

Sitecore Connect for SFMC 6.0 Content Exchange Container Deployment Guide

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



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
3.3. Update the Solr indexes	9
4. Add the SFMCCE 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
4.4. Update Solr indexes	12

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	<ul style="list-style-type: none">• Read• Write
Hub	Tags	<ul style="list-style-type: none">• 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 the Solr 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 these subfolders:
 - `mssql`
 - `cm`
 - `mssql-init`

NOTE

The `mssql-init` image is only necessary if you are deploying to Azure Kubernetes Services (AKS).

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

```
# escape=`
ARG BASE_IMAGE
ARG SFMCCE_IMAGE

FROM ${SFMCCE_IMAGE} as sfmcce
FROM ${BASE_IMAGE}

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

# Deploy SFMCCE dacpac file
COPY --from=sfmcce C:\module\db C:\sfmcce_data
RUN C:\DeployDatabases.ps1 -ResourcesDirectory C:\sfmcce_data; `
    Remove-Item -Path C:\sfmcce_data -Recurse -Force;
```

5. In the `cm` folder, 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\xdttransform\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
```

6. If you are deploying to AKS, in the `mssql-init` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG SFMCCE_IMAGE

FROM ${SFMCCE_IMAGE} as sfmcce
FROM ${BASE_IMAGE}

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

# Deploy SFMCCE dacpac file
COPY --from=sfmcce C:\module\db C:\resources\ce
```

7. In the `compose\<version>\<topology>\docker-compose.override.yml` file, add build instructions for each role. If you are using, for example, the XPO 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: ${MYSF_CONNECTIONSTRING}
  mssql:
    image: sitecore-sfmcce-xp0-mssql:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./mssql/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp0-mssql:${SITECORE_VERSION}
        SFMCCE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcce-xp0-assets:${
SFMCCE_VERSION}
```

NOTE

If you are deploying to AKS, remember to add build instructions for the `mssql-init` image as well.

8. 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.

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

3.3. Update the Solr indexes

When the Docker compose command has finished, you must update your Solr indexes.

To update the indexes:

1. When the Docker compose command finishes, browse to your Sitecore URL, for example, `https://xp0cm.localhost/`. Open the control panel and click on **Populate Solr Managed Schema**.
2. After Sitecore has populated the Solr Schema, click **Indexing Manager**.
3. Open the Content Editor with *Master* as the content database.
4. In the content tree, navigate to `/sitecore/system/settings`. Verify that the *SFMC Connector* item is available.
5. Navigate to *SFMC Connector/Configuration*. On the **Folder** tab, verify that the **SFMC Configuration** button is available.

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:

1. Build the SFMCCE images and push them to Azure.
2. Prepare files and folders for deployment.
3. Deploy the containers using *kubectl* commands.
4. Update your Solr indexes.

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 SFCRM connector module to Sitecore in Docker](#).
2. Tag the images with the `docker tag` command. For example:

```
docker tag sfmcce/sitecore-xp1-cm:10.1.1.0.005207.643-10.0.17763.1757-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 Developer Portal](#) 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`, `overrides\xp1\init`, 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 to deploy SFCRM to Sitecore containers](#).

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 SFCRM 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.

4.4. Update Solr indexes

To update your Solr indexes:

1. Browse to your Sitecore URL, for example, `https://cm.globalhost/`. Open the control panel and click **Populate Solr Managed Schema**.
2. After Sitecore has populated the Solr Schema, click **Indexing Manager**.
3. In the **Indexing Manager** dialog, select the indexes you want to update, and click **Rebuild**. When the indexes have been rebuilt, click **Close**.
4. Open the Content Editor with *Master* as the content database.
5. In the content tree, navigate to `/sitecore/system/settings`. Verify that the *SFMC Connector* item is available.
6. Navigate to *SFMC Connector/Configuration*. On the **Folder** tab, verify that the **SFMC Configuration** button is available.