

Sitecore Connect for SFMC 7.0 Content Exchange Container Deployment Guide

A guide to deploying Sitecore Connect for SFMC Content Exchange to Docker and Azure Kubernetes Service.

November 29, 2021

Table of Contents

1. Introduction	3
2. Prepare the installation	4
2.1. Requirements	4
2.2. Add an API integration package to your installation	4
2.3. Prepare an SFMC connection string to your Sitecore installation	5
3. Add the SFMCCE connector module to Sitecore in Docker	6
3.1. Prepare the installation files	6
3.2. Build the Docker images	6
4. Add the SFMCCE connector module to Sitecore in Azure Kubernetes Service	9
4.1. Build images and push them to Azure	9
4.2. Prepare files and folders for deployment	9
4.3. Deploy the containers	10
5. Rebuild the search indexes	12
6. Upgrade the SFMCCE connector to version 7.0 in Docker	13
6.1. Requirements	13
6.2. Upgrade process	13
6.3. Build new Docker images	13
6.4. Build the mssql-upgrade image	14
6.5. Perform upgrade process	15
7. Upgrade the SFMCCE connector to version 7.0 in Kubernetes	16
7.1. Requirements	16
7.2. Build and push the mssql-upgrade image	16
7.3. Perform upgrade process	16

1. Introduction

Sitecore Connect for Salesforce Marketing Cloud - Content Exchange (SFMC-CE) enables you to push Sitecore media item structures to corresponding asset structures in Salesforce.

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

2. Prepare the installation

This section explains how to prepare for deploying the Sitecore Connect software for Salesforce Marketing Cloud - Content Exchange (SFMC-CE) connector to Docker and Azure Kubernetes Service.

2.1. Requirements

Before you install SFMC-CE for containers, 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 the Sitecore containers, see the *Installation Guide for Developer Workstation with Containers* on the [Sitecore download site](#).
- 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 site](#).
- An SFMC account where you can create an installed package with appropriate access rights.

To prepare for the installation, you must:

- add an API integration package to your SFMC installation
- prepare an SFMC connection string.

2.2. Add an API integration package to your installation

To add an API integration package to your Salesforce Marketing Cloud (SFMC) installation:

1. Follow the steps in the Marketing Cloud Package Development documentation for Salesforce, in the *Create and Install Packages* section.
2. Select the `Server-to-Server` integration type for the API Integration package.
3. Add the following required rights for your connection:

Category	Type	Rights
Assets	Documents and images	<ul style="list-style-type: none">• Read• Write

Category	Type	Rights
Assets	Saved content	• Read
		• Write
Hub	Tags	• Write

2.3. Prepare an SFMC connection string to your Sitecore installation

To construct a Salesforce Marketing Cloud (SFMC) connection string to your Sitecore installation:

1. In Salesforce, in the component you created, in the **Components** area, make a note of the following values:
 - Client Id
 - Client Secret
 - Authentication Base URI
 - REST Base URI
 - SOAP Base URI
2. Use the values to construct a connection string with this format:

```
SFMC_CONNECTIONSTRING=client id=<Client Id>;  
                        client secret=<Client Secret>;  
                        auth endpoint=<Authentication Base URI>;  
                        rest endpoint=<REST Base URI>;  
                        soap endpoint=<SOAP Base URI>;
```

3. Add the SFMCCE connector module to Sitecore in Docker

To add Sitecore Connect for Salesforce Marketing Cloud - Content Exchange (SFMCCE) in Docker, you must:

1. Prepare the installation files.
2. Build the Docker images.
3. Update your search indexes.

3.1. Prepare the installation files

To prepare the files you need for the installation:

1. Download the SFMCCE container deployment package from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact.
2. Go to the folder into which you extracted the SFMCCE container deployment package. Go to the folder for the Windows version and topology you are using, for example, `compose\ltsc2019\xp1`.
3. Open the `.env-example` file in an editor. Copy all the variables to the clipboard.
4. Go to the Sitecore container deployment folder on your local machine. Go to the folder for the Windows version and topology you are using, for example, `compose\ltsc2019\xp1`.
5. Open the `.env` file in an editor, and paste in the variables from the SFMCCE `.env-example` file. Replace the default value for `SFMC_CONNECTIONSTRING` with the connection string you prepared in [Prepare to deploy SFCRM to Sitecore containers](#).
6. From the SFMCCE `compose\<version>\<topology>` folder, copy the `docker-compose.override.yml` file to the Sitecore container deployment `compose\<version>\<topology>` folder (where the `docker.compose.yml` file is).

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. Go to the Sitecore container deployment folder on your local machine. Go to the folder for the Windows version and topology you are using, for example, `compose/ltsc2019/xp1`. Create a folder and name it *module*.
2. In the *module* folder, create this subfolder:
 - `cm`
3. In the subfolder, create a new file and name it `Dockerfile`. In the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG SFMCCE_IMAGE
ARG TOOLING_IMAGE

FROM ${SFMCCE_IMAGE} as sfmcce
FROM ${TOOLING_IMAGE} as tooling
FROM ${BASE_IMAGE} as baseImage

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

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

FROM baseImage as toolsfmc

#Copy from image\module\cm\content
COPY --from=sfmcce \module\cm\content C:\inetpub\wwwroot

#Copy transformation files
COPY --from=sfmcce \module\xdtttransform\cm\transforms\ C:\transforms\role

# Add SFMCCE connection strings in Sitecore config file
RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\inetpub\wwwroot -XdtPath
C:\transforms\role
```

4. In the `compose\<version>\<topology>\docker-compose.override.yml` file, add build instructions for each role. If you are using, for example, the XP0 topology, the file will look like this:

```
services:
  cm:
    image: sitecore-sfmcce-xp0-cm:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./cm/Dockerfile
    args:
      BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp0-cm:${SITECORE_VERSION}
      SFMCCE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcce-xp0-assets:${
SFMCCE_VERSION}
      TOOLING_IMAGE: ${SITECORE_TOOLS_REGISTRY}sitecore-docker-tools-assets:${
TOOLS_VERSION}
    environment:
      Sitecore_ConnectionStrings_mysf: ${MYSF_CONNECTIONSTRING}
```

5. In the `compose\<version>\<topology>\.env` file, add the asset image version. For example:

```
SFMCCE_VERSION=<image version for your topology>  
SITECORE_TOOLS_REGISTRY=scr.sitecore.com/tools/  
TOOLS_VERSION=<image version for your topology>
```

NOTE

You can find the image version in the Sitecore Docker Images repository.

6. In the Windows console, go to the folder containing the `docker-compose.override.yml` file. Run the command `docker-compose build`.
7. Once the build completes, run the command `docker-compose up -d`.

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

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

To add the Sitecore Connect for Salesforce Marketing Cloud (SFMCCE) connector in Azure Kubernetes Service (AKS) you must:

- Build the SFMCCE images and push them to Azure.
- Prepare files and folders for deployment.
- Deploy the containers using *kubectI* commands.

4.1. Build images and push them to Azure

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

1. Build the images for SFMCCE as explained in [Add the SFMCCE connector module to Sitecore in Docker](#).
2. Tag the images with the `docker tag` command. For example:

```
docker tag sfmcce/sitecore-xp1-cm:10.1.0.005207.643-10.0.17763.1757-ltsc2019 $registry/  
sitecore-sfmcce-xp1-cm:<tag version>
```

3. Push the images to your Azure registry with the `docker push` command. For example

```
docker push $registry/sitecore-sfmcce-xp1-cm:<tag version>
```

4.2. Prepare files and folders for deployment

To prepare files and folders in your installation for deployment:

1. Download the SFMCCE container deployment package from the [Sitecore download page](#) and extract it to a folder on your local workstation.
2. Open the folder into which you extracted the SFMCCE container deployment package.
3. Navigate to the `k8s\<version>\<topology>` folder, for example, `k8s\ltsc2019\xp1`. Copy the `overrides` subfolder to the Sitecore Experience Platform (SXP) container deployment package folder `k8s\<version>` (on the same level as the `xp1` folder).
4. In the SXP container deployment package, in each of the `overrides` and `overrides\xp1\secrets` folders, locate the `kustomization.yaml` file. In each file, update the `bases` parameter with the appropriate folder names for your installation, for example, `../.. /xp1`.

NOTE

The `bases` parameter contains the placement of the original Sitecore container deployment files that the `kustomization.yaml` files override.

5. In each of the `kustomization.yaml` files, in the `images:` section, update the `newName` and `newTag` parameters with the values for the `mssql-init` and `cm` images you pushed to the Azure Registry.
6. In the `overrides\xp1\secrets` folder, in the `sitecore-salesforce-content-exchange-connection-string.txt` file, replace the content with the connection string you prepared in [Prepare the installation](#).

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 folder containing the `xp1` and `overrides` folders.
2. Deploy the secrets. Use this command:

```
kubectl apply -k ./overrides/xp1/secrets/
```

3. Run the `external` folder. Use this command:

```
kubectl apply -k ./xp1/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 ./overrides/xp1/init/
```

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

```
kubectl get pods
```

7. To create persistent volumes, run this command:

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

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

```
kubectl apply -k ./overrides/xp1/
```

9. Wait for all containers to have the status *Ok/Running*. You can check the status with the `kubectl get pods` command.
10. Update the local host file. For information on how to do this, 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, click **Select All**, then click **Rebuild**. When the indexes have been rebuilt, click **Close**.
5. Open the Content Editor with *Master* as the content database.
6. In the content tree, navigate to */sitecore/system/settings*. Verify that the *SFMC Connector* item is available.
7. Navigate to *SFMC Connector/Configuration*. On the **Folder** tab, verify that the **SFMC Configuration** button is available.

6. Upgrade the SFMCCE connector to version 7.0 in Docker

This section explains how you upgrade SFMCCE Container 6.0 to SFMCCE 7.0. SFMCCE 7.0 is compatible with Sitecore 10.2.

6.1. Requirements

Before you upgrade SFMCBDE for container 6.0 to SFMCCE for container 7.0 you must have the following:

- Sitecore Experience Platform (SXP) 10.1 deployed on Docker
- DCRM 6.0 deployed on Docker
- An up to date back up of the current mssql databases.

6.2. Upgrade process

You must upgrade your SXP installation to 10.2 and your SFMCCE installation to 7.0 together.

To do so, you must:

1. Build new docker images for SXP 10.2 and SMFCCE 7.0.
2. Build an mssql-upgrade image
3. Perform the upgrade

6.3. Build new Docker images

To build the SXP10.2 and SFMCCE 7.0 Docker images:

1. Download the Sitecore container deployment package for 10.2 from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact.
2. Download the SFMCCE container deployment package for 7.0 package from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact.
3. Go to the Sitecore container deployment 10.2 folder on your local machine. Go to the folder for the Windows version and topology you are using, for example, `compose\ltsc2019\xp1`. Create a folder and name it `module`.

4. Navigate to your Sitecore 10.1 container deployment folder. Copy the databases from the `mssql-data` folder and paste them into the Sitecore 10.2 container deployment folder.
5. Open the Windows console, go to Sitecore Container Deployment 10.2 folder and run the following commands.

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

6.4. Build the mssql-upgrade image

To upgrade a Sitecore solution that has SFMCCE installed, you must build a custom `mssql-upgrade` image. To do so:

1. From the **Resource files for Modules 1.0.0** section on the [Sitecore download page](#), download the SFMC - Content Exchange Upgrade resources package and extract it to your local machine.
2. From the upgrade resources, copy the `SFMC - Content Exchange Upgrade resources 1.0.0\6.0.0\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 `SFMC - CE Upgrade resources 1.0.0\6.0.0` 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=${SITECORE_DOCKER_REGISTRY}sitecore-xp1-mssql-upgrade:${SITECORE_VERSION}
FROM ${BASE_IMAGE}
SHELL ["powershell", "-Command", "$ErrorActionPreference = 'Stop'; $ProgressPreference = 'SilentlyContinue';"]
# ADD SFMCCE module
COPY <Data folder local path> "C:\data\ResourceItems\10.2.0\modules"
```

NOTE

In the Dockerfile, ensure that the `ARG BASE_IMAGE` value is pointing to the `mssql-upgrade` image and that the `Data` folder local path is set to the folder you created in [Step 1](#).

4. To build the `mssql-upgrade` image with SFMCCE resources, 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>" command
```

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

6.5. Perform upgrade process

To perform the upgrade:

1. On your local machine, in a PowerShell window, navigate to the Sitecore Container Deployment 10.2 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 `docker-compose.upgrade.yml` file, update the `image` setting with the `mssql-upgrade` image you created previously.
4. Open a new PowerShell window with administrator rights. Navigate to the upgrade directory.
5. To perform the upgrade, run this command:

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

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

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

7. When the upgrade process is completed, you can clean up your environment. If you ran the upgrade container in Docker Compose, from the Docker Compose folder for the topology that you upgraded, for example, `sitecore-xp1`, run the following PowerShell cmdlet:

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

7. Upgrade the SFMCCE connector to version 7.0 in Kubernetes

This section explains how you could upgrade from SFMCCE Container 6.0 to SFMCCE 7.0. SFMCCE 7.0 is compatible with Sitecore 10.2.

7.1. Requirements

Before you upgrade SFMCCE for container 6.0 to SFMCCE for container 7.0 you must:

- Back up the mssql databases.

7.2. Build and push the mssql-upgrade image

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

1. Build the images for SFMCCE as explained in the *Build the mssql-upgrade image* section in [Upgrade the SFMCCE connector to version 7.0 in Docker](#).
2. Open the Windows console and use the docker tag command to tag the images. For example:

```
docker tag sitecore-SFMCCE-xpl-cm:<imageVersionTag> $registry/sitecore-sfmcce-xpl-cm:<newTag>
```

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

```
docker push $registry/sitecore-sfmcce-xpl-cm:<newTag>
```

7.3. Perform upgrade process

To upgrade to SFMCCE 7.0:

1. Download the Sitecore container deployment package for 10.2 from the [Sitecore Developer Portal](#). 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\xpl. 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 SFMCCE container deployment package for 7.0 package from the [Sitecore download page](#). Extract it to your local workstation with the folder structure intact. Copy the `SFMCBDE\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 the `sitecore-salesforce-content-exchange-connection-string.txt` secret file.
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 by running this command:

```
az aks get-credentials --resource-group <10.1 resource group> --name <10.1 cluster>
```

8. Deploy the Sitecore upgrade job by moving to the folder in step 2 where the updated files are and running 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 folder you used in step 2, and run these commands:

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

NOTE

Upgrading with a custom mssql-image includes cleaning up SFMCCE items from the database since SFMCCE 7.0 has moved to using resources files. This requires rebuilding and upgrading the Sitecore role instances `cm`, `xconnect` and `xdbsearchworker`.

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