opentext™

OpenText Software Delivery Management

Software version: 25.1 & 25.3

Installation Guide for Windows

Go to Help Center online
https://admhelp.microfocus.com/octane/

Document release date: July 2025

Send Us Feedback

Let us know how we can improve your experience with the Installation Guide for Windows.

Send your email to: admdocteam@opentext.com

Legal Notices

Contents

Architecture

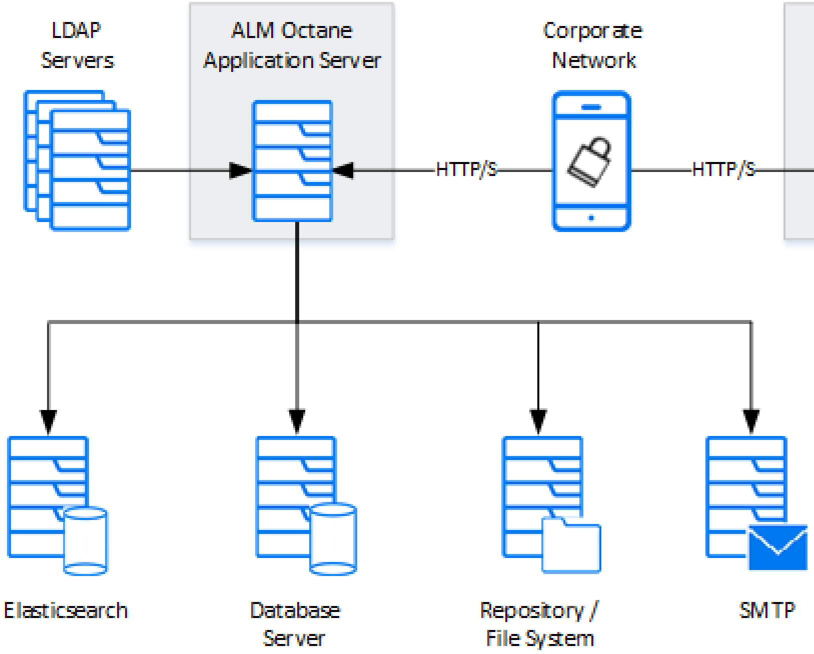
You can set up OpenText™ Software Delivery Management as a single node, or in a cluster configuration. The following diagrams illustrate the system architecture for both options. These are followed by descriptions of each of the components.

- "Basic configuration" below
- "Enterprise configuration" on the next page
- "Components" on page 7

Basic configuration

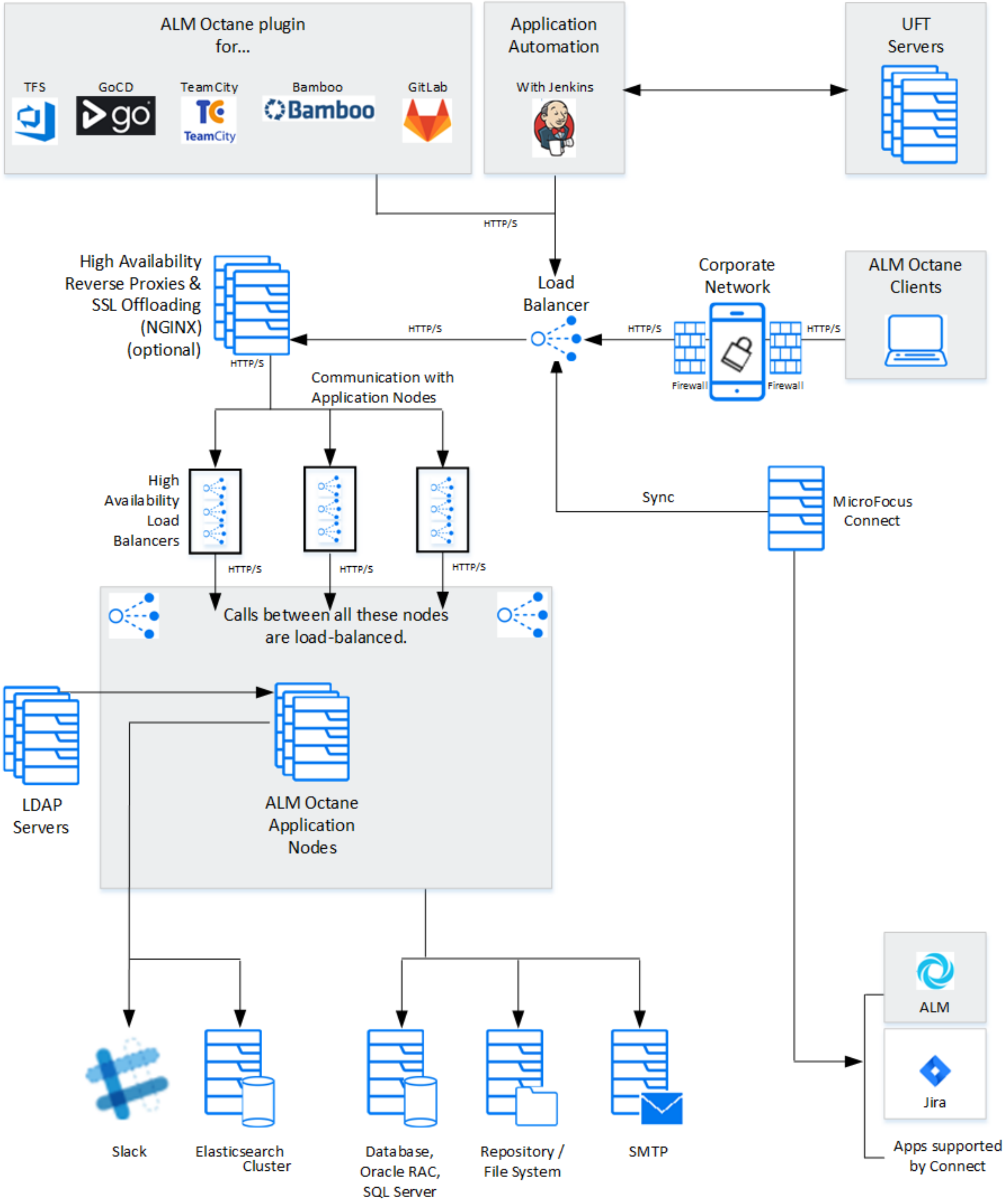
The following diagram illustrates the system architecture of a single-node configuration. Components in gray are OpenTextproducts.

> **Note:** The OpenText Software Delivery Management, database, and Elasticsearch servers should each reside on separate machines.



Enterprise configuration

The following diagram illustrates the system architecture of an enterprise, cluster configuration. Components in gray are OpenText products.



Components

Components	Description
OpenText Software Delivery Management clients	The clients communicate with the OpenText Software Delivery Management server over HTTP/S.
OpenText Software Delivery Management Server application nodes	Client requests from OpenText Software Delivery Management are dispatched to the deployed application. **Note:** The OpenText Software Delivery Management, database, and Elasticsearch servers should each reside on separate machines.
OpenText Software Delivery Management application additional cluster (sync) nodes	**Cluster configuration**: A cluster is a group of application servers that run as a single system. Each application server in a cluster is referred to as a "node." • All nodes must have access to the database server on which the site database schema resides. • All nodes must have access to the repository. Generally, the repository is located on an NFS or SAN server. • All nodes must have access to each other.
Repository / File system	Stores all files to be used by all the projects in the system, such as templates and attachments. **Cluster configuration**: When working in a clustered configuration, the repository must be accessible by all nodes. Also, the repository must be configured to use the same path on all nodes.

Components	Description
Database server	A relational database management system, either Oracle RAC or Microsoft SQL Server. The database server stores the following schemas: • **Site schema**. Stores all site-related information, such as database servers, cluster nodes, the SMTP servers, and configuration. • **Space schema**. All space information, such as workspaces, users, and roles. This server can be shared with other applications with the following constraints: • The database must be able to sustain the load of all the applications. • Future versions of OpenText Software Delivery Management might require a database upgrade. This may necessitate migration of data if other applications sharing the same database do not support the database version that OpenText Software Delivery Management requires. **Note:** The OpenText Software Delivery Management, database, and Elasticsearch servers should each reside on separate machines.

Components	Description
Elasticsearch server (or cluster)	A Java-based, open-source search engine. This component is used for various aspects of the application, such as global search and trends. This server can be shared with other applications with the following constraints: • The Elasticsearch engine must be able to sustain the load of all the applications. • Future versions of OpenText Software Delivery Management might require an Elasticsearch upgrade. This may necessitate migration of data if other applications sharing the same Elasticsearch do not support the Elasticsearch version that OpenText Software Delivery Management requires. **Note:** The OpenText Software Delivery Management, database, and Elasticsearch servers should each reside on separate machines. A working Elasticsearch server is a requirement for working with OpenText Software Delivery Management. For the supported version, see Database and Elasticsearch in the OpenText Software Delivery Management Help Center.
Load balancer	**Cluster configuration**: When working with a load balancer, client requests are transmitted to the load balancer and distributed according to server availability within the cluster. If you are using a load balancer, we recommend you utilize SSL offloading.
High availability load balancers	**Cluster configuration**: These can be "VIPs" (virtual IP addresses) of one physical load balancer.
DMZ	An optional, demilitarized zone.
High availability reverse proxies and SSL offloading	**Cluster configuration**: Optional configuration for load balancing using a software solution (for example, NGINX).
SMTP	A mail server.
Jenkins (with OpenText Software Delivery Management plugin)	**Enterprise configuration**: You can integrate OpenText Software Delivery Management with a Jenkins CI server using the Application Automation Tools Plugin on your CI server.

Components	Description
TFS, TeamCity, or Bamboo server (with OpenText Software Delivery Management plugin)	**Enterprise configuration**: You can integrate OpenText Software Delivery Management with a TFS, TeamCity, or Bamboo CI server using the OpenText Software Delivery Management plugin on your CI server.
Slack	Integration with Slack, which enables all stakeholders of a backlog item or pipeline run failure to collaborate and communicate. You can integrate with Slack by adding it as a collaboration tool associating it with a workspace.
Open Text testing tools: OpenText Functional Testing, OpenText Functional Testing for Developers, OpenText Core Performance Engineering, OpenText Enterprise Performance Engineering	You can integrate OpenText Software Delivery Management with Open Text testing tools. For details, see Integrations overview in the OpenText Software Delivery Management Help Center.

Installation types

This document describes the necessary requirements and procedures for the installation of OpenText Software Delivery Management server, and initial setup steps.

Type	Description
This Windows Installation	Instructions for installing on: • A single node. For details, see "Installation flow" on page 15. • A cluster configuration. For details, see "Cluster installation flow" on page 18.
Docker installation	A simplified installation of OpenText Software Delivery Management by deploying a Docker image. For details, see "Install using a Docker image" on page 67.
Upgrade	For details, see Upgrade in the Help Center.

Licensing flow

This topic provides a high-level flow for setting up your trial license.

This section includes:

- "Overview" below
- "Request a trial" below
- "Using Pro Edition" on the next page
- "Install a license" on the next page

Overview

To get started with OpenText Software Delivery Management, you begin with a 90-day on-premises free trial for 100 users. You can then install an OpenText Software Delivery Management license file, or allocate licenses from OpenText Application Quality Management.

Before you begin a trial, you should be familiar with the different editions. OpenText Software Delivery Management is available in Enterprise and Pro Editions. For details, see OpenText Software Delivery Management editions in the OpenText Software Delivery Management Help Center.

Request a trial

Submit a request for a free trial here: https://www.opentext.com/en-gb/products/alm-octane.

When you first start using OpenText Software Delivery Management, you automatically receive a **Trial** license which gives you a 90-day trial for 100 users.

By default, your trial is Enterprise Edition, which allows one shared space. If you create a shared space in an Enterprise Edition trial and then install a license for Pro Edition, the trial shared space should not be used in a production environment since the sharing capabilities may not be supported in future releases.

Using Pro Edition

There is no Pro Edition trial.

To work with Pro Edition:

1. Install OpenText Software Delivery Management Enterprise Edition as your trial type, but do not create shared spaces. If you create a shared space during an Enterprise Edition trial and then install a Pro Edition license, the shared space is deactivated.
2. Get an evaluation Pro Edition license from your Sales account manager, or create a support ticket for a one-time evaluation license.
3. In the OpenText Software Delivery Management **Settings** ⚙ area, apply your Pro Edition license. For details about applying licenses, see "Install a license" below.

Install a license

After you install and configure your trial instance, you can purchase licenses for Enterprise or Pro Edition. You then install your license key (.dat file) in OpenText Software Delivery Management.

Alternatively, you can allocate your current licenses from OpenText Application Quality Management and share them with OpenText Software Delivery Management. Licenses can be allocated from OpenText Application Quality Management(ALM.Net) Edition to OpenText Software Delivery Management Enterprise Edition, or from OpenText Application Quality Management (QC) Enterprise Edition to OpenText Software Delivery Management Pro Edition.

> **Note:** You can share up to 15% of your licenses from OpenText Application Quality Management, or up to 150 licenses, the lower of the two.

For more details, see Manage licenses in the OpenText Software Delivery Management Help Center.

Next steps:

- "Installation flow" on the next page

Installation flow

This document describes the overall flow for installing the OpenText Software Delivery Management server on Windows.

This section includes:

- "Prerequisites " below
- "Deployment " on the next page
- "Configuration" on the next page
- "Start the server" on page 17
- "Verify and log in " on page 17
- "Configure cluster (optional) " on page 17

Prerequisites

Verify your system meets hardware and software requirements.

This includes setting up permissions, opening ports, database configuration, and more.

You need three separate server machines.

- OpenText Software Delivery Management server
- Database server
- Elasticsearch server

For details, see "Prerequisites" on page 22.

> **Note:** We recommend that you review security considerations in OpenText Software Delivery Management Secure Deployment and Configuration Guidelines for instructions on setting up a secure configuration.

Deployment

Deploy OpenText Software Delivery Management on a machine dedicated for the OpenText Software Delivery Management server on Windows.

OpenText Software Delivery Management is deployed using an installation program.

The default deployment path is **C:\Program Files\octane**.

The command to deploy is: `octane-onprem-<version>.exe`

For details, see "Deployment" on page 34.

Configuration

This section describes the initial configuration.

To configure:

1. Edit the **octane.conf** file with your site's settings for initial configuration.
2. (Optional) Depending on your needs, configure optional configuration files:
   - **elasticsearch-security.conf** to configure secure Elasticsearch.
   - **proxy.conf** to use a proxy server.
   - **ldap.conf** to use LDAP authentication.
   - **sso.conf** to use SSO authentication.

   The path to these files is **<Repository folder>\conf**.

For details, see "Configure site settings" on page 37.

> **Note:** The .conf files do not support use of backslashes (\) in paths. Instead, use a regular slash (/) or double-slash (//).

Start the server

Select **Start > OpenText Software Delivery Management > Start OpenText Software Delivery Management Server**.

For details, see "Start the server" on page 63.

Verify and log in

Verify that OpenText Software Delivery Management was properly installed.

Log into OpenText Software Delivery Management. For details, see "Log in" on page 65.

Configure cluster (optional)

After starting the server on the first machine, configure and initialize each additional cluster node. For details, see "Cluster installation flow" on the next page.

Next steps:

- "Prerequisites" on page 22
- "Deployment" on page 34
- "Configure site settings" on page 37

Cluster installation flow

This section provides end-to-end instructions for installing an on-premises OpenText Software Delivery Management server in a cluster configuration on Windows. A cluster is a group of application servers that run as a single system. Each application server in a cluster is referred to as a "node."

To install in a cluster configuration:

1. For each node in the cluster, check requirements and access.

Area	Instructions
Check requirements	Verify that the all cluster nodes, including the first, meet all requirements and prerequisites. For details, see "Prerequisites" on page 22.
Check database server access	All cluster nodes, including the first, must have access to the database server on which the site database schema resides.
Check repository access	The repository directory has to be a shared directory visible to all cluster nodes. All nodes must have read and write access to the repository. Generally, the repository is located on an NFS or SAN server. The repository must be configured to use the same mount point (path) on all nodes. It is important that you enter the repository path using the same path name on all nodes.
Check access between nodes	All nodes must have access to each other. Verify ports are open in your firewall. OpenText Software Delivery Management needs to communicate between the nodes in the cluster on port 5701. Therefore, make sure that your firewall enables communication between the nodes of the cluster on the specified port.. By default, outbound ports are open. Check inbound ports.

2. Install OpenText Software Delivery Management on the first cluster node, as described in "Installation" on page 34.

   a. Deploy the installation files onto the first node. Make sure to set the **Repository folder** as a location that all cluster nodes can access.

   b. Configure initial site settings in **octane.conf** and optional configuration files.

      - Make sure to set the **database server name** to a value that all cluster nodes can access.

      - Enter values described in "Cluster settings" on page 47.

These settings are validated when starting. If they are not valid, the OpenText Software Delivery Management server does not start.

c. On the first node only, start the OpenText Software Delivery Management server. See "Start the server" on page 63.

3. (Optional) If you want to set up a secure configuration, follow the instructions in OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

4. Log in to the first node in the cluster. For details, see "Log in" on page 65.

5. Download and deploy the OpenText Software Delivery Management package on each cluster node. For details, see "Deployment" on page 34 and "Deploy in cluster environment" on page 37.

> **Caution:** Do not configure **octane.conf** or other configuration files. Each node is automatically configured using the configuration files located in the repository, as defined when you configured the first node.

6. On each node, start the OpenText Software Delivery Management server. See "Start the server" on page 63.

7. (Optional) If you want to set up a secure configuration for OpenText Software Delivery Management in a cluster configuration, follow these instructions on each other node: OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

8. Log in to make sure OpenText Software Delivery Management is running on each other node. For details, see "Log in" on page 65. Use the load balancer URL when you log in.

> **Tip:** For best performance, configure your load balancer with round-robin (stateless) configuration.

9. If you need to make changes in configuration settings later, edit the **<Repository folder>\conf\octane.conf** file, and not **octane.conf.new**, which is a temporary file that is for internal use only. After modifying these settings, restart the OpenText Software Delivery Management server on each node to pull the configuration changes from the repository.

Troubleshooting:

If the cluster was not properly defined, you may receive an error message when you start the OpenText Software Delivery Management server:

```
Cluster is unhealthy...
```

During installation, values in the **hazelcast.xml** file change according to the **octane.conf** configuration. The configuration of the **hazelcast.xml** file is the one that controls the cluster behavior.

Make sure that the **member** element of the **hazelcast.xml** file contains the same values that were defined in the **nodes** section of the **octane.conf** file.

Next steps:

- "Prerequisites" on the next page
- "Deployment" on page 34
- "Configure site settings" on page 37

Prerequisites

Verify that your system meets the requirements listed below, and the detailed Support matrix in the OpenText Software Delivery Management Help Center.

For security requirements, see the OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

This section includes:

Checklist

Use the following questions to make sure you are ready to install.

OpenText Software Delivery Management

Question	Answer
On which machine will you be installing OpenText Software Delivery Management?	
Does the machine have a Quad Core AMD64 processor or equivalent x86-compatible processor?	
How much memory does the machine have? You need a minimum of 8 GB. Contact customer support for site-specific recommendations.	
Does the machine have a minimum of 8 GB free disk space? Contact customer support for site-specific recommendations.	
What Microsoft Windows operating system is on the machine?	

Question	Answer
What is the user name and password you will use for the installation user? **Limitation:** The **$** character is not allowed in the user name or password.	
Are your browsers and screen resolutions compatible with OpenText Software Delivery Management?	
On-premises installation of OpenText Software Delivery Management supports only English characters for the names of schemas, operating systems, users, and so on. Did you check?	

Elasticsearch

Question	Answer
Does your Elasticsearch version match OpenText Software Delivery Management requirements? For details, see Support matrix in the OpenText Software Delivery Management Help Center.	
Do you need to download Elasticsearch? If you haven’t installed Elasticsearch, you can download it from the product's website.	
On which machine is Elasticsearch installed?	
Did you make sure that the port for outbound communication to Elasticsearch is open? By default, outbound ports are open.	
Did you make sure that the Elasticsearch ports (such as 9300 and 9200) are accessible directly from the OpenText Software Delivery Management server, not just by checking the HTTP connection?	
What is the name of the Elasticsearch cluster you have configured?	
Is the Elasticsearch accessible from the OpenText Software Delivery Management server?	
Was Elasticsearch configured according to OpenText Software Delivery Management requirements? These are described in detail in "Configure Elasticsearch" on page 31.	

Oracle

Question	Answer
Does your Oracle version match OpenText Software Delivery Management requirements? For details, see Support matrix in the OpenText Software Delivery Management Help Center.	
On which machine is the database installed?	
What is the Oracle database port? Default: 1521 You can modify the port in **octane.conf**.	
Did you make sure that the port for outbound communication to Oracle is open? By default, outbound ports are open.	
What is the URL for Java Database Connectivity (JDBC) for your database?	
What is the database admin's user name and password?	
Does the database admin have the necessary permissions? See "Oracle database permissions" on page 26.	
What table space and temporary table space can be used?	
Did the DBA add any objects to the schemas? If so, create an exception file before installing. For details, see "Using exception files for manual database changes" on page 77.	

Microsoft SQL Server

Question	Answer
Does your SQL Server version match OpenText Software Delivery Management requirements? For details, see Support matrix in the OpenText Software Delivery Management Help Center.	
On which machine is the database installed?	
Will you be using the SQL Server database port or instance name to connect to the database? • What is the SQL Server database port? Default: 1433 • What is the SQL Server instance name?	

Question	Answer
What is the database admin's user name and password?	
Does the database admin power user have the necessary permissions? See "SQL database permissions" on page 28.	
What MSSQL database login user, and password, can be used for OpenText Software Delivery Management?	
Did the DBA add any objects to the databases/schemas? If so, create an exception file before installing. For details, see "Using exception files for manual database changes" on page 77.	

Java

Question	Answer
Do you need to install the JDK on the OpenText Software Delivery Management server and other servers, such as the ElasticSearch server?	
Does your Java version match OpenText Software Delivery Management requirements? For details, see Support matrix in the OpenText Software Delivery Management Help Center.	

Jetty

Question	Answer
Did you make sure that the port for inbound communication with Jetty is open? By default, the port is 8080. For SSL, 8443. You can define the port during initial installation, in **octane.conf**.	

Hazelcast

Question	Answer
Did you make sure that OpenText Software Delivery Management can communicate between the nodes in the cluster, using inbound and outbound communication for clusters? By default, the port is 5701. You can define the port during initial installation, in **hazelcast.xml**.	

File system permissions

The user installing OpenText Software Delivery Management should be an administrator on the machine, and should be able to create services.

Oracle database permissions

Permissions depend on whether you want OpenText Software Delivery Management to create schemas, objects, and tables during installation, or if you prefer your DBA to prepare them.

Refer to the relevant section for your installation scenario:

- "Allow OpenText Software Delivery Management to create Oracle schemas automatically " below
- "Create your own Oracle schemas for OpenText Software Delivery Management" on the next page

Allow OpenText Software Delivery Management to create Oracle schemas automatically

To enable OpenText Software Delivery Management to create schemas, tables, and objects automatically during the installation, provide OpenText Software Delivery Management with an Oracle power user with the following admin privileges:

- CREATE USER
- CREATE SESSION WITH ADMIN OPTION
- CREATE TABLE WITH ADMIN OPTION
- CREATE SEQUENCE WITH ADMIN OPTION
- DROP USER (optional). If not provided, the DBA must take responsibility for cleaning up unnecessary schemas.

> **Note:** These permissions are for the user you specify in the **admin-user > name** setting in the **octane.conf** file. For details, see "admin-user > name" on page 41.
>
> When defining your site action in the **octane.conf** file, you will specify **CREATE_NEW**. For details, see "CREATE_NEW" on page 44.

This power user can also be created temporarily, for installation purposes only. You can remove this user if:

- The installation is complete, and login to OpenText Software Delivery Management is successful.
- The OpenText Software Delivery Management site admin intends to create spaces using an existing schema, which can be selected when creating a space in the OpenText Software Delivery Management Settings area for the site. For details, see Manage spaces - site admins in the OpenText Software Delivery Management Help Center.

Create your own Oracle schemas for OpenText Software Delivery Management

If you do not want OpenText Software Delivery Management to create schemas, tables, and objects automatically, perform the following:

1. Before installation, create two schemas with the same password.
2. Provide OpenText Software Delivery Management with a regular Oracle user with the following permissions, for both the site and space schemas:
   - CREATE TABLE
   - CREATE SESSION
   - CREATE SEQUENCE
   - The QUOTA clause on the user's default tablespace should be unlimited.

> **Note:** To allow OpenText Software Delivery Management to use schemas you have created, you will specify the **FILL_EXISTING** site action when defining your **octane.conf** file. For details, see "FILL_EXISTING" on page 44.

SQL database permissions

Permissions depend on whether you want OpenText Software Delivery Management to create databases during the installation, or if you prefer your DBA to prepare them.

Refer to the relevant section for your installation scenario:

- "Allow OpenText Software Delivery Management to create SQL databases automatically" below
- "Allow OpenText Software Delivery Management to create SQL databases when using Windows Authentication" on the next page
- "Create your own SQL databases for OpenText Software Delivery Management" on the next page
- "Create your own SQL databases when using Windows Authentication" on page 30

Allow OpenText Software Delivery Management to create SQL databases automatically

To enable OpenText Software Delivery Management to create databases automatically during the installation, use the **sa** user, or an OpenText Software Delivery Management database admin power user.

Install OpenText Software Delivery Management with a database admin power user if you cannot use the SQL **sa** user for security reasons. This user can be a temporary user, for installation purposes only.

Request that the SQL Server database admin create a temporary power user with the following privileges (roles), which are required to install OpenText Software Delivery Management:

- Database Creators **dbcreator** role
- Security Administrator **securityadmin** role

> **Note:** These permissions are for the user you will specify in the **admin-user > name** setting in the **octane.conf** file. For details, see "admin-user > name" on page 41.

> To allow OpenText Software Delivery Management to create databases, you will specify the **CREATE_NEW** site action when defining your **octane.conf** file. For details, see "CREATE_NEW" on page 44.

It is important that the database administrative user is not the same as the admin user. The SQL Server database admin could name this power user **octane_install_power_user**, for example. For details on removing this temporary power user, see "Handle database-related issues" on page 70.

Allow OpenText Software Delivery Management to create SQL databases when using Windows Authentication

1. If you are using Windows authentication, create a new Windows domain login user in your database before installing OpenText Software Delivery Management. Select the **Windows authentication** option, and and use the credentials of a Windows domain user. Provide this user with the **sysadmin** or **dbcreator** role.
2. After installing OpenText Software Delivery Management, use this user to run the OpenText Software Delivery Management service. In the OpenText Software Delivery Management service properties, do not use Local system account to run the service, but rather use this user.

When defining your **octane.conf** file, you will enter **WINDOWS** as your authentication method. For details, see "Authentication Type" on page 51.

Create your own SQL databases for OpenText Software Delivery Management

If you do not want OpenText Software Delivery Management to create databases, create two databases before installation: one for the site and one for the space.

Associate the login user to ‘octane’ user in both databases.

The default collation is **SQL_Latin1_General_CP1_CI_AS** (must be case-insensitive).

> **Example: Create a database and grant user access**

```
Use master
CREATE DATABASE <database_name>
GO
alter database <database_name> SET READ_COMMITTED_SNAPSHOT ON
GO
CREATE LOGIN <login_name> WITH PASSWORD = 'thepassword'
GO
USE <database_name>
CREATE SCHEMA [octane]
GO
CREATE USER [octane] FOR LOGIN WITH DEFAULT_SCHEMA= [octane]
GO
ALTER AUTHORIZATION ON Schema::octane TO [octane]
GO
ALTER ROLE [db_ddladmin] ADD MEMBER [octane]
GO
```

Run the previous commands separately for each database (site schema and space schema).

> **Note:** During installation when you define the **octane.conf** file, you will enter the name of the site schema in **schemas > site**, the space schema in **schemas > initial-shared-space**, and the password in **schema-password**.
>
> To allow OpenText Software Delivery Management to use databases you have created, you will specify the **FILL_EXISTING** site action when defining your **octane.conf** file. For details, see "FILL_EXISTING" on page 44.

Create your own SQL databases when using Windows Authentication

1. If you are using Windows authentication, create two databases.
2. Assign the **db_owner** role to the Windows authentication user for these databases.

You do not need to associate the login user to ‘octane’ user in the databases.

When defining your **octane.conf** file, you will enter **WINDOWS** as your authentication method. For details, see "Configure site settings" on page 37.

Configure Elasticsearch

Before installing OpenText Software Delivery Management, there are a number of settings you must configure in Elasticsearch.

> **Note:** Elasticsearch supports indexes that were created in the current Elasticsearch main version, or one earlier version. Each time OpenText Software Delivery Management extends support for a new Elasticsearch main version, the OpenText Software Delivery Management upgrade includes a reindex process for the older indexes.

To configure Elasticsearch settings:

1. In the **elasticsearch.yml** file, configure the following:

   - **cluster.name**. Assign a unique name which will be used when you configure OpenText Software Delivery Management to connect to the cluster. Note that even a single-server installation is considered a cluster.
   - **node.name**. If you do not assign the node a name, Elasticsearch generates a random name on every reboot.
   - **network.host**. The node binds to this hostname or IP address and publishes this host to other nodes in the cluster. You can enter an IP address, hostname, a special value, or an array of any combination of these. Defaults to **_local_**.
   - **action.auto_create_index**. In each of your Elasticsearch cluster nodes, you must have the following line in the elasticsearch.yml files:

     ```
     action.auto_create_index: "-mqm_*,*"
     ```

     > **Note:** If you already have an **action.auto_create_index** line in the yml file, add the **-mqm_*** phrase to the beginning of its specified value. For example, if you have the following line:
     >
     > ```
     > action.auto_create_index: "-index*,*"
     > ```

> You would change that to:
>
> ```
> action.auto_create_index: "-mqm_*,-index*,*"
> ```

2. You can configure Elasticsearch securely using TLS. For details, see https://softwaresupport.softwaregrp.com/doc/KM03712315.

3. In the **jvm.options** file, set the following parameters: **-Xms<*value*>g** and **-Xmx<*value*>g**.

   Define *value* as half of memory available on the machine - 1, but no more than 31GB.

Configuring an Elasticsearch cluster

Elasticsearch can run on a single node but it is designed to run as a cluster. We do not recommend running a production environment on a single host Elasticsearch instance.

Elasticsearch clusters should have at least 3 nodes, or a larger odd number.

To configure an Elasticsearch cluster, modify the following parameters in the **elasticsearch.yml**:

- **cluster.name**. This name should be identical on all nodes of the cluster to make sure they join the same cluster.
- **discovery.seed_hosts**. To form a cluster with nodes on other hosts, use the static **discovery.seed_hosts** setting to provide a list of other nodes in the cluster that are master-eligible, and likely to be live in order to seed the discovery process.

> **Note:** The cluster nodes should be able to communicate with each other, meaning, the ports should be open in the firewall.

Restart Elasticsearch

After changing Elasticsearch setting files (for example elasticsearch.yml or jvm.options), you must restart the Elasticsearch service.

Backing up Elasticsearch

We recommend performing ELS snapshot at the same time as database backup and file repository backup.

OpenText Software Delivery Management does not need to be stopped for this operation.

Consider creating a batch to back up Elasticsearch data on a regular basis.

Next steps:

- "Deployment" on the next page

Installation

This section describes how to install an on-premises OpenText Software Delivery Management server using Microsoft Windows.

Before installing:

- Verify that your server fulfills all prerequisites. For details, see System Requirements in the OpenText Software Delivery Management Help Center.
- Review the OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

> ! **Language support:** On-premises installation of OpenText Software Delivery Management supports only English. This means only English characters can be specified for the names of schemas, operating systems, and users.

This section includes:

Deployment

This section describes how to deploy the files necessary for installing an OpenText Software Delivery Management server.

This section includes:

- "Overview" below
- "Prerequisites" on the next page
- "Deploy" on the next page
- "Deploy in cluster environment" on page 37

Overview

Installing OpenText Software Delivery Management does the following:

- Creates the correct folder structure and copies all the files to the correct locations.
- Installs the OpenText Software Delivery Management service so that the operating system recognizes it.

Prerequisites

Before installing:

- Verify that your server fulfills all prerequisites. For details, see System Requirements in the OpenText Software Delivery Management Help Center.
- Review the OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

Deploy

This section describes how to deploy the OpenText Software Delivery Management package.

To deploy:

1. Download the package:

   https://sld.microfocus.com/mysoftware/download/downloadCenter

   > **Tip:** To verify the digital signature of the RPM package, see "Installation Security" in the OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

2. Install the package, by running as an administrator:

   ```
   setup.exe
   ```

   Click **Next**.

3. In the installation wizard panes, set the following.

Field	Description
Installation folder	The folder in which to install OpenText Software Delivery Management. The default is **C:\Program Files\octane**. Do not enter a name with spaces for the folder.
Log folder	The folder in which to create log files. The default is **C:\Program Files\octane\log**.
Repository folder	• **Single node**: Full path of the repository folder. The default is **C:\Program Files\octane\repo**. • **Cluster installation**: The repository folder has to be a shared directory visible to all cluster nodes. For example, **MACHINE_NAME\FOLDER_NAME\repo.** ◦ All nodes must have read and write access to the repository. ◦ You must enter the repository folder using the same path name on all nodes.
Service user	Whether the service should use the local system account or a specific user.
Service user domain	The domain of the user that starts the service. Available when the **Service user** is **Custom user**.
Service user name	The name of the user that starts the service. This user must have administrative permissions if using Microsoft SQL Server, and must be a local administrator. **Limitation:** The **$** character is not allowed in the user name or password. Available when the **Service user** is **Custom user**.
Password	Password for the user that will start the service. **Limitation:** The **$** character is not allowed in the user name or password. Available when the **Service user** is **Custom user**.
Start Menu folder	Location of the shortcuts in the **Start** menu. The default is **Start > ALM Octane**.

Click **Next**. The installation starts deploying files.

4. Click **Finish**.

5. Verify that you have full administrator permissions for the following.

Default folder	Description
C:\Program Files\octane	The installation folder and all its sub-directories and files. These files are used for configuring the server.
C:\Program Files\octane\repo	The repository folder, and its site and spaces sub-directories.
C:\Program Files\octane\log	Log file folder.

6. If planning to install on additional cluster nodes, perform the steps described under "Deploy in cluster environment" below.

Deploy in cluster environment

This section describes how to deploy in cluster environment.

To deploy in cluster environment:

1. Configure the IP addresses (or fully qualified domain names) of the cluster nodes.

   Configure the node IP addresses or fully qualified domain names in the **octane.conf** file. For details, see "Configure site settings" below.

2. Verify ports are open in your firewall.

   When deploying over a cluster, OpenText Software Delivery Management needs to communicate between the nodes in the cluster located on port 5701. Therefore, make sure that your firewall enables communication between the nodes of the cluster on the specified port.

Configure site settings

Configure site settings using the configuration files:

- The **octane.conf** settings are mandatory for all environments.
- In addition, there are other settings that are required in complex environments. These

include secure Elasticsearch, proxy settings, and LDAP or SSO authentication, as described below.

These settings are configured during installation, and can also be changed any time, whenever necessary.

This section includes:

Workflow

1. Configure basic settings by editing the **<Repository folder>\conf\octane.conf** file.

   In addition, depending on your environment, configure the optional files described in the following sections.

   > **Note:** The **.conf** files do not support use of backslashes (\) in paths. Instead, use a regular slash (/) or double-slash (//).

2. If you are installing OpenText Software Delivery Management, after editing your configuration files proceed with "Start the server" on page 63.

3. If you need to make changes in configuration files later, make sure you edit the **<Repository folder>\conf\octane.conf** file, and not **octane.conf.new**, which is a temporary file that is for internal use only.

   After modifying these settings, restart the OpenText Software Delivery Management server on each node to pull the configuration changes from the repository. For details, see Modify site settings in the OpenText Software Delivery Management Help Center.

   For example, you might initially install OpenText Software Delivery Management to use native user management, and at a later time, decide to implement LDAP authentication for user management instead.

> **Tip:** We recommend that you save a local copy of the **octane.conf** file before making changes to it. Also, for security purposes, **octane.conf** should be stored in a secure, off-site location.

Database server settings

The following are the database server settings.

Setting	Description
db-type	Enter **ORACLE** or **MSSQL**.
connection-string	The Java Database Connectivity (JDBC) database connection string. It includes the following details: database type, database server name, database server port number, service name. Oracle connection-string The instructions below demonstrate how to set up the string with non-secured database access. To configure secure access to the database, see "Using SSL/SSO in Oracle (optional)" on the next page. **Syntax using service names:** `jdbc:oracle:thin:@//DB_SERVER_NAME:DB_SERVER_PORT/DB_SERVICE_NAME` **Examples:** - `jdbc:oracle:thin:@//dbserver1.net:1521/orcl` - `jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=dbserver1.net)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=orcl)))` **Note:** To connect to Oracle RAC, use the Single Client Access Name (SCAN) instead of the database server name.

	SQL connection-string
	- **Syntax using port:**
	`jdbc:sqlserver://DB_SERVER_NAME:DB_SERVER_PORT`
	Example:
	`jdbc:sqlserver://dbserver1:1433`
	- **Syntax using instance:**
	`jdbc:sqlserver://DB_SERVER_NAME;instanceName=INSTANCE_NAME`
	Example:
	`jdbc:sqlserver://dbserver1;instanceName=my_instance`

Setting	Description
admin-user > name	OpenText Software Delivery Management uses the **admin-user** both to create objects during installation and also to check that the database server is accessible. • For Oracle, enter the name of the database admin user. • For SQL Server, enter the **sa** user, or an SQL Server power user with the correct permissions. For details about **admin-user** permissions, see "Prerequisites" on page 22.
admin-user > password	The password of the database admin user. Do not include a pound sign (**#**) or accented characters (such as, **ä**, **ç**, **ñ**).
schemas > site	The name of the site schema that will be created by the **admin-user** during the installation, or supplied by the organization's DBA. Enter the supplied name.
schemas > initial-shared-space	This parameter is relevant only for the **FILL_EXISTING** site action. If you are using **FILL_EXISTING**, set the **initial-shared-space** to the name of the schema that is designated for the space.

Using SSL/SSO in Oracle (optional)

You can configure a secure connection from the OpenText Software Delivery Management server to the database server using SSL or SSO.

1. On the Oracle database server, convert the client wallet to jks keystore:

   ```
   orapki wallet pkcs12_to_jks -wallet "<path to client wallet folder>/<client wallet folder name>" -pwd <wallet_password> -jksKeyStoreLoc <name of your jks file>.jks -jksKeyStorepwd <jks_pass>
   ```

   For example:

   ```
   orapki wallet pkcs12_to_jks -wallet
   "/home/oracle19/wallets/client_wallet" -pwd aaa123456 -
   jksKeyStoreLoc clientstore.jks -jksKeyStorepwd test123#456
   ```

2. Check the content of the newly created jks keystore:

   ```
   keytool -list -keystore <name of your jks file>.jks -storepass <jks_pass>
   ```

For example:

```
keytool -list -keystore clientstore.jks –storepass test123#456
```

3. Copy the client wallet file from the Oracle database server to the OpenText Software Delivery Management Server. Place the newly-created keystore jks file in a location on the OpenText Software Delivery Management app server that is accessible to all, such as **C:\Program Files\Octane\conf\<name of your jks file>**.
4. Copy the following to **octane.conf**, after the **connection-string** parameter. Replace the values with those specific to your installation:

```
connection-properties : [
      {
          "key" : "javax.net.ssl.trustStore",
          "value" : "<full path to keystore file>/<jks keystore
file name>.jks"
      }
  ,
      {
          "key" : "javax.net.ssl.trustStoreType",
          "value" : "JKS"
      }
  ,
      {
          "key" : "javax.net.ssl.trustStorePassword",
          "value" : "<jks keystore password>"
      }
  ]
```

Oracle server settings

The following are the Oracle database server settings.

Setting	Description
schema-password	The password of the site schema. When installing using existing site schemas (with the **FILL_EXISTING** site action), make sure that the passwords that the DBA defines for the site schema and the space schema both match this **schema-password**.

Setting	Description
table-space	The tablespace in the Oracle database where the site schema segment will be created. Case-sensitive.
temp-table-space	The temporary tablespace in the Oracle database. Case-sensitive.
user-default-sort	Defines whether the standard Oracle binary sort (**NLS_SORT="BINARY_CI"**) should be overridden for non-Latin language support. Valid values: **yes**, **no**, or blank **Default**: blank (yes)

SQL Server settings

The following are the Microsoft SQL Server settings.

Setting	Description
app-user > name	MSSQL database login authentication user for OpenText Software Delivery Management. This is the user for day-to-day OpenText Software Delivery Management use. This login is associated with the OpenText Software Delivery Management site and space databases. **Note:** This should be different from the **admin-user > name**. However if you are using **FILL_EXISTING**, this must be the same as the **admin-user** name.
app-user > password	The password for the app-user. If you are using **FILL_EXISTING**, this must be the same as the **admin-user** password.
authentication-method	Enter the authentication method used: **Windows** or **DB** (SQL Server Authentication).

Site actions

The **SiteAction** setting determines how the installation should handle databases.

Property	Description
CREATE_ NEW	Use this site action for new installations. • Creates a new site schema, creates a new space schema, and configures the current node. • Only an **admin-user** with **create schema** permissions can create a new schema. • The **CREATE_NEW** site action fails when the schema already exists.
FILL_ EXISTING	Use this site action for new installations, in cases where the database administrator does not give permissions to create a schema (for Oracle) or a database (for SQL Server). In this case, the organization's DBA must create a new site and space schema/database and users **before** installation. See the following for details: • "Create your own Oracle schemas for OpenText Software Delivery Management" on page 27 • "Create your own SQL databases for OpenText Software Delivery Management" on page 29 **Handling schema exceptions** If the organization's DBA made changes to schemas, such as the addition of tables or columns, you can define an exception file. The exception file instructs OpenText Software Delivery Management to ignore manual changes to the database user schema during installation and upgrade. For details, see "Using exception files for manual database changes" on page 77.

Space settings

The following are the space settings.

Property	Description
initial-space-mode	The mode in which the initial space is created when the OpenText Software Delivery Management server starts. Valid values are: • **isolated**. Workspaces associated with the initial space do not share entities or customization settings. • **shared**. Workspaces associated with the initial space can share entities or customization settings.

Elasticsearch settings

A working Elasticsearch server is a requirement for working with OpenText Software Delivery Management. For details on Elasticsearch prerequisites, see "Configure Elasticsearch" on page 31.

Property	Description
hosts	The name of the host running Elasticsearch. If running an Elasticsearch cluster, all node host names should be separated by commas, as follows: **["host1","host2","host3"]**
http-port	Port configured in Elasticsearch for incoming HTTP requests. Default in Elasticsearch is 9200.
cluster-name	The name of the Elasticsearch cluster.

Elasticsearch security (optional)

You can connect with Elasticsearch securely using TLS. For details, see Setting up TLS for OpenText Software Delivery Management and Elasticsearch.

1. Make sure you have the following line in your **octane.conf** file:

   ```
   include "elasticsearch-security.conf"
   ```

2. Set up the **elasticsearch-security.conf** file as follows.

Property	Description
user	• **name**: The username to use when authenticating against Elasticsearch. • **password**: The password of the Elasticsearch user.
key-store	• **file**: The name of the PKCS12 keystore file. The file should be placed in the configuration folder. • **password** (optional, encrypted): The password to use to open the keystore file if the store is password protected. • **keystore type**: Certificate files should be in the PKCS12 format and should be put in the configuration folder.

Property	Description
trust-store	• **file**: The name of the PKCS12 truststore file. The file should be placed in the configuration folder. • **password** (optional, encrypted): The password to use to open the truststore file if the store is password protected. • **keystore type**: Certificate files should be in the PKCS12 format and should be put in the configuration folder.
verification-mode	Determine the level used when verifying the certificate. We recommend using the default setting. • **none**: No certificate verification checks are made. This means that any certificate can be accessed and should only be sued to debug issues. • **certificate**: Only checks that the certificate is signed by a trusted CA. Should be used when hosts are dynamic. • **full**: In addition to certificate, also checks that the host name reported by the certificate matches the host the request is coming from. Should be used whenever possible and is the default.

Site admin credentials

Use the following settings for the Site admin credentials.

Property	Description
site-administrator > name	The email of the site admin user that the installation creates. The email address can be specified now and created later. This is the only user available after installation. Other users can be added later. When using external user authentication, such as LDAP or SSO, this admin should be an existing user in the external system (LDAP or the IdP, respectively).
site-administrator > password	The site admin's password. The password must be at least 8 characters long, and contain at least one uppercase letter, one lowercase letter, and one number or symbol. Do not include a pound sign (**#**) or accented characters (such as, **ä**, **ç**, **ñ**). When using external user authentication, such as LDAP or SSO, this password should be defined as a "dummy" password. This password is not used when OpenText Software Delivery Management is configured for external authentication.

Cluster settings

Use the following settings to establish whether you are installing a standalone OpenText Software Delivery Management server or a cluster configuration. For details on cluster configurations, see "Cluster installation flow" on page 18.

Property	Description
single-server	Whether your server is standalone or in a cluster configuration. Mandatory. • For a standalone server, set this value to **true** and do not enter any host names using the **nodes** setting. • For a cluster configuration, set this value to **false**. You must enter node host names in the **nodes** setting.
nodes	Configure the IP addresses or fully qualified domain names for each cluster node. Enter a comma-separated list of node host names or IPs, in the cluster, for example: ["host1","host2","host3"] Make sure **single-server** is set to **false**.

Heap size

Use the following for changing heap size.

Property	Description
heap-size	Before starting the OpenText Software Delivery Management server the first time, change the heap memory values on all active cluster nodes. For example, you may need to increase the heap size if there is an increase in the number of active workspaces in OpenText Software Delivery Management, or an increase in the number of concurrent user sessions. Set **heap-size** to half of available server memory on a dedicated server, regardless of load. Heap size should not exceed 31 GB. Values should be specified in MB (for example, 4096 for 4 GB). Default: **4096**

Proxy settings (optional)

If OpenText Software Delivery Management is behind a firewall, and needs to access an outside server, you might need to use a proxy server.

To configure the proxy settings:

1. Make sure you have the following line in your **octane.conf** file:

   ```
   include "proxy.conf"
   ```

2. Set up the **proxy.conf** file as follows:

Property	Description
http	• **host**: The proxy host (if using HTTP). • **port**: The proxy port (if using HTTP).
https	• **host**: The proxy host (if using HTTPS). • **port**: The proxy port (if using HTTPS).
user	• **name**: User name accessing the proxy. • **password**: Password for proxy user.
non-proxy hosts	Any non-proxy hosts.

Public URL and Server Ports

In production systems, only secure configuration (HTTPS) is supported. In staging, we do not recommend using non-secure configuration. For details, see "Configuration tips" on page 62.

Enter the following in the **server-binding** section.

Property	Description
app-url	The fully-qualified domain name and port for the OpenText Software Delivery Management server. This is used for SSO configuration, reverse proxy configuration, SSL offloading configuration, and so on. This URL is also inserted as a link in emails that OpenText Software Delivery Management sends. Email recipients can click the link to access the relevant entity directly in OpenText Software Delivery Management. Use this pattern: `http://<Server URL>:[Port]` **Basic configuration:** Usually the URL of the server on which you installed the OpenText Software Delivery Management server. **Cluster configuration:** The Virtual IP URL. **Note:**If you have a URL with a top-level domain (TLD) that is not listed in the Internet Assigned Numbers Authority, for example, http://a.b.corp, where **corp** is not listed, see "Configure site settings" on page 37.

http-port
https-port
	The value of a Jetty port for HTTP, or a Jetty secure port for HTTPS.
	After you install OpenText Software Delivery Management, you might need to change the OpenText Software Delivery Management server port number.
	Because the installation uses a non-root user, common ports (below 1024) cannot be used with OpenText Software Delivery Management.
	By default, the installation uses port 8080 for HTTP or port 8443 for HTTPS (SSL).
	`httpPort: 8080`
	`httpsPort: 8443`
	Leaving any of these ports empty disables the access using the specified http schema server.
	It is possible that the default application server port is used by another application that is running on the same machine. In this case, you can either locate the application that is using the port and stop it, or you can change the OpenText Software Delivery Management server port.

allow-http-requests-if-ssl-enabled	By default, if you define your **app-url** as using HTTPS protocol, users cannot access OpenText Software Delivery Management via HTTP. If you need to enable HTTP access (for example for internal tools inside your network), you can set this parameter to **true**. This allows HTTP access to OpenText Software Delivery Management even though your protocol is set to HTTPS.
java-default-trust-store-password	By default, the Java trust store password is **changeit**. If you changed this password, enter the Java trust store password here. When OpenText Software Delivery Management starts, it encrypts this password. This is useful when OpenText Software Delivery Management server trust is configured.

Property	Description
force-disable-http2	By default, the HTTP/2 protocol is disabled, and this parameter is **true**. To use HTTP/2, change this parameter to **false**. In this case, you must configure HTTPS using the **key-store** fields. If you are using a load balancer or proxy server, make sure that they support HTTP/2.
file	Enter the absolute path to the keystore file, or the file name if the keystore is in OpenText Software Delivery Management's configuration folder. **Note:** Keystore fields are mandatory for HTTPS.
password	Password used to protect the keystore file. When OpenText Software Delivery Management starts, it encrypts this password. **Note:** Keystore fields are mandatory for HTTPS.
keystore type	Enter JKS or PKCS12. **Note:** • This field must be populated (default: **JKS**). • Keystore fields are mandatory for HTTPS.

Troubleshooting non-standard top-level-domains

OpenText Software Delivery Management validates that the top-level domain (TLD) entered in the **app-url** parameter is listed in the Internet Assigned Numbers Authority. If you enter a URL with a TLD that is not listed there (for example `http://a.b.corp`, where `corp` is not listed), server startup fails. In this case, perform the following steps:

1. Enter the default app-url: **https://localhost:8080**.
2. Start OpenText Software Delivery Management.
3. In the configuration parameters, define the parameter **ADDITIONAL_ALLOWED_TLD** with the value of your TLD (for example `corp`).
4. Restart OpenText Software Delivery Management.
5. In the configuration parameters, define the parameter **SERVER_BASE_URL** with the correct value of your server URL (for example `http://a.b.corp`).

License settings

Use the following settings for configuring licenses.

Property	Description
trial-edition	The trial edition is always **enterprise**. For details, see the information about OpenText Software Delivery Management editions in the OpenText Software Delivery Management Help Center.
license-mode	• If you are using a standalone OpenText Software Delivery Management license, enter **standalone**. You can then skip the remaining fields in the **License** section. Default. • If you are allocating licenses from OpenText Application Quality Management to OpenText Software Delivery Management, enter **ALM_SHARING**. You then need to fill in the following fields as described below. For details, see Manage licenses (on-premises) in the OpenText Software Delivery Management Help Center.
url	Enter the full path that you use to access OpenText Application Quality Management. Typically, this includes the suffix **qcbin**. Mandatory for **ALM_SHARING** mode.
integration-user > name	Enter the user name for accessing OpenText Application Quality Management. This user was defined in OpenText Application Quality Management for integration purposes. Mandatory for **ALM_SHARING** mode.
integration-user > password	Enter the password for the **integration-user**. This password is automatically encrypted after you restart the OpenText Software Delivery Management server. Mandatory for **ALM_SHARING** mode.

Authentication Type

Specify whether the installation should use native user management (default), LDAP, or SSO authentication for user management.

Property	Description
authentication-type	Values are: **internal**. Use internal, native OpenText Software Delivery Management user management. Default. **ldap**. Use LDAP authentication. Define LDAP settings as described in "LDAP authentication settings (optional)" on the next page. **sso**. Use SSO authentication. Define SSO settings as described in "SSO authentication settings (optional) " on page 57.

LDAP authentication settings (optional)

If you plan on authenticating users using LDAP, we recommend that you configure LDAP settings using the OpenText Software Delivery Management Settings UI after installation, rather than in the **ldap.conf** file. When you configure LDAP in the Settings UI, your settings are automatically validated and updated in the **ldap.conf** file. For details, see Configure LDAP in the OpenText Software Delivery Management Help Center.

If you prefer to work directly in the configuration files rather than in the Settings UI:

1. Make sure you have the following line in your **octane.conf** file:

   ```
   include "ldap.conf"
   ```

2. In the **ldap.conf** file, configure the LDAP settings as described below.
3. Later, after OpenText Software Delivery Management installation, import users from LDAP into OpenText Software Delivery Management.

> **Tip:** LDAP settings are validated when you start OpenText Software Delivery Management. If there are errors in your LDAP configuration which prevent the OpenText Software Delivery Management server from starting, have a site admin check the wrapper, site, and app logs.

General LDAP settings

Field	Description
connection-timeout	Connection timeout in seconds. Optional. Default: 30 seconds

Field	Description
admin-dn	The user that signs in to OpenText Software Delivery Management after initially setting up LDAP authentication. Its purpose is to make sure that one workable user exists to start configuring LDAP user authentication. When the OpenText Software Delivery Management server starts, it checks LDAP configuration settings, verifies that this user exists, and validates this user against the LDAP data. If this attribute is not defined correctly, the server does not start. Correct the user details and restart the server. This user can be same user as the user entered in the **octane.conf** file, or a different user. After entering the value for this user, and then restarting the OpenText Software Delivery Management server, the admin user entered in the **octane.conf** file is overwritten. This becomes the OpenText Software Delivery Management site admin user that can be used to log into OpenText Software Delivery Management the first time. **Note**: If the **admin-dn** is changed and the server is restarted, both the original **admin-dn** and the new **admin-dn** exist as site admins. Modifying the **admin-dn** does not remove the original one.

LDAP server settings

Enter the following settings for each LDAP server separately.

> **Caution:** Back up all passwords set below because they are encrypted after the OpenText Software Delivery Management server is initialized.

Field	Description
servers	Header row to delineate that the information below is for each LDAP server. Do not enter a value.
host	The LDAP server host name or IP address. Mandatory.
port	LDAP server connection port. Mandatory.
ssl	Whether the LDAP server uses SSL. Mandatory. Enter **Y** or **N**. If **Y**, establish trust to the certificate authority that issued the LDAP server certificate. For details, see "Configure trust on the server" on page 73.

Field	Description

base-directories	Root of the LDAP path to use to search for users when including new LDAP users in OpenText Software Delivery Management spaces. This can be a list of common names and domain components (cns and dns), a list of organizational units (ou), and so on.
	Optional. Default: Blank.
	Example:
	`"base-directories" : [`
	`      "dc=maxcrc,dc=com",`
	`      "ou=Administrative,dc=maxcrc,dc=com"`
	`      ],`

base-filters	Filters to use to refine the search for users when including new LDAP users in OpenText Software Delivery Management spaces. This is generally a semi-colon delimited list of LDAP **objectClasses**. Optional. Default: (objectClass=*)
description	Description of the LDAP server. Optional.
authentication:	Header row to delineate that the information below is for authentication. Do not enter a value.
method	The LDAP authentication method supported by the LDAP server. Authentication method used by the LDAP server. The following methods are supported: • **anonymous**. In this case, skip the next two parameters, **name** and **password**. • **simple**. **name** and **password** are mandatory.
user name	Only required if you set the **authentication** parameter to **simple**. User name for accessing the LDAP server. This user must have at least read permissions for the LDAP server.
password	Only required if you set the **authentication** parameter to **simple**. Password for accessing the LDAP server. This password will be encrypted.

LDAP server mapping settings

Enter the following mapping settings for each LDAP server separately.

Values used in the mapping section are case-sensitive.

OpenText Software Delivery Management attribute in ldap.conf	Sample LDAP attribute that can be used	Values and descriptions
mapping	**mapping**	Header row to delineate that the information below is for mapping of LDAP attributes. Do not enter a value.
dn	• **distinguishedName** (for Active Directory) • **entryDN** (for other LDAP systems)	The LDAP distinguished name attribute. Unique. Mandatory. This attribute is typically in a format that contains the common name and organization details, such as: **cn=<common_name>,ou=<organizational_unit>,dc=<part_of_domain>** The **dn** is a unique string that typically contains other LDAP attributes, such as **cn**, **ou**, and **dc**. **Example** 1. If in LDAP, the **entryDN** attribute value is: **cn=<common_name>,ou=<organizational_unit>,dc=<part_of_domain>** 2. In the **ldap.conf**, the dn value would be mapped to: **entryDN** 3. When exporting users from LDAP, the **dn** string representation of each LDAP user would be the common name, followed by the organizational unit, followed by a part of the domain, such as: **cn=Joe_Smith@nga,ou=my_org,dc=com**

OpenText Software Delivery Management attribute in ldap.conf	Sample LDAP attribute that can be used	Values and descriptions
uid	• **objectGUID** (for Active Directory) • **entryUUID** (for other LDAP systems)	The LDAP attribute that should be used as the immutable, globally-unique identifier. Mandatory. In this documentation, we also refer to this as the UUID (universally unique ID). • For Active Directory, we use **objectGUID**. • For other LDAP systems, we generally use **entryUUID** for OpenLDAP. However, depending on your LDAP, this attribute might be different, such as **GUID** or **orclguid**. This is an attribute by which OpenText Software Delivery Management identifies each user internally for synchronization between OpenText Software Delivery Management and LDAP, including when importing users into OpenText Software Delivery Management. You can configure other values, such as GUID or orclguid, or any other unique value.
first-name	**givenName**	LDAP attribute for first name, such as **givenName**. Mandatory.
last-name	**sn**	LDAP attribute for last name, such as **sn**. Mandatory.
full-name	**cn**	LDAP attribute for full name, such as **cn**. Optional.
logon-name	**mail**	This is the unique identifier between all OpenText Software Delivery Management users, and this attribute is used for logging on. In some cases, this attribute is used to identify each user internally for synchronization between OpenText Software Delivery Management and LDAP, including when importing users into OpenText Software Delivery Management. **mail** is usually unique for each user, so **mail** is an appropriate LDAP attribute to use to map to **logon-name**. Mandatory. You can change the **logon-name** attribute mapping at any time, but make sure the **logon-name** is unique across all OpenText Software Delivery Management users.
email	**mail**	The LDAP attribute for email address, such as **mail**. Mandatory.

OpenText Software Delivery Management attribute in ldap.conf	Sample LDAP attribute that can be used	Values and descriptions
phone1	**telephoneNumber**	The LDAP attribute for the primary phone number, such as **telephoneNumber**. Optional.

SSO authentication settings (optional)

Use these settings to set up SSO authentication for connecting with an external IDP.

To configure the SSO authentication settings:

1. Make sure you have the following line in your **octane.conf** file:

   ```
   include "sso.conf"
   ```

2. Set up the **sso.conf** file as follows:

key-pair settings

The following describe the key-pair settings.

Setting	Description and usage
alias	Unique identifier for the SSO public/private key pair used by the OpenText Software Delivery Management service provider for signing and encrypting authentication information. Required. Example: **sso-osp-keypair**
password	Password for protecting and encrypting the key pair defined with **key-pair alias**. When OpenText Software Delivery Management starts, it encrypts this password. Required. Example: **my-secret**

key-store settings

The following describe the key-store settings.

Setting	Description and usage
file	The absolute path to the keystore file identified with **key-pair alias**. The path should be under OpenText Software Delivery Management's configuration folder to avoid permission issues. Required.

Setting	Description and usage
password	Password used to protect the keystore file defined with **keystore file**. When OpenText Software Delivery Management starts, it encrypts this password. Required. Example: **my-password** **Note:** If you are using pkcs12, you must use the same password for both the keystore and the key(s). This is a Java limitation.
keystore-type	This defines the keystore type. The default format for this file is **PKCS12**. You can change the format to Java KeyStore (JKS) by specifying this type here.

oauth settings

The following describe the oauth settings.

Setting	Description and usage
client-id	Client ID used for internal OAuth2 configuration, and by which the integration that accesses OpenText Software Delivery Management identifies itself. Regular expressions are not supported (meaning, no asterisk wildcards). Must be the same on all OpenText Software Delivery Management cluster nodes. Required. Example: **my-client-ID**
client-secret	The OAuth client secret for the integration's client ID defined with **oauth client-id**. Can be any value. We recommend that the secret be complex and hard to guess. Must be the same on all OpenText Software Delivery Management cluster nodes. When OpenText Software Delivery Management starts, it encrypts this password. Required. Example: **secret**

Setting	Description and usage
authentication-timeout	The SSO authentication timeout in seconds. Optional. Default: **10800** seconds (3 hours). **Other timeout settings when working with SSO** The following configuration parameters can be used to set other timeouts when working with SSO. These parameters are defined in the Settings area in OpenText Software Delivery Management, not in the **sso.conf** file. They do not have any effect on the SSO authentication timeout. • **MINUTES_UNTIL_IDLE_SESSION_TIMEOUT**. Defines license consumption in minutes. • **MINUTES_UNTIL_GLOBAL_SESSION_TIMEOUT**. Defines API key authorization timeout in minutes. For details on setting these configuration parameters, see Configuration parameters in the OpenText Software Delivery Management Help Center.

saml settings

The following describe the saml settings.

Section	Setting	Description and usage
set-request-subject	n/a	Defines if the SAML subject should be set when available in the authentication request. Some IDPs do not accept the Subject property in the authentication request, therefore the default value is false. **Default:** false
idp	**metadata-url**	The IdP's URI for publishing IdP metadata. Part of the pairing process. If this is set, there is no need to set metadata. Using this option, the URL must be available and respond with a valid XML, otherwise OpenText Software Delivery Management will not start. Any valid URL is accepted. You can define the SAML metadata descriptor resource with either this setting, or the **saml idp metadata** setting. Mandatory, if **saml idp metadata** is not defined. Example: **http://my-server.company-infra.net:8080/auth/realms/Dev/protocol/saml/descriptor** **Note:** Only one of the parameters **metadata** or **metadata-url** should be defined. If the sso_configuration validator is disabled and both parameters are defined, the URL defined in **saml idp metadata-url** will be ignored.

Section	Setting	Description and usage
idp	**metadata**	Base 64 encoded XML of the SAML metadata descriptor from the IdP. This should be used if the IdP metadata URL cannot be accessed from the OpenText Software Delivery Management server. You can define the SAML metadata descriptor resource with either this setting, or the **saml idp metadata-url** setting. Mandatory, if **saml idp metadata-url** is not defined. **Note:** Only one of the parameters **metadata** or **metadata-url** should be defined. If the sso_configuration validator is disabled and both parameters are defined, the URL defined in **saml idp metadata-url** will be ignored.
mapping	**user-name**	The parameter in the SAML response which maps to the user name. Valid values are: • **'{$id}'**. Mapping is to the **NameID** in the SAML response's subject. Default. • **userName**. Mapping is to the **username** in the SAML attribute statement. Changing the default to a property name, such as **userName**, in the SAML response, does not require quotes.
mapping	**uuid**	The attribute in the SAML response's attribute statement that maps to the user's UUID. Optional. Default: **uuid**
mapping	**mail**	The attribute in the SAML response's attribute statement that maps to the user's email address. Optional. Default: **mail**
mapping	**first-name**	The attribute in the SAML response's attribute statement that maps to the user's first name. Optional. Default: **firstName**
mapping	**last-name**	The attribute in the SAML response's attribute statement that maps to the user's last name. Optional. Default: **lastName**
mapping	**full-name**	The attribute in the SAML response's attribute statement that maps to the user's full name. Optional. Default: **fullName**

Token-exchange settings

The following describe the Token-exchange settings.

Setting	Description and usage
token-exchange-enabled	Activates the federated identity option for authenticating APIs within an organization's SSO system. Required. Default: **false**
issuer	Used to define the <baseUrl> in any OpenID Connect (OIDC) endpoint when authorizing against the external OAuth 2.0 authorization server. Required. Use the following endpoints to review the metadata and find the issuer: **https://<OAuth2 Authorization Server>/.wellknown/openid-configuration**
treat-access-token-as-opaque	If **true**, any access token returned from the OIDC provider is treated as an opaque token even if it appears to be a JWT token. Set to **true** only if the provider returns an access token that appears to be a JWT, but which is invalid. Required. Default: **false**
max-clock-skew	The maximum time difference between the OpenText Software Delivery Management system and the OIDC provider system in milliseconds. The value can be suffixed with "s", "m", "h", or "d" to indicate that the value is seconds, minutes, hours, or days. **Note:**If systems are time-synchronized using NTP, there is no need to set maximum skew time to more than a couple of seconds. Required. Default: **1s**
oidc	The oidc section contains the following settings. - **client-id** and **client-secret**. The OIDC client ID and secret to use in your organization's tool for the token exchange. Required. The OIDC client ID and secret should be placed in the Authorization header as Basic: `Authorization: Basic Base64(clientId:clientSecret)` **Note:** - The OIDC client ID is not the same client ID that is used by the tool for authentication. - OIDC client ID and secret differ from: ◦ The API key used for authentication in the authorization server. ◦ The client ID and secret defined in the **sso.oauth** section because of the different usage scenarios. Client ID and secret from **sso.oauth** are used by OpenText Software Delivery Management during the SSO authentication flow, while client ID and secret from the **token-exchange.oidc** section are used by the tool that performs the token exchange. - **authentication-timeout**. The federated SSO authentication timeout in seconds. Required. Default: **10800** seconds (3 hours).

Setting	Description and usage
introspect	The introspect section contains the following settings. • **enabled**. Set to **true** to use the introspect method for OAuth 2.0 authentication for API keys. Default: **false** • **auth-server-client-id**. Client ID returned by the authorization server which is used by OpenText Software Delivery Management to access the token introspection endpoint. Required. • **auth-server-client-secret**. The OAuth client secret returned by the authorization server which is used by OpenText Software Delivery Management to access the token introspection endpoint. Required.
mapping	The mapping section contains the following settings. Only the standard OIDC claims are supported. • **user-name**. Defines the claim in the access token from the authorization server that holds the name of the authenticated API key. This is used for mapping the authenticated API key with its role in OpenText Software Delivery Management. Required. • **session-identifier**. Defines the claim in the access token from the authorization server that holds a unique authentication identifier (for example "txn", Transaction Identifier). Required. Default: **jti**

Logging settings

The following describe the logging settings.

Setting	Description and usage
directory	The directory in which to create the SSO log files. Optional. If the value is empty then the default logging directory will be used. Default: **<log folder>/sso**
logging-level	Logging level. Optional. Possible values are: • **SEVERE** • **INFO** • **WARNING** • **ALL** Default: **WARNING**

Configuration tips

The following are tips for configuring site settings:

- In production systems, only secure configuration (HTTPS) is supported. In staging, we do not recommend using non-secure configuration as the industry standard is to always use secure communication. Non-secure configuration results in poorer client performance, which does not fully represent what will happen in the production environment.
- OpenText Software Delivery Management uses the TLS version 1.2 secure protocol. To configure a secure connection using TLS (SSL), obtain the server certificate issued to the name of this server in java keystore format (.jks) issued to the fully qualified domain name of server. It must contain a private key and the certificate authority that issued it. For details on creating certificates using the Certificate Authority, see OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

  You then enter certificate details in the section "Public URL and Server Ports" on page 48.
- When installing a single node configuration for the Jetty server, use the full address to access it.

  Example: If the Jetty server is installed on a machine named `myserver.mydomain.com`, access it using: `http[s]://myserver.mydomain.com:<port>` and not `http[s]://myserver:<port>` if there are client-side DNS shortcuts installed.
- When you install a cluster Jetty server environment, the load balancer and all Jetty nodes should all be accessible from one another. The same rules for accessing the server using the load balancer from the client side apply. This means that the full address of the load balancer should be used for access.

Start the server

When you finish defining your configuration settings as described in "Configure site settings" on page 37, start OpenText Software Delivery Management.

To start the server:

1. Select **Start > OpenText Software Delivery Management > Start OpenText Software Delivery Management Server**.

   Alternatively, start the **OpenText Software Delivery Management** service.

   The installation is complete when the "Server is ready!" message is shown in the **C:\Program Files\octane\log\wrapper.log** file. If you do not see the "Server is ready!" message, correct the errors shown in the log.

2. You are now ready to:

   - **Single-node configuration**: Log in and create additional users. For details, see "Log in" on the next page.

     Check connectivity by logging in, after initializing the first node and before installing the remaining cluster nodes.

   - **Cluster configuration**: Optional.

     For details on installing on a cluster, see "Cluster installation flow" on page 18.

Next steps:

- "Log in" on the next page
- "Cluster installation flow" on page 18

Log in

This section describes how to log into OpenText Software Delivery Management.

> **Tip:** When you first start using OpenText Software Delivery Management, you automatically receive a Trial license which gives you a 90-day trial for 100 users. For details, see Trial license in the OpenText Software Delivery Management Help Center.

To log in:

1. In a browser, go to **<serverURL>:<serverport>/ui**.

   Make sure to specify a fully-qualified domain name for the server. The name must include at least one period. Do not specify an IP address.

   Cluster configuration: Use the load balancer URL.

2. Log in with the site admin user name and password you provided in the **octane.conf** file using settings **site-administrator name** and **password**.

> **Note:** Errors might be listed even if the OpenText Software Delivery Management server initializes and starts. If you encounter problems initializing, check for errors in the log files.

Next steps:

- **Cluster configuration**: If you successfully installed and logged into OpenText Software Delivery Management on the first cluster node, continue installing on additional cluster nodes. See "Cluster installation flow" on page 18.
- Set configuration parameters, such as FORGET_USER_ON_DELETE and SMTP_NOTIFICATION_SENDER_EMAIL. For details, see Configuration parameters in the OpenText Software Delivery Management Help Center.
- Create spaces. For details, see Create a space in the OpenText Software Delivery Management Help Center.

- After you have logged on as the space admin, you can create other users and workspaces. For details, see Users and Create workspaces in the OpenText Software Delivery Management Help Center.

Install using a Docker image

This section describes how to install OpenText Software Delivery Management using a Docker image.

> **Note:** The OpenText Software Delivery Management Docker container is based on a Linux image. This section relates to running this container on a Windows host.

1. Download and install the latest version of Docker Desktop.
2. In the Docker Hub, search for **Octane**.
3. Select **lifecyclemanagement/octane**. The description should say: The official repository for OpenText Software Delivery Management.
4. Select **Tags**.
5. Choose the version you want to install, and copy the download command.

   > **Note:** The list you see includes both SaaS versions and on-premises versions of OpenText Software Delivery Management, but only on-premises versions are supported. Select an on-premises version now.

6. In a command line, run the command you copied.
7. In Docker Desktop, select **Images**.
8. Locate the OpenText Software Delivery Management version you want to install. To configure the container, click **Run**.
9. Open the **Optional Settings**, and enter the following:
   - In **Container name**, enter a name of your choice.
   - In **Ports**, enter 8080 to use HTTP, or 8443 to use HTTPS.
   - In **Volumes**, enter the following:

Host path	Container path
C:\OctaneDocker\conf	opt/octane/conf
C:\OctaneDocker\log	opt/octane/log

Host path	Container path
C:\OctaneDocker\repo	opt/octane/repo

10. Click **Run** to run the Docker image for the first time.

    The run fails with errors, because OpenText Software Delivery Management has not yet been configured.

11. Open the **octane.conf** file located in **C:\OctaneDocker\repo\conf-discover**.
12. Configure OpenText Software Delivery Management as described in "Configure site settings" on page 37.

    > **Note:** If you want to use resources from your local machine instead of localhost, use **host.docker.internal**.

13. When you are done, run the container from **Containers/Apps**.
14. Open the **C:\OctaneDocker\log** folder and check for errors in the **wrapper.log** and **octane.log** files.
15. OpenText Software Delivery Management is now ready for use.

Management

Here are some management tasks you may have to perform during or after installation.

> **Note:** In addition to these management tasks, you can also set configuration parameters to define how your site operates. Configuration parameters for the site are set using Settings. For details, see Configuration parameters in the OpenText Software Delivery Management Help Center.

Start the server manually

If you need to start the OpenText Software Delivery Management server manually, perform the following.

To start (or restart) the server:

Select **Start > OpenText Software Delivery Management > Start OpenText Software Delivery Management Server**

The service runs in the background.

To start (or restart) in a cluster configuration:

All nodes must be restarted.

See also:

- "Management" above

Handle database-related issues

This topic provides instructions for handling database-related management tasks.

This section includes:

- "Change site schema settings and reinitialize" below
- "Update database password in the site schema and configuration files" on the next page

Change site schema settings and reinitialize

If you need to make changes to the site schema settings, make the changes in the **octane.conf** file.

To change site schema settings and reinitialize:

1. Obtain the names of the indexes related to your instance of OpenText Software Delivery Management in the **sharedspace_logical_name.txt** in the **C:\Program Files\octane\server\conf\** folder.
2. Delete the database site schema.
3. Delete the repository.
4. Delete the **mqm_<sp_logical_name>** indexes from Elasticsearch. From the command prompt on the OpenText Software Delivery Management server, run:

   ```
   curl -XDELETE 'http://<server address>:9200/mqm_<sp_logical_
   name>/'
   ```

5. Select **Start > OpenText Software Delivery Management > Start OpenText Software Delivery Management Server**.

Update database password in the site schema and configuration files

If you change your database password, you can use the database password update tool to update the database password in Software Delivery Management's site schema, and in the octane.conf configuration files. Note that this does not update the database user's password, but only Software Delivery Management's configuration.

> **Note:** The tool operates offline. Credential outputs are disabled for security.

1. Stop the OpenText Software Delivery Management server.

   After stopping the server, wait 30 seconds before running the tool. The cluster is considered offline when there is no activity from any node for 30 seconds.

2. Run the following command as an administrator on your OpenText Software Delivery Management server:

   ```
   \opt\octane\install\updatedbcreds.bat
   ```

3. Enter values as described in the sections below.
4. When you are done, start the Software Delivery Management server.

Usage

The tool can operate in file mode or interactive mode.

```
updatedbcreds.bat <-m mode> <-f path | -t target>
```

Where	Equals
`mode`	{file | interactive} • **File.** If mode is set to file, use `-f` to specify the path to the password definition file. Credentials are taken from the provided file. • **Interactive.** If mode is set to interactive, use `-t` to specify the target whose password you want to change - either admin or user. You then enter credentials interactively.
`target`	{admin | user}
`path`	valid absolute or relative path to file

Example: If you want to update the db.admin-user in the config file, the target should be **admin** (in Interactive mode).

If you want to update the db.<db-vendor>.app-user-name in the config file, the target should be **user** (in Interactive mode).

File mode

You can use the CLI in file mode, which allows granular definitions for admin, user, or space passwords.

Using a tool in file mode looks like this:

```
updatedbcreds.bat -m file -f /path/to/definition.json
```

This is done using a JSON password definition file, in the following format:

~~~
```json
{
  "admin" : {
    "password" : "PasswordForAdminUser"
  },
  "appUser" : {
    "password" : "PasswordForAppUser"
  },
  "spaces": {
    "default_shared_space": {
      "password": "PasswordForSpecificSpace"
    }
  }
}
~~~

In SQL Server, you can delete the **spaces** section. In this case all spaces get the appUser password.

Caution: Before the tool runs, your file contains passwords in clear text. It is your responsibility as administrator to secure the file according to your organization's policies. The tool encrypts the file when running. The tool can read the encrypted password if you want to rerun the tool.

> For improved security, use interactive mode.

Interactive mode

In interactive mode, you update only the admin or user password. This is useful when you do not need extensive password definition and just want to change a password for a single user.

Using a tool in interactive mode looks like this:

```
updatedbcreds.bat -m interactive -t admin
```

Enter the following:

- New password for ADMIN: Enter a new password for admin user. Output is disabled.
- DB authentication username: Enter a user for CLI database connection. Output is disabled.
- DB authentication password: Enter a password for CLI database connection. Output is disabled.

See also:

- "Management" on page 69

Configure trust on the server

Configure trust on the OpenText Software Delivery Management server when you connect to any remote server (such as a database server, an LDAP server, or license sharing with OpenText Application Quality Management) over a secure channel.

> **Note:** When connecting to a database server with SSL, or an LDAP server, over a secure channel, you must configure trust before starting the OpenText Software Delivery Management server.

To configure trust:

1. Obtain the certificate of the root and any intermediate Certificate Authority that issued the remote server certificate.
2. Import each certificate into the OpenText Software Delivery Management java truststore using a keytool command.
   - Locate your **<java_home>** folder. One way to check the location of the **<*java_home*>** folder is to check the environment information settings in the **C:\Program Files\octane\log\wrapper.log** file.

     Example: **C:\Program Files\java\<*jdkversion*>\jre**
   - Locate your keystore **cacerts** file, which is usually here: **<*java_home*>\jre\lib\security\cacerts**
   - Import each certificate.

     Example:

     ```
     cd <java_home>\bin

     .\keytool -import -trustcacerts -alias <CA> -file <path to the
     CA certificate file> -keystore ..\lib\security\cacerts
     ```
3. If the OpenText Software Delivery Management service is running, restart it.

> **Tip:** For general details on configuring HTTPS, see "Secure configuration and deployment" in the OpenText Software Delivery Management Secure Deployment and Configuration Guidelines.

Next steps:

- "Management" on page 69

Advanced server configuration

This section describes advanced configuration tasks for the OpenText Software Delivery Management server.

Configure secure database access

This section describes how to configure a secure connection from the OpenText Software Delivery Management server to the database server. The secure connection is protected with SSL/TLS for encryption and authentication.

This section includes:

- "Defining the connection-string for secure database access" below
- "To configure a secure database connection for a previously-unsecured database " on the next page
- "To configure a secure database connection for a new installation" on page 77

Defining the connection-string for secure database access

SQL Server

Use the following to define the connection-string for secure SQL Server access.

SQL Server Scenario	Instructions
SSL/TLS is required	Add the encryption method to the end of the **ConnectionString** value. **jdbc:sqlserver://<server>:<port>;encrypt=true;trustServerCertificate=true**
SSL without certificate validation	When using SSL, disable validation of the certificate sent by the database server. Add the encryption method to the end of the **ConnectionString** value, and apply the certificate into the java certs file located under **<JAVA_HOME>\jre\lib\security\certs**. **jdbc:sqlserver://<server>:<port>;encrypt=true;trustServerCertificate=false;trustStore=<Java Certs file>;trustStorePassword=<JKS password>**

Oracle

Use the following to define the connection-string for secure Oracle database access.

Oracle scenario	Instructions
SSL/TLS required	To configure a secure connection from the OpenText Software Delivery Management server to the database server using SSL or SSO, refer to the section "Using SSL/SSO in Oracle (optional)" on page 41. The connection string should include the port defined in the Oracle database as the port for SSL connections. The protocol should be set to TCPS: `connection-string = "jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=<hostname>)(PORT=<ssl port>)) (CONNECT_DATA=(SERVICE_NAME=<ORA_SERVICENAME>)))"`

To configure a secure database connection for a previously-unsecured database

This step provides instructions for configuring the site schema connection.

Skip this section if you have a separate database server for your workspaces and you only want a secure connection to that database.

This section is relevant if the database server that was configured for a secure connection contains your site schema.

1. Edit the **octane.conf** file. The default location is **/opt/octane**):
   a. Set the value of **site-action** to **CONNECT_TO_EXISTING**:

      ```
      site-action=CONNECT_TO_EXISTING
      ```

   b. Edit the line with **connection-string**.
2. If SSL/TLS is required, make sure the trust on the OpenText Software Delivery Management server has been established. For details, see "Configure trust on the server" on page 73.
3. Run the service to start the OpenText Software Delivery Management server.

   ```
   systemctl start octane
   ```

To configure a secure database connection for a new installation

1. After installing OpenText Software Delivery Management, start the server:

   ```
   systemctl start octane
   ```

2. In the Database Server step, select the **connection-string** option and set the values for your database.
3. Make sure the trust on OpenText Software Delivery Management the OpenText Software Delivery Management server has been established. For details, see "Configure trust on the server" on page 73.

See also:

- "Management" on page 69

Using exception files for manual database changes

This topic provides instructions for defining exception files. Use exception files if the organization's DBA added objects to database schemas, such as tables, indexes, stored procedures, columns, or other objects.

This section includes:

- "Overview" below
- "Define exception files" on the next page
- "Set up use of the exception file" on page 80

Overview

Exception files instruct OpenText Software Delivery Management to ignore any errors issued because of manual additions to the database schema. These errors would typically stop the installation or upgrade process.

You can use exception files to ignore errors for extra tables, views, columns, and sequences. For any other problem, consult with your database administrator.

> **Caution:** Using the exception file to ignore errors for objects that are added manually to the schema may compromise stability and the validity of the database user schema.

You can use the exception files during a new installation, when upgrading, and when creating a space.

Define exception files

Define exception files before installation, before upgrading, or before you create the new spaces.

To define exception files:

1. Copy both of the **mqm_exception.xml** files from the installation directories. You can rename them.
2. Locate the MQM_EXCEPTIONS part of the file.

   ```
   <MQM_EXCEPTIONS>
       <exceptions>
           <declaration>
               <!--<object pattern="TABLE_1_EXAMPLE" type="missing"
   />-->
               <!--<object pattern=" TABLE_1_EXAMPLE" type="extra"
   />-->
           </declaration>
       </exceptions>
   </MQM_EXCEPTIONS>
   ```

3. Change the <declaration> to one of the following. Add as many declarations as you need.
   - TableMissing
   - ViewMissing
   - ColumnMissing

- ConstraintMissing
- IndexMissing
- PartitionFunctionMissing
- PartitionSchemeMissing
- ProcedureMissing
- SequenceMissing
- TriggerMissing

4. For each object pattern, you can specify one of the following types.

Object	Description
missing	The object is needed but is missing.
extra	The object is extra because it was created after installation or before upgrading.

Examples

- For an extra table:

```
<TableMissing>
         <object pattern="MY_Table" type="extra"/>
</TableMissing>
```

- For an extra view:

```
<ViewMissing>
         <object pattern="MY_VIEW" type="extra"/>
</ViewMissing>
```

- For an extra column:

```
<ColumnMissing>
         <object pattern="MY_COLUMN" type="extra"/>
</ColumnMissing>
```

- For an extra sequence:

```
<SequenceMissing>
         <object pattern="MY_SEQUENCE" type="extra"/>
</SequenceMissing>
```

Set up use of the exception file

This topic explains how to use the exception file when installing OpenText Software Delivery Management, when upgrading, or when creating a new space.

Use of the exception files during first-time installation

You can use exception files when installing OpenText Software Delivery Management using existing schemas/databases instead of having OpenText Software Delivery Management create new schemas for you. This is the FILL_EXISTING installation option and it is set in the **octane.conf** file.

1. During installation, when configuring the **octane.conf** file in the configuration folder, add these two settings using an editor.

Setting	Description
MqmExceptionsSiteAdminPath	The exception file for the site. **C:/temp/site_admin/mqm_exception.xml**
MqmExceptionsSharedSpacePath	The exception file for the default space. **C:/temp/shared_space/mqm_exception.xml**

2. Continue installing.
3. Check that the server is up and that you have proper access to the site and the default space.

Use of the exception files when upgrading

You can use exception files when upgrading OpenText Software Delivery Management.

After installation, the exception files are copied to the repository folder. So when upgrading, modify the copies of the exception files in the repository folder instead of the files in the configuration folder.

1. During the upgrade, when configuring the **octane.conf** file in the repository folder, add or modify these two settings using an editor.

Exception file	Path
File for the site	**C:\Program Files\octane\repo\storage\schema\maintenance\exceptions\site_admin\mqm_exception.xml**
File for the new space	**C:\Program Files\octane\repo\storage\schema\maintenance\exceptions\shared_space\mqm_exception.xml**

2. Continue upgrading.
3. Check that the server is up and that you have proper access to the site and the default space.

Use of the exception files when creating a space

The exception files are also processed when adding new spaces.

After installation, the exception files are copied to the repository folder.

Before adding a new space, modify the copies of the exception files in the repository folder instead of the files in the configuration folder.

1. Add exceptions as necessary to the exception files using an editor.

Exception file	Path
File for the site	**C:\Program Files\octane\repo\storage\schema\maintenance\exceptions\site_admin\mqm_exception.xml**
File for the new space	**C:\Program Files\octane\repo\storage\schema\maintenance\exceptions\shared_space\mqm_exception.xml**

2. In OpenText Software Delivery Management **Settings** ⚙ area, add the space using an existing schema. For details, see Create a space in the OpenText Software Delivery Management Help Center.
3. Check that you have proper access to the space.

See also:

- "Configure site settings" on page 37
- "Management" on page 69

Uninstall

To uninstall the OpenText Software Delivery Management server, use the uninstall feature from the Windows Control Panel.

The uninstall process does not delete the repository, log, and configuration directories, in case you want to reinstall. Delete them if necessary.

See also:

- "Installation" on page 34