

# Sitecore Connect for Content Hub 5.0 Container Deployment Guide

A guide to deploying the Sitecore Connect for Content Hub connector on Docker and Azure Kubernetes Service.

February 8, 2022

## Table of Contents

|  |    |
|--|----|
| 1. Introduction .....  | 3  |
| 2. Prepare to deploy SCCH to Sitecore containers .....                         | 4  |
| 2.1. Requirements .....  | 4  |
| 3. Add the SCCH connector module to Sitecore in Docker .....                   | 5  |
| 3.1. Prepare the installation files .....                                      | 5  |
| 3.2. Build the Docker images .....   | 5  |
| 4. Add the SCCH connector module to Sitecore in Azure Kubernetes Service ..... | 10 |
| 4.1. Build images and push them to Azure .....                                 | 10 |
| 4.2. Prepare files and folders for deployment .....                            | 10 |
| 4.3. Deploy the containers .....   | 11 |
| 5. Rebuild the search indexes .....  | 13 |
| 6. Upgrade SCCH 4.0 to 5.0 on Docker .....                                     | 14 |
| 6.1. Requirements .....  | 14 |
| 6.2. Upgrade process .....   | 14 |
| 6.2.1. Build new Docker images .....   | 14 |
| 6.2.2. Build the mssql-upgrade image .....                                     | 15 |
| 6.2.3. Perform upgrade process .....   | 15 |
| 7. Upgrade SCCH 4.0 to 5.0 on Kubernetes .....                                 | 17 |
| 7.1. Requirements .....  | 17 |
| 7.2. Build and push the mssql-upgrade image .....                              | 17 |
| 7.3. Perform upgrade process .....   | 17 |

# 1. Introduction

The Sitecore Connect for Content Hub (SCCH) gives web editors and content/digital marketers the ability to work with content created in Sitecore Content Hub. When you make changes in Sitecore Content Hub, the connector automatically creates and updates content in Sitecore.

It also provides access to your digital assets in Sitecore DAM and allows you to easily embed them in your CMS solution.

For more information on Sitecore Content Hub see [the Sitecore documentation](#).

This guide shows you how to add the SCCH connector to Sitecore container installations for Docker and Azure Kubernetes Service.

## 2. Prepare to deploy SCCH to Sitecore containers

This section explains what you need to deploy the Sitecore Connect for Content Hub (SCCH) connector to Sitecore containers 10.1.3 and 10.2 for Docker and Azure Kubernetes Service.

### 2.1. Requirements

Before you add the SCCH module for Docker or AKS, you must have the following:

- Docker Desktop installed and running. For instructions on how to set up the Docker environment, see the [Containers in Sitecore development](#) documentation.
- If the installation is done on Docker, you must have the Sitecore Docker container files deployed on a local machine.. For instructions on how to prepare a Sitecore container, see the *Installation Guide for Developer Workstation with Containers* on the [Sitecore download page](#).
- If the installation is done on Kubernetes, you must have the Sitecore AKS container files deployed on a local machine. For instructions on how to prepare a Sitecore environment with Kubernetes, see the *Installation Guide for Production Environment with Kubernetes* on the [Sitecore download page](#).
- You must have installed the Sitecore certificates, and added the Sitecore domain name to your host file in Windows.

## 3. Add the SCCH connector module to Sitecore in Docker

To add Sitecore Connect for Content Hub (SCCH) in Docker, you must do the following in this order:

- Prepare the installation files.
- Build the Docker images.
- Update the Solr indexes.

### 3.1. Prepare the installation files

To prepare the files you need for the installation:

1. Download the SCCH container deployment package from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact.
2. Download the Sitecore Experience Platform container deployment package from the [Sitecore download page](#). Extract it to a new folder on your local workstation with the folder structure intact. Name the new folder, for example, SCCH.
3. In the folder where you extracted the SCCH package in [Step 1](#), navigate to the folder for the Windows version and topology you are using, for example, SCCH\compose\ltsc2019\xp1. Copy the `docker-compose.override.yml` file to the SCCH\compose\<<version>\<topology> folder you created in [Step 2](#).
4. In the SCCH folder, navigate to the folder for your Windows version and topology you are using, for example, `compose\ltsc2019\xp1`. Open the `.env` file in a text editor. At the bottom of the file, add a definition for the SCCH image. For example:

```
SCCH_IMAGE= scr.sitecore.com/sxp/modules/sitecore-chub-assets:5.0.0-<target-OS-to-deploy>
```

5. In the folder where you extracted the SCCH package, open the `.env-example` file in a text editor, and copy all the contents.
6. At the bottom of the `.env` file from [Step 4](#), after the SCCH image data, paste in the contents from the `.env-example` file. Save the `.env` file.

### 3.2. Build the Docker images

When you have prepared the installation files, you must create Docker files for each role and build the Docker images.

**NOTE**

For more information on image assets, see the documentation on how to [Add Sitecore Modules](#).

To build the images:

1. In the `SCCH` folder, navigate to the folder for your Windows version and topology you are using, for example, `compose\ltsc2019\xp1`. Create a subfolder and name it `module`.
2. In the `module` folder, create these subfolders:
  - `cm`
  - `cd`

**NOTE**

You only need the `cd` subfolder, if you are deploying to an XP1 or XM1 topology.

- `mssql`

**NOTE**

You only need the `mssql` subfolder, if you are deploying to Sitecore 10.1.3 on Docker.

- `mssql-init`

**NOTE**

You only need the `mssql-init` subfolder, if you are deploying to Sitecore 10.1.3 on Kubernetes.

3. In each subfolder, create a new file and name it `Dockerfile`.
4. In the `cm` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG SCCH_IMAGE
ARG TOOL_IMAGE

FROM ${SCCH_IMAGE} as scch
FROM ${TOOL_IMAGE} as tooling
FROM ${BASE_IMAGE}

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]

WORKDIR C:\inetpub\wwwroot

# Add SCCH module
COPY --from=scch \module\cm\content .\

# Add tools from itecore-docker-tools-assets
```

```
COPY -from=tooling \tools\ C:\tools\

# Apply SCCH transformation files
RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\inetpub\wwwroot -XdtPath
\inetpub\wwwroot\App_Data\Transforms\scch\xdts
```

5. If you are deploying an XP1 or XM1 topology, in the `cd` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG SCCH_IMAGE
ARG TOOL_IMAGE

FROM ${SCCH_IMAGE} as scch
FROM ${TOOL_IMAGE} as tooling
FROM ${BASE_IMAGE}

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference =
'SilentlyContinue';"]

WORKDIR C:\inetpub\wwwroot

# Add SCCH module
COPY -from=scch \module\cm\content .\

# Copy SCCH transformation files
COPY -from=scch \module\transforms\ C:\transforms\

# Add tools from itecore-docker-tools-assets
COPY -from=tooling \tools\ C:\tools\

# Apply SCCH transformation files
RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\inetpub\wwwroot -XdtPath
\inetpub\wwwroot\App_Data\Transforms\scch\xdts
RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\inetpub\wwwroot -XdtPath
\transforms\cd
```

6. If you are deploying to Sitecore 10.1.3 in Docker, in the `mssql` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG SCCH_IMAGE

FROM ${SCCH_IMAGE} as scch
FROM ${BASE_IMAGE}

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference =
'SilentlyContinue';"]

# Copy SCCH dacpac files
COPY -from=scch C:\module\db C:\resources\scch_data

# Copy SCCH tool files
COPY -from=scch C:\module\tools\ C:\scch\

# Deploy SCCH dacpac files
RUN C:\scch\Set-DbContainmentLevel.ps1 -moduleDacpacSource C:\resources\scch_data
-isContainer $true; `
Remove-Item -Path C:\scch -Recurse -Force; `
C:\DeployDatabases.ps1 -ResourcesDirectory C:\resources\scch_data; `
Remove-Item -Path C:\resources\scch_data -Recurse -Force;
```

7. If you are deploying to Sitecore 10.1.3 in Kubernetes, in the `mssql-init` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG SCCH_IMAGE

FROM ${SCCH_IMAGE} as scch
FROM ${BASE_IMAGE}

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]

# Copy SCCH dacpac files
COPY --from=scch C:\module\db C:\resources\scch_data

# Copy SCCH tool files
COPY --from=scch C:\module\tools\ C:\scch\

# Prepare SCCH dacpac files
RUN C:\scch\Set-DbContainmentLevel.ps1 -moduleDacpacSource C:\resources\scch_data
-isContainer $true; `
Remove-Item -Path C:\scch -Recurse -Force;# escape=`
ARG BASE_IMAGE
ARG DCRM_CNN_IMAGE

FROM ${DCRM_CNN_IMAGE} as dcrm
FROM ${BASE_IMAGE}

SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]

# Deploy DCRM Staging database file
COPY --from=dcrm C:\module\db C:\resources\dcrm
```

8. Navigate to the `SCCH\compose\<version>\<topology>` folder. Open the `docker-compose.override.yml` file for edit.
9. In the `docker-compose.override.yml` file, add these instructions for the `cm` image:

```
cm:
  image: sitecore-chub-${TOPOLOGY}-cm:${SITECORE_VERSION}
  build:
    context: ./module/cm
    args:
      BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-${TOPOLOGY}-cm:${SITECORE_VERSION}
      SCCH_IMAGE: ${SCCH_IMAGE}
      TOOL_IMAGE: ${TOOL_IMAGE}
```

10. If you are deploying to an XP1 or XM1 topology, add these instructions for the `cd` image:

```
cd:
  image: sitecore-chub-${TOPOLOGY}-cd:${SITECORE_VERSION}
  build:
    context: ./module/cd
    args:
      BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-${TOPOLOGY}-cd:${SITECORE_VERSION}
      SCCH_IMAGE: ${SCCH_IMAGE}
      TOOL_IMAGE: ${TOOL_IMAGE}
```



11. If you are deploying to Sitecore 10.1.3 in Docker, add these instructions for the `mssql` image:

```
mssql:
  image: sitecore-chub-${TOPOLOGY}-mssql:${SITECORE_VERSION}
  build:
    context: ./module/mssql
    args:
      BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-${TOPOLOGY}-mssql:${
SITECORE_VERSION}
      SCCH_IMAGE: ${SCCH_IMAGE}
```

12. If you are deploying to Sitecore 10.1.3 in Kubernetes, add these instructions for the `mssql-init` image:

```
mssql-init:
  image: sitecore-chub-${TOPOLOGY}-mssql-init:${SITECORE_VERSION}
  build:
    context: ./module/mssql-init
    args:
      BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-${TOPOLOGY}-mssql-init:${
SITECORE_VERSION}
      SCCH_IMAGE: ${SCCH_IMAGE}
```

13. In the Windows console, go to the `SCCH\compose\<version>\<topology>` folder where the `docker-compose.override.yml` file is. Run the command `docker-compose build`.
14. After the build has completed, run the command `docker-compose up -d`.

#### NOTE

Some modifications to Sitecore deployments, such as adding connection strings or changing the web configuration files, require you to use configuration transforms to change the configuration files. For information on how to apply configuration transforms, see the Sitecore [container development documentation](#).

When the Docker compose command has finished, [rebuild your search indexes](#).

## 4. Add the SCCH connector module to Sitecore in Azure Kubernetes Service

To add the Sitecore Connect for Content Hub (SCCH) connector in Azure Kubernetes Service (AKS) you must do the following in this order:

- Build the SCCH images and push them to Azure.
- Prepare files and folders for deployment.
- Deploy the containers using *kubectl* commands.
- Update your Solr indexes.

### 4.1. Build images and push them to Azure

To build the images for SCCH and push them to Azure:

1. Build the images for SCCH as explained in [Add the SCCH connector module to Sitecore in Docker](#).
2. Open the Windows console and use the `docker tag` command to tag the images. For example:

```
docker tag sitecore-chub-xpl-cm:<tag version> $registry/sitecore-chub-xpl-cm:<tag version>
```

3. In the console, use the `docker push` command to push the images to your Azure registry. For example:

```
docker push $registry/sitecore-chub-xpl-cm:<tag version>
```

### 4.2. Prepare files and folders for deployment

To prepare files and folders in your installation for deployment:

1. In the folder where you extracted the SCCH container deployment package, navigate to the `SCCH\<Sitecore version>\k8s\<windows version>` folder, for example, `SCCH\k8s\ltsc2019`. Copy the `overrides` subfolder to the Sitecore Experience Platform (SXP) container deployment package folder `k8s\<version>` (on the same level as the `xpl` folder).
2. If you are deploying to Sitecore 10.1.3 on Kubernetes, in the `k8s\<version>\overrides\<topology>` folder, create a subfolder and name it `init`. In

the `init` subfolder, create a file, name it `kustomization.yaml`, and add the following instructions to it:

```
apiVersion: kustomize.config.k8s.io/v1beta1
kind: Kustomization
bases:
  - ../../../../<topology>/init
images:
- name: scr.sitecore.com/sxp/sitecore-<topology>-mssql-init
  newName: "{registry}/sitecore-chub-<topology>-mssql-init"
  newTag: tag
```

3. In the `overrides\<topology>\secrets` folder, update each of the following files with connection string details:

- `sitecore-cmp-content-hub.txt`
- `sitecore-cmp-service-bus-entity-path-in.txt`
- `sitecore-cmp-service-bus-entity-path-out.txt`
- `sitecore-cmp-service-bus-subscription.txt`
- `sitecore-dam-content-hub.txt`
- `sitecore-dam-external-redirect-key.txt`
- `sitecore-dam-search-page.txt`

#### NOTE

Each file contains an example of how the connection string should look.

## 4.3. Deploy the containers

Prepare the AKS cluster configuration and deploy the ingress controller. For information on how to do this, see the *Installation Guide for Production Environment for Kubernetes*, which is available on the [Sitecore download page](#).

To deploy the containers and the necessary Kubernetes components:

1. Open the Windows console, and navigate to the `k8s\<version>` folder.
2. Deploy the secrets. Use this command:

```
kubectl apply -k ./overrides/<topology>/secrets/
```

3. Run the external folder. Use this command:

```
kubectl apply -k ./<topology>/external/
```

4. Wait for all containers to have the status *Ok/Running*. You can check the status with this command:

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

5. Run the `init` folder. Use this command:

```
kubectl apply -k ./<topology>/init/
```

**NOTE**

If you are deploying to Sitecore 10.1.3, use this command instead:

```
kubectl apply -k ./overrides/<topology>/init/
```

6. Wait for all containers to have the status *Completed*. You can check the status with this command:

```
kubectl get pods
```

7. If you are using the `XP1` topology, and you want to create persistent volumes, run this command:

```
kubectl apply -f ./xp1/volumes/azurefile
```

**NOTE**

Persistent volumes are only available for the `XP1` topology

8. Run the Sitecore containers with the `SCCH` changes. Use this command:

```
kubectl apply -k ./overrides/<topology>/
```

9. Wait for all containers to have the status *Ok/Running*. You can check the status with the `kubectl get pods` command.

10. In order to enable external access, you need to update the local host file with the external IP address. To obtain the external IP address, use this command:

```
kubectl get service -l app=nginx-ingress
```

**NOTE**

For information on how to update the local host file, see the *Installation Guide for Production Environment for Kubernetes*, which is available on the [Sitecore download page](#).

When the containers have been deployed, [rebuild your search indexes](#).

## 5. Rebuild the search indexes

When you have deployed the containers, you must rebuild your search indexes.

To rebuild the indexes:

1. Browse to your Sitecore URL, for example, `https://xp1cm.localhost/`. Open the control panel.
2. In the **Indexing** section, click **Populate Solr Managed Schema**.
3. In the **Schema Populate** dialog box, click **Select All**, then click **Populate**. Wait for the process to finish.
4. On the Control Panel, in the **Indexing** section, click **Indexing Manager**. In the **Indexing Manager** dialog, select the following indexes:
  - `sitecore_master_index`
  - `sitecore_core_index`
  - `sitecore_web_index`
  - `sitecore_marketing_asset_index_master`
  - `sitecore_marketing_asset_index_web` databases
5. Click **Rebuild**. When the indexes have been rebuilt, click **Close**.

## 6. Upgrade SCCH 4.0 to 5.0 on Docker

This section explains how to upgrade Sitecore Connect for Content Hub (SCCH) to version 5.0 on a container platform.

### 6.1. Requirements

Before you perform the upgrade for SCCH container to version 5.0, you must have the following:

- Sitecore Experience Platform (SXP) 10.1 deployed on Docker
- SCCH 4.0 deployed on Docker
- An up to date back up of the current SCCH 4.0 container files

### 6.2. Upgrade process

You must upgrade your SXP installation to 10.2 and your SCCH installation to 5.0 together.

To do so, you must:

1. Build new docker images for SXP 10.2 and SCCH 5.0.
2. Build an mssql-upgrade image
3. Perform the upgrade

#### 6.2.1. Build new Docker images

To build the SXP 10.2 and SCCH 5.0 Docker images:

1. Download the SXP container deployment 10.2 package from the [Sitecore download page](#). Extract it to a new folder on your local workstation with the folder structure intact. Name the new folder, for example, SCCH7.
2. Download the SCCH container deployment 5.0 package from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact. Copy the files in the `Chub.Asset\compose\<windows version>\<topology>` folder and paste them into the `\compose\<windows version>\<topology>` folder in the SXP 10.2 deployment structure.
3. Navigate to your Sitecore 10.1 container deployment folder. Copy the databases from the `mssql-data` folder and paste them in the Sitecore 10.2 container deployment `mssql-data` folder.
4. To deploy Sitecore 10.2, open a PowerShell window with administrator rights, navigate to the SC 10.2 Container deployment folder, and run the following commands:

```
docker-compose build
docker-compose up
```

5. Verify that Sitecore Container 10.2 is up and running successfully.

## 6.2.2. Build the mssql-upgrade image

You use a custom mssql-upgrade image to upgrade a Sitecore solution that has SCCH installed. To build a custom mssql-upgrade image, you must download the latest Sitecore XP mssql-upgrade image from the container registry and create a custom Dockerfile on top of it.

1. From the **Resource files for Modules 1.0.0** section on the [Sitecore download page](#), download the Sitecore Connect for Content Hub Upgrade resources 1.0.0 package. Extract it to a folder on your local machine.
2. From the upgrade resources, copy the `\<current version>\Data` folder and paste it into the upgrade folder for the Windows version and topology you are using, for example, `ltsc2019\upgrade\xp1`.
3. In the `<current version>` folder, create a docker file and name it `Dockerfile`. In the file, add instructions to point its base image to the 10.2 mssql-upgrade image. The file will look, for example, like this:

```
ARG BASE_IMAGE=ideftdevacr.azurecr.io/sxp/sitecore-xp1-mssql-
upgrade:10.2.0.006572.994-10.0.17763.2183-ltsc2019-unstable
FROM ${BASE_IMAGE}
SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference =
'SilentlyContinue';"]
# Add chub module
COPY < SCCH Module Upgrade Resource file path> "C:\data\ResourceItems\10.1.0\modules\"
```

### NOTE

In the Dockerfile, ensure that ARG BASE\_IMAGE parameter points to the mssql-upgrade image, and that the Data folder local path is set to the 6.0.0\Data folder in the folder you created in step 1.

4. To build the mssql-upgrade image, open a PowerShell window, navigate to the folder where you placed the Dockerfile, and run the following command:

```
docker build . -t "<imageName>:<available port number>"
```

5. Verify that Docker has created an image with the name you specified.

## 6.2.3. Perform upgrade process

To perform the upgrade:

1. On your local machine, in a PowerShell window, navigate to the SCCH Deployment 5.0 folder. Navigate to the upgrade folder for the Windows version and topology you are using, for example, `ltsc2019\upgrade\xp1`.
2. In the topology folder, run the `compose-init.ps1` script. This script updates the environment configuration file with the appropriate values for all the environment variables including the SQL username, SQL password, SQL Server address, and the Sitecore license file.

**NOTE**

For more information about running the script to prepare for the deployment, see the Installation Guide for Developer Workstation with Containers on the [Sitecore download page](#).

3. In the upgrade folder, verify that the `upgrade.env` file has the correct details. For more details, see the [Upgrade Container Deployment Guide](#) for SXP 10.2.0.
4. In the `docker-compose.upgrade.yml` file, update the `image` setting with the `mssql-upgrade` image you created previously.
5. Open a new PowerShell window with administrator rights. Navigate to the upgrade directory.
6. Verify that the `mssql` container is up and running.
7. To perform the upgrade, run this command:

```
docker-compose.exe -f .\docker-compose.upgrade.yml --env-file .\upgrade.env up
```

8. To check the status of the upgrade, run this command:

```
docker-compose.exe -f .\docker-compose.upgrade.yml --env-file .\upgrade.env ps
```



## 7. Upgrade SCCH 4.0 to 5.0 on Kubernetes

This section explains how you upgrade Sitecore Connect for Content Hub (SCCH) container in Kubernetes to 5.0.

### 7.1. Requirements

Before you perform the upgrade for Container SCCH Container to version 5.0, you must:

- Have a Sitecore AKS 10.1 deployment with SCCH 4.0
- Back up the 10.1 database.

### 7.2. Build and push the mssql-upgrade image

To upgrade SCCH you must build an mssql-upgrade image. To do so:

1. Build the images for SCCH as explained in the *Build the mssql-upgrade image* section in [Upgrade SCCH 4.0 to 5.0 on Docker](#).
2. In a PowerShell window, navigate to the folder containing the Dockerfile and build and tag the image. For example:

```
docker build . -t sitecore-scch-xp1-cm:<imageVersionTag> $registry/sitecore-scch-xp1-cm:<newTag>
```

3. Check that Docker has created the image.
4. Push the created image to your Azure registry. For example:

```
docker push $registry/sitecore-scch-xp1-cm:<newTag>
```

### 7.3. Perform upgrade process

To upgrade to SCCH 5.0 on Sitecore 10.2:

1. Download and extract Sitecore 10.2 Sitecore Container Deployment Package from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact.
2. Navigate to the upgrade folder for the Windows version and topology you are using, for example, `k8s\ltsc2019\upgrade\xp1`. In the `kustomization.yaml` file, update the

images section with `newName` and `newTag` of the custom `mssql-upgrade` image you created and pushed previously.

3. In the `configuration` folder, update the secrets files. For more information on the secrets files, please refer to the *Installation Guide for Production Environment with Kubernetes* guide available on the [Sitecore download page](#).
4. Download the SCCH container deployment 5.0 package from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact. Copy the `SCCH\k8s\<windows version>\overrides` folder and paste it into the `\k8s\<windows version>\` folder in the SXP 10.2 deployment structure.
5. In the `secrets` folder in the SXP 10.2 structure, for example, `k8s\ltsc2019\overrides\xp1\secrets`, update these secrets files:

- `sitecore-cmp-content-hub.txt`
- `sitecore-cmp-service-bus-entity-path-in.txt`
- `sitecore-cmp-service-bus-entity-path-out.txt`
- `sitecore-cmp-service-bus-subscription.txt`
- `sitecore-dam-content-hub.txt`
- `sitecore-dam-external-redirect-key.txt`
- `sitecore-dam-search-page.txt`

6. Log in to the Azure CLI and set a subscription.

```
az login
az account set --subscription "Your Subscription"
```

7. Get the credentials for the Kubernetes cluster that was created with the AKS cluster.

```
zaks get-credentials --resource-group <10.1 resource group>--name <10.1 cluster>
```

8. To deploy the Sitecore upgrade job, go to the folder where the updated files are, for example `k8s\ltsc2019\upgrade\xp1`, and run this command:

```
kubectl apply -k .\
```

9. To check if the job has completed, run this command:

```
kubectl get pod
```

10. When the upgrade process is completed, you can delete the Kubernetes upgrade job and upgrade secrets. In the console, go to the upgrade folder from step 2, and run these commands:

```
kubectl delete -f .\
kubectl delete -k .\
```

## NOTE

Upgrading with a custom mssql-image includes cleaning up SCCH items from the database since SCCH 5.0 has moved to using resources files. This requires rebuilding and upgrading the Sitecore role instances cm, xconnect and xdbsearchworker.

For detailed instructions on how to build and deploy the Sitecore role instances please refer to the Sitecore *Upgrade Container Deployment Guide* on the [Sitecore download page](#).