opentext™

OpenText™ Documentum™ Content Management

Cloud Upgrade and Migration Guide

Upgrade OpenText Documentum Content Management (CM) on certified cloud platforms. Migrate OpenText Documentum Content Management (CM) from the on-premises environment to certified cloud platforms.

EDCSYCD250400-AUM-EN-02

OpenText™ Documentum™ Content Management
Cloud Upgrade and Migration Guide

EDCSYCD250400-AUM-EN-02
Rev.: 2026-Feb-20

This documentation has been created for OpenText™ Documentum™ Content Management CE 25.4.
It is also valid for subsequent software releases unless OpenText has made newer documentation available with the product, on an OpenText website, or by any other means.

Open Text Corporation

275 Frank Tompa Drive, Waterloo, Ontario, Canada, N2L 0A1

Tel: +1-519-888-7111
Toll Free Canada/USA: 1-800-499-6544 International: +800-4996-5440
Fax: +1-519-888-0677
Support: https://support.opentext.com
For more information, visit https://www.opentext.com

Table of Contents

Chapter 1

Overview

This guide is intended for system, repository, and cloud administrators, application programmers, and any other user who wishes to obtain an understanding of upgrading OpenText Documentum CM on different cloud platforms. In addition, this guide provides information about migrating OpenText Documentum CM from the on-premises environment to cloud platforms.

Users who read this guide must have working knowledge of OpenText Documentum CM components and orchestration in clusters such as Kubernetes using container images and Helm, and have an understanding of databases and services of different cloud platforms.

OpenText encourages you to choose the cloud over an on-premises solution.

OpenText Documentum CM offers a single All-In-One Helm package for the supported containers. This approach defines a common layout within the OpenText Documentum CM solutions and includes all containers with the ability to upgrade and migrate using a single Helm package.

Chapter 2

Upgrading OpenText Documentum CM on cloud platforms

This documentation provides the information about upgrading OpenText Documentum CM to 25.4.

2.1 Prerequisites

This section provides the information about the prerequisites tasks.

To download the software binaries to upgrade OpenText Documentum CM using manual steps:

- Download the 25.4 version of the Helm charts from OpenText Container Registry and extract the contents to a temporary location.

For more information, see "Upgrading OpenText Documentum CM using manual steps" on page 8.

To download the software binaries to upgrade OpenText Documentum CM using Documentum™ Cloud Assist:

1. Download the `Documentum CM - Utilities <version>.zip` and `Documentum CM - Utilities <version>.tar.gz` files from My Support and extract the contents to a temporary location.
2. From the temporary location, extract the contents of the `Documentum_Cloud_Accelerator_windows-<version>.zip` and `Documentum_Cloud_Accelerator_linux-<version>.tar.gz` files from the temporary location.
3. Extract the contents of the `Documentum-Cloud-Assist-25.4.zip` and `Documentum-Cloud-Assist-25.4.tar.gz` files.

For more information, see "Upgrading OpenText Documentum CM using Documentum™ Cloud Assist" on page 39.

2.2 Upgrading OpenText Documentum CM using manual steps

This section provides information to upgrade OpenText Documentum CM using the manual steps.

Note: Before upgrading OpenText Documentum CM from 24.4 or 25.2 to 25.4, ensure that you run the following command to delete the Documentum xPlore statefulsets:

```
kubectl delete statefulset <Xplore_Statefulsets> -n <namespace>
```

2.2.1 Upgrading Documentum CM Server

This section provides information about upgrading Documentum CM Server from 24.4 or 25.2 to 25.4.

2.2.1.1 Prerequisites

This section provides information about the important tasks that you must perform and notes that you must consider before upgrading Documentum CM Server to 25.4.

- Using the Helm charts of previous deployment

  Before upgrading the Documentum CM Server pod, make sure that you have the Helm charts used for the previous deployment ready for your reference. The configuration values defined in the Helm charts used for the previous deployment need to be used in the upgrade process.

  Caution

  - Retain the previous deployment password information as is for `cs-secrets.docbase.password`, `cs-secrets.contentserver.globalRegistry.password`, `cs-secrets.contentserver.aek.passphrase`, and `cs-secrets.contentserver.install.appserver.admin.password` in the `documentum/config/passwords.yaml` file.
  - Retain the Graylog and Fluentd variables during the upgrade process.
  - If a mismatch exist in the configuration values, it impacts the deployment of 25.4 Helm chart and the upgrade process may fail.

- Using the correct release name

  Ensure that you use the same release name used in the previous deployment of Documentum CM Server for upgrading the previous deployment of Documentum CM Server to 25.4.

- Using the previous release ingress annotations

If you have modified any default ingress annotations in the previous release, ensure that you copy all those ingress annotations from the previous release to the `documentum/platforms/<cloud platform>.yaml` file of 25.4.

- Updating new migrated database host details

  If you have migrated the database to a new database host, then provide the new database login credentials for `databaseAdminPassword` in the `documentum/config/<passwords or passwords_vault or passwords_k8api>.yaml` file.

  In addition, provide the new host value of the migrated database for `global.db_hostname` and the new port value for `global.dbport` in the `documentum/values.yaml` file.

- Modifying service account name

  1. Upgrade the Documentum CM Server 24.4 or 25.2 pod to 25.4 using the steps described in "Upgrading from 24.4 or 25.2 to 25.4" on page 9 with the same service account name that was used in the previous deployment.
  2. Set the value of `docbroker.serviceAccount.createserviceaccount` to `true` in the `documentum/config/configuration.yml` file.
  3. Provide a new service account name for `documentumserviceaccount` in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.
  4. Run the Helm upgrade command.

2.2.1.2 Upgrading from 24.4 or 25.2 to 25.4

This section provides information about upgrading Documentum CM Server from 24.4 or 25.2 to 25.4.

To upgrade from non-HashiCorp Vault 24.4 or non-HashiCorp Vault 25.2 to non-HashiCorp Vault 25.4:

1. Update the new image location details of `db`, `docbroker`, and `content-server` in the `documentum/dockerimages-values.yaml` file of 25.4.
2. Copy all the required configuration values from the `global variables` section of the previous deployment to the `global variables` section in the `documentum/dockerimages-values.yaml`, `documentum/config/passwords.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.
3. Copy all the required configuration values from the `db` section of the previous deployment to the `db` section in the `documentum/config/passwords.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

Important

Ensure that the database version is same for both the existing and new deployment.

For example, if your 25.2 deployment is running with PostgreSQL 14.4 database container, then in your 25.4 upgrade, ensure that you change the database version to 14.4 as follows:

- Value of `version` in the `documentum/charts/db/Chart.yaml` file.
- Value of `db.images.db.tag` in the `documentum/dockerimages-values.yaml` file.

4. Copy all the required configuration values from the `docbroker` section of the previous deployment to the `docbroker` section in the `documentum/config/passwords.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

Notes

- Upgrade process is in descending order. The upgrade process starts from the second connection broker (for example, `dbr-1`) followed by the first connection broker (for example, `dbr-0`).
- If you encounter any problems during the upgrade process with the new image, then the upgrade process stops automatically. In addition, you can roll back to the previous image. "Rolling back the upgrade process" on page 21 contains detailed information.
- The time for the upgrade process is approximately four minutes for each pod. If the replica count of the connection broker pod is more than one, then there is no downtime. It is because, when one pod is in the process of upgrade, the other pod serves the requests. However, if the client is connected through external connection broker, a downtime occurs for approximately four minutes.

5. Copy all the required configuration values from the `content-server` section of the previous deployment to the `content-server` section in the `documentum/config/passwords.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

Notes

- Replica count should not be modified during the initial upgrade. To modify replica count, perform another upgrade exclusively for replica count change.
- Upgrade process is in descending order. The upgrade process starts from the second Documentum CM Server (for example, `dcs-pg-1`) followed by the first Documentum CM Server (for example, `dcs-pg-0`).

- While upgrading the Documentum CM Server pod, the existing Documentum CM Server pod is deleted and new Documentum CM Server pod is created. VCTs and PVCs remain as is and the new pods continue to mount the old VCTs and PVCs.
- If you encounter any problems during the upgrade process with the new image, then the upgrade process stops automatically. In addition, you can roll back to the previous image. "Rolling back the upgrade process" on page 21 contains detailed information.

6. Copy all the required configuration values from the `cs-logging-configMap` section of the previous deployment to the `cs-logging-configMap` section in the `documentum/config/passwords.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

7. Copy all the required configuration values from the `dctm-ingress` section of the previous deployment to the `dctm-ingress` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

8. Copy all the required configuration values from the `cs-dfc-properties` section of the previous deployment to the `cs-dfc-properties` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

9. Copy all the required configuration values from the `otds` section of the previous deployment to the `otds` section in the `documentum/dockerimages-values.yaml`, `documentum/config/passwords.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

10. Optional If you want to enable the Event Hub feature during the upgrade process, ensure that you follow the information provided in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

11. Upgrade the 24.4 or 25.2 Documentum CM Server pod using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/passwords.yaml --values <location where Helm charts are
extracted>/platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
extracted>/documentum-resources-values-test-small.yaml --values <location where
Helm charts are extracted>/documentum-components.yaml --namespace <name of
namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

12. Verify the status of the successful upgrade.

    a. Verify if all the pods are recreated successfully after the upgrade using the following example command:

    ```
    kubectl get pods
    ```

    If the pods are recreated correctly after the upgrade process, then the READY state of each pod is displayed as `3/3` (one for Documentum CM Server, one for Graylog, and one for Event Hub (if enabled)) and the STATUS state is shown as `Running`.

    b. Verify the installation log files using the following example command:

    ```
    kubectl logs dctmcs-0 -c dctmcs
    ```

    If the upgrade is successful, no errors are reported in the log files.

    c. Verify if the repository is updated successfully. Log in to the pod and run a command in the following command format:

    ```
    kubecl exec -ti <name of pod> -c <name of container> bash
    ```

    Verify the `<repository_name>.log` file located at `/opt/dctm/dba/logs`. If the upgrade is successful, no errors are reported in the log file.

    d. Verify the Documentum CM Server version. Log in to the pod and run a command in the following command format:

    ```
    kubectl exec -ti <name of pod> -c <name of container> bash
    ```

    Run the following command:

    ```
    documentum -version
    ```

    If the upgrade is successful, the upgraded version of Documentum CM Server is displayed.

    e. Verify if all the existing ingress URLs are working correctly.

    f. Verify if you can create a sample document and check the document successfully into the upgraded repository using the IAPI commands.

    Note: Verification can also be done for the secondary repository. Ensure that you specify the name of the repository in the following format in the IAPI command:

    ```
    <name of repository>.<server_config_name>
    ```

    g. Verify if New Relic monitors are created for Documentum CM Server and Java Method Server and that you are able to generate the metrics.

    h. Verify if the logs are generated on the Graylog server console for Documentum CM Server and connection broker.

    i. Verify the status of the pod. If the upgrade is successful, the status of the pod is active.

    j. After the successful upgrade, do the following:

i. Perform all the tasks as described in "Licensing OpenText Documentum CM" on page 45.

ii. Upgrade AEK as described in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

Note: The `DM_DOCBROKER_E_NETWORK_ERROR` and `DM_DOCBROKER_E_CONNECT_FAILED_EX` errors are reported in the repository log files of the secondary Documentum CM Server pod after an upgrade process. You can ignore these errors.

To upgrade from non-HashiCorp Vault 24.4 or non-HashiCorp Vault 25.2 to HashiCorp Vault-enabled 25.4:

1. Update the new image location details of `db`, `docbroker`, and `content-server` in the `documentum/dockerimages-values.yaml` file of 25.4.

2. Copy all the required configuration values from the `global variables` section of the previous deployment to the `global variables` section in the `documentum/dockerimages-values.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4.

   Update the value of `global.isVaultEnabled` to `true` and the value of `global.aekLocation` to `Remote_Vault` in the `documentum/values.yaml` file of 25.4. Then, update the value of `global.vaultType` to `HashiCorp`.

   Ensure that you copy all the password information from the previous deployment and store them in the HashiCorp Vault server in the *<secret_name>*/*<key_name>* format as mentioned in the `documentum/config/passwords_vault.yaml` file.

   In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

3. Copy all the required configuration values from the `db` section of the previous deployment to the `db` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4.

   Ensure that you copy all the password information from the previous deployment and store them in the HashiCorp Vault server in the *<secret_name>*/*<key_name>* format as mentioned in the `documentum/config/passwords_vault.yaml` file.

   In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

   Important

   Ensure that the database version is same for both the existing and new deployment.

   For example, if your 25.2 deployment is running with PostgreSQL 14.4 database container, then in your 25.4 upgrade, ensure that you change the database version to 14.4 as follows:

- Value of `version` in the `documentum/charts/db/Chart.yaml` file.
- Value of `db.images.db.tag` in the `documentum/dockerimages-values.yaml` file.

4. Copy all the required configuration values from the `docbroker` section of the previous deployment to the `docbroker` section in the `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4.

   Ensure that you copy all the password information from the previous deployment and store them in the HashiCorp Vault server in the *<secret_name>*/*<key_name>* format as mentioned in the `documentum/config/passwords_vault.yaml` file.

   In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

   Notes

   - Upgrade process is in descending order. The upgrade process starts from the second connection broker (for example, `dbr-1`) followed by the first connection broker (for example, `dbr-0`).
   - If you encounter any problems during the upgrade process with the new image, then the upgrade process stops automatically. In addition, you can roll back to the previous image. "Rolling back the upgrade process" on page 21 contains detailed information.
   - The time for the upgrade process is approximately four minutes for each pod. If the replica count of the connection broker pod is more than one, then there is no downtime. It is because, when one pod is in the process of upgrade, the other pod serves the requests. However, if the client is connected through external connection broker, a downtime occurs for approximately four minutes.

5. Copy all the required configuration values from the `content-server` section of the previous deployment to the `content-server` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4.

   Ensure that you copy all the password information from the previous deployment and store them in the HashiCorp Vault server in the *<secret_name>*/*<key_name>* format as mentioned in the `documentum/config/passwords_vault.yaml` file.

   In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

   Notes

   - Replica count should not be modified during the initial upgrade. To modify replica count, perform another upgrade exclusively for replica count change.

- Upgrade process is in descending order. The upgrade process starts from the second Documentum CM Server (for example, `dcs-pg-1`) followed by the first Documentum CM Server (for example, `dcs-pg-0`).
- While upgrading the Documentum CM Server pod, the existing Documentum CM Server pod is deleted and new Documentum CM Server pod is created. VCTs and PVCs remain as is and the new pods continue to mount the old VCTs and PVCs.
- If you encounter any problems during the upgrade process with the new image, then the upgrade process stops automatically. In addition, you can roll back to the previous image. "Rolling back the upgrade process" on page 21 contains detailed information.

6. Copy all the required configuration values from the `cs-logging-configMap` section of the previous deployment to the `cs-logging-configMap` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4.

   Update the HashiCorp Vault configuration information in the `cs-logging-configMap.vault.configmap` section in the `documentum/config/configuration.yml` file.

   Ensure that you copy all the password information from the previous deployment and store them in the HashiCorp Vault server in the *<secret_name>*|*<key_name>* format as mentioned in the `documentum/config/passwords_vault.yaml` file.

   In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

7. Copy all the required configuration values from the `dctm-ingress` section of the previous deployment to the `dctm-ingress` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4.

   In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

8. Copy all the required configuration values from the `cs-dfc-properties` section of the previous deployment to the `cs-dfc-properties` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4.

   In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

9. Copy all the required configuration values from the `otds` section of the previous deployment to the `otds` section in the `documentum/dockerimages-values.yaml`, `documentum/config/passwords_vault.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

10. Perform all the relevant tasks and provide the appropriate values for all variables as described in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

11. Upgrade the AEK key. To upgrade the AEK key, do the following:

    a. Ensure that you update the new AEK key name (`CSaek`) in `content-server.contentserver.aek.name` in the `documentum/config/configuration.yml` file.

    b. Upload the AEK passphrase from the previous deployment and the new AEK passphrase to the HashiCorp Vault server.

    c. In the `documentum/config/passwords_vault.yaml` file, do the following:

        i. Provide the secret name value for `cs-secrets.contentserver.aek.oldPassphrase`.

        ii. Set the value of `cs-secrets.contentserver.aek.passphrase` with the new key name.

        For example:

```
aek:
        oldPassphrase: AEK_PASSWORD/aek_name
        passphrase: AEK_PASSWORD/CSaek
```

12. Optional If you want to enable the Event Hub feature during the upgrade process, ensure that you follow the information provided in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

13. Upgrade the 24.4 or 25.2 Documentum CM Server pod using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/passwords_vault.yaml --values <location where Helm charts are
extracted>/platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
extracted>/documentum-resources-values-test-small.yaml --values <location where
Helm charts are extracted>/documentum-components.yaml --namespace <name of
namespace>
```

    where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

    The available resource YAML files contain pod sizing values, including CPU and memory.

14. Verify the status of the successful upgrade using step 12.

To upgrade from HashiCorp Vault-enabled 24.4 or HashiCorp Vault-enabled 25.2 to HashiCorp Vault-enabled 25.4:

1. Update the new image location details of `db`, `docbroker`, and `content-server` in the `documentum/dockerimages-values.yaml` file of 25.4.

2. Copy all the required configuration values from the `global variables` section of the previous deployment to the `global variables` section in the `documentum/dockerimages-values.yaml`, `documentum/config/passwords_vault.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. Then, update the value of `global.vaultType` to `HashiCorp`. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

3. Copy all the required configuration values from the `db` section of the previous deployment to the `db` section in the `documentum/config/passwords_vault.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

   Important

   Ensure that the database version is same for both the existing and new deployment.

   For example, if your 25.2 deployment is running with PostgreSQL 14.4 database container, then in your 25.4 upgrade, ensure that you change the database version to 14.4 as follows:

   - Value of `version` in the `documentum/charts/db/Chart.yaml` file.
   - Value of `db.images.db.tag` in the `documentum/dockerimages-values.yaml` file.

4. Copy all the required configuration values from the `docbroker` section of the previous deployment to the `docbroker` section in the `documentum/config/passwords_vault.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

   Notes

   - Upgrade process is in descending order. The upgrade process starts from the second connection broker (for example, `dbr-1`) followed by the first connection broker (for example, `dbr-0`).
   - If you encounter any problems during the upgrade process with the new image, then the upgrade process stops automatically. In addition, you can roll back to the previous image. "Rolling back the upgrade process" on page 21 contains detailed information.
   - The time for the upgrade process is approximately four minutes for each pod. If the replica count of the connection broker pod is more than one, then there is no downtime. It is because, when one pod is in the process of upgrade, the other pod serves the requests. However, if the client is connected through external connection broker, a downtime occurs for approximately four minutes.

5. Copy all the required configuration values from the `content-server` section of the previous deployment to the `content-server` section in the `documentum/config/passwords_vault.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

   Notes

   - Replica count should not be modified during the initial upgrade. To modify replica count, perform another upgrade exclusively for replica count change.
   - Upgrade process is in descending order. The upgrade process starts from the second Documentum CM Server (for example, `dcs-pg-1`) followed by the first Documentum CM Server (for example, `dcs-pg-0`).
   - While upgrading the Documentum CM Server pod, the existing Documentum CM Server pod is deleted and new Documentum CM Server pod is created. VCTs and PVCs remain as is and the new pods continue to mount the old VCTs and PVCs.
   - If you encounter any problems during the upgrade process with the new image, then the upgrade process stops automatically. In addition, you can roll back to the previous image. "Rolling back the upgrade process" on page 21 contains detailed information.

6. Copy all the required configuration values from the `cs-logging-configMap` section of the previous deployment to the `cs-logging-configMap` section in the `documentum/config/passwords_vault.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

7. Copy all the required configuration values from the `dctm-ingress` section of the previous deployment to the `dctm-ingress` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

8. Copy all the required configuration values from the `cs-dfc-properties` section of the previous deployment to the `cs-dfc-properties` section in the `documentum/config/configuration.yml` and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

9. Copy all the required configuration values from the `otds` section of the previous deployment to the `otds` section in the `documentum/dockerimages-values.yaml`, `documentum/config/passwords_vault.yaml`, `documentum/config/configuration.yml`, and `documentum/values.yaml` files of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

10. Optional If you want to enable the Event Hub feature during the upgrade process, ensure that you follow the information provided in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

11. Upgrade the 24.4 or 25.2 Documentum CM Server pod using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/passwords_vault.yaml --values <location where Helm charts are
extracted>/platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
extracted>/documentum-resources-values-test-small.yaml --values <location where
Helm charts are extracted>/documentum-components.yaml --namespace <name of
namespace>
```

    where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

    The available resource YAML files contain pod sizing values, including CPU and memory.

12. Verify the status of the successful upgrade using step 12.

To upgrade from HashiCorp Vault-enabled 24.4 or HashiCorp Vault-enabled 25.2 to Kubernetes native secrets-enabled 25.4:

1. Upgrade from HashiCorp Vault-enabled 24.4 or HashiCorp Vault-enabled 25.2 to HashiCorp Vault-enabled 25.4 environment. For instructions, see "To upgrade from HashiCorp Vault-enabled 24.4 or HashiCorp Vault-enabled 25.2 to HashiCorp Vault-enabled 25.4:" on page 16.

2. Upgrade from 25.4 HashiCorp Vault-enabled environment to 25.4 Kubernetes native secrets-enabled environment.

    a. Copy all the required configuration values from the `documentum/dockerimages-values.yaml` file of the HashiCorp Vault-enabled 25.4 environment to the `documentum/dockerimages-values.yaml` file of Kubernetes native secrets-enabled 25.4 environment.

    b. Copy all the required configuration values from the `documentum/values.yaml` file of the HashiCorp Vault-enabled 25.4 environment to the `documentum/values.yaml` file of Kubernetes native secrets-enabled 25.4 environment.

    c. Set the value of `global.vaultType` to `K8API` in the `documentum/values.yaml` file of Kubernetes native secrets-enabled 25.4 environment.

    d. Set the value of `global.secretConfigName` to the name of the Kubernetes secret created using the `documentum/config/vault_secret.yaml` file in the `documentum/values.yaml` file of Kubernetes native secrets-enabled 25.4 environment.

    e. Set the value of `otds.otdsws.vault.enabled` to `false` in the `documentum/values.yaml` file of Kubernetes native secrets-enabled 25.4 environment.

f. Optional For AppWorks Gateway, set the value of `appworks-gateway.vault.enabled` to `false` in the `documentum/values.yaml` file of Kubernetes native secrets-enabled 25.4 environment.

g. Optional For Intelligent Viewing, set the value of `otiv.global.secretlink.enabled` to `false` in the `documentum/values.yaml` file of Kubernetes native secrets-enabled 25.4 environment.

h. Copy all the required configuration values from the `documentum/config/configuration.yml` file of the HashiCorp Vault-enabled 25.4 environment to the `documentum/config/configuration.yml` file of Kubernetes native secrets-enabled 25.4 environment.

3. Upgrade the 24.4 or 25.2 Documentum CM Server pod using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/passwords_k8api.yaml --values <location where Helm charts are
extracted>/platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
extracted>/documentum-resources-values-test-small.yaml --values <location where
Helm charts are extracted>/documentum-components.yaml --namespace <name of
namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

4. Verify the status of the successful upgrade using step 12.

Note: Upgrade from vault-enabled environment to non-vault environment is not supported.

2.2.1.3 Enabling certificate-based communication in upgraded 25.4 environment

If you want to upgrade from a non-certificate 24.4 or 25.2 environment to a non-certificate 25.4 environment where both the environments were deployed and then you want to enable certificate-based communication in the upgraded 25.4 environment, you must do the following:

1. Upgrade the non-certificate 24.4 or 25.2 environment to a non-certificate 25.4 environment.

   "Upgrading from 24.4 or 25.2 to 25.4" on page 9 contains the instructions.

2. Set the value of enable of `content-server.certificate.customUpgrade.enable` to `true` in the `documentum/config/configuration.yml` file of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4. Then, upgrade the updated 25.4 environment using the Helm upgrade command.

3. Set the value of `useCertificate` to `true` in *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)* in the `documentum/values.yaml` file of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

4. Provide all the certificate-related configuration values for `docbroker` and `content-server` in the `documentum/values.yaml` file of 25.4. In addition, provide the appropriate values for the new variables, if any and required according to your requirement, in 25.4.

5. Modify the value of `content-server.certificate.customUpgrade.enable` to `false` in the `documentum/config/configuration.yml` file of 25.4.

6. Upgrade the updated 25.4 environment using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/<passwords.yaml or passwords_vault.yaml or passwords_k8api.yaml>
--values <location where Helm charts are extracted>/platforms/<cloud platform>.yaml
--values <location where Helm charts are extracted>/dockerimages-values.yaml --
values <location where Helm charts are extracted>/documentum-resources-values-test-
small.yaml --values <location where Helm charts are extracted>/documentum-
components.yaml --namespace <name of namespace>
```

   where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

   The available resource YAML files contain pod sizing values, including CPU and memory.

   Note: If you want to use vault, use `passwords_vault.yaml` or `passwords_k8api.yaml` in the command instead of `passwords.yaml`.

7. Verify if the upgraded environment uses certificate-based communication.

2.2.1.4 Rolling back the upgrade process

Rolling back the upgrade process (rolling back to the previous image) is recommended when the upgrade process fails or when you encounter errors using the new image.

Note: For information about rolling back the Documentum CM Server pod enabled with vault to a Documentum CM Server pod that is not enabled with vault, see the product *Release Notes*.

To roll back to the 24.4 or 25.2 image if AEK is not upgraded:

1. Retrieve the details of history using the following command format:

```
helm history <release name>
```

2. Roll back to the previous image using the following command format:

```
helm rollback <release name> <revision>
```

To roll back to the 24.4 or 25.2 image if AEK is upgraded:

1. Retrieve the details of history using the following command format:
   ```
   helm history <release name>
   ```
2. Upgrade AEK.
   a. Change the value of `content-server.contentserver.aek.name` in the `documentum/config/configuration.yml` file to the same key name mentioned in the 24.4 or 25.2 image.
   b. Change the value of `content-server.contentserver.aek.location` to the same location mentioned in the 24.4 or 25.2 image.
   c. Change the value of `cs-secrets.contentserver.aek.algorithm` to the same key algorithm mentioned in the 24.4 or 25.2 image.
   d. Do not change the value of `cs-secrets.contentserver.aek.passphrase`. It should be same for the upgraded image and the previous image.
   e. Run the Helm upgrade command.

   Ensure that the Helm upgrade is successful and the Documentum CM Server pod is up and running.
3. Roll back to the previous image using the following command format:
   ```
   helm rollback <release name> <revision>
   ```

To roll back to the 24.4 or 25.2 image if installOwner value is changed and upgraded:

1. Retrieve the details of history using the following command format:
   ```
   helm history <release name>
   ```
2. Upgrade the installation owner (`installOwnerUsername`).
   a. Change the value of `installOwnerUsername` in 25.4 in `global variables` to the same value mentioned in `cs-secrets.contentserver.installowner.userName` in the `documentum/values.yaml` file) of 24.4 or 25.2.
   b. Run the Helm upgrade command.

   Ensure that the Helm upgrade is successful and the Documentum CM Server pod is up and running.
3. Roll back to the previous image using the following command format:
   ```
   helm rollback <release name> <revision>
   ```

To roll back to the 24.4 or 25.2 image if service account name is modified and upgraded:

If you have modified the service account name as described in "Prerequisites" on page 8 in "Upgrading Documentum CM Server" on page 8, then to roll back the previous image release version, do the following:

1. Retrieve the details of history using the following command format:

   ```
   helm history <release name>
   ```

   For example:

REVISION	UPDATED	STATUS	CHART	APP VERSION	DESCRIPTION
1	Wed Oct 4 13:19:54 2023	superseded	documentum-24.4	24.4	Install complete
2	Wed Oct 4 14:10:40 2023	superseded	documentum-25.2	25.4	Upgrade complete //major upgrade
3	Wed Oct 4 19:02:54 2023	deployed	documentum-25.2	25.4	Upgrade complete // service account name modified version

2. Roll back to the previous deployment release version with previous service account name (revision 2).

   For example:

   ```
   helm rollback <release name> 2
   ```

3. Roll back to the previous release image version (revision 1).

   For example:

   ```
   helm rollback <release name> 1
   ```

Notes

- Database schema changes are not reverted when you roll back.
- Rolling back within the same Documentum CM Server version is not supported if you have upgraded AEK during the upgrade process.

2.2.2 Upgrading client

This section provides information about upgrading client.

2.2.2.1 Upgrading or patching client single Helm deployment

This section includes instructions about how to upgrade from one version of client to a newer version.

Prerequisites:

- Get the required version of the Helm charts.
- Foundation SOAP API, Foundation REST API, Foundation CMIS API, Intelligent Viewing, BPS, xDA, Records Client/Records Queue Manager, OpenText Documentum CM for Microsoft 365, and Content Connect components are enabled by default. Enable or disable the components in `documentum/documentum-components.yaml` according to your requirement during upgrade.

- Update `documentum/dockerimages-values.yaml` and `documentum/Chart.yaml` with the newer versions.
- Modify the `documentum/values.yaml` file according to your old deployment.
- Ensure that you back up the database before proceeding to the upgrade. Before the database backup, set the repository to the Dormant state.

  Note: After the database backup is done, revert the Dormant state to Active state, log in to Documentum Administrator with `backupadmin` user credentials and go to **Administration > Basic Configuration > Repository**, right-click the repository name and click **Make Active**.

Upgrade notes:

- Verify the supported Kubernetes version for the required client version before upgrade. Check the product *Release Notes* for details, which is available on My Support (support.opentext.com).
- If you used the `d2-resources-values-<config>.yaml` file during deployment, before upgrading, ensure that the replica counts for all the pods are the same as the existing deployment's `d2-resources-values-<config>.yaml` file.
- Ensure that the `configNameOption` variable in the `documentum/values.yaml` file is the same as the previous deployment, that is, if it is `HOSTNAME`, retain it as `HOSTNAME` and if it is `HA`, retain it as `HA`.

```
content-server:
 otds:
  configNameOption: HOSTNAME
```

  Note: If the `configNameOption` variable is not available in the `documentum/values.yaml`, check and update the value of the `configNameOption` variable in the `documentum/charts/content-server/values.yaml` file.

- Ensure that the PVC sizes for all components are equal or greater than the PVC sizes in the existing deployment's `d2-resources-values-<config>.yaml` file.
- If Documentum Administrator is part of the older deployment, do the following steps to clear the existing Documentum Administrator deployment before proceeding with the update:

```
kubectl delete deployment.apps/da -n <namespace>
kubectl delete pvc da-pvc -n <namespace>
```

- From the 23.4 release onwards, the client master Helm chart has been renamed to `documentum`. If you are upgrading to version 23.4 and later, read `d2` as `documentum` for the Helm chart name and folder structure in any of the upgrade steps.

From OTDS 22.1.0 onwards, the OTDS architecture has changed. OpenDJ will no longer be used as a directory server. Instead, the existing or new PostgreSQL or SQL Server or Oracle or SAP database can be used. If you are directly upgrading OTDS from 21.3.x or earlier to 22.1.0 or later, data will be lost. Data migration is needed before upgrading from the previous release.

Migrating data from a running OpenDJ container:

- When migrating data from a running OpenDJ container, an OTDS 22.1.0 temporary instance is needed to connect to the previous OpenDJ service and migrate the data. To do that, scale down the existing `otdsws` replicas to `0` and scale the OpenDJ replicas to `1` using the following commands:

```
kubectl scale deployment otdsws --replicas=0 kubectl scale statefulset opendj --
replicas=1
```

- Download the 22.1.0 OTDS Helm chart or copy the `otds` folder from 22.2 version of `documentum/charts/otds` to a temporary location and update the `values.yaml` file with the following values:
  - Set the value of `otdsUseReleaseName` to `true`.
  - Set the value of `otdsws.publicHostname` to `otds-migrate`.
  - Set the value of `otdsws.migration.enabled` to `true`.
  - Provide existing database details:

```
otdsdb.url = jdbc:postgresql://<db_host>:<db_port>/postgres
otdsdb.username = <db_username>
otdsdb.password = <db_password>
```

Note: `otdsdb.url` must be given according to the database being used. Here are the sample values:

  - `jdbc:postgresql://postgres.domain.local:5432/otdsdb`
  - `jdbc:sqlserver://ms-sql.domain.local:1433;databaseName=otdsdb`
  - `jdbc:oracle:thin:@oracle.domain.local:1521:otdsdb`
  - `jdbc:sap://hana.domain.local:30015/?databaseName=otdsdb`

- Provide the container image and tag details:

```
image.source = <artifactory_location>
image.name = <otds-server_image_name>
image.tag = <otds-server_image_version>
```

- Deploy the modified OTDS Helm chart using the following command:

```
helm install otds-migrate <path_to_the_values_yaml_directory_of_otdschart>
```

- If upgrading an existing Helm v2 deployment, migrate the existing deployment to Helm v3. *Helm* documentation contains detailed information.
- Verify the otdsws container logs (of the temporary deployment), access the OTDS frontend using the newly created ingress, and check if the data/ functionality is intact. If everything is working fine, uninstall the temporary deployment using the following command and proceed with the client upgrade, making sure that migration is set to `false` in the Helm charts while upgrading:

```
helm uninstall otds-migrate
```

The following table contains additional information that is required when you upgrade client from specific versions:

From client version / To client version	20.4	21.1	21.2	21.3	21.4	22.1 or later
20.3	Section 1	Section 2				
20.4	N/A	Section 2				
21.1	N/A		Section 2		Section 3	Section 4
21.2	N/A				Section 3	Section 4
21.3	N/A				Section 3	Section 4
21.4	N/A					Section 4

For example, if you are upgrading client from 21.2 to 22.2, follow the instructions in Section 3 and Section 4.

2.2.2.1.1 Section 1: From 20.3 to 20.4

If you are upgrading an existing Helm v2 deployment, then migrate the existing deployment to Helm v3. *Helm* documentation contains detailed information.

2.2.2.1.2 Section 2: From 20.4 to 21.2 or a later version

If you are upgrading client to version 21.2 or later, you must also upgrade the Documentum CM Server to version 21.2 or later.

client image versions 21.2 and later work with Documentum CM Server version 21.2 and later, which uses Tomcat as the Java Method Server. Documentum CM Server 21.1 and earlier uses WildFly as the Java Method Server.

2.2.2.1.3 Section 3: From 21.2 to 21.4

1. Delete the otdsws service using the following command:

   ```
   kubectl delete svc otdsws
   ```

2. Delete the `d2config`, `d2rest`, and `d2smartview` statefulsets using the following commands:

   ```
   kubectl delete statefulset --cascade=orphan d2config
   kubectl delete statefulset --cascade=orphan d2rest
   kubectl delete statefulset --cascade=orphan d2smartview
   ```

3. Edit the PVC size of the following PVCs to 1Gi:

   ```
   d2config-vct- <pod-name>
   d2rest-vct- <pod-name>
   d2smartview-vct- <pod-name>
   ```

   For example:

```
kubectl edit pvc d2config-vct-d2config-0
spec:
     accessModes:
     - ReadWriteOnce
     resources:
           requests:
                    storage: 107374182400m
     storageClassName: trident-nfs
     volumeMode: Filesystem
     volumeName: pvc-a32685ea-4b50-400d-b416-93e43f79acd0
status:
     accessModes:
     - ReadWriteOnce
     capacity:
           storage: 103Mi
     phase: Bound
```

4. For `resources.requests`, change storage from 107374182400m to 1Gi, and save the file.

2.2.2.1.4 Section 4: Upgrade from 22.1 or later

Follow these steps after a successful upgrade:

1. If the existing deployment is with Google object store or S3 store, set the default filestore to filestore_01 using the following IAPI commands by logging into Documentum CM Server pod:

```
?,c,alter type dm_sysobject set DEFAULT STORAGE = 'filestore_01'
?,c,alter type dm_document set DEFAULT STORAGE = "filestore_01'
```

2. In 23.4, New Relic is disabled by default, so if you are upgrading to 23.4 and if New Relic is enabled in existing deployment, then enable New Relic in 23.4 `documentum/values.yaml`. Change the value of `newrelic` to `true`.

```
newrelic: &newrelic_enabled true
```

3. If Content Connect is already enabled in the existing deployment, set the following value to `false` in `documentum/config/configuration.yml`. This is required after the upgrade process to use the same database instead of creating a new one.

```
contentconnect:
  contentconnectdb:
     value: false
```

4. Uncomment and update the values in `dctm-workflow-designer` ConfigMap template YAML with existing deployment details (ensure to use correct ingress with respect to 22.2) in the `d2\charts\charts\dctm-workflow-designer\templates\dctm-workflow-designer-configmap.yaml` file.

```
# OTDS SSO authentication settings
rest.security.auth.mode-otds-basic
rest.security.otds.login.url={otds-idp}/otdsws/login?
response_type=token&client_id={client_id}&client_secret={client_secret}&logon_appnam
e={app name to display in log on screen}
rest.signon.logout.url=/otds-signin.jsp?logout=yes
rest.security.otds.reverseproxy.url=<Workflow Designer URL/base URL of the reverse
proxy/load balancer>
```

5. AppWorks Gateway upgrade from 21.3.0 to 21.3.1 fails when upgrading client from 21.3 to 21.4. Due to this known issue in AppWorks Gateway, the upgrade

is not supported for client 21.4 and AppWorks Gateway must be deleted before doing the upgrade.

6. If OTDS auto-configuration is not used in the previous deployment, then update the Redirect URLs in the Oauth clients section from `d2ingress` to the ingress host.

7. Update the trusted sites to ingress in the OTDS admin portal, if any.

8. If Content Connect is enabled in the previous deployment, the URLs in the Content Connect Admin portal must be updated to the ingress host and new manifest files must be used in Office365 and Outlook to work with the Content Connect plug-in.

9. If you are upgrading client to 22.1 or later, then update the ingress annotation values in the `documentum/platforms/<platform>.yaml` file according to your platform's ingress if necessary, and then run the following command:

```
helm dependency update .
helm upgrade <d2deployment> . --values config\configuration.yml --values
dockerimages-values.yaml --values d2-resources-values-<config>.yaml --values
platforms/<platform>.yaml -n <namespace>
```

   where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files and `platform` can be `azure`, `aws`, `gcp`, or `openshift`.

   The available resource YAML files contain pod sizing values, including CPU and memory.

   Note: If you are upgrading to version 23.4 and later, `documentum/documentum-components.yaml` must be used for enabling or disabling the components. The resource YAML names begin with `documentum` and not `d2`.

   An example Helm upgrade command:

```
helm upgrade <d2deployment> . --values config\configuration.yml --values
dockerimages-values.yaml --values documentum-resources-values-<config>.yaml --values
platforms/<platform>.yaml --values documentum-components.yaml -n <namespace>
```

10. By default, client URLs are added as part of the deployment. You must click **Tools > Refresh Cache**, if you do not see the client URLs in D2-Config.

11. After the deployment is complete and configured with native filestore and the functional testing is completed, make the Google Cloud store as default store using the following IAPI command in the Documentum CM Server pod:

```
kubectl exec - it cedcs-pg-0 -nxcp
iapi <docbase_name>
API> ?,c.alter type dm_sysobject set DEFAULT STORAGE = '<gcp>'
API> ?,c,alter type dm_document set DEFAULT STORAGE = 'gcp>'
API> reinit,c
```

12. To check the functionality of Google Cloud Platform object store with default as `true`, do the following:

- Try to push or pull the content through the IAPI session. Use the following steps (while pushing the content, do not set `a_storage_type` as content will go to the default store by default):

```
API> create,c,dm_document
...
0901e24080002dee
API> setfile,c,1,/opt/dctm/share/sample.txt,crtext
...
OK
API> save,c,1
...
OK
API> getpath,c,1
...
https://storage.googleapis.com/testidlit/01e240/80/00/11/ce.dat
API> getfile,c,1
```

Note: The content is now stored in the Google Cloud store, `https://storage.googleapis.com/testidlit/01e240/80/00/11/ce.dat`, and the content will be in an encrypted format. Before the DAR installation, the default store must be switched to the filestore.

2.2.2.2 Rolling back the upgrade process

Notes

- Rollback is not supported in 22.2 to previous versions since the d2installer is moved as an init container and there is no need for shared PVC.
- AppWorks Gateway rollback or downgrade is not supported.

To roll back to any previous revision of the release:

1. If you want to roll back to any previous revision of the release, run the following command:

```
helm rollback <release> [REVISION]
```

The first argument of the rollback command is the name of a release, and the second is a revision (version) number. If this argument is omitted, it will roll back to the previous release. To see the revision numbers, run the `helm history release name` command.

Note: DAR rollback is not supported.

2. After the Helm rollback is successful, restore the database backed up prior to the upgrade process.

Caution

Do not restore the database without bringing the repository to Dormant state. Uninstall Documentum Administrator and stop the Documentum Content Intelligence Services and OpenText™ Documentum™ Content Management Transformation Services services. Then, restore the Documentum CM Server database backed

up prior to the upgrade process. Let the repository be in the Dormant state until all the components are rolled back.

3. If you are rolling back from client 21.3 or later to 21.2, then do the following before rollback:

   a. If Documentum Administrator is part of the deployment, run the following commands to clear the existing Documentum Administrator deployment before proceeding with the rollback:

   ```
   kubectl delete deployment.apps/da -n <namespace>
   kubectl delete pvc da-pvc -n <namespace>
   ```

   b. Delete the otdsws service using the following command:

   ```
   kubectl delete svc otdsws -n <namespace>
   ```

   c. Delete the `d2config`, `d2rest`, and `d2smartview` statefulsets using the following commands:

   ```
   kubectl delete statefulset --cascade=false d2config - n <namespace>
   kubectl delete statefulset --cascade=false d2rest - n <namespace>
   kubectl delete statefulset --cascade=false d2smartview - n <namespace>
   ```

   Note: If you see the `warning: --cascade=false is deprecated (boolean value) and can be replaced with --cascade=orphan` warning message, then use the `orphan` value instead of `false`.

2.2.3 Upgrading Reports

This section provides information about upgrading Reports from 23.4 and later versions to 25.4.

Before upgrading the Reports pod, you must remove the following marker and log files in the `d2config-0` pod for a seamless deployment.

```
> kubectl exec -ti d2config-0 -n <namespace> -- bash
> rm /opt/D2CS-install/custom_config_import_DCTM-Reports-Application-x.x.x-Export-
Config
> rm /opt/D2CS-install/DCTM-Reports-Application-x.x.x-Export-Config.log
```

To upgrade Reports from 24.4 or 25.2 to 25.4:

1. Update the `documentum/dockerimages-values.yaml` with the image tag according to the following table:

Product or component name	Image name	Image tag
OpenText™ Documentum™ Content Management Reports	`ot-dctm-reports-client`	25.4.0 or 25.4

Product or component name	Image name	Image tag
OpenText™ Documentum™ Content Management Reports	ot-dctm-reports-base	25.4.0 or 25.4
OpenText™ Documentum™ Content Management Reports	ot-dctm-reports-installer	25.4.0 or 25.4

2. Upgrade the deployment using the following command format:

```
> cd <path_to_documentum_directory>>
helm upgrade <documentumDeployment> . --values .\config\configuration.yml --
values .\config\constants.yaml --values .\config\passwords.yaml -- values platforms/
<platform>.yaml --values dockerimages-values.yaml --values <resources_yaml> -n
<namespace>
```

To upgrade Reports from 23.4 or 24.2 to 25.4:

1. Set the value of `dtrbase.enabled` to `false` in the `dtrbase` section of 23.4 or 24.2 version in the `documentum-components.yaml` file.
2. Upgrade the deployment using the following command format:

```
> cd <path_to_documentum_directory>>
helm upgrade <documentumDeployment> . --values .\config\configuration.yml --
values .\config\constants.yaml --values .\config\passwords.yaml -- values platforms/
<platform>.yaml --values dockerimages-values.yaml --values <resources_yaml> -n
<namespace>
```

   This deletes the Reports pod along with its dependent PVC and SVC.

3. Set the value of `dtrbase.enabled` to `true` to deploy the Reports pod and re-run the Helm upgrade with the applicable Reports image details.

2.2.4 Upgrading OpenText Documentum CM for Microsoft 365

Before upgrading the OpenText Documentum CM for Microsoft 365 pod, do the following:

- Delete the `CCSmartviewTeamsDarInstallSuccessMarker` marker file from the `dcs-pg` pod from the following location: `/opt/dctm_docker/customscriptpvc/`
- Revoke the allocated license to the `xecmserviceuser` user and remove the account as the `xecmserviceuser` user is no longer required. The OTDS user with the `xecmserviceuser` user name account is replaced by `M365_SERVICE`.

To upgrade OpenText Documentum CM for Microsoft 365 from version 25.2 to 25.4:

1. Update the app registration details in Microsoft Entra admin center according to the latest version requirements.

   Note: Ensure that the app registration details in the Microsoft Entra admin center comply with the latest version requirements defined by Microsoft.

2. Update the image tag according to the following table:

OpenText™ Documentum™ Content Management for Microsoft® 365™	`ot-dctm-smartviewm365`	25.4.0 or 25.4
OpenText™ Documentum™ Content Management for Microsoft® 365™	`ot-dctm-smartviewm365customjar`	25.4.0 or 25.4
OpenText™ Documentum™ Content Management for Microsoft® 365™	`ot-dctm-smartviewm365-ns`	25.4.0 or 25.4

3. Keep the database details according to the previous deployment.

4. Upgrade the pod using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/passwords_vault.yaml --values <location where Helm charts are
extracted>/platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
extracted>/documentum-resources-values-test-small.yaml --values <location where
Helm charts are extracted>/documentum-components.yaml --namespace <name of
namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

2.2.5 Upgrading Documentum Connector for Core Share

Note: Back up the database before you begin to upgrade or rollback.

1. In the existing deployed chart, go to the `documentum/charts/dctm-dcc/templates` location.

2. Update the `jobs.yaml` file with the following details:

Existing value	Replace with
`{{- if and (eq .Values.coreNotification.enabled true) (eq .Values.coreNotification.dbSchemaInit.enabled true) }}`	`{{- if and (eq .Values.coreNotification.enabled true) (eq .Values.coreNotification.dbSchemaInit.enabled true) .Release.IsInstall }}`

Existing value	Replace with
{{- if and (eq .Values.metadata.enabled true) (eq .Values.metadata.dbSchemaInit.enabled true) }}	{{- if and (eq .Values.metadata.enabled true) (eq .Values.metadata.dbSchemaInit.enabled true) .Release.IsInstall }}
{{- if and (eq .Values.syncagent.enabled true) (eq .Values.syncagent.dbSchemaInit.enabled true) }}	{{- if and (eq .Values.syncagent.enabled true) (eq .Values.syncagent.dbSchemaInit.enabled true) .Release.IsInstall }}
{{- if and (eq .Values.syncnshareManual.enabled true) (eq .Values.syncnshareManual.dbSchemaInit.enabled true) }}	{{- if and (eq .Values.syncnshareManual.enabled true) (eq .Values.syncnshareManual.dbSchemaInit.enabled true) .Release.IsInstall }}

3. In the `documentum/config/configuration.yml` file, set the value of the following variables to `false`:

   - `dctm-dcc.dcc.metadata.dbSchemaInit.enabled`
   - `dctm-dcc.dcc.syncagent.dbSchemaInit.enabled`
   - `dctm-dcc.dcc.syncnshareManual.dbSchemaInit.enabled`
   - `dctm-dcc.dcc.coreNotification.dbSchemaInit.enabled`

4. Upgrade the existing deployment to apply the changes using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/passwords_vault.yaml --values <location where Helm charts are
extracted>/platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
extracted>/documentum-resources-values-test-small.yaml --values <location where
Helm charts are extracted>/documentum-components.yaml --namespace <name of
namespace>
```

   where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

   The available resource YAML files contain pod sizing values, including CPU and memory.

5. Download the 25.4 or 25.4.2 Helm chart, provide the appropriate values for the relevant variables, if required for your environment.

   To provide the appropriate values, open the YAML files, and read the descriptions of the variables.

6. Before deploying the Documentum Connector for Core Share application, you must register the mail service on Microsoft Entra admin center. For more information, see *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

7. Update the image tag as per the following table:

Product or component name	Image name	Image tag
OpenText™ Documentum™ Connector for Core Share	`ot-dctm-dcc-dbschema`	25.4.2 or 25.4
OpenText™ Documentum™ Connector for Core Share	`ot-dctm-dcc`	25.4.2 or 25.4
OpenText™ Documentum™ Connector for Core Share	`ot-dctm-dcc-darinitcontainer`	25.4.2 or 25.4

8. In the `documentum/config/configuration.yml` file, provide the **Application (client) ID** value from the **App registrations** page in Microsoft Entra admin center to `dctm-dcc.mailService.configMap.mailconfig.clientID`.

9. In the `documentum/config/configuration.yml` file, provide the **Directory (tenant) ID** value from the **App registrations** page in Microsoft Entra admin center to `dctm-dcc.mailService.configMap.mailconfig.tenantID`.

10. In the `documentum/config/configuration.yml` file, provide the **Client credentials** (encrypted client secret) value from the **App registrations** page in Microsoft Entra admin center to `dctm-dcc.mailService.configMap.mailconfig.clientSecret`.

    Note: For more information about client ID, tenant ID, and encrypted client secret, see *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)*.

11. In the `documentum/config/configuration.yml` file, to upgrade from any version to 25.4, set the value of the following variables to `false`:

    - `dctm-dcc.dcc.metadata.dbSchemaInit.enabled`
    - `dctm-dcc.dcc.syncagent.dbSchemaInit.enabled`
    - `dctm-dcc.dcc.syncnshareManual.dbSchemaInit.enabled`
    - `dctm-dcc.dcc.coreNotification.dbSchemaInit.enabled`

12. Ensure that *<DB_NAME>* used for each service is same as in the previous deployment to maintain data continuity.

    For example:

```
dbSchemaInit:
      enabled: false
      dbname: <DB_NAME>
```

13. Upgrade the pod using the following command format:

```
helm upgrade <release name> /opt/temp/documentum --values <location where Helm
charts are extracted>/config/configuration.yml --values <location where Helm charts
are extracted>/config/constants.yaml --values <location where Helm charts are
extracted>/config/passwords_vault.yaml --values <location where Helm charts are
extracted>/platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
```

```
extracted>/documentum-resources-values-test-small.yaml --values <location where
Helm charts are extracted>/documentum-components.yaml --namespace <name of
namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

2.2.5.1 Rolling back the upgrade process

Rolling back the upgrade process (rolling back to the previous image) is recommended when the upgrade process fails or when you encounter errors using the new image.

```
helm rollback <release name> <revision>
```

Note: In case you want to roll back, you can roll back only to the previously used version.

2.2.6 Upgrading other products and components

1. Update the new image details in the `documentum/dockerimages-values.yaml` file of 25.4.
2. Modify the relevant variables in all the required YAML files.
3. Upgrade the pod using the following command format:

```
helm upgrade <release name> <location where Helm charts are extracted> --values
<location where Helm charts are extracted>/config/configuration.yml --values
<location where Helm charts are extracted>/config/constants.yaml --values <location
where Helm charts are extracted>/config/<passwords.yaml or passwords_vault.yaml or
passwords_k8api.yaml> --values <location where Helm charts are extracted>/platforms/
<cloud platform>.yaml --values <location where Helm charts are extracted>/
dockerimages-values.yaml --values <location where Helm charts are extracted>/
documentum-resources-values-<config>.yaml --values <location where Helm charts are
extracted>/documentum-components.yaml --namespace <name of namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

Notes

- If vault is enabled, use `passwords_vault.yaml` or `passwords_k8api.yaml` in the command instead of `passwords.yaml`.
- When `ConfigMap` is updated, an additional parameter, `--recreate-pods`, is required for the upgrade command as shown in the following command format:

```
helm install --recreate-pods <release name> /opt/temp/documentum --values
<location where Helm charts are extracted>/config/configuration.yml --values
<location where Helm charts are extracted>/config/constants.yaml --values
<location where Helm charts are extracted>/config/<passwords or
```

```
passwords_vault>.yaml --values <location where Helm charts are extracted>/
platforms/<cloud platform>.yaml --values <location where Helm charts are
extracted>/dockerimages-values.yaml --values <location where Helm charts are
extracted>/documentum-resources-values-test-small.yaml --values <location
where Helm charts are extracted>/documentum-components.yaml --namespace <name
of namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

Important

If you have deployed an earlier version of Documentum xPlore, you must upgrade to the latest supported version. For more information, see the latest version of Documentum xPlore patch notes available on OpenText My Support.

2.2.6.1 Rolling back the upgrade process

Rolling back the upgrade process (rolling back to the previous image) is recommended when the upgrade process fails or when you encounter errors using the new image.

```
helm rollback <release name> <revision>
```

2.2.7 Using Helm Merge utility

The Helm Merge utility is an optional tool that streamlines the Helm upgrade process for all Kubernetes deployment upgrades. It eliminates the manual effort of updating the complete Helm chart for each component.

Note: When an exact key match is found in the source Helm chart, the utility updates the values in the target Helm chart. You can only consider merging files if the source and target file names match. The `Chart.yaml` and `dockerimages-values.yaml` files and all subcharts are excluded from merging.

- Ensure you have the latest version of Python 3 installed.
- Set the Python environment variable in `PATH`.

Note: Ensure that the out-of-the-box file names of deployed (old) Helm charts are retained.

To install the Merge utility:

1. Download `mergeutility.zip` from My Support and extract the contents.
2. Open the Merge utility folder in the Helm chart and run the following command:

```
python -m pip install lib/merge-25.4-py3-none-any.whl --force-reinstall
```

3. Run either of the following commands to verify the installation:

```
merge --help
```

or

```
merge --version
```

To run the Merge utility:

1. Explore the options using the following command:

```
merge --help
```

2. Explore the options for merging or comparing Helm charts using the following command:

```
merge folder --help
```

3. Explore the options for merging or comparing YAML files using the following command:

```
merge file --help
```

You can use the file and folder options to merge or compare YAML files or Helm charts.

Table 2-1: File options

Options	Descriptions
`--debug`	Get detailed logs on the terminal.
`--compare-folder`	Use this option to save the command report by providing the folder path. By default, compared reports are saved in the current working directory under `/mergeutility/compare-report`.
`--log-folder`	Use this option to provide folder path to store the log reports. By default, this option stores the log report in the current working directory under `/mergeutility/logs`.
`--compare-only`	This option only compares the YAML files.
`-c`, `--compare`	Compares the YAML files before merging.
`--output`	Avoids any manipulations to the destination YAML file. The updated file is saved to the current working directory.

Options	Descriptions

`--updated-key`	Matches a specific key in the source file to a key in the target file for merging. Use `source_key:target_key` format for the updated keys:
	
	You can specify this option multiple times to provide multiple keys. For example:
	
	`--updated-key key1:key2 key3:key4`
	
	or
	
	`--updated-key key1:key2 --updated-key key3:key4`

Table 2-2: Folder options

Options	Descriptions
`--debug`	Prints logs on the terminal.
`-Y, -y`	Merges all files. If you do not use this option, a prompt appears asking if you want to merge a given file.
`--output`	Avoids any manipulations to the destination YAML file. The updated file is saved to the current working directory under `/mergeutility`.
`--compare-folder`	Use this option to save the command report by providing the folder path. By default, log reports are saved in the current working directory under `/mergeutility/compare-report`.
`--log-folder`	Use this option to save the log report by providing the folder path. By default, log reports are saved in the current working directory under `/mergeutility/logs`.
`-c,--compare`	Enables comparison before and after merge for all files.
`--compare-only`	This option only compares the YAML files.
`-C,--compare-all`	Compares all the files without a prompt.

Options	Descriptions

`--updated-key`	Matches a specific key in the source file to a key in the target file for merging. Use `source_key:target_key` format for the updated keys:
	
	You can specify this option multiple times to provide multiple keys. For example:
	
	`--updated-key key1:key2 key3:key4`
	
	or
	
	`--updated-key key1:key2 --updated-key key3:key4`

`--updated-filename`	Use this option to provide updated `values.yaml` file name. For example, `--updated-filename values.yaml:updated_values.yaml`
`--configuration`	Use this option to provide all the files passed during Helm install with the `--values` option. For example, if the Helm install command is `helm install <existing_deployment>.--values config/configurations.yml`, then the option to be passed to merge utility will be `--configuration config/configuration.yml`.
`--otds-config`	Merges OTDS bootstrap config file in `documentum\charts\otds\charts\otdsws\config`.
`--merge-disabled-components`	Use this option to merge the values of the components that are not enabled.

2.3 Upgrading OpenText Documentum CM using Documentum™ Cloud Assist

This section provides information to prepare and generate updated Helm chart using `Documentum™ Cloud Assist` and to upgrade OpenText Documentum CM at the command prompt. The upgrade process is supported from the 23.4 version.

Supported upgrade

To upgrade from 23.4 to 25.4:

1. Upgrade from 23.4 to 24.4.
2. Extract the `editor-config.zip` file from step 3 to a temporary location.
3. Copy the `.editor-config` folder from the 24.4 folder, paste in the 24.4 `documentum` folder of extracted Helm charts, and then upgrade to 25.4.

To upgrade from 24.4 to 25.4:
You can upgrade from 24.4 to 25.4 directly.

To upgrade from 24.4 to 25.2:

1. Upgrade from 24.4 to 25.2.
2. Copy the `.editor-config` folder from the 25.2 folder and paste in the 25.2 `documentum` folder of extracted Helm charts.

To upgrade from 25.2 to 25.4:
You can upgrade from 25.2 to 25.4 directly.

For more information about the upgrade instructions, see "To prepare and generate updated Helm chart using Documentum™ Cloud Assist and to upgrade OpenText Documentum CM at command prompt:" on page 40.

Caution

- Retain the previous deployment password information as is for `cs-secrets.docbase.password`, `cs-secrets.contentserver.globalRegistry.password`, `cs-secrets.contentserver.aek.passphrase`, and `cs-secrets.contentserver.install.appserver.admin.password` in the `documentum/config/passwords.yaml` file.
- Retain the Graylog and Fluentd variables.
- If a mismatch exist in the configuration values, it impacts the deployment of 25.4 Helm chart and the upgrade process may fail.

To prepare and generate updated Helm chart using Documentum™ Cloud Assist and to upgrade OpenText Documentum CM at command prompt:

1. Run `dctm-cloud-assist-<version>.exe` on Windows or `dctm-cloud-assist-<version>.AppImage` on Linux.
2. On the home page in the **Upgrade** tile, click **Upgrade**.
3. On the **Upgrade** dialog box, complete all the relevant prerequisite tasks such as downloading the container images, Helm charts, and so on.

   Do the following:

   a. For **New Helm Chart**, click **Upload** to upload the 25.4 Helm chart downloaded from My Support.

   b. For **Existing Helm Chart**, click **Upload** to upload your version of Helm chart that you want to upgrade.

   Make sure that you upload your 23.4, or 24.4, or 25.2 Helm chart for **Existing Helm Chart**.

Note: If you have a chart with custom changes and want to merge, select the **Merge and Customize** check box. For more information about merge and customization, click the information icon.

4. Click **Compare**.

   Notes

   - Click **Reset** to upload the new and existing Helm charts again.
   - Click **Cancel** to go to the home page to restart the upgrade process.

5. In the **Uploaded Charts** section, the version of the new Helm chart, existing Helm chart, and the deployed Helm chart without changes are displayed.

   In the **Reports** section, click **View** in each tile to view and understand the reports information for the following categories:

   - **Files Compared**: Lists the files compared for the upgrade process.
   - **To Be Merged**: Lists the variables customized in your Helm chart that will be merged during the upgrade process.
   - **New Changes**: Lists the variables from the new Helm chart.
   - **Removed**: Lists the variables removed from the new Helm chart.
   - **To Be Excluded**: Lists the variables that will be excluded from the upgrade process.
   - **Subchart Modifications**: Lists the variables modified within the Helm subcharts.

     Note: The **Subchart Modifications** tile appears only if you select the **Merge and Customize** check box in step 3.

   Click **Merge**.

6. On the **Component Selection** page, do the following:

   a. In the **Cloud Platform** list, click the required cloud platform.
   b. Optional To use vault, turn on the **Secure Secrets** switch.
   c. To configure HashiCorp Vault, turn on the **Secure Secrets** switch.
   d. If you turn on the **Secure Secrets** switch, select a secret type.

      If you click **Kubernetes Native Secrets**, ensure that the name provided for **Secret Name** is same as you mentioned while creating a secret using the `documentum/config/vault_secret.yaml` file.

   e. By default, all components available in the `documentum/documentum-components.yaml` file along with the following mandatory components appears:

      - **Documentum CM Server**

- **Connection Broker**
- **OTDS**

Select or clear the check boxes for the relevant components based on your license and according to your requirement.

> **Important**
> You can select only either **DCM for Engineering** or **DCM for Life Sciences** along with other add-on components.

If you select add-on components, the configuration information related to the add-on components YAML file takes precedence instead of `documentum/values.yaml` and `documentum/config/configuration.yml` files.

f. Click **Next**.

7. On the **Container Images** page, by default, all values from the uploaded Helm chart related to container images, image tags, repository path, and so on are pre-populated for all the variables in the following category tabs:

   - **Global Variables**: Lists the global variables related to containers such as repository path, pull policy type, and so on.
   - **Public Components**: Lists the container image information for the third-party components.
   - **OpenText Components**: Lists the container image information such as image name and image version for the components that you enabled in the **Component Selection** page. This page also lists the subcomponents and init containers information.

   Review the existing values for all the variables across all categories. You can retain the existing values or modify them or provide new values for the required variables.

   The values from the existing Helm chart are pre-populated and those variable fields appears dimmed. To provide or modify the value for variables, move your mouse pointer over the field, click the edit icon, and then provide a new value or modify an existing value.

   For example, you can provide or modify the value for **Repository Path**, **Image Name**, **Image Version** and so on across the category tabs.

   > **Notes**
   > - Modifying the component name and its type is not supported.
   > - Click **Previous** to go the previous page.
   > - Click **Cancel** to go to the home page to restart the upgrade process.

8. Click **Next**.

- If you have provided the appropriate values for all the mandatory and other variables for your requirement, the **Deployment Configuration** page is displayed.
- If you have not provided the appropriate values for the mandatory variables, an error message is displayed.

  Click **Close** to see the error information icon displayed against the category name.

  Click the name of the category. The application highlights all the mandatory variables that need appropriate values. Provide the appropriate values and then click **Next**.

9. On the **Deployment Configuration** page, you can choose to use the default values for the variables according to your requirement for the enabled components or modify them.

   To modify the value, do one of the following:

   - Move your mouse pointer over the field, click the field or click the edit icon as relevant, and then modify the value.

     Note: If the variable field is from the new Helm chart, you can modify the value directly. If the variable field is from your Helm chart, the variable fields appears dimmed and you can click the edit icon to modify the value.
   - Search for a variable in the **Search** box or use the search filters to modify the value.
   - Use the find and replace option. Type the value of the variable in the **Search** box, click the toggle replace icon, provide the new value, and then click **Replace All**.
   - By default in 25.4, the **Proxy** switch is enabled when you upload a Helm chart for the first time or later too. You can choose to turn on or turn off the switch according to your requirement.

   Note: Click **Show all passwords** and **Hide all passwords** to view and hide all the passwords respectively.

10. Click **Next**.

    - If you provide the appropriate values for all the required variables, the **Chart Generation** page is displayed.
    - If you have not provided the appropriate values for the mandatory variables, an error message appears.

      Click **Close** to see the error information icon displayed against the component name.

      Click the name of the component. The application highlights all the mandatory variables that need appropriate values. Provide the appropriate values and then click **Next**.

11. On the **Chart Generation** page, click the name of the YAML file to review the updated values.

    To modify the values, click the edit icon of the YAML files to open the **Editor** page, modify the values directly or use the find and replace option, and then click **Save**.

    Note: Click **Previous** to go to the previous pages and make modification to the components and values.

    After reviewing the updated values, click **Generate Chart** and confirm to generate the Helm charts.

    Notes

    - Click **Generate Report** to generate the override report that lists the original values and the updated values for each YAML file.

      The password information is hidden in the report for security reasons.
    - Click **Exit Deployment** to go to the home page to restart the upgrade process.
    - Click **Close** to return to the **Chart Generation** page.

12. On the **Chart Generated** dialog box, click **Continue** to proceed with the upgrade process.

    If you did not opt for **Secure Secrets** in the **Component Selection** page, proceed to step 15.

13. Optional To configure Kubernetes native secrets, do the following:

    a. Create the Kubernetes secret before deploying, upgrading, or migrating as OpenText Documentum CM Helm charts depend on this secret.

       Create a Kubernetes secret using the `kubectl apply -f vault_secret.yaml` command.

    b. Provide the appropriate values for all variables to pass them to your templates.

       To provide the appropriate values, open the YAML files, and read the descriptions of the variables.

       Note: Ensure to update the passwords information for all the appropriate variables in the `documentum/config/vault_secret.yaml` file.

14. Click **Next**.

15. To upgrade OpenText Documentum CM at the command prompt, do the following:

    a. Provide the appropriate values for the **Release Name**, **Namespace**, and **Select Resource YAML file** variable fields.

Ensure that you use the same configuration in the resource YAML file used for the previous deployment.

b. Click **Generate Command** to generate the `helm upgrade` command.

c. Click **Copy** to copy the generated command.

d. Open the command prompt window and go to the updated Helm chart location.

e. Run the copied Helm upgrade command to complete the upgrade process.

16. Click **Done** and then click **Exit** to exit the application.

Note: Click **Cancel** only if you want to cancel all the changes and go to the home page to restart the upgrade process.

2.4 Post-upgrade task

2.4.1 Licensing OpenText Documentum CM

This section provides the information to license OpenText Documentum CM.

Note: If your existing deployment is already having the OpenText Documentum CM license, ignore this section.

2.4.1.1 Procuring license file from OpenText

Procure the license file from OpenText as described in OpenText Documentum Content Management License Management (https://support.opentext.com/csm?id=kb_article_view&sysparm_article=KB0834991).

2.4.1.2 Configuring OTDS and license

This section provides the information about configuring OTDS and license.

- For information about configuring OTDS using an automated method and configuring license using a manual method, see "To configure OTDS using an automated method and to configure license using a manual method: " on page 45.
- For information about configuring OTDS and license using a manual method, see "To configure OTDS and license using a manual method:" on page 49.

To use these instructions, ensure that you have updated the `documentum/charts/otds/charts/otdsws/config/config.yml` file to configure OTDS. If you have not updated the `config.yml` file, see "To configure OTDS and license using a manual method:" on page 49.

To configure OTDS using an automated method and to configure license using a manual method:

1. Sign in to the OTDS Admin website using the following information:

- URL: URL to access the OTDS Admin website.

  `<Ingress URL>/otds-admin`

- User name: Value for the OTDS Admin user name.

  For example: `otadmin@otds.admin`

- Password: Value provided for `otdsAdminPassword` in the `documentum/config/<passwords or passwords_vault or passwords_k8api>.yaml` file.

2. Synchronize the resources to the Documentum CM repository using the following steps:

   a. Click **Resources**.
   b. Click **<***your resource name***> > Actions > Consolidate**.

      For example: Your resource name can be `otdctmresource`.

3. Import the license file in OTDS using the following steps:

   a. On the **License Keys** page, click **Add**.
   b. On the **General** tab, provide values for the following fields:

      - **License Key Name**: Unique name.

        For example: `otdctmlicense`

      - **Resource ID**: License linked to the resource.

   c. On the **License Key** tab, click **Get License File**, browse and select the license file.
   d. Click **Save**.

4. Create a `businessadmin` user in the `otds.admin` partition using the following steps:

   a. Click **Partitions**.
   b. Click **otds.admin > Actions > View Members.**.
   c. Click **Add > New User**.
   d. On the **General** page, in the **User Name** box, type a name for this user as `businessadmin`.
   e. On the **Account** page, in the **Password Options** area, do the following:

      i. Click **Do not require password change on reset** from the list.
      ii. Clear the **User cannot change password** check box, if selected.
      iii. Select the **Password never expires** check box and click **Save**.

   ### Additional information and tasks

   Documentum CM client

   Create OTDS users with the user name as `d2_mail_manager` and `d2_wf_notification_user` and keep the passwords as blank in a new non-

synchronized partition which is not associated to any resources in OTDS.

By default, OTDS is enabled for client configuration. If you disable OTDS, add another user with the user name as `install_owner_user` to the same partition with the password as blank.

In the **Password Options** area, click **Do not require password change on reset** from the list, and do the following:

- Select the **User cannot change password** check box.
- Select the **Password never expires** check box.

This is applicable only for the system account user partition.

Workflow Designer

If you want to use Workflow Designer with the skip SSO feature, only a user with the `install_owner` privilege can access without a license.

Advanced Workflow

Advanced Workflow is available with advanced license or as an add-on. As a user, you can build processes using Advanced Workflow but to install these processes, the xDA Documentum CM repository endpoint user must have advanced Documentum CM or an add-on license.

To start a workflow at runtime from any Documentum CM client components, you must have advanced Documentum CM or an add-on license and the required transaction capability. The transaction counter increments at runtime when a user creates a new instance of workflow irrespective of the state of the workflow.

Reports

Create an OTDS user with the user name as `dctmreports` and keep the password as blank. Ensure that the OTDS user name is same as the name provided for `drServiceAccountUser` in the `documentum/config/<passwords or passwords_vault or passwords_k8api>.yaml` file.

In the **Password Options** area, click **Do not require password change on reset** from the list, and do the following:

- Select the **User cannot change password** check box.
- Select the **Password never expires** check box.

OpenText Documentum CM for Microsoft 365, OpenText Documentum CM Online Editing Service, and Notification Service

Create an OTDS user with the user name as `M365_SERVICE` with the password as `Xecmserviceuser@123`. The password is case-sensitive.

In the **Password Options** area, click **Do not require password change on reset** from the list, and do the following:

- Clear the **User cannot change password** check box.
- Select the **Password never expires** check box.

Documentum Archive Services for SAP Solutions

Create an OTDS user with the user name as `installownerusername` with the password as `installownerpassword`. This is a service account.

The system automatically changes the password after the service is started.

5. Add the `businessadmin` user to the `otdsbusinessadmins` group in OTDS using the following steps:
   a. Click **Users & Groups**, and select the **Groups** tab.
   b. In the **Search** box, type `otdsbusinessadmins` to find the `otdsbusinessadmins@otds.admin` group.
   c. On the **Groups** tab, click **Actions > Edit Membership**.
   d. On the **otdsbusinessadmins@otds.admin** page, on the **Members** tab, click **Add Member**.
   e. In the **Search** box, type `businessadmin` to find the `businessadmin@otds.admin` member.
   f. Select the **businessadmin@otds.admin** check box, and click **Add Selected**.
6. Run the following command in IAPI to create the `dm_otds_license_config` object in the Documentum CM Server pod:

```
create,c,dm_otds_license_config
set,c,l,otds_url
<otds_url_including_rest>
set,c,l,license_keyname
<license_keyname>
set,c,l,business_admin_name
<business_adminname>
set,c,l,business_admin_password
<password>
save,c,l
```

For example:

```
create,c,dm_otds_license_config
set,c,l,otds_url
http://documentumcm:8080/otdsws/rest
set,c,l,license_keyname
otdctmlicense
set,c,l,business_admin_name
businessadmin
set,c,l,business_admin_password
Password-1234567890
save,c,l
```

Notes

- Alternatively, you can create the `dm_otds_license_config` object using the Documentum Administrator graphical user interface.

  For instructions, see *OpenText Documentum Content Management - Administrator User Guide (EDCAC-UGD)*.

- If you modify the `dm_otds_license_config` object after the first time, you must run the `apply,c,NULL,FLUSH_OTDS_CONFIG` command in IAPI.

7. Allocate a license in OTDS to a partition using the following steps:

   a. Click **Partitions > <***your desired partition***> > Actions > Allocate to License**.

   b. On the **Allocate to License** page, click the relevant counter from the list.

      You can also allocate the license to users and groups. When you allocate a license, ensure that you allocate the relevant counter to a user or group.

   c. Click **Allocate to License**.

   Notes

   - If you modify the allocated license file after the initial deployment, you must restart OTDS.
   - You can allocate users to your license any time, but a user must exist before you can allocate the user to a license. The changes to the allocation, deallocation, and revocation take effect after 24 hours for releases prior to and including the 24.4 release.
     - For the 25.2 release, run the following command in IAPI to get the allocation, deallocation, revocation, or deletion with immediate effect:

       ```
       apply,c,NULL,FLUSH_JMS_OTDS_CACHE
       ```

     - From the 25.4 release, run the following command in IAPI to get the allocation, deallocation, revocation, or deletion with immediate effect:

       ```
       apply,c,NULL,FLUSH_OTDS_CACHE
       ```

8. On the Documentum CM Server pod, open the `otdsauth.properties` file in the `$DM_HOME\OTDSAuthLicenseHttpServerBin\config` folder and verify if information for `certificate`, `otds_rest_credential_url`, `otds_rest_ticket_url`, and `admin_username` are updated.

9. Sign in to a deployed application (for example, Documentum Administrator) with any licensed user to verify the license configuration.

   Ensure that the user authentication is successful.

To configure OTDS and license using a manual method:

1. Sign in to the OTDS Admin website using the following information:

   - URL: URL to access the OTDS Admin website.

     <Ingress URL>`/otds-admin`

   - User name: Value for the OTDS Admin user name.

     For example: `otadmin@otds.admin`

- Password: Value provided for `otdsAdminPassword` in the `documentum/config/<passwords or passwords_vault or passwords_k8api>.yaml` file.

2. Create a non-synchronized user partition in OTDS using the following steps:

   a. Click **Partitions > Add > New Nonsynchronized User Partition**.
   b. In the **Name** box, type a name for your user partition.

      For example: `otdctmpartitions`
   c. Click **Save**.

   For information to create a synchronized user partition in OTDS, see *OpenText Directory Services - Installation and Administration Guide (OTDS250400-IWC)*.

3. Create a synchronized resource in OTDS using the following steps:

   a. Click **Resources > Add**.
   b. On the **General** page, do the following:

      i. In the **Resource name** box, type a name for the resource.

         For example: `otdctmresource`
      ii. In the **Description** box, type a description for the resource.
      iii. On the **Synchronization** tab, select the **User and group synchronization** check box and click **REST(Generic)** for **Synchronization connector**.
      iv. Click **Next**.
   c. On the **Resources > <***your resource name***> > Actions** page, select **Properties**.

      For example: Your resource name can be `otdctmresource`.

      i. On the **<***your resource name***>** page, select the **Connection Information** tab.
      ii. On the **Connection Information** page, provide the value for the following fields:

         - **Base URL**: URL endpoint for user or group provisioning REST API.

           Use the following format:

           `http://<value provided for global.jmsServiceName in documentum/values.yaml>:<value provided for global.jmsPort in documentum/values.yaml>/dmotdsrest`

           For example: `http://dcs-pg-jms-service:9080/dmotdsrest`
         - **Username**: Installation owner name for the repository.

           Use the following format:

           `<repository name>\<installation owner name>`

           For example: `docbase1\dmadmin`
         - **Password**: Installation owner password for the repository.

d. Click **Test Connection**.
e. On the **User Attribute Mappings** tab, click **Reset to Default**.
f. For **User Attribute Mappings**, add the `client_capability` attribute with `Format` value of `2`.

   Add the `default_folder` attribute with `OTDS Attribute` value of `cn` and `Format` value of `/%s`.
g. Retain the default value for all other settings.
h. Click **Next**.
i. On the **Group Attribute Mappings** tab, click **Reset to Default**.
j. Click **Save**.

4. Click **Access Roles** and go to **Access to <***your resource name***>** (for example, `otdctmresource`).

   a. Click **Actions** and select **Include Groups**.
   b. Click **Actions** and select **View Access Role Details**.
   c. On the **User Partitions** tab, add the partition you created.
   d. Click **Save**.

5. Click **Resources > <***your resource name***> > Actions > Consolidate** to synchronize the members to your repository.

6. Click **OAuth clients** and create an OAuth client.

   a. For **Redirect URLs**, add the following to the **Redirect URLs** list:

      - `<Ingress URL>/D2/d2_otds.html`
      - `<Ingress URL>D2/OTDSLogoutResponse.html`
      - `<Ingress URL>/D2-Config/d2config_otds.html`
      - `<Ingress URL>/D2-Config/OTDSLogoutResponse.html`
      - `<Ingress URL>D2-Smartview/ui`
      - `<Ingress URL>oes-connector`
      - `<Ingress URL>AdminConsole`
      - `<Ingress URL>d2-rest`
      - `<Ingress URL>`

      Note: If you have disabled OTDS for client configuration, do not add the following URLs to the **Redirect URLs** list:

      - `<Ingress URL>/D2-Config/d2config_otds.html`.
      - `<Ingress URL>/D2-Config/OTDSLogoutResponse.html`.

   b. Click **Save**.

7. Click **Auth Handlers**, select **http.negotiate**, click **Action**, and select **Disable**.
8. Create roles using the following steps:
   a. Click **Partitions > <***your desired partition***> > Actions > View Members**.
   b. On the **Roles** tab, click **Add > New Role**.
   c. Add the following roles:
      - Client Capabilities: `Client_Consumer`, `Client_Contributor`, `Client_Coordinator`, and `Client_System_Administrator`.
      - User Privileges: `privilege_createtype`, `privilege_createcabinet`, `privilege_creategroup`, `privilege_sysadmin`, and `privilege_superuser`.

      Note: The role name is case-sensitive.
   d. Go to **Application Roles** to view the list of added roles.
9. Import the license file in OTDS using the following steps:
   a. On the **License Keys** page, click **Add**.
   b. On the **General** tab, provide values for the following fields:
      - **License Key Name**: Unique name.

        For example: `otdctmlicense`
      - **Resource ID**: License linked to the resource.
   c. On the **License Key** tab, click **Get License File**, browse and select the license file.
   d. Click **Save**.
10. Create a `businessadmin` user in the `otds.admin` partition using the following steps:
    a. Click **Partitions**.
    b. Click **otds.admin > Actions > View Members.**.
    c. Click **Add > New User**.
    d. On the **General** page, in the **User Name** box, type a name for this user as `businessadmin`.
    e. On the **Account** page, in the **Password Options** area, do the following:
       i. Click **Do not require password change on reset** from the list.
       ii. Clear the **User cannot change password** check box, if selected.
       iii. Select the **Password never expires** check box and click **Save**.

Additional information and tasks

Documentum CM client

Create OTDS users with the user name as `d2_mail_manager` and `d2_wf_notification_user` and keep the passwords as blank in a new non-synchronized partition which is not associated to any resources in OTDS.

By default, OTDS is enabled for client configuration. If you disable OTDS, you must add another user with the user name as `install_owner_user` to the same partition with the password as blank.

In the **Password Options** area, click **Do not require password change on reset** from the list, and do the following:

- Select the **User cannot change password** check box.
- Select the **Password never expires** check box.

This is applicable only for the system account user partition.

Workflow Designer

If you want to use Workflow Designer with the skip SSO feature, only a user with the `install_owner` privilege can access without a license.

Advanced Workflow

Advanced Workflow is available with advanced license or as an add-on. As a user, you can build processes using Advanced Workflow but to install these processes, the xDA Documentum CM repository endpoint user must have advanced Documentum CM or an add-on license.

To start a workflow at runtime from any Documentum CM client components, you must have advanced Documentum CM or an add-on license and the required transaction capability. The transaction counter increments at runtime when a user creates a new instance of workflow irrespective of the state of the workflow.

Reports

Create an OTDS user with the user name as `dctmreports` and keep the password as blank. Ensure that the OTDS user name is same as the name provided for `drServiceAccountUser` in the `documentum/config/<passwords or passwords_vault or passwords_k8api>.yaml` file.

In the **Password Options** area, click **Do not require password change on reset** from the list, and do the following:

- Select the **User cannot change password** check box.
- Select the **Password never expires** check box.

OpenText Documentum CM for Microsoft 365, OpenText Documentum CM Online Editing Service, and Notification Service

Create an OTDS user with the user name as `M365_SERVICE` with the password as `Xecmserviceuser@123`. The password is case-sensitive.

The password is updated when the notification service pod is up and running.

In the **Password Options** area, click **Do not require password change on reset** from the list, and do the following:

- Clear the **User cannot change password** check box.
- Select the **Password never expires** check box.

Documentum Archive Services for SAP Solutions

Create an OTDS user with the user name as `installownerusername` with the password as `installownerpassword`. This is a service account.

The system automatically changes the password after the service is started.

11. Add the `businessadmin` user to the `otdsbusinessadmins` group in OTDS using the following steps:
    a. Click **Users & Groups**, and select the **Groups** tab.
    b. In the **Search** box, type `otdsbusinessadmins` to find the `otdsbusinessadmins@otds.admin` group.
    c. On the **Groups**, click **Actions > Edit Membership**.
    d. On the **otdsbusinessadmins@otds.admin** page, on the **Members** tab, click **Add Member**.
    e. In the **Search** box, type `businessadmin` to find the `businessadmin@otds.admin` member.
    f. Select the **businessadmin@otds.admin** check box, and click **Add Selected**.
12. Run the following command in IAPI to create the `dm_otds_license_config` object in the Documentum CM Server pod:

```
create,c,dm_otds_license_config
set,c,l,otds_url
<otds_url_including_rest>
set,c,l,license_keyname
<license_keyname>
set,c,l,business_admin_name
<business_adminname>
set,c,l,business_admin_password
<password>
save,c,l
```

For example:

```
create,c,dm_otds_license_config
set,c,l,otds_url
http://documentumcm:8080/otdsws/rest
set,c,l,license_keyname
otdctmlicense
set,c,l,business_admin_name
```

```
businessadmin
set,c,l,business_admin_password
Password-1234567890
save,c,l
```

Notes

- Alternatively, you can create the `dm_otds_license_config` object using the Documentum Administrator graphical user interface.

  For instructions, see *OpenText Documentum Content Management - Administrator User Guide (EDCAC-UGD)*.

- If you modify the `dm_otds_license_config` object after the first time, you must run the `apply,c,NULL,FLUSH_OTDS_CONFIG` command in IAPI.

13. Allocate a license in OTDS to a partition using the following steps:

    a. Click **Partitions > <***your desired partition***> > Actions > Allocate to License**.

    b. On the **Allocate to License** page, click the relevant counter from the list.

    Note: You can also allocate the license to users and groups. When you allocate a license, ensure that you allocate the relevant counter to a user or group.

    c. Click **Allocate to License**.

    Notes

    - If you modify the allocated license file after the initial deployment, you must restart OTDS.
    - You can allocate users to your license any time, but a user must exist before you can allocate the user to a license. The changes to the allocation, deallocation, and revocation take effect after 24 hours for releases prior to and including the 24.4 release.

      – For the 25.2 release, run the following command in IAPI to get the allocation, deallocation, revocation, or deletion with immediate effect:

      ```
      apply,c,NULL,FLUSH_JMS_OTDS_CACHE
      ```

      – From the 25.4 release, run the following command in IAPI to get the allocation, deallocation, revocation, or deletion with immediate effect:

      ```
      apply,c,NULL,FLUSH_OTDS_CACHE
      ```

14. Import all the inline and Lightweight Directory Access Protocol (LDAP) users to OTDS:

    - On Windows:

      On the Documentum CM Server pod, copy the `dfc.properties` file to the `$DOCUMENTUM\Shared` folder. At the command prompt, go to the `$DOCUMENTUM\Shared` folder, run the following command:

```
java --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/
java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-
opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/
sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/
sun.security.provider=ALL-UNNAMED --add-exports=java.base/
sun.security.pkcs=ALL-UNNAMED --add-exports=java.base/sun.security.x509=ALL-
UNNAMED --add-exports=java.base/sun.security.util=ALL-UNNAMED --add-
exports=java.base/sun.security.tools.keytool=ALL-UNNAMED -
cp .;dfc.jar;dfc.properties;* com.documentum.fc.tools.MigrateInlineUsersToOtds
<repository name> <installation owner name> <installation owner password> <non-
synchronized partition name>
```

- On Linux:

  On the Documentum CM Server pod, copy the `dfc.properties` file to the `$DOCUMENTUM\dfc` folder. At the command prompt, go to the `$DOCUMENTUM\dfc` folder, run the following command:

```
java --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/
java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-
opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/
sun.rmi.transport=ALL-UNNAMED --add-exports=java.base/
sun.security.provider=ALL-UNNAMED --add-exports=java.base/
sun.security.pkcs=ALL-UNNAMED --add-exports=java.base/sun.security.x509=ALL-
UNNAMED --add-exports=java.base/sun.security.util=ALL-UNNAMED --add-
exports=java.base/sun.security.tools.keytool=ALL-UNNAMED -
cp .:dfc.jar:dfc.properties:* com.documentum.fc.tools.MigrateInlineUsersToOtds
<repository name> <installation owner name> <installation owner password> <non-
synchronized partition name>
```

15. On the Documentum CM Server pod, open the `otdsauth.properties` file in the `$DM_HOME\OTDSAuthLicenseHttpServerBin\config` folder and verify if information for `certificate`, `otds_rest_credential_url`, `otds_rest_ticket_url`, and `admin_username` are updated.
16. To verify the license configuration, sign in to any deployed application (for example, Documentum Administrator) with any licensed user.

    Ensure that user authentication is successful.

2.4.1.3 Creating new users, allocating license, and applying roles in OTDS

1. Go to **Partitions > <*partition name*> > Actions > View Members > Add > New user**.
2. Create a new user with a desired name (for example, `otdctmuser`), and set all the attributes including password.
3. Click **Save**.
4. Select the user you created and click **Actions > Allocate to License**.
5. On the **Allocate to License** dialog box, select the relevant counter and click **Allocate to License**.
6. Select the user you created and click **Actions > Edit Application Roles**.
7. Click **Assign Roles** and select a desired role for the user.

   For more information about the list of roles, see step 8.c.

Note: Ensure `da_privilege_enabled=T` and `lss_cc_enabled=T` are added in the `<Java Method Server Home>/webapps/dmotdsrest/WEB-INF/classes/dmotds.properties` file in the Documentum CM Server pods.

8. Click **Add Selected** and then click **Close**.

2.4.1.4 Troubleshooting license configuration

The following table describes the license-related errors captured in the `$DOCUMENTUM/dba/log/otdsauth.log` file.

Error code	Description	Solution
`DM_LICENSE_E_NO_LICENSE_CONFIG`	The license configuration is not created in the repository.	Ensure that you activate the license from IAPI or Documentum Administrator.
`DM_LICENSE_E_CONFIG_MISSING_PARAMS`	One of the following mandatory parameters is not available: OTDS URL, or OTDS business administrator credentials, or License key	Ensure that you have provided valid values in IAPI or Documentum Administrator.
`DM_LICENSE_E_SERVER_NOTREACHABLE`	The OTDS URL is not reachable.	Ensure that the OTDS deployment is active and reachable.
`DM_LICENSE_E_BAD_ADMIN_CRED`	OTDS business administrator credentials are incorrect or locked.	Before you activate the license, ensure that you have provided valid values in IAPI or Documentum Administrator.
`DM_LICENSE_E_REQ_BUSINESS_ADMIN`	The OTDS business administrator user is not added to the `otdsbusinessadmins` group in OTDS.	Ensure that you add the business administrator user to the `otdsbusinessadmins` group in OTDS.
`DM_LICENSE_E_NO_LICENSE`	The license key is not configured with the license file in OTDS.	Ensure that you upload the license file in OTDS to generate the license key and provide the correct license key.
`DM_LICENSE_E_INVALID_LICENSE`	The license file is invalid.	Ensure that you have uploaded a valid license file in OTDS.
`DM_LICENSE_E_USER_NOT_FOUND_OR_DUPLICATE`	The user is not found in OTDS or the user is not unique in OTDS.	Ensure that the user exists in OTDS and is unique.
`DM_LICENSE_E_USER_NO_LICENSE_ALLOCATED`	The user is not allocated with a counter.	Ensure that you allocate a counter to the user in OTDS.

Error code	Description	Solution
DM_LICENSE_E_USER_NO_ACCESS	The user is not allocated to the relevant counter or all the licenses for users or transactions are consumed.	Ensure that you allocate the user to the relevant counter in OTDS.
DM_LICENSE_E_UNEXPECTED_ERROR	There is an unexpected error in the licensing code.	Analyze the otdsauth.log file to troubleshoot the issue.
DM_LICENSE_E_SYSTEMACCT_MISUSE	When a system account user is allocated to any other counters.	Remove the SYSTEMACCT or other counters.
DM_LICENSE_E_SERVICEACCT_MISUSE	When a service account user is allocated to any other counters.	Remove the SERVICEACCT or other counters.

Chapter 3

Migrating on-premises OpenText Documentum CM to cloud platforms

This documentation provides the information about migrating OpenText Documentum CM from the on-premises environment including VM-based environment to cloud platforms.

3.1 Overview of migration sequence

This section provides the high-level sequence to migrate OpenText Documentum CM from versions 24.4 or 25.2 or 25.4 in on-premises environment to the 24.4 or 25.2 or 25.4 cloud platforms.

Run the pre-migration report generator utility to assess the migration requirements, update Helm with the original repository ID and name, restore the database and file stores to the Kubernetes storage, and AEK key for encryption.

Run scripts on the database to update configurations for High-Availability Documentum CM Server, Java Method Server, and Accelerated Content Services and then validate the new containerized environment to ensure successful migration.

3.2 Using the pre-migration report generator utility

The `Documentum Migration Report Generator` utility is a Java-based tool that runs DQL using Foundation Java API to produce migration-ready reports for on-premises OpenText Documentum CM. It discovers the repositories dynamically, allows users to select one or more repositories, handles credentials securely on Windows, and works using the cross-platform scripts.

The reports include detailed Microsoft Excel workbooks (summary, errors, per-query sheets, and so on), per-repository PDF summary, and a consolidated PDF for all repositories.

To generate reports using the pre-migration report generator utility:

1. Download the `Documentum CM - Utilities <version>.zip` and `Documentum CM - Utilities <version>.tar.gz` files from My Support and extract them to a temporary location.
2. From the temporary location, extract the contents of the `Documentum_Cloud_Accelerator_windows-<version>.zip` and `Documentum_Cloud_Accelerator_linux-<version>.tar.gz` files.
3. Extract the contents of the `documentum-migration-report-<version>.zip` and `documentum-migration-report-<version>.tar.gz` files.

4. Ensure that you have the `dfc.properties` and `dfc.keystore` files in the `DFC_CONFIG_PATH` environment variable's path.
5. On Windows, at the command prompt, go the extracted location and run `run_migration_report.bat`.

   On Linux, at the terminal window, go to the extracted location and run `run_migration_report.sh`.
6. Read the `README.MD` file for detailed usage instructions.

   If you select all repositories, the script generates the report as a consolidated PDF. However, if you select a specific repository, the utility generates the report in both the Microsoft Excel and PDF formats.

Notes

- You can customize the built-in queries through the `queries.properties` file. If you customize the `queries.properties` file, ensure that you place the `queries.properties` file in the `DFC_CONFIG_PATH` environment variable's path.
- Only the SELECT statements are supported. The UPDATE, DELETE and other statements are not supported. In addition, nested statements within SELECT such as UPDATE, DELETE and so on are not supported.

3.3 Prerequisites

This section provides the prerequisites information to migrate OpenText Documentum CM from the versions 24.4 or 25.2 or 25.4 in on-premises environment to the 24.4 or 25.2 or 25.4 cloud platforms.

- Stop the repository.
- Take a backup of the database with your desired tools.

Caution

- Migration is supported only from the PostgreSQL database from on-premises environment to the PostgreSQL database in cloud platform.
- Migration is supported only from the Oracle database from on-premises environment to the Oracle database in cloud platform.

- Take a backup of the `dba`, `config`, and the `data` folders in the existing 24.4 or 25.2 or 25.4 Documentum CM Server.
- Download the 24.4 or 25.2 or 25.4 versions of Helm charts from OpenText Container Registry, according to your requirement, and extract the contents to a temporary location.
- Perform the migration in the following sequence:

1. Migrate from the 24.4 on-premises environment to the 24.4 cloud platform or from the 25.2 on-premises environment to the 25.2 cloud platform or from the 25.4 on-premises environment to the 25.4 cloud platform.

   For instructions to migrate *from* and *to the same version*, perform all the tasks as described in "Migrating OpenText Documentum CM from 24.4 or 25.2 or 25.4 on-premises to 24.4 or 25.2 or 25.4 cloud platform" on page 61.

2. Upgrade from 24.4 to 25.4 or 25.2 to 25.4.

   For instructions to upgrade after the successful migration, perform all the tasks as described in "Upgrading OpenText Documentum CM on cloud platforms" on page 7.

Note: If you are migrating from the 25.4 on-premises environment to the 25.4 cloud platform, the upgrade step is not required.

3.4 Migrating OpenText Documentum CM from 24.4 or 25.2 or 25.4 on-premises to 24.4 or 25.2 or 25.4 cloud platform

This section provides the information about migrating OpenText Documentum CM from 24.4 or 25.2 or 25.4 on-premises to 24.4 or 25.2 or 25.4 cloud platform.

The following migration paths are supported:

- Migration from the 24.4 on-premises environment to 24.4 cloud platform.
- Migration from the 25.2 on-premises environment to 25.2 cloud platform.
- Migration from the 25.4 on-premises environment to 25.4 cloud platform.

To migrate OpenText Documentum CM from one version of on-premises environment including VM-based environment to same version of cloud platform:

1. In the `documentum/values.yaml` file, do the following:

   a. Set the value of `docbase` to the same name you provided for the repository in your on-premises environment.
   b. Set the value of `installOwnerUsername` to the same name you provided for the installation owner in your on-premises environment.
   c. Set appropriate values for all Ingress-related variables according to your requirement.
   d. Set values for all other variables according to your requirement.

   For more information about the list of mandatory variables for 24.4, 25.2, and 25.4, see *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD240400-IGD)*, *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250200-IGD)*, and *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)* respectively.

For more information about the description of variables to provide appropriate values, see the respective YAML files.

2. In the `documentum/configuration.yml` file, do the following:

   a. Set the value of `content-server.docbase.id` to the same ID you provided for the repository in your on-premises environment.

   b. Set the value of `content-server.docbase.existing` to `true`.

   c. Set the value of `content-server.docbase.index` to the same name you provided for the repository in your on-premises environment.

   d. Set the value of `content-server.contentserver.aek.name` to the same name you provided in your on-premises environment.

   e. Set the value of `content-server.persistentVolume.createPVC` to `false`.

   f. Set appropriate values for all Ingress-related variables according to your requirement.

   g. Set values for all other variables according to your requirement.

   For more information about the list of mandatory variables for 24.4, 25.2, and 25.4, see *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD240400-IGD)*, *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250200-IGD)*, and *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)* respectively.

   For more information about the description of variables to provide appropriate values, see the respective YAML files.

3. In the `documentum/passwords.yaml` file, do the following:

   

   Caution

   Migration is supported only from non-vault on-premises environment to non-vault cloud platform.

   Important

   Ensure that the password complies with the password complexity rules.

   a. Set the value of `installOwnerPassword` to the same password you provided in your on-premises environment.

   b. Set values for all other variables according to your requirement.

   For more information about the list of mandatory variables for 24.4, 25.2, and 25.4, see *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD240400-IGD)*, *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250200-IGD)*, and *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)* respectively.

   For more information about the description of variables to provide appropriate values, see the respective YAML files.

4. In the `documentum/dockerimages-values.yaml` file, set values for all variables according to your requirement.

   For more information about the list of mandatory variables for 24.4, 25.2, and 25.4, see *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD240400-IGD)*, *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250200-IGD)*, and *OpenText Documentum Content Management - Cloud Deployment Guide (EDCSYCD250400-IGD)* respectively.

   For more information about the description of variables to provide appropriate values, see the respective YAML files.

5. In the `documentum/documentum-components.yaml` file, set the value of `cs-secrets.enabled` to `true`, set the value of all other components to `false`, and run the following command:

   ```
   helm install <release name> <location where Helm charts are extracted> --values
   <location where Helm charts are extracted>/config/configuration.yml --values
   <location where Helm charts are extracted>/config/constants.yaml --values <location
   where Helm charts are extracted>/config/passwords.yaml --values <location where
   Helm charts are extracted>/platforms/<cloud platform>.yaml --values <location where
   Helm charts are extracted>/dockerimages-values.yaml --values <location where Helm
   charts are extracted>/documentum-resources-values-test-small.yaml --values
   <location where Helm charts are extracted>/documentum-components.yaml --namespace
   <name of namespace>
   ```

   where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

   The available resource YAML files contain pod sizing values, including CPU and memory.

   Example 3-1: Migrating cs-secrets

   ```
   helm install cs-secrets-mig /opt/temp/documentum --values /opt/temp/documentum/
   config/configuration.yml --values /opt/temp/documentum/config/constants.yaml --
   values /opt/temp/documentum/config/passwords.yaml --values /opt/temp/documentum/
   platforms/gcp.yaml --values /opt/temp/documentum/dockerimages-values.yaml --
   values /opt/temp/documentum/documentum-resources-values-test-small.yaml --
   values /opt/temp/documentum/documentum-components.yaml --namespace docu
   ```

6. In the `documentum/documentum-components.yaml` file, set the value of `db.enabled` to `true`, set the value of all other components to `false`, and run the following command:

   ```
   helm install <release name> <location where Helm charts are extracted> --values
   <location where Helm charts are extracted>/config/configuration.yml --values
   <location where Helm charts are extracted>/config/constants.yaml --values <location
   where Helm charts are extracted>/config/passwords.yaml --values <location where
   Helm charts are extracted>/platforms/<cloud platform>.yaml --values <location where
   Helm charts are extracted>/dockerimages-values.yaml --values <location where Helm
   charts are extracted>/documentum-resources-values-test-small.yaml --values
   <location where Helm charts are extracted>/documentum-components.yaml --namespace
   <name of namespace>
   ```

   where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

   The available resource YAML files contain pod sizing values, including CPU and memory.

Example 3-2: Migrating database

```
helm install db-mig /opt/temp/documentum --values /opt/temp/documentum/config/
configuration.yml --values /opt/temp/documentum/config/constants.yaml --values /opt/
temp/documentum/config/passwords.yaml --values /opt/temp/documentum/platforms/
gcp.yaml --values /opt/temp/documentum/dockerimages-values.yaml --values /opt/temp/
documentum/documentum-resources-values-test-small.yaml --values /opt/temp/
documentum/documentum-components.yaml --namespace docu
```

7. To use the existing PostgreSQL database in your cloud platform, do the following:

   a. Sign in as a `postgres` user.

   Example 3-3: Sign in as postgres user

   ```
   #su postgres
   ```

   b. Run the following command to create new user with an encrypted password:

   ```
   # psql
   # create role <name of user> noinherit login password '<password to
   authenticate user>';
   ```

   Example 3-4: Create new PostgreSQL user with an encrypted password

   ```
   # psql
   # create role testenv noinherit login password '024$2356*651';
   ```

   c. Run the following command to create new database and grant the permissions:

   ```
   # psql
   # create database dm_<repository name>_docbase with encoding='UTF8'
   LC_COLLATE='C' LC_CTYPE='C' CONNECTION LIMIT=-1 owner <repository name>
   TEMPLATE template0;
   # ALTER DATABASE dm_<repository name>_docbase SET SEARCH_PATH to <repository
   name>;
   # GRANT ALL ON database dm_<repository name>_docbase to <repository name>;
   ```

   Example 3-5: Create new PostgreSQL database and grant permissions

   ```
   # psql
   # CREATE DATABASE dm_testenv_docbase WITH ENCODING='UTF8' LC_COLLATE='C'
   LC_CTYPE='C' CONNECTION LIMIT=-1 OWNER testenv TEMPLATE template0;
   # ALTER DATABASE dm_testenv_docbase SET SEARCH_PATH to testenv;
   # GRANT ALL ON DATABASE dm_testenv_docbase to testenv;
   ```

   d. Run the following command to verify the path mentioned in the `file_system_path` column from the `dm_location_s` table:

```
SELECT * FROM dm_location_s;
```

If you encounter an invalid object path error, update the path in the `file_system_path` column from the Windows format, `C:\<DOCUMENTUM_HOME>\`, to the Linux format, `/<DOCUMENTUM_HOME>`. For example, `/opt/dctm/`.

Note: Ensure to change all backward slash (`\`) to the forward slash (`/`).

e. Run the following `UPDATE` command:

```
# UPDATE dm_location_s SET file_system_path='/opt/dctm/dba/log' where
r_object_id='<dm_location_s>';
```

8. To use the existing Oracle database in your cloud platform, do the following:

a. Sign in as an `oracle` user.

Example 3-6: Sign in as oracle user

```
su oracle
```

b. Run the following command to create new user with an encrypted password:

```
sqlplus /nolog
connect /as sysdba
CREATE USER <name of user> IDENTIFIED BY '<password to authenticate user>;
```

Example 3-7: Create new Oracle user with an encrypted password

```
CREATE USER docbase1 IDENTIFIED BY 024$2356*651;
```

c. Run the following command to create new database and grant the permissions:

```
CREATE TABLESPACE DM_<name of user>_docbase DATAFILE '/u01/app/oracle/oradata/
ORA19C/dm_<name of user>20220322104051_db.dbf' SIZE 180M REUSE;
ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/ORA19C/dm_<name of
user>20220322104051_db.dbf' AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
GRANT CREATE SESSION TO <name of user> IDENTIFIED BY "<password to
authenticate user>";
GRANT CREATE SYNONYM TO <name of user>;
GRANT CREATE VIEW TO <name of user>;
GRANT CREATE ANY TABLE, ALTER ANY TABLE, DROP ANY TABLE, LOCK ANY TABLE,
SELECT ANY TABLE TO <name of user>;
GRANT CREATE ANY TRIGGER TO <name of user>;
GRANT CREATE ANY INDEX TO <name of user>;
GRANT CREATE SEQUENCE TO <name of user>;
GRANT CREATE PROCEDURE TO <name of user>;
GRANT SELECT_CATALOG_ROLE TO <name of user>;
GRANT EXECUTE_CATALOG_ROLE TO <name of user>;
ALTER USER <name of user> DEFAULT TABLESPACE DM_<name of user>_docbase
TEMPORARY TABLESPACE TEMP;
CREATE TABLESPACE DM_<name of user>_index DATAFILE '/u01/app/oracle/oradata/
ORA19C/dm_<name of user>20220322104050_ind.dbf' SIZE 180M REUSE;
ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/ORA19C/dm_<name of
user>20220322104050_ind.dbf' AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
```

```
ALTER USER <name of user> quota unlimited on DM_<name of user>_docbase;
ALTER USER <name of user> quota unlimited on DM_<name of user>_index;
```

Example 3-8: Create new Oracle database and grant permissions

```
CREATE TABLESPACE DM_docbase1_docbase DATAFILE '/u01/app/oracle/oradata/ORA19C/
dm_docbase120220322104051_db.dbf' SIZE 180M REUSE;
ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/ORA19C/
dm_docbase120220322104051_db.dbf' AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
GRANT CREATE SESSION TO docbase1 IDENTIFIED BY "024$2356*651";
GRANT CREATE SYNONYM TO docbase1;
GRANT CREATE VIEW TO docbase1;
GRANT CREATE ANY TABLE, ALTER ANY TABLE, DROP ANY TABLE, LOCK ANY TABLE,
SELECT ANY TABLE TO docbase1;
GRANT CREATE ANY TRIGGER TO docbase1;
GRANT CREATE ANY INDEX TO docbase1;
GRANT CREATE SEQUENCE TO docbase1;
GRANT CREATE PROCEDURE TO docbase1;
GRANT SELECT_CATALOG_ROLE TO docbase1;
GRANT EXECUTE_CATALOG_ROLE TO docbase1;
ALTER USER docbase1 DEFAULT TABLESPACE DM_docbase1_docbase TEMPORARY
TABLESPACE TEMP;
CREATE TABLESPACE DM_docbase1_index DATAFILE '/u01/app/oracle/oradata/ORA19C/
dm_docbase120220322104050_ind.dbf' SIZE 180M REUSE;
ALTER DATABASE DATAFILE '/u01/app/oracle/oradata/ORA19C/
dm_docbase120220322104050_ind.dbf' AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
ALTER USER docbase1 quota unlimited on DM_docbase1_docbase;
ALTER USER docbase1 quota unlimited on DM_docbase1_index;
```

d. Run the following command to verify the path mentioned in the `file_system_path` column from the `dm_location_s` table:

```
SELECT * FROM dm_location_s;
```

If you encounter an invalid object path error, update the path in the `file_system_path` column from the Windows format, `C:\<DOCUMENTUM_HOME>\`, to the Linux format, `/<DOCUMENTUM_HOME>`. For example, `/opt/dctm/`.

Note: Ensure to change all backward slash (`\`) to the forward slash (`/`).

e. Run the following `UPDATE` command:

```
# UPDATE dm_location_s SET file_system_path='/opt/dctm/dba/log' where
r_object_id='<dm_location_s>';
```

9. Import the database from step 7 or step 8 to your cloud platform. You can use any third-party tools of your choice to import the database. Ensure that all the tables are migrated successfully.

10. In the `documentum/documentum-components.yaml` file, set the value of `docbroker.enabled` to `true`, set the value of all other components to `false`, and run the following command:

```
helm install <release name> <location where Helm charts are extracted> --values
<location where Helm charts are extracted>/config/configuration.yml --values
<location where Helm charts are extracted>/config/constants.yaml --values <location
where Helm charts are extracted>/config/passwords.yaml --values <location where
Helm charts are extracted>/platforms/<cloud platform>.yaml --values <location where
Helm charts are extracted>/dockerimages-values.yaml --values <location where Helm
charts are extracted>/documentum-resources-values-test-small.yaml --values
<location where Helm charts are extracted>/documentum-components.yaml --namespace
<name of namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

Example 3-9: Migrating connection broker

```
helm install dbr-mig /opt/temp/documentum --values /opt/temp/documentum/config/
configuration.yml --values /opt/temp/documentum/config/constants.yaml --values /opt/
temp/documentum/config/passwords.yaml --values /opt/temp/documentum/platforms/
gcp.yaml --values /opt/temp/documentum/dockerimages-values.yaml --values /opt/temp/
documentum/documentum-resources-values-test-small.yaml --values /opt/temp/
documentum/documentum-components.yaml --namespace docu
```

11. (Applicable only for migrating 24.4 on-premises environment to 24.4 cloud platform) In the `documentum/documentum-components.yaml` file, set the value of `cs-logging-configMap.enabled` to `true`, set the value of all other components to `false`, and run the following command:

```
helm install <release name> <location where Helm charts are extracted> --values
<location where Helm charts are extracted>/config/configuration.yml --values
<location where Helm charts are extracted>/config/constants.yaml --values <location
where Helm charts are extracted>/config/passwords.yaml --values <location where
Helm charts are extracted>/platforms/<cloud platform>.yaml --values <location where
Helm charts are extracted>/dockerimages-values.yaml --values <location where Helm
charts are extracted>/documentum-resources-values-test-small.yaml --values
<location where Helm charts are extracted>/documentum-components.yaml --namespace
<name of namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

Example 3-10: Migrating cs-logging-configMap

```
helm install cs-logging-configmap-mig /opt/temp/documentum --values /opt/temp/
documentum/config/configuration.yml --values /opt/temp/documentum/config/
constants.yaml --values /opt/temp/documentum/config/passwords.yaml --values /opt/
temp/documentum/platforms/gcp.yaml --values /opt/temp/documentum/dockerimages-
values.yaml --values /opt/temp/documentum/documentum-resources-values-test-
small.yaml --values /opt/temp/documentum/documentum-components.yaml --namespace docu
```

12. (Applicable only for migrating 24.4 on-premises environment to 24.4 cloud platform) In the `documentum/documentum-components.yaml` file, set the value of `cs-dfc-properties.enabled` to `true`, set the value of all other components to `false`, and run the following command:

```
helm install <release name> <location where Helm charts are extracted> --values
<location where Helm charts are extracted>/config/configuration.yml --values
<location where Helm charts are extracted>/documentum/config/constants.yaml --
values <location where Helm charts are extracted>/documentum/config/passwords.yaml
--values <location where Helm charts are extracted>/documentum/platforms/<cloud
platform>.yaml --values <location where Helm charts are extracted>/documentum/
dockerimages-values.yaml --values <location where Helm charts are extracted>/
documentum/documentum-resources-values-test-small.yaml --values <location where
```

```
Helm charts are extracted>/documentum/documentum-components.yaml --namespace <name
of namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

Example 3-11: Migrating cs-dfc-properties

```
helm install cs-dfc-properties-mig /opt/temp/documentum --values /opt/temp/
documentum/config/configuration.yml --values /opt/temp/documentum/config/
constants.yaml --values /opt/temp/documentum/config/passwords.yaml --values /opt/
temp/documentum/platforms/gcp.yaml --values /opt/temp/documentum/dockerimages-
values.yaml --values /opt/temp/documentum/documentum-resources-values-test-
small.yaml --values /opt/temp/documentum/documentum-components.yaml --namespace docu
```

13. To create the PVC pod, do the following:

Note: OpenText recommends that you have the local copy of the file or folder and run the command from the file or folder location, where applicable.

a. Create PVC for `data`, `dba`, and `config` using the following sample YAML file:

```
#Migrating PVC
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  #name: csServiceName-pvc
  name: dcs-pg-pvc
spec:
  accessModes:
    - ReadWriteMany
  storageClassName: trident-nfs
  resources:
    requests:
      storage: 3Gi
---
##Dummy pod to copy the configs and data
apiVersion: v1
kind: Pod
metadata:
  name: migrationcs #can be changed with a precise name like tempPod
spec:
  containers:
  - name: migrationcs
    image: alpine:latest
    command:
    - sleep
    - "999999"
    volumeMounts:
        - mountPath: /opt/dbaconfig
          name: migpvc
          #subPath: dba_mig/csServiceName
          subPath: dba_mig/dcs-pg
        - mountPath: /opt/migdata
          name: migpvc
          #subPath: data/csServiceName
          subPath: data/dcs-pg
  volumes:
```

```
  - name: migpvc
    persistentVolumeClaim:
      #claimName: csServiceName-pvc
      claimName: dcs-pg-pvc
```

Note: Replace `csServiceName` with the Documentum CM Server service name used while creating the Documentum CM Server pod.

b. Run the following command to create PVC with mounted paths for `data`, `dba`, and `config` using the `pvc.yaml` created in step 13.a.

c. Run the following command to copy the `dba` folder from your on-premises environment to the `migrationcs` container located at `/opt/dbaconfig/`:

```
# kubectl cp C:\helm\cs254\install\cs254_115_mig\mig\dba migrationcs:/opt/
dbaconfig -n <name of namespace>
```

d. Run the following command to copy the `config` folder from your on-premises environment to the `migrationcs` container located at `/opt/dbaconfig/`:

```
# kubectl cp C:\helm\cs254\install\cs254_115_mig\mig\config migrationcs:/opt/
dbaconfig -n <name of namespace>
```

e. Run the following command to copy the repository folder, `data/<repository name>`, from your on-premises environment to the `migrationcs` container located at `/opt/migdata/`:

```
# kubectl cp C:\helm\cs254\install\cs254_115_mig\mig\data\<repository
name> migrationcs:/opt/migdata -n <name of namespace>
```

f. Run the following command to connect to `migrationcs` pod you created:

```
# kubectl exec -it migrationcs -- sh
```

Note: The owner of the `migdata` and `dbaconfig` folders should be same as the installation owner name.

Otherwise, run the following command to change the owner:

```
# chown -R <installation owner name>:<installation owner name> <name of the
folder>
```

If you encounter the `Unknown user/group <installation owner name>:<installation owner name>` error, run the following command to add the user:

```
# adduser <installation owner name>
```

After the user is added, ensure that you change the owner.

g. Optional Run the following command to change the permission of the `migdata` and `dbaconfig` folders to `777`:

```
# chmod -R 777 <name of the folder>
```

14. In the `documentum/charts/content-server/values.yaml` file, do the following:

a. Set the value of `migration.migratecs` to `true`.

b. Set the value of `migration.oldhostname` to the same host name in your on-premises environment.

c. Set the value of `migration.oldDocumentumHome` to the same path of Documentum CM Server in your on-premises environment.

d. (Only if you want to migrate from Windows) Set the value of `migration.migfromwindows` to `true`.

15. In the `documentum/documentum-components.yaml` file, do the following:

a. Set the value of `content-server.enabled` to `true`.

b. Set the value of other components also such as `d2config`, `d2classic`, `records`, and so on, according to your requirement, to `true`.

c. Set the value of all other components to `false`.

16. In the `documentum/dockerimages-values.yaml` file, enable all the init containers for the components that you set to `true` in step 15.b.

17. Run the following command to migrate Documentum CM Server:

```
helm install <release name> <location where Helm charts are extracted> --values
<location where Helm charts are extracted>/config/configuration.yml --values
<location where Helm charts are extracted>/config/constants.yaml --values <location
where Helm charts are extracted>/config/passwords.yaml --values <location where
Helm charts are extracted>/platforms/<cloud platform>.yaml --values <location where
Helm charts are extracted>/dockerimages-values.yaml --values <location where Helm
charts are extracted>/documentum-resources-values-test-small.yaml --values
<location where Helm charts are extracted>/documentum-components.yaml --namespace
<name of namespace>
```

where `<config>` can be `extra-large`, `large`, `medium`, `medium-large`, `small`, `small-medium`, or `test-small` resource value YAML files.

The available resource YAML files contain pod sizing values, including CPU and memory.

Example 3-12: Migrating Documentum CM Server

```
helm install dcm-server-mig /opt/temp/documentum --values /opt/temp/documentum/
config/configuration.yml --values /opt/temp/documentum/config/constants.yaml --
values /opt/temp/documentum/config/passwords.yaml --values /opt/temp/documentum/
platforms/gcp.yaml --values /opt/temp/documentum/dockerimages-values.yaml --
values /opt/temp/documentum/documentum-resources-values-test-small.yaml --
values /opt/temp/documentum/documentum-components.yaml --namespace docu
```

18. If Transformation Services is already installed and configured in your on-premises environment, do the following:

a. Remove the Transformation Services configuration.

b. Uninstall Transformation Services.

c. Reinstall Transformation Services.

d. Reconfigure Transformation Services.

For more information, see *OpenText Documentum Content Management - Transformation Services Installation Guide (EDCCT250400-IGD)*.

19. If Documentum xPlore is already installed and configured in your on-premises environment, do the following:

a. Create a new administrator account in your on-premises environment as on-premises and cloud deployment have different default administrator accounts.
b. Make sure that you have sufficient space in `<XPLORE_HOME>/rtdata/` for configuration and data file backup on cloud.
c. Deploy the Documentum xPlore pod.

   For more information, see *OpenText Documentum xPlore - Cloud Deployment Guide (EDCSRCCD220100-CGD)*.
d. Sign in to `http://<xPlore host machine IP address>:9300/dsearchadmin` to access your on-premises environment.
e. Go to **Home > Administration**.
f. Configure an administrator user with user name as `admin` and password as `password` to authenticate `dsearchadmin` on cloud.
g. In the `documentum/charts/xPlore-OneD/values.yaml` file, remove the comment from the `PASSWORD` variable in the `indexserver.extraEnv` section, and modify the default value `password` to the Index Server password provided in your on-premises environment while installing Documentum xPlore.
h. In the `documentum/charts/xPlore-OneD/values.yaml` file, modify the default value for the `docbaseUser` variable in the `indexagent` section to the repository user name provided in your on-premises environment while installing Documentum xPlore.
i. To copy data from your on-premises environment, shut down the following services:

   - `Documentum xPlore Watchdog`
   - `Documentum IndexAgent`
   - `Documentum xPlore PrimaryDSearch`

j. To copy the data and configuration from your on-premises environment, do the following:

   i. Copy the following folders to a backup folder for cloud configuration:

      - `<XPLORE_HOME>/Data/`
      - `<XPLORE_HOME>/Config/`
      - `<XPLORE_HOME>/Config/wal/`

   ii. Create a TAR or ZIP version of the folders from step 19.j.i.
   iii. Create a folder named `temp` in `<XPLORE_HOME>/rtdata/` in the Index Server pod and transfer the TAR or ZIP version of the folders created in step 19.j.ii.
   iv. To update the Index Server statefulset, do the following:

      A. Run the following command:

      ```
      kubectl edit sts indexserver
      ```

B. Take a copy of the Index Server statefulset file to a temporary location.

C. Remove the contents of the `livenessProbe` and `readinessProbe` sections.

D. Save and close the Index Server statefulset file.

v. To update the Index Agent statefulset, do the following:

A. Run the following command:

```
kubectl edit sts indexagent
```

B. Take a copy of the Index Agent statefulset file to a temporary location.

C. Remove the contents of the `livenessProbe` and `readinessProbe` sections.

D. Save and close the Index Agent statefulset file.

vi. Shut down the following Documentum xPlore services using the script available in `<XPLORE_HOME>` in each pod:

- `Documentum xPlore IndexAgent`
- `Documentum xPlore IndexServer`

vii. Take a backup of the following folders available in the Index Server pod:

- `<XPLORE_HOME>/rtdata/config/`
- `<XPLORE_HOME>/rtdata/data/`
- `<XPLORE_HOME>/rtdata/wal/indexserver-0/indexserver-0_wal/`

viii. Extract the transferred data and configurations folders in the same `temp` folder created in step 19.j.iii.

ix. In the Index Server pod, copy and replace the content in the following locations:

- Configuration files to `<XPLORE_HOME>/rtdata/config/`.
- Data files to `<XPLORE_HOME>/rtdata/data/`.
- Wal files to `<XPLORE_HOME>/rtdata/wal/indexserver-0/indexserver-0_wal/`.

k. To update the `XhiveDatabase.bootstrap` file in `<XPLORE_HOME>/rtdata/config/`, do the following:

i. Update all relative path information to the respective Index Server pod deployment path.

ii. Retrieve the Index Server pod IP address information and update in the `XhiveDatabase.bootstrap` file.

Run the following command to retrieve the Index Server pod IP address:

```
kubectl get pods -o wide
```

Update the Index Server pod IP address.

```
<node name="primary" host="<Index Server pod IP address>" port="9330">
```

iii. Update the Wal log path.

Example 3-13: Wal log path

```
<log path="/root/xPlore/rtdata/wal/indexserver-0/indexserver-0_wal"
usable="true"
primary="true"
id="1767596772404"
keep-log-files="false"/>
```

l. To update the `indexserverconfig.xml` file in `<XPLORE_HOME>/rtdata/config/`, do the following:

i. Update all relative path information to the respective Index Server pod deployment path.

ii. Update the Index Server pod host name and URL.

Example 3-14: Index Server pod host name and URL

```
<node hostname="indexserver-0"
admin-rmi-port="9331"
url="http://indexserver:9300/dsearch/"
status="normal"
primaryNode="true"
xdb-listener-port="9330"
appserver-instance-name="indexserver-0"
name="indexserver-0">
```

iii. Update the storage location path.

Example 3-15: Storage location path

```
<storage-location path="/root/xPlore/rtdata/data"
quota_in_MB="10"
status="not_full"
name="default"/>
```

iv. Update the administrator backup path.

Example 3-16: Administrator backup path

```
<backup-location path="/root/xPlore/dsearch/backup"/>
```

v. Update the value of all `name`, and `binding-node name` from `PrimaryDsearch` to the Index Server pod name.

Example 3-17: Changing the value from PrimaryDsearch to Index Server pod name

```
<domain default-document-category="" storage-location-name="default"
name="SystemData">
        <collection usage="Internal" document-category="metricsdata"
name="MetricsDB">
            <properties>
                <property value="false" name="index-required"/>
            </properties>
            <collection usage="Internal" document-category="metricsdata"
name="indexserver-0">
                <properties>
                    <property value="metrics.xml" name="load-file-on-
startup_0"/>
                </properties>
                <binding-node name="indexserver-0"/>
            </collection>
        </collection>
        <collection usage="Internal" document-category="auditdata"
name="AuditDB">
            <properties>
                <property value="false" name="index-required"/>
            </properties>
            <collection usage="Internal" document-category="auditdata"
name="indexserver-0">
                <properties>
                    <property value="auditRecords.xml" name="load-file-on-
startup_0"/>
                </properties>
                <binding-node name="indexserver-0"/>
            </collection>
        </collection>
    </domain>
```

m. To update the Index Server statefulset file, do the following:

   i. Run the following command:

   ```
   kubectl edit sts indexserver
   ```

   ii. From the Index Server statefulset file copy taken in step 19.j.iv.B, add the contents of the `livenessProbe` and `readinessProbe` sections.

   iii. Save and close the Index Server statefulset file.

n. To update the Index Agent statefulset file, do the following:

   i. Run the following command:

   ```
   kubectl edit sts indexagent
   ```

   ii. From the Index Agent statefulset file copy taken in step 19.j.v.B, add the contents of the `livenessProbe` and `readinessProbe` sections.

   iii. Save and close the Index Agent statefulset file.

o. Start the following Documentum xPlore services using the script available in `<XPLORE_HOME>` in each pod:

   - `Documentum xPlore IndexAgent`
   - `Documentum xPlore IndexServer`

3.5 Post-migration tasks

1. Perform all the relevant tasks as described in "Licensing OpenText Documentum CM" on page 45.

2. After the Documentum CM Server pod is up and running, verify if you can access the containers, client components, and connect to the repository using IAPI or IDQL.

   - To verify the IAPI connectivity, connect to the Documentum CM Server pod, and run the following commands:

```
kubectl exec -it dcs-pg-0 -- bash

[dmadmin@dcs-pg-0 /]$ iapi testenv

Please enter a user (dmadmin):
Please enter password for dmadmin:
```

Example 3-18: Sample output to verify the IAPI connectivity

```
        OpenText Documentum iapi - Interactive API interface
        Copyright (c) 2024. OpenText Corporation
        All rights reserved.
        Client Library Release 24.4.0006.0155

Connecting to Server using docbase testenv
[DM_SESSION_I_SESSION_START]info:  "Session 010340dd800026c1 started for user
dmadmin."

Connected to OpenText Documentum Server running Release 24.4.0009.0232
Linux64.Postgres
Session id is s0
API>
```

   - To verify the IDQL connectivity, connect to the Documentum CM Server pod, and run the following commands:

```
kubectl exec -it dcs-pg-0 -- bash
[dmadmin@dcs-pg-0 /]$ idql testenv

Please enter a user (dmadmin):
Please enter password for dmadmin:
```

Example 3-19: Sample output to verify the IDQL connectivity

```
        OpenText Documentum idql - Interactive document query interface
        Copyright (c) 2024. OpenText Corporation
        All rights reserved.
        Client Library Release 24.4.0006.0155

Connecting to Server using docbase testenv
[DM_SESSION_I_SESSION_START]info:  "Session 010340dd800026cc started for user
dmadmin."

Connected to OpenText Documentum Server running Release 24.4.0009.0232
Linux64.Postgres
1>
```

3. After the Documentum xPlore pod is up and running, to verify if you can access the pod, sign in to the cloud platform using the new administrator user created in your on-premises environment in step 19.a and the password provided in the `indexserver.extraEnv` section in the `documentum/charts/xPlore-OneD/values.yaml` file.

4. Remove the redundant entry based on `r_creation_date` having older date from the `dm_acs_config` object.

   a. Run the following command to retrieve the `dm_acs_config` object information:

   ```
   ?,c,select r_object_id,object_name, acs_base_url, r_creation_date from
   dm_acs_config
   ```

   b. Run the following command to remove the object:

   ```
   destroy,c,<r_object_id>
   ```

5. Remove the redundant entry based on `r_creation_date` having older date from the `dm_jms_config` object.

   a. Run the following command to retrieve the `dm_jms_config` object information:

   ```
   ?,c,select r_object_id,object_name,r_creation_date from dm_jms_config
   ```

   b. Run the following command to remove the object:

   ```
   destroy,c,<r_object_id>
   ```

6. Remove the redundant entry from the `dm_server_config` object.

   a. Run the following command to retrieve the `dm_server_config` object information:

   ```
   ?,c,select r_object_id, object_name , r_creation_date, jms_config_id from
   dm_server_config order by r_object_id ASC, i_position DESC
   ```

   b. Run the following commands to remove the `jms_config_id` object in step 5 from the `dm_server_config` object:

   ```
   remove,c,<r_object_id>,jms_config_id[0]
   remove,c,<r_object_id>,jms_mode[0]
   save,c,<r_object_id>
   ```

7. Retrieve and update the `dm_otds_license_config` object.

   a. Run the following commands to retrieve and dump the `dm_otds_license_config` object information:

   ```
   retrieve,c,dm_otds_license_config
   dump,c,l
   ```

   b. Set the values for the following variables according to your requirement:

   - `otds_url`
   - `license_keyname`
   - `business_admin_name`

- `business_admin_password`

c. Optional Run the following commands to remove the on-premises environment license object if it exists in Documentum Administrator:

```
retrieve,c,dm_otds_license_config
destroy,c,l
```

Then, upload a valid license, configure OTDS, and recreate the license object in Documentum Administrator.

Chapter 4

Backing up and restoring OpenText Documentum CM on cloud platforms

Back up OpenText Documentum CM deployment on cloud platforms such as Amazon Web Services (AWS) and Google Cloud Platform (GCP) before initiating upgrade process. This ensures that the deployment can be restored to previous version if any issues arise during the upgrade process, preserving the data integrity and minimizing the data loss.

The OpenText Documentum CM deployment backup also include Persistent Volume Claims (PVCs) in Kubernetes environments deployed on AWS and GCP.

4.1 Backing up and restoring OpenText Documentum CM on AWS

The *Amazon Web Services* documentation describes the standardized procedure for backing up and restoring AWS-based deployments with a focus on Kubernetes workloads, persistent storage, and AWS-managed services such as Relational Database Service (RDS) and Elastic File System (EFS).

The backing up and restoring process of OpenText Documentum CM on AWS is a combination of backup steps described in *Amazon Web Services* documentation and the customized OpenText steps in the given sections.

4.1.1 Prerequisites

Before running the AWS rollback and restore scripts, ensure the following:

- Verify that your environment has access to the Kubernetes cluster.
- AWS rollback and restore shell scripts are downloaded and extracted to the `aws_rollback_scripts` folder.
- `Object Store S3` is available for storing customer data.
- Verify that native backups are configured and scheduled for `RDS` and `EFS`.
- Verify that OpenText Documentum CM is deployed and running.
- Before running the rollback scripts verify the following:
  - Verify that all `*.sh` files have the required run access (Set executable permissions on all `*.sh` files if required).
  - Verify that your Kubernetes (kubectl) context is configured to the current deployment.

– Verify that the cluster name and AWS region are correctly set:

```
export CLUSTER=<EKS CLUSTER NAME>
export AWS_REGION=<AWS REGION>
```

- Back up the OpenText Documentum xPlore. For more information, see the *OpenText Documentum xPlore - Administration and Development Guide (EDCSRC-AGD)*.

4.1.2 Deleting Kubernetes objects

Delete the relevant Kubernetes objects to prevent additional document creation or modification in the OpenText Documentum CM repository during the upgrade process.

To delete Kubernetes objects:

1. Run the following commands to delete the Kubernetes objects:

```
kubectl delete sts --all -n <namespace>
kubectl delete deployments --all -n <namespace>
kubectl delete jobs --all -n <namespace>
kubectl delete pod --all -n <namespace>
```

2. Run the `kubectl get all` command to ensure that the deployments and StatefulSets do not exist and the pods are deleted before upgrading:

```
kubectl get all -n <namespace>
```

4.1.3 Backing up OpenText Documentum CM on AWS

Backing up OpenText Documentum CM ensures that the deployment can be restored if the upgrade fails.

To back up OpenText Documentum CM:

1. Run the `kubectl get sc` command to get the list of storage classes and update the list of storage classes in the `sc-backup.list` file located in the `aws_rollback_scripts` folder:

```
kubectl get sc -o=jsonpath="{range .items[*]}{.metadata.name}{'\n'}"
```

Note: `sc-backup.list` contain only the storage classes that are referenced in the Helm charts.

2. Run the `kubectl get pvc` command to retrieve the list of PVCs:

```
kubectl get pvc -n <namespace> -o=jsonpath="{range .items[*]}{.metadata.name}{'\n'}"
```

Based on the number of `dcs-pg` replicas, update the list of PVCs in the `pvc-backup.list` file located in the `aws_rollback_scripts` folder.

For example, if two `dcs-pg` pods are running, the following PVCs must be added in the `pvc-backup.list` file:

- dcs-vct-dcs-pg-0

- dcs-vct-dcs-pg-1

3. Run the following shell command to back up the storage classes and the PVCs manifest:

```
./aws_rollback_scripts/scripts/create-sc-pv-templates.sh <namespace> <target_dir>
```

For example:

```
./scripts/create-sc-pv-templates.sh d2namespace backup244
```

After the shell script runs, ensure that the log file contains no errors.

4. After completing the AWS backup, use the AWS Backup service to take a backup of the database and EFS. For more information, see the *Amazon Web Services* documentation.

5. Run the Helm `upgrade` command to upgrade the Helm charts to the latest version.

4.1.4 Restoring OpenText Documentum CM deployment on AWS

If the upgrade fails, you can restore the EFS backup into a new file system using AWS Backup. Ensure that restore role has sufficient permission to perform the restore operation. For more information, see the *Amazon Web Services* documentation.

Note: When EFS is restored to new EFS, it will show the total size as 6 kibibytes (KiB). Ensure the EFS size is greater than or equal to the original EFS size before proceeding to restoration.

To restore EFS backup:

1. Run the following command to create EFS mount targets:

```
./aws_rollback_scripts/scripts/create-efs-mount-target.sh <file-system-id> <efs-sg-id>
```

For example:

```
./aws_rollback_scripts/scripts/create-efs-mount-target.sh fs-03572ea926af08bf3 sg-0107ec2591f46b410
```

Note: After running this command, mount targets are automatically created for the restored EFS under the **Network > EFS**.

2. Run the following command on the EC2 instance to install the nfs-common package:

```
sudo apt update
sudo apt install nfs-common -y
```

3. Run the following commands on the same EC2 instance to prevent the restored file system from using `aws-backup-restore_<timestamp>` as its parent folder. To auto-generate the sudo mount command, open the restored EFS and click **Attach**.

```
mkdir /efs
sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
<filesystemID>.efs.<region>.amazonaws.com:/ efs
cd efs && sudo mv /efs/aws-backup-restore_<timestamp>/<BASE-PATH-REFERENCED-IN-
STORAGE-YAML> ./
sudo chown -R <USER>:<USER> <BASE-PATH-REFERENCED-IN-STORAGE-YAML>
```

For example:

```
mkdir /efs
sudo mount -t nfs4 -o
nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport
fs-03572ea926af08bf3.efs.us-east-1.amazonaws.com:/ efs
cd efs && sudo mv /efs/aws-backup-restore_2025-06-23T14-43-10-651995093Z/testdctm ./
sudo chown -R ubuntu:ubuntu testdctm
```

4. Run the following command to recreate StorageClasses, PVs, and PVCs using the new EFS ID:

   Note: Before running this command, ensure that no pods are running.

```
./aws_rollback_scripts/scripts/create-sc-pv-pvc.sh <file-system-id> <namespace>
<source_dir>
```

For example:

```
./scripts/create-sc-pv-pvc.sh fs-03572ea926af08bf3 d2namespace backup244
```

5. Restore the AWS RDS backup to a new RDS instance. For more information, see *Amazon Web Services* documentation.
6. Update the Helm charts with the new RDS instance details.
7. Perform Helm upgrade using the previous version of the Helm charts.

4.1.5 Cleaning up unused backup files

After successful rollback and validation of OpenText Documentum CM deployment, delete the unused backups from AWS and Helm charts:

1. Delete the unused PV and PVC template YAML files from the `/aws_rollback_scripts/` folder.
2. Delete old EFS and RDS, if the retention period is not set.

4.2 Backing up and restoring OpenText Documentum CM on GCP

The backup and recovery process uses native Google Cloud Platform (GCP) services such as Cloud SQL and Google Cloud Storage (GCS), along with Kubernetes volume snapshots. Although GCS is the preferred object store for customer data, its use is optional. The process involves use of shell scripts to automate snapshot creation, deployment restoration, and cleanup tasks.

OpenText Documentum CM deployment backup and restore on GCP is a combination of backup steps described in *Google Cloud Platform* documentation and customized OpenText shell scripts.

4.2.1 Prerequisites

- Verify that OpenText Documentum CM is deployed and running.
- Enable and schedule periodic native backups for Cloud SQL and GCS.
- GCP backup and rollback scripts are downloaded and extracted to `gcp_rollback_scripts` folder. Verify the following steps before running the rollback scripts:
  - Executable permissions set on all `*.sh` files.
  - Kubernetes (kubectl) context is configured to the current deployment.
  - CLUSTER name in the environment variable is configured.

    ```
    export CLUSTER=<GKE CLUSTER NAME>
    ```
- Back up the OpenText Documentum xPlore. For more information, see the *OpenText Documentum xPlore - Administration and Development Guide (EDCSRC-AGD)*.

4.2.2 Deleting Kubernetes objects

Delete the relevant Kubernetes objects to prevent the creation or modification of documents in the OpenText Documentum CM repository during the upgrade process:

1. Run the following commands to delete the Kubernetes objects:

   ```
   kubectl delete sts --all -n <namespace>
   kubectl delete deployments --all -n <namespace>
   kubectl delete jobs --all -n <namespace>
   kubectl delete pod --all -n <namespace>
   ```
2. Run the `kubectl get all` command to verify that the deployments and statefulsets do not exist and the pods are not in running state:

   ```
   kubectl get all -n <namespace>
   ```

4.2.3 Backing up and upgrading OpenText Documentum CM on GCP

OpenText Documentum CM backup ensures that the deployment can be restored if the upgrade process fails.

To backup and upgrade OpenText Documentum CM on GCP:

1. Run the `kubectl get pvc` command to retrieve the list of PVCs from the `gcp_rollback_scripts\pvc-backup.list` file:
   ```
   kubectl get pvc -n <namespace> -o=jsonpath="{range .items[*]}{.metadata.name}{'\n'}"
   ```
2. Add information about all the `dcs-pg` replicas in the `gcp_rollback_scripts\scripts\pvc-backup.list` file.
3. Set the **ReadyToUse** field for all the created VolumeSnapshots to `true`.
4. Run the `create-volume-snapshot.sh` command to backup PVCs manifest:
   ```
   ./gcp_rollback_scripts/scripts/10-create-volume-snapshot.sh <namespace> <nfs-driver>
   ```
5. Use GCP Backup service to backup the Cloud SQL database and GCS object store. Ensure that the backup status is marked as `Completed` before starting the upgrade process. The time required to update the backup status depends on the volume of data.
6. Run the `helm upgrade` command to upgrade the Helm charts to the latest version.

   Notes

   - If the Helm upgrade process fails at the Helm chart level, resolve the Helm chart issues and proceed with the upgrade process.
   - If the Helm deployment fails and the pods are not running, follow the instructions in "Restoring OpenText Documentum CM deployment on GCP" on page 84.

4.2.4 Restoring OpenText Documentum CM deployment on GCP

If the upgrade process fails, you must restore the PVCs, Cloud SQL, and GCS object store.

1. Run the following shell command to backup PVCs:
   ```
   ./gcp_rollback_scripts/scripts/20-restore-pvc.sh <namespace>
   ```
   For example:
   ```
   ./gcp_rollback_scripts/scripts/20-restore-pvc.sh dctm_namespace
   ```
2. Run the `kubectl get pvc` command to verify that all the restored PVCs are in `Bound` state:

```
kubectl get pvc
```

3. Restore Cloud SQL and Object store GCS. For more information, see *Google Cloud Platform* documentation.
4. Run the `Helm upgrade` command to restore the deployment.

4.2.5 Cleaning up the OpenText Documentum CM deployment on GCP

After successful rollback and validation of OpenText Documentum CM deployment, delete the unused backups from GCP and Helm charts:

1. Delete the Kubernetes volume snapshots.

```
./gcp_rollback_scripts/scripts/30-delete-snapshot.sh <namespace>
```

2. Delete the PV and PVC template YAML files from the `gcp_rollback_scripts` folder.
3. Delete the associated GCS instance and Cloud SQL backups to complete the cleanup process.

Chapter 5

Limitations and troubleshooting

This documentation provides information about limitations and troubleshooting when you upgrade the pods on different cloud platforms.

5.1 Limitations and troubleshooting on Microsoft Azure

Table 5-1: Limitations on Microsoft Azure

Product or component name	Limitation
Documentum CM Server	• Installation path is predefined and cannot be changed. The value is `/opt/dctm/product/<product version>`. • Upgrading of schema is not supported. • When you run Tomcat on Linux, the console output is redirected to the `catalina.out` file. This is a Tomcat limitation. • Host name must have the FQDN and it must not be greater than 59 characters. • You must change the storage class according to the Azure Kubernetes service offering. The *default* storage class provisions a standard Azure disk while the *managed-premium* storage class provisions a premium Azure disk.

Table 5-2: Troubleshooting on Microsoft Azure

Product or component name	Problem	Cause	Solution

Documentum CM Server	When you check the status of the upgrade process, it results in an error.	Unsuccessful upgrade.	Use the `describe` command to find the cause.
			For example:
			`kubectl describe pod <name of the pod>`
			Or
			`kubectl describe statefulset <name of the statefulset>`
			If any pod is down or not available, then the upgrade or rollback is not started at the pod level. Recreate the pod for the upgrade or rollback to start.

Documentum CM Server	All the Documentum CM Server pods gets recreated and the upgrade or rollback process is stuck for a long period than the usual time.	Kubernetes platform issue.	Log in to the primary Documentum CM Server pod and delete the `UpgradeInitiated<version>` file located in the `/opt/dctm/data` folder.
Documentum CM Server	Upgrade process is stuck because of change in the installation owner.	Unsuccessful upgrade.	Go to the `/opt/dctm/kube/` folder, change the installation owner name that belongs to the previous deployment, and recreate the pod.

Product or component name	Problem	Cause	Solution
Documentum Connector for Core Share	DBschema jobs failed during deployment with the following error: `1 _Error: UPGRADE FAILED: cannot patch "dbschema-core-notification" with kind Job: Job.batch "dbschema-core-notification" is invalid: spec.template: Invalid value: core.PodTemplateSpec{ObjectMeta:v1.ObjectMeta{Name:"", GenerateName:"", Namespace:"", SelfLink:"", UID:"", ResourceVersion:"", Generation:0, CreationTimestamp:time.Date(1, time.January, 1, 0, 0, 0, 0, time.UTC), DeletionTimestamp:DeletionGracePeriodSeconds:(*int64)(nil), Labels:map[string]string{"batch.kubernetes.io/controller-uid":"0902f2e5-6f8b-4257-8092-1404ed322d87", "batch.kubernetes.io/job-name":"dbschema-core-notification", "controller-uid":"0902f2e5-6f8b-4257-8092-1404ed322d87", "job-name":"dbschema-core-notification"},`	Upgrading the DBschema jobs.	Helm upgrade is not applicable to Documentum Connector for Core Share jobs. If the jobs complete successfully, the DBschema jobs must be disabled in the subsequent Helm upgrade. If the jobs fail, the DBschema jobs must be deleted before the Helm upgrade and ensure that the DBschema jobs are reinstalled. **Note:** If some immutable properties are changed in jobs and a upgrade is performed on the existing jobs, then an error is encountered.

5.2 Limitations and troubleshooting on Amazon Web Services

This section provides information about limitations and troubleshooting on Amazon Web Services.

Table 5-3: Limitations on Amazon Web Services

Product or component name	Limitation
Documentum CM Server	• Installation path is predefined and cannot be changed. The value is `/opt/dctm/product/<product version>`. • Upgrading of schema is not supported. • When you run Tomcat on Linux, the console output is redirected to the `catalina.out` file. This is a Tomcat limitation. • Host name must have the FQDN and it must not be greater than 59 characters.

Table 5-4: Troubleshooting on Amazon Web Services

Product	Problem	Cause	Solution

Documentum CM Server	When you check the status of the upgrade process, it results in an error.	Unsuccessful upgrade.	Use the `describe` command to find the cause.
			For example:
			`kubectl describe pod <name of the pod>`
			Or
			`kubectl describe statefulset <name of the statefulset>`
			If any pod is down or not available, then the upgrade or rollback is not started at the pod level. Recreate the pod for the upgrade or rollback to start.

Product	Problem	Cause	Solution
Documentum CM Server	All the Documentum CM Server pods gets recreated and the upgrade or rollback process is stuck for a long period than the usual time.	Kubernetes platform issue.	Log in to the primary Documentum CM Server pod and delete the `UpgradeInitiated<version>` file located in the `/opt/dctm/data` folder.
Documentum CM Server	Upgrade process is stuck because of change in the installation owner.	Unsuccessful upgrade.	Go to the `/opt/dctm/kube/` folder, change the installation owner name that belongs to the previous deployment, and recreate the pod.
Documentum Connector for Core Share	DBschema jobs failed during deployment with the following error: `1 _Error: UPGRADE FAILED: cannot patch "dbschema-core-notification" with kind Job: Job.batch "dbschema-core-notification" is invalid: spec.template: Invalid value: core.PodTemplateSpec{ObjectMeta:v1.ObjectMeta{Name:"", GenerateName:"", Namespace:"", SelfLink:"", UID:"", ResourceVersion:"", Generation:0, CreationTimestamp:time.Date(1, time.January, 1, 0, 0, 0, 0, time.UTC), DeletionTimestamp:<nil>, DeletionGracePeriodSeconds:(*int64)(nil), Labels:map[string]string{"batch.kubernetes.io/controller-uid":"0902f2e5-6f8b-4257-8092-1404ed322d87", "batch.kubernetes.io/job-name":"dbschema-core-notification", "controller-uid":"0902f2e5-6f8b-4257-8092-1404ed322d87", "job-name":"dbschema-core-notification"},`	Upgrading the DBschema jobs.	Helm upgrade is not applicable to Documentum Connector for Core Share jobs. If the jobs complete successfully, the DBschema jobs must be disabled in the subsequent Helm upgrade. If the jobs fail, the DBschema jobs must be deleted before the Helm upgrade and ensure that the DBschema jobs are reinstalled. **Note:** If some immutable properties are changed in jobs and a upgrade is performed on the existing jobs, then an error is encountered.

Product	Problem	Cause	Solution

OpenText Documentum CM	The script fails while deleting PVCs.	PVCs are not deleted completely.	Delete all PVCs using the following command:
			`kubectl delete pvc --all -n <namespace>`
			Verify that the system automatically removes the associated PVs.
			In the `CLAIM` column, ensure that the system does not reference any PVs for the <*namespace*> used during the backup.
			• If `ReclaimPolicy` is set to `Delete`, the system automatically deletes the associated PVs.
			• If `ReclaimPolicy` is set to `Retain`, run the following commands:
			`kubectl get pv`
			`kubectl delete pv <PV NAME>`
			• If a PV remains in the terminating state, replace <*PV_NAME*> with the actual name of the PV and run the following command to remove its finalizers:
			`kubectl patch pv <PV NAME> -p '{"spec": {"claimRef": null}}'`

5.3 Limitations and troubleshooting on Google Cloud Platform

This section provides information about limitations and troubleshooting on Google Cloud Platform.

Table 5-5: Limitations on Google Cloud Platform

Product or component name	Limitation
Documentum CM Server	• Installation path is predefined and cannot be changed. The value is `/opt/dctm/product/<product version>`. • Upgrading of schema is not supported. • When you run Tomcat on Linux, the console output is redirected to the `catalina.out` file. This is a Tomcat limitation. • Host name must have the FQDN and it must not be greater than 59 characters.

Table 5-6: Troubleshooting on Google Cloud Platform

Product or component name	Problem	Cause	Solution

Documentum CM Server	When you check the status of the upgrade process, it results in an error.	Unsuccessful upgrade.	Use the `describe` command to find the cause.
			For example:
			`kubectl describe pod <name of the pod>`
			Or
			`kubectl describe statefulset <name of the statefulset>`
			If any pod is down or not available, then the upgrade or rollback is not started at the pod level. Recreate the pod for the upgrade or rollback to start.

Product or component name	Problem	Cause	Solution
Documentum CM Server	All the Documentum CM Server pods gets recreated and the upgrade or rollback process is stuck for a long period than the usual time.	Kubernetes platform issue.	Log in to the primary Documentum CM Server pod and delete the `UpgradeInitiated<version>` file located in the `/opt/dctm/data` folder.
Documentum CM Server	Upgrade process is stuck because of change in the installation owner.	Unsuccessful upgrade.	Go to the `/opt/dctm/kube/` folder, change the installation owner name that belongs to the previous deployment, and recreate the pod.

Product or component name	Problem	Cause	Solution
Documentum Connector for Core Share	DBschema jobs failed during deployment with the following error: `1 _Error: UPGRADE FAILED: cannot patch "dbschema-core-notification" with kind Job: Job.batch "dbschema-core-notification" is invalid: spec.template: Invalid value: core.PodTemplateSpec{ObjectMeta:v1.ObjectMeta{Name:"", GenerateName:"", Namespace:"", SelfLink:"", UID:"", ResourceVersion:"", Generation:0, CreationTimestamp:time.Date(1, time.January, 1, 0, 0, 0, 0, time.UTC), DeletionTimestamp:DeletionGracePeriodSeconds:(*int64)(nil), Labels:map[string]string{"batch.kubernetes.io/controller-uid":"0902f2e5-6f8b-4257-8092-1404ed322d87", "batch.kubernetes.io/job-name":"dbschema-core-notification", "controller-uid":"0902f2e5-6f8b-4257-8092-1404ed322d87", "job-name":"dbschema-core-notification"},`	Upgrading the DBschema jobs.	Helm upgrade is not applicable to Documentum Connector for Core Share jobs. If the jobs complete successfully, the DBschema jobs must be disabled in the subsequent Helm upgrade. If the jobs fail, the DBschema jobs must be deleted before the Helm upgrade and ensure that the DBschema jobs are reinstalled. **Note:** If some immutable properties are changed in jobs and a upgrade is performed on the existing jobs, then an error is encountered.