you to upgrade.

Before you upgrade, you should consider the following:

- Whether you want to replace all the connectors you currently have installed on the Agent host server. For more information, see "Planning to Upgrade Connectors" on page 25.
- Whether you want to make changes in your high availability environment. For more information, see "Planning for High Availability Before Upgrade" on page 25.

NOTE: Upgrading your Agent environment does not remove old Agent images. As a best practice, after you have upgraded successfully you should periodically review and remove unneeded old images to free up disk space. For more information, see "Removing Old Agent Images" on page 46.

Planning to Upgrade Connectors

Before you begin upgrading your Agent environment, review your existing connectors and take the appropriate action as outlined below. When you upgrade the OpenText Cloud Bridge Agent, the connector `.jar` files that reside in the `<Agent_install_dir>/agent/bridgelib/` folder on the Agent host server take precedence over the connector `.jar` files that are included in the `/collectors` folder.

IMPORTANT: Work with your SaaS support team to address any questions or concerns you have about upgrading connectors.

Consider the following:

- If you have older patched connectors in the `/bridgelib` folder and the newer Agent has better built-in `.jar` files, you should remove the outdated files in the `/bridgelib` folder before you install the Agent. Remove any copies of dist-collectors, daas-ldap, or any other updated `.jar` files provided by your SaaS support team.
- If your `/bridgelib` folder contains files such as a supporting `.jar` file for JDBC or other custom collector `.jar` files, you should leave those files alone.
- If you have older connectors in your `/bridgelib` folder that are working well and you do not want to risk upgrading them, you can choose to keep an older version in `/bridgelib` and not use the updated version included with the Agent in the `/collectors` folder.

Planning for High Availability Before Upgrade

If you currently have only one Agent instance in your environment, we strongly recommend that you install and configure one or more secondary or backup Agent servers for failover. For more information to help you plan, see "Planning for High Availability" on page 13.

Consider the following possible high availability upgrade scenarios:

- You want to keep your *current* Agent as your primary Agent instance. For more information, see "Upgrading with Your Current Agent as the Primary Instance" on page 26.
- You want to use *another* server for your primary Agent. For more information, see "Upgrading with a New Agent as the Primary Instance" on page 26.

NOTE: You can reconfigure your Agent instances as needed after installation by rerunning the installation script on the appropriate servers. For more information, see "Updating High Availability and Other Agent Settings" on page 47.

Upgrading with Your Current Agent as the Primary Instance

If you are upgrading your Agent environment and want to assign your current Agent as your primary instance, complete the following steps:

1. Run the Agent installation script on your Agent server. For more information, see "Installing Your Primary Agent Instance" on page 21.
2. At the upgrade prompt, enter `y`.

   IMPORTANT: OpenText Cloud Bridge saves your Agent credentials when you upgrade. If you respond `n` at the upgrade prompt and do a fresh installation, you lose those credentials.
3. Set the instance and site to `Primary`.

   Wait for the server to restart.
4. Run a fresh installation for each secondary and backup Agent that you want to use in your environment. Ensure that you use the encryption key and IV from your first Agent instance for all secondary and backup Agent instances. For more information, see "Installing Secondary and Backup Agent Instances" on page 23.
5. Configure your primary Agent instance with the credentials for all data source connections you plan to use, then export those credentials to a file. For more information, see "Manage Credentials for Data Source Connections" on page 28.
6. Import the data source credentials to each secondary and backup Agent instance one at a time.

Upgrading with a New Agent as the Primary Instance

If you are upgrading your Agent environment and want to use a new host server as your primary Agent instance, complete the following steps:

1. Configure the host server that you plan to use as your primary Agent instance with all of the Agent prerequisites. For more information, see "Understanding Requirements" on page 11 and "Setting Up the Agent Server" on page 19.
2. Run the Agent installation script on the new host server and set the instance and site to `Primary`. Since this is a new installation, there is no prompt to upgrade. For more information, see "Installing Your Primary Agent Instance" on page 21.

   Wait for the server to restart.
3. Run the Agent installation script on your *existing* Agent server, enter `y` at the upgrade prompt, and set the instance and site to `Secondary` or `Backup` as appropriate for your environment.
4. Run a fresh installation for any additional secondary and backup Agents that you want to use in your environment. Ensure that you use the encryption key and IV from your first Agent instance for all secondary and backup Agent instances. For more information, see "Installing Secondary and Backup Agent Instances" on page 23.
5. Configure your primary Agent instance with the credentials for all data source connections you plan to use, then export those credentials to a file. For more information, see "Manage Credentials for Data Source Connections" on page 28.
6. Import the data source credentials to each secondary and backup Agent instance one at a time.

4 Configuring OpenText Cloud Bridge

After you install your OpenText Cloud Bridge Agent ("the Agent") and configure your external repository and data source connections, you must add credentials for the data source connections in the Agent console. You can also manage your connectors, logging, and administrator accounts in the Agent console.

Logging In to the OpenText™ Cloud Bridge Agent

When you install the OpenText Cloud Bridge Agent, the installer sets up a bootstrap administrator account named `cbadmin`. You set a password for this account during installation and use those credentials to log in to the Agent console.

NOTE: You cannot delete the bootstrap administrator account, but you can change the account password as needed. For more information, see "Managing the Bootstrap Administrator Account" on page 46.

If you need to share Agent administrator responsibilities with other users, OpenText recommends that you create additional administrator accounts and roles. Those users can log in using their own user name and password, and can also change their own password. Currently all Agent administrator accounts have full admin rights. For more information about adding administrator accounts, see "Manage Administrators and Roles" on page 32.

To log in to the Agent:

1. Open a supported browser and go to the OpenText Cloud Bridge Agent URL:

   `http://localhost (Agent_IP_address_or_DNS_name):8080`

2. Provide the user name and password for the `cbadmin` account or another administrator account and click **Sign In**.

View OpenText Cloud Bridge Agent Details

OpenText™ Cloud Bridge Agent > Dashboard

Use this page to view details of the current Agent instance, including its site and instance priority in your high availability environment. This page also indicates whether any connector updates are available or any connector conflicts exist. For more information about connectors, see “Manage and Update Connectors” on page 30.

You cannot make any changes to the Agent on this page. However, if you need to modify your Agent settings, you can rerun the installation script. For more information, see “Updating High Availability and Other Agent Settings” on page 47.

Search for Data or Change Data Layout

You can view and manage the data on most tabs in the Agent console using either a table or tile layout according to your preference. Click the icons in the top right section of the page to toggle between layouts as needed.

TIP: If you want to use the same layout for all tabs, make this global change in the Agent console settings. Click your login name in the top right corner of the window, then click **Settings**. Select your preferred layout, then click **OK**.

You can customize the table layout view to sort or filter data as follows:

- To change the sort order of a column, click the AZ column arrow or the column heading.
- To show only items that meet specific search parameters, click the **Filter** icon, specify parameters, then close the dialog box. Click **Reset** in the Filter dialog box if you want to return to the default display.
- To pin or autosize a column, click the **Column Settings** icon.
- To add or remove columns, and to autosize columns, click the **Table Settings** icon above the table and make changes in the Table Settings dialog box. Click **Reset** to return to the default view.

You can customize the tile layout as follows:

- To change the sort order of tiles, click the **Show Sort Settings** icon.
- To show only tiles that meet specific search parameters, click the **Show Advanced Filter** icon above the tiles, specify search parameters, then close the dialog boxes. Use the filter reset icon above the tiles to return to the unfiltered view.

Manage Credentials for Data Source Connections

OpenText™ Cloud Bridge Agent > Data Sources

Use this page to add, view, modify, or delete credentials for your data source connections. You can also export and import credentials from one OpenText™ Cloud Bridge Agent to another to reuse them in a high availability environment.

For more information, see the following topics:

- “Add New Credentials” on page 29
- “Export or Import Existing Credentials” on page 29

Add New Credentials

Complete the following steps to add new credentials for data source connections.

1. Click the plus (+) sign.
2. In the **Unique Id** box, paste the data source connection ID you copied from the application to which you want to connect. For example, for OpenText Advanced Authentication, paste the data source connection credential Id you copied in Step 7b on page 22.
3. In the **Ordinal** box, specify the appropriate ordinal for the authentication method. This value varies depending on whether you are entering credentials for the OpenText Identity Manager AE Permission collector or one of the SCIM collectors and fulfillers. For information about ordinals, see the documentation for the data source for which you are adding credentials.
4. (Optional) In the **Description** box, specify a unique description of the data source to help you easily identify it.
5. Under **Data Source Credentials** in the **Username** box, specify the service account ID. For LDAP accounts, specify the full DN. For example, `CN=svc-id-admin,CN=Users,DC=support,DC=test`.

   NOTE: This box is limited to 255 characters.

6. In the **Password** box, enter the password for the service account.

   NOTE: This box is limited to 255 characters.

7. Click **Save**.
8. Repeat these steps to add credentials for other identity or application collectors in OpenText Identity Governance or external repositories in OpenText Advanced Authentication.

You can change these credentials anytime if your service account or password has changed. Click the credential name in the **Unique Id** column to open the Update Credential window where you can make changes.

After you have added credentials on an Agent, you can reuse those credentials on other Agents in a high availability environment by exporting and importing the credentials. For more information, see “Export or Import Existing Credentials” on page 29.

Export or Import Existing Credentials

After you have added credentials on an Agent instance, you can reuse those credentials on other Agents in your high availability environment by exporting and importing the credentials. For information about adding new credentials, see “Add New Credentials” on page 29. As a best practice, you should verify that all data source credentials are correct before you export them for reuse.

To export or import existing credentials:

1. (Conditional) If this is the first Agent in your environment and you plan to install additional Agents for high availability, export your data source credentials as follows:
   - **1a** Click the **Export** icon.
   - **1b** Provide a name for the credentials file with a `.json` extension, then click **Download**.
2. (Conditional) If you previously saved data source credentials and want to reuse them now, import them as follows:
   - **2a** Click the **Import** icon.
   - **2b** Click **Select file to import** and locate the `.json` file that you previously saved to the Downloads folder, then click **Import**.

IMPORTANT:
- If you recreate your OpenText Advanced Authentication external repositories, your Data Source Connection credential ID will change and you will need to delete the previous credential and add the new credential.
- If you delete and recreate any in-use data source connection credential ID in OpenText Identity Governance, the associated collector will stop communicating through the Agent and you will need to enter the new credential in the Agent using the steps above.

Manage and Update Connectors

OpenText™ Cloud Bridge Agent > Connectors

Use this page to perform the following tasks:

- Add a Connector
- Update a Connector
- Delete a Connector
- Resolve Connector Conflicts

This section provides steps for performing tasks using the default table layout. If you are using the tile layout, click the **More Actions** menu on any tile to perform actions specific to that connector, such as editing or deleting that connector, canceling a pending action, resolving a connector conflict, or upgrading a connector.

By default, the Agent synchronizes with the OpenText™ Cloud Bridge Client to get new and updated connectors once a day. The update check also runs automatically whenever you restart the Agent.

IMPORTANT: Connector additions, deletions, or updates take effect only when you restart the Agent. Before you restart the Agent, as a best practice you should view all pending actions to verify that they are still appropriate. If you no longer want to perform a particular action, click the **Cancel** icon next to the connector, then confirm at the prompt.

To view the results of the last connector updates at any time, click **Last Update Result**. OpenText recommends that you click the trashcan icon to delete this information after you have verified that the actions you selected have completed successfully.

Add a Connector

To add a new connector:

1. Click the plus sign (+) icon to view available connectors.

   The Add Connector window shows the last date and time when the Agent checked for connector updates.
2. (Conditional) If you do not see an expected connector listed in this window, you can click the **Sync** icon to force a refresh of the list.
3. Select one or more connectors that you want to install and click **Mark for Install**.
4. When you are ready to install the connectors, restart the Agent. For more information, see "Restarting the OpenText Cloud Bridge Agent" on page 45.

NOTE: If you want the Agent to check for connector updates more or less often than the daily default, you can change this setting in the `<Agent_install_dir>/agent/conf/bridge-agent.yml` file. Under the `notification:` section, locate the `connectorUpdateCheckInterval` entry and specify a value that represents the appropriate number of days between update checks.

Update a Connector

When there are updates available for installed connectors, an ***X* UPDATES AVAILABLE** indicator appears above the table or tiles. To see all available updates at a glance, click the indicator.

A blue arrow icon also marks each affected connector. You can view information about the available updates by hovering over the arrow.

To update a connector:

1. Click the blue arrow icon.
   - If there is a single update available, this action marks the connector for update on the next restart of the Agent.
   - If there is more than one update available, a window opens that allows you to select the version of the connector that you want. Select the appropriate version and click **Mark for Update**.
2. When you are ready to apply the updates, restart the Agent. For more information, see "Restarting the OpenText Cloud Bridge Agent" on page 45.

Delete a Connector

To delete an installed connector, select the checkbox for the connector in the left column, then click the trashcan icon. The deletion does not actually occur until you restart the Agent. For more information, see "Restarting the OpenText Cloud Bridge Agent" on page 45.

Resolve Connector Conflicts

The Agent detects conflicts or duplicates of any installed connector `.jar` files. The Dashboard indicates when there is a conflict and the Connectors tab also marks conflicting connectors with a red icon.

To resolve connector conflicts:

1. On the Connectors tab, locate the connectors that display the same red icon in the left column, or click the **Filter Status** button and select the appropriate option.
2. Click any of the conflict icons to view details of the duplicate connectors, select the connector you want to keep, then click **Resolve**.

   The Conflict Resolution window closes, and each of the connectors to be deleted has a red icon with an **X** in the left column.
3. (Conditional) If you change your mind about the connector you want to keep:
   - **3a** Cancel a deletion by clicking the red icon with the X for the connector you want to keep, then click **Yes**.

     This action reverts the connectors to a conflicted state.
   - **3b** Click one of the conflict icons again, select the connector to be retained, then click **Resolve**.
4. When you are ready to complete the selected actions, restart the Agent. For more information, see "Restarting the OpenText Cloud Bridge Agent" on page 45.

Configure Agent Logging

OpenText™ Cloud Bridge Agent > Loggers

The default Agent logging settings are appropriate for standard installations. However, you can customize these settings if necessary to:

- Add or remove loggers using the plus (+) sign or trashcan icon
- Edit the logging level for any logger to collect more or less data
- Reset all loggers to their default collection levels
- Clear logs

To change the logging level of a logger file:

1. Click the name of the logger.
2. In the Update Logger dialog box, select the appropriate log level.
3. Click **Save**.

Manage Administrators and Roles

OpenText™ Cloud Bridge Agent > Administrators

Use this page to view, create, and manage administrator accounts and roles. By adding administrator accounts, you can provide access to the OpenText Cloud Bridge Agent console without sharing the bootstrap administrator account credentials.

IMPORTANT: Roles that you create on this page have no connection to roles in OpenText Identity Governance or other applications that use OpenText Cloud Bridge. They are intended to help you manage your Agent users.

Add an Administrator Account

1. On the Administrators page, click the **Administrators** tab, then click the plus (+) icon.
2. On the Add Administrator window, provide an administrator name and password, select the appropriate role, then click **Add**.

   NOTE: New administrator accounts are assigned to the built-in `admin` role by default. If you want to assign different roles, you must first create those additional roles so they are available for selection on this window.

Add a Role

You can create roles in addition to the built-in `admin` role to help you organize your administrator users, but this is optional. If you do not create additional roles, all new administrator accounts that you create are assigned to the default `admin` role.

1. On the Administrators page, click the **Roles** tab, then click the plus (+) icon.
2. Provide a Role Name and Description, then click **Add**.

Modify an Administrator Account or Role

You can modify administrator accounts and roles that you created. However, you cannot modify the `cbadmin` bootstrap administrator account except to change the account password. For more information, see "Managing the Bootstrap Administrator Account" on page 46.

- To modify an administrator account, click the name of the account to open the Edit Administrator window. Make the desired changes, then click **Update**.
- To modify a role, click the name of the role to open the Edit Administrator Role window. Make the desired changes, then click **Update**.

Delete an Administrator Account or Role

You can delete administrator accounts and roles that you created. You cannot delete the `cbadmin` bootstrap administrator account, but you can change the account password if necessary. For more information, see “Managing the Bootstrap Administrator Account” on page 46.

1. On the Administrators page, click the **Administrators** or **Roles** tab.
2. Select the administrator or role that you want to delete.
3. Click the trashcan icon.

5 Configuring Secure Communication for Your OpenText Cloud Bridge Agent

This section provides an introduction to the essential components of secure communication, instructions for configuring HTTPS connections in your OpenText Cloud Bridge Agent ("the Agent") environment, and reference information about the Agent Transport Layer Security (TLS) policy.

Understanding the Components of Secure Communication

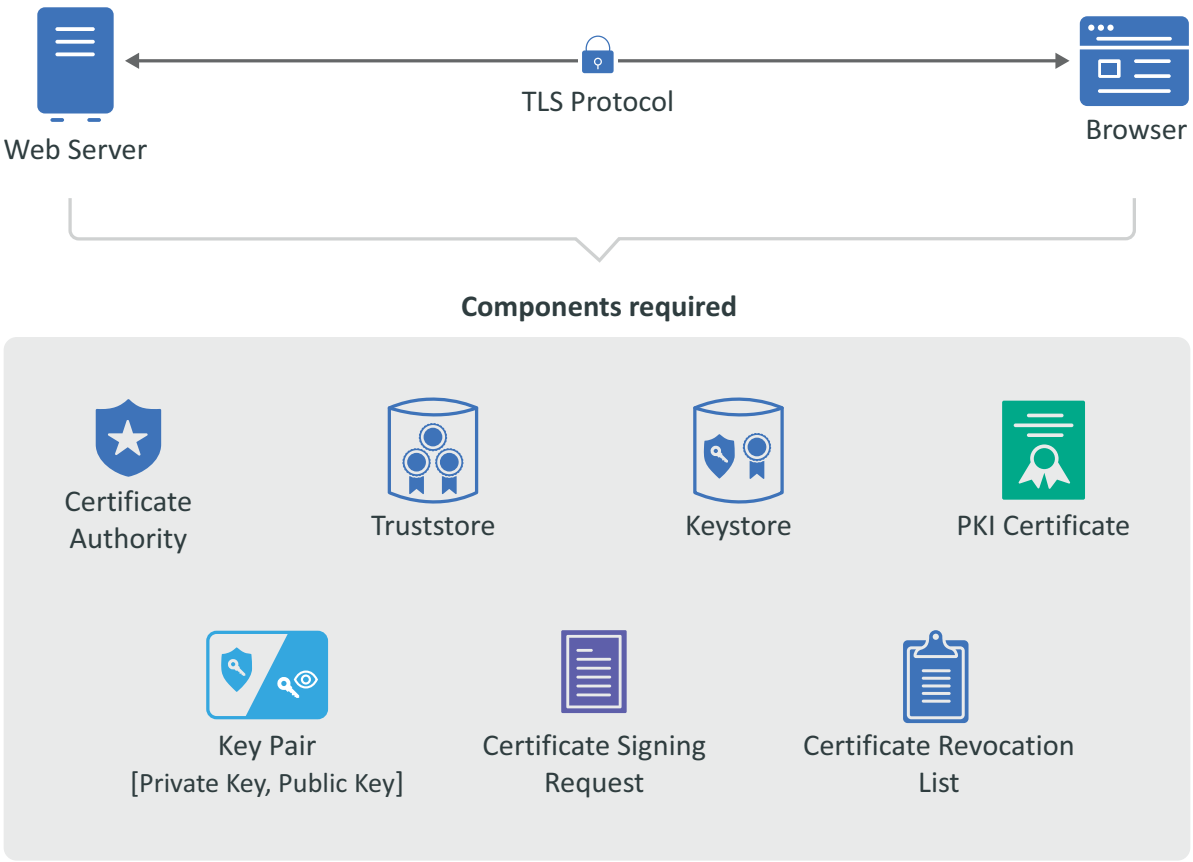
Configuring secure communication between your OpenText products, infrastructure products, and any administrative tools requires that you have a good understanding of the components that allow the secure communication to occur. OpenText products use industry standards of X.509 certificates, public key infrastructure (PKI), and transport layer security (TLS).

This section provides a basic introduction to these components. For more detailed information, see the following online resources:

- Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
- The Transport Layer Security (TLS) Protocol Version 1.2
- The Transport Layer Security (TLS) Protocol Version 1.3

You should secure the communication channels between servers and clients to protect your data and stop security breaches from occurring in your environment. The following graphic depicts the different components required for secure communication using certificates, PKI, TLS, and tools to manage the keys.

Figure 5-1 *Components of Secure Communication*



The secure communication occurs between a server and a client. In Figure 5-1, the web server represents the server and the browser represents the client. The following are important terms that you need to understand to create a secure connection between a server and a client.

- **Certificate Authority:** An entity that issues digital certificates. A certificate authority (CA) acts as a trusted third party—trusted both by the subject (owner) of the certificate and by the party relying upon the certificate. There are two different types of CAs:
  - **Well-known:** A certificate authority that provides server certificates signed by well-known CAs such as IdenTrust or DigiCert.
  - **Self-signed:** A certificate authority that generates self-signed certificates through other products such as OpenSSL®, OpenText™ eDirectory, and Microsoft® Active Directory™. You can create self-signed certificates through the certificate authorities in these other products to use in test environments.

  NOTE: A security best practice is to use a well-known CA to issue certificates in production environments.

- **Public Key Infrastructure (PKI) Certificates:** Digital certificates that the CA issues that prove ownership of the certificate. The CA can issue certificates for users, applications, or devices. PKI certificates contain the following information:
  - Version number
  - Unique serial number

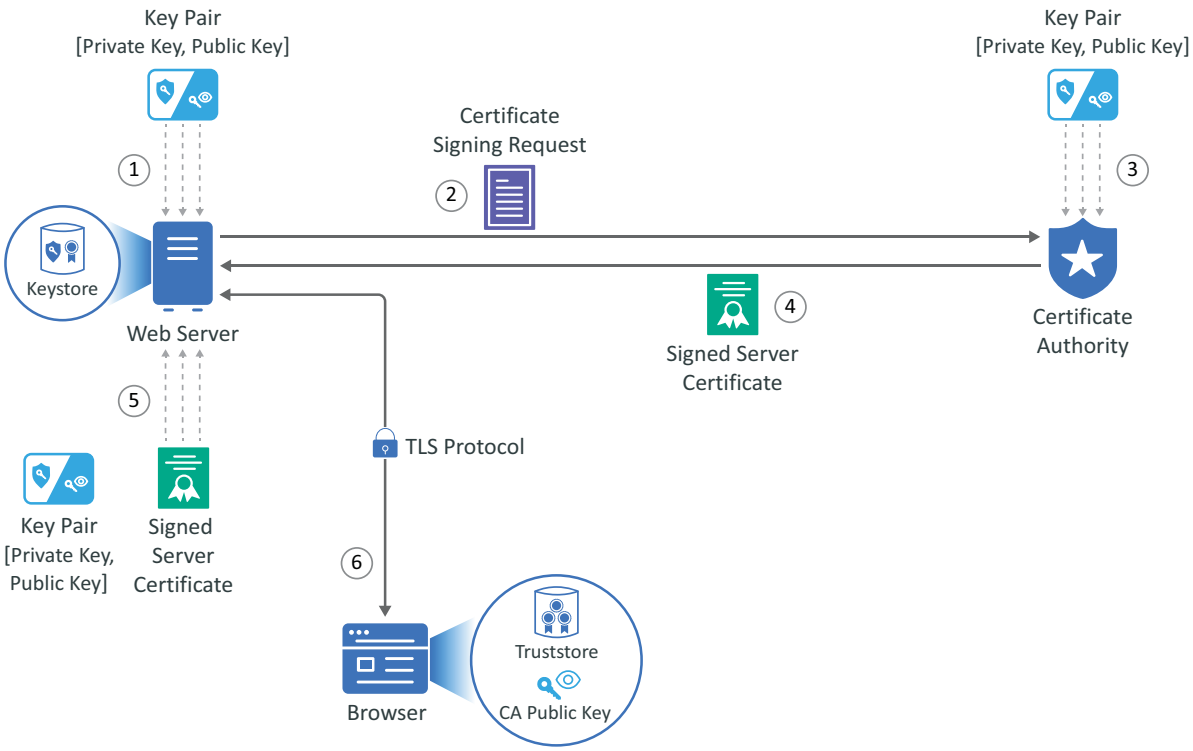
- CA digital signature and algorithm used
- Validity period
- Certificate usage
- Subject name, URL, and email address
- Public and private keys (sometimes only the public key)

- **Key Pair:** A private key and public key that work together to encrypt and decrypt messages. PKI is based on the fact that everyone will trust any communication encrypted with a public key or trust any certificate signed by a private key.
  - **Private Key:** A cryptographic key that you use to decrypt any communication encrypted by the public key. Only the private key of the key pair can decrypt the communication encrypted with the corresponding public key. You keep the private key private and do not share it.
  - **Public Key:** A cryptographic key that you use to encrypt communications to keep the communication secure. Only someone with the private key can decrypt the communications. You share the public key so that anyone with access to the public key can verify that any communication signed with this public key is really from the sending source.
- **Certificate Revocation List (CRL):** A list that the CA creates and manages that contains a list of unique serial numbers that it has revoked. The CA uses the certificate revocation list (CRL) to deny requests from any user, application, or device that has a serial number listed on the CRL.
- **Certificate Signing Request (CSR):** A message sent from an applicant to the CA to apply for a PKI certificate. Usually the certificate signing request (CSR) contains a copy of the public key of the applicant making the request, identifying information such as a domain name, and a digital signature.
- **KeyStore:** A secure Oracle® Java® repository that stores the private key and identity certificate for the server in the trust relationship. The information is stored encrypted on the server with a KeyStore password that you set and manage. Use either the keytool or keytoolgui tool to set and manage the KeyStore passwords.
- **TrustStore:** A secure Oracle® Java® repository that stores the certificates signed by a CA in a secure repository on the client. The information is stored and encrypted on the client with a TrustStore password that you set and manage. Use either the keytool or keytoolgui tool to set and manage the TrustStore passwords.
- **Transport Layer Security (TLS) Protocol:** The secure protocol created by all of the components defined in this section. It allows the server and client to communicate securely using certificates and key pairs to prove identity on the server and client.

Example of Establishing Secure Communication for a Web Server

When you install a web server, the communication is not secure by default. Or, if the communication is secure, it is usually using a self-signed certificate. The following example shows how the web server obtains a server certificate signed by a well-known certificate authority (CA) to use in establishing secure communication with any client.

Figure 5-2 *Obtaining a Signed Server Certificate from a Well-known Certificate Authority*



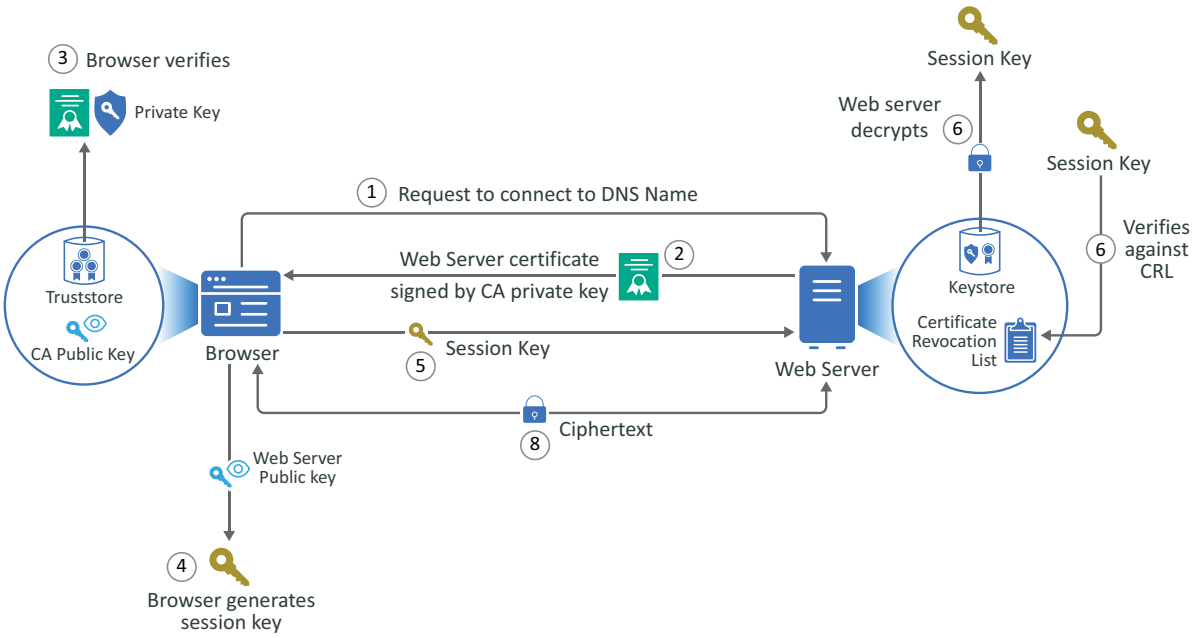
In this example, Adam the administrator requests a signed server certificate from the well-known CA and uses the certificate to establish secure communication with a client that is a web application.

1. Adam generates a key pair on the web server using keytool. Adam uses the key pair to create a certificate signing request (CSR) using keytool. The CSR contains the fully qualified DNS name of the server, the key pair, and other such information to help identify the web server.
2. Adam sends the CSR containing the web server's information to a well-known CA such as DigiCert.
3. The CA uses the CSR to generate a server certificate for the web server. The server certificate contains the key pair and the web server's information included in the CSR. The CA signs the certificate with its private key.
4. The CA sends the signed web server certificate back to Adam.
5. Adam imports the signed web server certificate into the web server, and the web server's certificate and private key are stored in the KeyStore on the web server.
6. When a browser accesses the web server, the web server sends a certificate signed by the private key of the CA to the browser. The browser has a copy of the CA's public key in its TrustStore and uses the public key to decrypt the signature of the CA. The browser now trusts any communication coming from this web server. For more information, see "Example of a Secure Handshake for the Client" on page 39.

Example of a Secure Handshake for the Client

The following example shows how a server receives a certificate from a well-known certificate authority (CA) to be able to communicate securely with any client. This example shows how the secure handshake occurs between a client and a server so that they can create their own secure communication channel that no other entities can use or access.

Figure 5-3 *A Browser Establishes a Secure Communication Channel to a Web Server*



In this example, Adam the administrator logs in to the administration console that is a web application. Every action except for Adam entering the URL of the web browser happens automatically between the browser and the web server. No user interaction is required.

1. Adam enters the URL into the browser. The browser sends a request to connect to the fully qualified DNS name of the web server.
2. The web server sends a copy of its server certificate that has been signed by the private key of a well-known CA.
3. The browser accesses the public key of the well-known CA that is stored in the browser's TrustStore. The browser uses the public key of the well-known CA to decrypt the signature on the web server's certificate to verify that the certificate is valid.
4. The browser generates a session key using the public key in the web server's certificate.
5. The browser sends the newly generated session key back to the web server.
6. The web server uses its private key stored in the KeyStore to decrypt the session key.
7. The web server verifies that the session key is not on the certificate revocation list (CRL). At this point the secure handshake between the browser and web server is established.
8. The web server encrypts the data using the session key and sends the data back in ciphertext to the browser. The browser uses the session key to decrypt the data, then uses the session key to encrypt data, and finally sends the data back in ciphertext. This secure communication continues until the session ends.

Enabling HTTPS Connections to Your OpenText Cloud Bridge Agent

In some instances, a company's security policy for on-premises applications requires that all HTTP connections be protected by the HTTPS protocol. This section describes how to generate a self-signed HTTPS certificate and import it into an Oracle® Java® keystore, and how to enable the OpenText Cloud Bridge Agent ("the Agent") to use that keystore to secure HTTP connections to the Agent user interface and APIs. The use of a self-signed certificate should be a temporary solution, and OpenText recommends that you take steps to replace the initial, self-generated certificate with a purchased certificate from a reputable certificate authority as soon as possible.

Prerequisites

Before you enable HTTPS for the OpenText Cloud Bridge Agent, verify that you have taken the following steps:

1. You have received and executed the installation script for the OpenText Cloud Bridge Agent.
2. The OpenText Cloud Bridge Agent is up and running on the Docker® or Podman host server.
3. You have the appropriate rights to the host server file system to add and edit some files.
4. You know the location of the OpenText Cloud Bridge Agent installation on the host server.
5. The host server has Oracle® Java® 11 keytool installed. You can check your version using the `java -version` command.

Enabling HTTPS

The process of enabling HTTPS includes the following steps:

1. Generate the new certificate and keystore.
2. Enable HTTPS connections.
3. Verify HTTPS connectivity.

Generate the New Certificate and Keystore

When you install the OpenText Cloud Bridge Agent, the `agent/conf/` directory is created below the installation location <*`Agent_install_dir`*>.

From the installation location:

1. Use the `cd` command to navigate to the <*`Agent_install_dir`*>`/agent/conf` directory.
2. Using keytool:
   - **2a** Generate the new certificate and Oracle® Java® keystore.

     ```
     keytool -genkey  -keyalg RSA -alias localhost -keystore
     bridge.keystore -storepass changeme -validity 360 -keysize 2048
     ```

   - **2b** Enter the appropriate information when prompted.

     When completed, a new file named `bridge.keystore` is created in the directory.

3 (Optional) If you are requesting a Certificate Authority (CA) signed (paid) certificate:

3a Using keytool, generate a key pair and use it to create a certificate signing request (CSR):

```
keytool -certreq -alias localhost -keystore bridge.keystore -
storepass changeme -file mycert.csr
```

The CSR contains the fully qualified DNS name of the server, the key pair, and other such information to help identify the web server.

3b Send the CSR to a well-known CA such as DigiCert.

The CA uses the CSR to generate a server certificate for the web server. The server certificate contains the key pair and the web server's information included in the CSR. The CA signs the certificate with its private key. The CA sends the signed web server certificate back to you.

3c When you get the real certificate from the CA, import it into the `bridge.keystore` using keytool.

When a browser accesses the web server, the web server sends a certificate signed by the private key of the CA to the browser. The browser has a copy of the CA's public key in its TrustStore and uses the public key to decrypt the signature of the CA. The browser now trusts any communication coming from this web server.

For more information about certificates, see "Understanding the Components of Secure Communication" on page 35.

Enable HTTPS Connections

To allow HTTPS connections to the OpenText™ Cloud Bridge Agent console and REST endpoints, you must edit the <*Agent_install_dir*>/agent/conf/bridge-agent.yml file. You must also update the container definition in the `create.sh` script file to allow connections to the HTTPS ports.

1 Edit the `bridge-agent.yml` file:

1a OpenText recommends that you copy the `bridge-agent.yml` file to something like `bridge-agent.yml.bak` in case you need to restore the file.

1b For both the `applicationConnectors:` and `adminConnectors:` sections in the file, remove the comment indicators (#) for the `– type:`, `port`, and `keystore` entries.

1c Save the changes.

2 Edit the `create.sh` file:

2a Use the `cd` command to return to the <*Agent_install_dir*>/agent/ directory.

2b Edit the `create.sh` script file:

2b1 Duplicate the entry for port mapping `-p "8080:8080" -p "8081:8081" \`

2b2 Copy this line into the file. Change the ports in this new line from `"8080:8080"` to `"8443:8443"` and `"8081:8081"` to `"8444:8444"`.

These will be the HTTPS connection ports.

3 Stop the OpenText Cloud Bridge Agent, recreate the container with the new ports mapped, then start the Agent:

3a From the <*Agent_install_dir*>/agent directory, stop the Agent:

```
sh stop.sh
```

3b Remove the `bridge-agent` container:

```
sh remove.sh
```

3c Recreate the container:

```
sh create.sh
```

3d (Conditional) If the `sh create.sh` command did not start the container, start the container:

```
sh start.sh
```

Verify HTTPS Connectivity

You should now be able to connect to the OpenText Cloud Bridge Agent console using the following URL:

`https://<hostname>:8443/`

Understanding the OpenText Cloud Bridge Agent TLS Security Policy

This section provides reference information for the OpenText Cloud Bridge Agent Transport Layer Security (TLS) policy. TLS is the successor to Secure Sockets Layer (SSL).

TLS Settings

Oracle® Java® ships with a security policy codified in the `java.security` policy file. It is typically located in the `conf/security` folder relative to `$JAVA_HOME`. OpenText Cloud Bridge containers are based upon the CAFapi Java 11 image. The CAFapi Java 11 container image disables weak TLS cipher suites in the `disableWeakTlsAlgorithms.patch` patch file. This is an excellent, if strict, security posture and meets the OpenText security requirements.

Terminology

This section uses the following terms:

- RSA (Rivest Shamir Adleman)
- ECC (Elliptic Curve Cryptography)
- Symmetric Cryptography (for example, AES)
- Asymmetric Cryptography (Public/Private Key pair)
- DH & ECDH (Diffie-Hellman & Elliptic-Curve Diffie-Hellman)
- Hash Function (for example, SHA1, SHA256, SHA512)
- CA (Certificate Authority)
- Host Certificate (Certificate used for a web server)

Disabled Algorithms

Ideally, we would like to restrict communication to TLS 1.3. However, we must also make it possible to use the TLS 1.2 protocol safely, which causes us to exclude cipher suites that are available but should not be used.

Highlights

1. Remove the TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256 cipher from the list.
2. Increase RSA key size: “RSA keySize < 2048”

Key Lengths

- RSA less than 2048-bit
- Diffie-Hellman less than 2048-bit
- Elliptic Curve less than 224-bit

The following algorithms are disallowed and will not successfully negotiate the "handshake" process:

- Camellia 128-bit with Cipher Block Chaining (CBC)
- AES 256-bit with CBC
- AES 128-bit with CBC
- TLS DH DSS with AES 256-bit GCM SHA384 (under translation...)

Oracle® Java® Security Policy

The following algorithms are disallowed:

```
TLSv1.1
TLSv1
SSLv3
SSLv2
DHE_DSS
RSA_EXPORT
DHE_DSS_EXPORT
DHE_RSA_EXPORT
DH_DSS_EXPORT
DH_RSA_EXPORT
DH_anon
ECDH_anon
DH_RSA
DH_DSS
ECDH
AES_256_CBC
AES_128_CBC
3DES_EDE_CBC
DES_CBC
RC4_40
RC4_128
DES40_CBC
RC2
HmacMD5
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
```

Additional Resources

For more information about configuring secure communication, see the following resources:

- The ROBOT Attack website
- *Testing for Weak SSL TLS Ciphers Insufficient Transport Layer Protection* on the OWASP website
- *Security/Server Side TLS* on the Mozilla wiki page

6 Managing the OpenText Cloud Bridge Agent

After the OpenText Cloud Bridge Agent ("the Agent") has been installed and configured, it does not typically require a lot of administrative maintenance, and the OpenText SaaS operations team has visibility into the overall health of your Agent. This section provides some basic information to help you maintain your Agent environment, and some troubleshooting tips for common issues.

Restarting the OpenText Cloud Bridge Agent

When you are ready to apply changes you have selected for an Agent instance, such as connector updates, you must restart the Agent.

To restart an Agent instance:

1. Log in to the host server where the Agent instance is installed.
2. Open a command line and run the following commands:
   - **2a** `cd <Agent_install_dir>`
   - **2b** `sh agent/stop.sh`
   - **2c** `sh agent/start.sh`
3. (Conditional) If you restarted the Agent because you added, updated, or deleted connectors, the Connector Update Results window opens.
   - **3a** Verify that the connectors updated successfully.
   - **3b** (Optional) If you do not want to see this information again, click the trashcan icon.

NOTE: The Connector Update Results window appears once per session or if you restart the Agent, at which time a new results file is generated. When you delete the information in this window, the information is deleted from the Agent console, but it is retained in backup files in the `/backup` folder.

Managing the Bootstrap Administrator Account

When you install the OpenText Cloud Bridge Agent, the installer sets up a bootstrap administrator account named `cbadmin`. You set a password for this account during installation and use those credentials to log in to the Agent console.

You cannot delete the bootstrap admin account, but you can change the password value if needed. As a best practice, change the bootstrap admin password in the `.env` file rather than entering it directly into the `bridge-agent.yml` configuration file.

To change the password:

1. Log in to the Agent host server.
2. Use the `cd` command to navigate to the `<Agent_install_dir>/agent` directory.
3. Open the `.env` file.
4. In the `Authorization` section, change the value for the `BOOTSTRAP_PWD` entry.
5. Restart the Agent. For more information, see "Restarting the OpenText Cloud Bridge Agent" on page 45.

Maintaining Your OpenText Cloud Bridge Agent Environment

Old Agent images and old connectors can consume valuable space on your host server. As a best practice, you should periodically review and remove unneeded files.

Removing Old Agent Images

The OpenText Cloud Bridge Agent upgrade process does not automatically remove older Agent images from Docker® or Podman in your environment. This allows you to keep images until you are sure your upgraded environment is working as expected. As a best practice, you should periodically review and manually remove unneeded old images so they do not consume space on the host server.

To remove old Agent images:

1. Log in to the Agent host server where you need to free up space.
2. Open a command line and run the appropriate command for your environment:
   - Docker®: `docker image prune [options]`
   - Podman: `podman image prune [options]`

For more information about available options, see the Docker® or Podman documentation.

Removing Old Connectors

When you delete a connector, the Agent deletes the old connector from the `/bridgelib` folder and copies it to the `/backup` folder. Similarly, when you update a connector the Agent copies the old connector to the `/backup` folder before installing the new one in the `/bridgelib` folder. The `/backup` folder contains both the connector `.jar` files and some backup `.json` files. The `.json` files contain the history of the most recent and older actions that have been taken.

When you are satisfied that your installed connectors are working properly, as a best practice you should periodically remove old connector `.jar` files from the `/backup` folder. You can also remove the `.json` files if you choose, though those files are very small.

To remove old connectors:

1. Log in to the Agent host server.
2. Open a command line and enter the following commands:
   - **2a** `cd <Agent_install_dir>/backup`
   - **2b** `rm <connector_jar_file>`

Uninstalling an OpenText Cloud Bridge Agent

You can uninstall an OpenText Cloud Bridge Agent instance if, for example, your business requirements change and you no longer need an Agent instance in a particular location or you are retiring a data center.

IMPORTANT: Before you uninstall an Agent, ensure that your environment is configured appropriately for high availability and you understand the order in which failover will occur. If you have not configured a secondary Agent instance, you will lose connectivity when you uninstall the Agent. For more information, see "Planning for High Availability" on page 13.

To uninstall an Agent instance:

1. Log in to the host server where the Agent instance is installed.
2. Open a command line and run the following commands:
   - **2a** `cd <Agent_install_dir>`
   - **2b** `sh agent/stop.sh`
   - **2c** `sh agent/remove.sh`
   - **2d** `rm -rf agent/`

Updating High Availability and Other Agent Settings

When you run the Agent installation script the first time, it is saved to the installation folder on the Agent instance. To make changes to any Agent settings, for example to change the instance or site priority in a high availability environment, you can rerun the `cb_agent.sh` installation script on the appropriate Agent instances.

To rerun the Agent installation script:

1. Log on to the host server of the Agent instance whose settings you want to change.
2. Run the following command:

   ```
   bash cb_agent.sh 2>&1 | tee -a install.log
   ```

3. Make the desired changes.

   For more information about high availability settings, see "Planning for High Availability" on page 13.

   The Agent instance restarts and applies your changes.

Repeat these steps for all Agent instances whose settings you need to change.

Troubleshooting Common Issues

To begin troubleshooting an error, you can check the main log file for the OpenText Cloud Bridge Agent. This file is located on the host server at `<Agent_install_dir>/agent/conf/daas_remote.log`. You can adjust the logging level in the Agent console so this log collects the appropriate amount of data for your environment.

Logger Configuration Issues

If you have too many Docker® container loggers running or logging levels configured too high, you might encounter disk space issues. In addition, the Agent can exceed the log limits so quickly that the error conditions are lost when the logs roll over. You can prevent or address these issues by manually changing the appropriate settings for the Docker® container in the `<Agent_install_dir>/agent/conf/bridge-agent.yml` file.

You can also configure logging levels in the `<Agent_install_dir>/agent/conf/daas-remote.yml` file. In the second section of the file, appenders can control the format of the log, the number of files, the size, and its rotation behavior.

NOTE: Avoid changing the log level for the `org.apache.kafka` file because it is very verbose and does not provide useful information.

Credential Issues

Data source credentials that appear in red on the Data Sources tab are credentials that the Agent has been unable to decrypt because they were encrypted using different keys. If you change the encryption IV and key for your Agent instances, all credentials will be displayed in red. This will also happen if you import the credentials from an Agent with different keys. Ensure that you use the same encryption IV and key values for all Agent instances in your high availability environment. For more information, see "Understanding Encryption IV and Key Values" on page 16.

Podman Red Hat® Enterprise Linux® 8.3 Does Not Automatically Restart the Container After a Server Restart

To fix this issue and ensure that your container automatically runs on Red Hat® Enterprise Linux® 8.3 Podman after a server restart, complete the steps in the following procedure. For more information, see the `podman generate systemd` reference page on the Podman website.

1. Stop your container:

   ```
   podman stop bridge-agent
   ```

2. Use the `podman generate systemd` command to create your system file:

   ```
   podman generate systemd --restart-policy=always --files --name bridge-agent
   ```

3. Locate the file with the name `container-bridge-agent.service`, then copy the file to the system folder:

   ```
   cp container-bridge-agent.service /etc/systemd/system/container-bridge-agent.service
   ```

4. Enable the service:

   ```
   systemctl enable container-bridge-agent.service
   ```

5. Start the service:

   ```
   systemctl start container-bridge-agent.service
   ```

6. Check the status:

   ```
   systemctl status container-bridge-agent.service
   ```

7. Restart your server to test it. For more information, see “Restarting the OpenText Cloud Bridge Agent” on page 45.

8. After a server restart, run the command `podman ps` and you should find your bridge-agent container running.