Sitecore Connect for SFMC 7.0 Behavioral Data Exchange Container Deployment Guide

A guide to deploying the Sitecore Connect for SFMC Behavioral Data Exchange to Docker and Azure Kubernetes Service.

November 29, 2021

Table of Contents

1. Introduction

Sitecore Connect for Salesforce Marketing Cloud - Behavioral Data Exchange (SFMCBDE) enables you to push Sitecore contact data and marketing list data to corresponding data structures in Salesforce Marketing Cloud (SFMC). The connector only synchronizes data from Sitecore to SFMC.

This basic functionality does not require you to deploy the tenant service.

In addition to the basic SFMCBDE functionality, you can add two plugins that enable advanced synchronization. The plugins are:

- SFMC Activity for Sitecore Sync. This enables you to synchronize xConnect contact status from SFMC to Sitecore using custom Journey Builder activities.
- SFMC Activity for Marketing Automation (MA). This enables you to synchronize xConnect contact details to SFMC using MA campaigns. With it, you can synchronize the contacts from Sitecore to SFMC Centralized Data Extension. You can also trigger SFMC Journey Builder API events.

Both of these plugins require you to deploy the tenant service.

This guide shows you how to add the SFMCBDE connector, with or without the tenant service, to Sitecore container installations for Docker and Azure Kubernetes Service.

2. Prepare to deploy SFMCBDE to Sitecore containers

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

2.1. Requirements

Before you install SFMCBDE 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. Choose the `Server-to-Server` integration type for the API Integration package.
3. Add the following required rights for your connection:

Category	Type	Rights
Data	Data Extensions	• Read • Write

4. If you use the SFMCBDE Activity for Marketing Automation plugin, you must also add the following rights for your connection:

Category	Type	Rights
Automation	Journey	• Read • Write • Execute
Contacts	List and Subscribers	• Read • 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:

```
ConnectionString_sfmc=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 SFMCBDE to Sitecore in Docker without the tenant service

You can deploy Sitecore Connect for Salesforce Marketing Cloud - Behavioral Data Exchange (SFMCBDE) to Docker without the tenant service, if you do not need the Activity for Marketing Automation and Sitecore Sync functionalities.

If you need the tenant service functionality, follow the instructions in Add SFMCBDE to Sitecore in Docker with the tenant service instead.

To add Sitecore Connect software for Sitecore Connect for Salesforce Marketing Cloud - Behavioral Data Exchange (SFMCBDE) in Docker, you must:

- 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 SFMCBDE 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 SFMCBDE container deployment package. Go to the SFMCBDE folder for the Windows version and topology you are using, for example, `SFMCBDE\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 SFMCBDE `.env-example` file. Replace the default value for `ConnectionString_sfmc` with the connection string you prepared in Prepare an SFMC connection string to your Sitecore installation.
6. From the SFMCBDE `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:
   - `cm`
   - `xconnect`
   - `xdbautomationworker`
   - `xdbsearchworker`
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 SFMCBDE_IMAGE
ARG DEF_IMAGE
ARG TOOLING_IMAGE

FROM ${DEF_IMAGE} as def
FROM ${SFMCBDE_IMAGE} as sfmcbde
FROM ${TOOLING_IMAGE} as tooling
FROM ${BASE_IMAGE}

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

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

# Add DEF module
COPY --from=def \module\cm\content C:\inetpub\wwwroot

# Add SFMCBDE module
COPY --from=sfmcbde \module\cm\content C:\inetpub\wwwroot

# Copy transformation file to add SFMCBDE connection strings in Sitecore config file
COPY --from=sfmcbde \module\xdttransform\cm\transforms\App_Config
\ConnectionStrings.config.xdt  C:\transforms\role\App_Config\ConnectionStrings.config.xdt

# Copy transformation file to update value for Sitecore.Services.SecurityPolicy in
Sitecore.Services.Client.config
COPY --from=sfmcbde \module\xdttransform\cm\transforms\App_Config\Sitecore\Services.Client
```

```
C:\transforms\role\App_Config\Sitecore\Services.Client

RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\inetpub\wwwroot -XdtPath
C:\transforms\role
```

5. In the `xconnect` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG SFMCBDE_IMAGE

FROM ${SFMCBDE_IMAGE} as sfmcbde
FROM ${BASE_IMAGE}

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

# Copy models file into xconnect
COPY --from=sfmcbde \module\xconnect\content C:\inetpub\wwwroot
```

6. In the `xdbsearchworker` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`

ARG BASE_IMAGE
ARG SFMCBDE_IMAGE

FROM ${SFMCBDE_IMAGE} as sfmcbde
FROM ${BASE_IMAGE}

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

# Copy models file into index worker
COPY --from=sfmcbde \module\xdbsearchworker\content C:\service\
```

7. In the `xdbautomationworker` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`

ARG BASE_IMAGE
ARG SFMCBDE_IMAGE
ARG DEF_IMAGE
ARG TOOLING_IMAGE

FROM ${DEF_IMAGE} as def
FROM ${SFMCBDE_IMAGE} as sfmcbde
FROM ${TOOLING_IMAGE} as tooling
FROM ${BASE_IMAGE}

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

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

# Add DEF module
COPY --from=def \module\xdbautomationworker\content C:\service

# Add SFMCBDE module
COPY --from=sfmcbde \module\xdbautomationworker\content C:\service
```

```
#Copy transformation files
COPY --from=sfmcbde \module\xdttransform\xdbautomationworker\transforms\ C:\transforms\role

# Add SFMCBDE connection strings in Sitecore config file
# Add SfmcJourneyInformation node in sc.MarketingAutomation.ContactLoader.xml file
RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\service -XdtPath C:\transforms\role
```

8. In the `compose\<version>\<topology>\docker-compose.override.yml` file, add build instructions for each role. If you are using, for example, the XP1 topology, the file looks like this:

```
services:
  cm:
    image: sitecore-sfmcbde-xp1-cm:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./cm/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp1-cm:${SITECORE_VERSION}
        DEF_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-def-xp1-assets:$
{DEF_VERSION}
        SFMCBDE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcbde-xp1-assets:$
{SFMCBDE_VERSION}
        TOOLING_IMAGE: ${SITECORE_TOOLS_REGISTRY}sitecore-docker-tools-assets:$
{TOOLS_VERSION}
    environment:
      Sitecore_ConnectionStrings_sfmc: ${ConnectionString_sfmc}
      Sitecore_ConnectionStrings_TenantService: ${ConnectionString_TS}

  xdbautomationworker:
    image: sitecore-sfmcbde-xp1-xdbautomationworker:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./xdbautomationworker/Dockerfile
      args:
       BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp1-xdbautomationworker:$
{SITECORE_VERSION}
        DEF_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-def-xp1-assets:$
{DEF_VERSION}
        SFMCBDE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcbde-xp1-assets:$
{SFMCBDE_VERSION}
        TOOLING_IMAGE: ${SITECORE_TOOLS_REGISTRY}sitecore-docker-tools-assets:$
{TOOLS_VERSION}
    environment:
      Sitecore_ConnectionStrings_TenantService: ${ConnectionString_TS}

  xdbsearchworker:
    image: sitecore-sfmcbde-xp1-xdbsearchworker:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./xdbsearchworker/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp1-xdbsearchworker:$
{SITECORE_VERSION}
        SFMCBDE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcbde-xp1-assets:$
{SFMCBDE_VERSION}

  xdbsearch:
    image: sitecore-sfmcbde-xp1-xdbsearch:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./xconnect/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp1-xdbsearch:${SITECORE_VERSION}
```

```
        SFMCBDE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcbde-xp1-assets:$
{SFMCBDE_VERSION}

  xdbcollection:
    image: sitecore-sfmcbde-xp1-xdbcollection:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./xconnect/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp1-xdbcollection:$
{SITECORE_VERSION}
        SFMCBDE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcbde-xp1-assets:$
{SFMCBDE_VERSION}
```

9. If you are not deploying the tenant service, remove the `id` and `cd` images from the `docker-compose.override.yml` file.
10. In the `compose\<version>\<topology>\.env` file, add the asset image version. For example:

```
DEF_VERSION=<image version for your topology>
SFMCBDE_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.

If you are deploying the tenant service, stop here, and continue with the instructions in Add SFMCBDE to Sitecore in Docker with the tenant service.

If you are not deploying the tenant service, continue with the following steps:

1. In the Windows console, go to the folder containing the `docker-compose.override.yml` file. Run the command `docker-compose build`.
2. When the build completes, run the command `docker-compose up -d`.
3. When the Docker compose command has finished, rebuild your search indexes.

4. Add SFMCBDE to Sitecore in Docker with the tenant service

In order to use the Activity for Marketing Automation and Sitecore Sync functionality in Sitecore Connect for Salesforce Marketing Cloud - Behavioral Data Exchange (SFMCBDE), you must deploy a tenant service.

To add the tenant service in Docker, you must:

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

4.1. Prepare the installation files for the tenant service

To prepare the files you need for the installation:

1. Complete the steps in the *Prepare the installation files* and *Build the docker images* sections in Add SFMCBDE to Sitecore in Docker without the tenant service.
2. Download the SFMCBDE-TS container deployment package from the Sitecore download page. Extract it to your workstation with the folder structure intact.
3. Navigate to the folder into which you extracted the package. Navigate to the `TenantService.SFMCBDE` folder for the Windows version and topology you are using, for example, `TenantService.SFMCBDE\compose\ltsc2019\xp1`.
4. Open the `.env-example` file in an editor. Copy all the variables to the clipboard.
5. 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`.
6. Open the `.env` file in an editor, and paste in the variables from the `.env-example` file. Update the variables according to your installation. Save the `.env` file.

   NOTE
   After you create the tenant in Sitecore, you can update the TENANT_SERVICE_CONFIG_EXECUTE_URL variable in the `.env` file with the tenant service endpoint ID. When you do this you must then redeploy the tenant service.

7. From the `TenantService.SFMCBDE\compose\<version>\<topology>` folder, copy the `tenant-service.override.yml` file to the Sitecore container deployment `compose \<version>\<topology>` folder (where the `docker.compose.yml` file is).

4.2. Build the Docker images for the tenant service

When you have prepared the installation files and the Docker image files for the SFMCBDE roles, you must add Docker files for each tenant service role, and build the Docker images.

To build the images:

1. In the `module\cm` folder, in the `Dockerfile` file, before the `invoke-XdtTransform.ps1` command, add the following instructions:

```
# Copy transformation file to update tenant service name into
Sitecore.DataExchange.MarketingAutomation.Client.config
COPY --from=sfmcbde \module\xdttransform\cm\transforms\App_Config\Sitecore
\MarketingAutomation C:\transforms\role\App_Config\Sitecore\MarketingAutomation

# Copy transformation file to add tenant service host name into web.config access control
allow
COPY --from=sfmcbde \module\xdttransform\cm\transforms\Web.config.xdt C:\transforms\role
\Web.config.xdt
```

2. In the `module` folder, create these subfolders:
   - `tenantservice`
   - `cd`
   - `id`
3. In each subfolder, create a new file and name it `Dockerfile`.
4. In the `tenantservice` folder, create a new subfolder and name it `Config`. In the `Config` subfolder, create a new file and name it `Update-ConfigJson.ps1`.
5. In the `Update-ConfigJson.ps1` file, enter the following code, and then save the file:

```
param(
    [Parameter(Mandatory = $true)][string]$executeUrl
)
$configJsonPath = "./sync-activity/config.json";
$jsonDepth = 20
Write-Host "Start Updating config.json with $executeUrl"

$snapshot = (Get-Content $configJsonPath | ConvertFrom-Json )
$snapshot.arguments.execute.url = $executeUrl
$snapshot.configurationArguments.publish.url = $executeUrl

$snapshot | ConvertTo-Json -Depth $jsonDepth  | set-content $configJsonPath
Write-Host "Done Updating config.json"
```

> **NOTE**
> This script updates the `config.json` file in the tenant service. It is run by the `Dockerfile` file for the `cm` image.

6. In the `tenantservice` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
```

```
ARG SFMCBDE_IMAGE
ARG TOOLING_IMAGE

FROM ${SFMCBDE_IMAGE} as sfmcbde
FROM ${TOOLING_IMAGE} as tooling
FROM ${BASE_IMAGE}

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

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

# Add SFMCBDE module files for tenant service
COPY --from=sfmcbde \module\tenantservice\content .\

#Copy transformation files
COPY --from=sfmcbde \module\xdttransform\tenantservice\transforms\ C:\transforms\role

# Copy connection string updater script file to wwwroot folder
# Add reference data connection strings
# Add SFMC connection strings in tenant service connection strings
RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\inetpub\wwwroot -XdtPath
C:\transforms\role

# Copy Update-ConfigJson.ps1 into scripts folder
COPY tenantservice/Config/Update-ConfigJson.ps1 C:\tools\scripts
ENTRYPOINT "C:\tools\scripts\Update-ConfigJson.ps1" "-executeUrl" "$
($env:TenantService_SfmcJourney_ExecuteUrl)"; `
        C:\ServiceMonitor.exe w3svc;
```

7. In the `cd` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`

ARG BASE_IMAGE
ARG SFMCBDE_IMAGE

FROM ${SFMCBDE_IMAGE} as sfmcbde
FROM ${BASE_IMAGE}

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

WORKDIR C:\inetpub\wwwroot

# Add SFMCBDE config
COPY --from=sfmcbde \module\cm\content\App_Config\Sitecore\DataExchange
\SalesforceMarketingCloud .\App_Config\Sitecore\DataExchange\SalesforceMarketingCloud

# Add SFMCBDE dll
COPY --from=sfmcbde \module\cm\content\bin
\Sitecore.DataExchange.XConnect.SalesforceMarketingCloud.dll .\bin
```

8. In the `id` folder, in the `Dockerfile` file, enter the following instructions:

```
# escape=`
ARG BASE_IMAGE
ARG DEF_IMAGE
ARG TOOLING_IMAGE

FROM ${DEF_IMAGE} as def
FROM ${TOOLING_IMAGE} as tooling
FROM ${BASE_IMAGE}
```

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

WORKDIR C:\Identity

# Add DEF module
COPY --from=def \module\transforms\ C:\transforms\

COPY --from=tooling \tools\ \tools\
RUN C:\tools\scripts\Invoke-XdtTransform.ps1 -Path C:\Identity -XdtPath c:\transforms\id
```

9. In the `compose\<version>\<topology>\docker-compose.override.yml` file, add build instructions for the `id` and `cd` roles. If you are using, for example, the XP1 topology, the file will look like this:

```
services:
  id:
    image: sitecore-sfmcbde-id:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./id/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-id6:${SITECORE_VERSION}
        DEF_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-def-xp1-assets:$
{DEF_VERSION}
        TOOLING_IMAGE: ${SITECORE_TOOLS_REGISTRY}sitecore-docker-tools-assets:$
{TOOLS_VERSION}

  cd:
    image: sitecore-sfmcbde-xp1-cd:${SITECORE_VERSION}
    build:
      context: ./module
      dockerfile: ./cd/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}sitecore-xp1-cd:${SITECORE_VERSION}
        SFMCBDE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcbde-xp1-assets:$
{SFMCBDE_VERSION}
```

10. In the `compose\<version>\<topology>\tenant-service.override.yml` file, add build instructions for the `tenant-service` role. For example:

```
services:
  tenant-service:
    build:
      context: ./module
      dockerfile: ./tenantservice/Dockerfile
      args:
        BASE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-tenant-service:$
{MODULE_VERSION}
        SFMCBDE_IMAGE: ${SITECORE_DOCKER_REGISTRY}modules/sitecore-sfmcbde-xp1-assets:$
{SFMCBDE_VERSION}
        TOOLING_IMAGE: ${SITECORE_TOOLS_REGISTRY}sitecore-docker-tools-assets:$
{TOOLS_VERSION}
    environment:
      TenantService_ConnectionStrings_sitecore: database=master;tenant id=$
{SITECORE_TENANT_ID};host=http://cm;timeout=20;auth endpoint=http://id/;client secret=$
{SITECORE_CLIENT_SECRET};client id=${SITECORE_CLIENT_ID}
      TenantService_ConnectionStrings_xconnect.collection: http://xdbcollection
      TenantService_ConnectionStrings_Xdb.ReferenceData.Client: http://xdbrefdata
      TenantService_ConnectionStrings_sfmc: ${ConnectionString_sfmc}
      TenantService_SfmcJourney_ExecuteUrl: ${TENANT_SERVICE_CONFIG_EXECUTE_URL}
```

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

```
DEF_VERSION=<image version for your topology>
SFMCBDE_VERSION=<image version for your topology>
SITECORE_TOOLS_REGISTRY=scr.sitecore.com/tools/
TOOLS_VERSION=<image version for your topology>
MODULE_VERSION=<tenant service image version for your topology>
```

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

12. In the Windows console, go to the folder containing the `docker-compose.override.yml` file. Run the command `docker-compose -f docker-compose.yml -f docker-compose.override.yml -f tenant-service.override.yml up –detach`.
13. When the Docker compose command has finished, rebuild your search indexes.

> **NOTE**
> To test if the tenant service is running, follow the *Deploy the Tenant Service container* section of the *Installation Guide for Data Exchange Tenant Web Service*, which is available on the Sitecore download page.

5. Add SFMCBDE to Sitecore in Azure Kubernetes Service

To add the Sitecore Connect for Salesforce Marketing Cloud - Behavioral Data Exchange (SFMCBDE) connector in Azure Kubernetes Service (AKS), with or without the tenant service, you must:

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

> **NOTE**
> The only topology that is supported for SFMCBDE in AKS is `XP1`. The `XM1` topology is not supported.

5.1. Build images and push them to Azure

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

1. Build the images for SFMCBDE as explained in Add SFMCBDE to Sitecore in Docker without the tenant service.
2. If you are installing the tenant service, update the following value in the `web.config` file with the tenant service URL for your installation:

   ```
   <add
   name="Access-Control-Allow-Origin" value="http://<tenant service url>"
   xdt:Locator="Match(name)"
   xdt:Transform="SetAttributes(value)" />
   ```

   For more information, see the Sitecore documentation on building images.
3. If you are installing the tenant service, build the images for the SFMCBDE tenant service as explained in Add SFMCBDE to Sitecore in Docker with the tenant service.
4. Open the Windows console, and use the `docker tag` command to tag the images. For example:

   ```
   docker tag sfmcce/sitecore-xp1-cm:10.1.0.005207.643-10.0.17763.1757-ltsc2019 $registry/
   modules/sitecore-sfmcbde-xp1-cm:sfmcbde
   ```

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

   ```
   docker push $registry/modules/sitecore-sfmcbde-xp1-cm:sfmcbde
   ```

5.2. Prepare files and folders for deployment

To prepare files and folders in your installation for deployment:

1. If you are not installing the tenant service, download the SFMCBDE container deployment package from the Sitecore download page. If you are installing the tenant service, download the SFMCBDE-TS container deployment package. Extract the package you chose to a folder on your local workstation.
2. Navigate to the `SFMCBDE\k8s\<version>` folder, for example, `SFMCBDE\k8s\ltsc2019`. Copy the `overrides` subfolder to the Sitecore Experience Platform (SXP) container deployment package folder `k8s\<version>` (on the same level as the `xp1` folder).
3. In the SXP container deployment package, in each of the `overrides\xp1` 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.

4. In the `overrides\xp1\secrets` folder, in the `sitecore-salesforce-bde-connection-string.txt` file, replace the content with the connection string you prepared in Prepare to deploy SFMCBDE to Sitecore containers.

The rest of the procedure depends on whether you are installing the tenant service or not.

If you are not installing the tenant service:

1. In each of the `kustomization.yaml` files, in the `images:` section, update the `newName` and `newTag` parameters with the values for the `cm`, `xdbcollection`, `xdbsearch`, `xdbsearchworker`, and `xdbautomationworker` images you pushed to the Azure Registry.
2. In the Sitecore container deployment package, open the `overrides\xp1\kustomization.yaml` file and remove the `cd` and `id` images.

If you are installing the tenant service:

1. Prepare the tenant service configuration. For information on how to do this, see the *Add Tenant Service module to Sitecore in Azure Kubernetes Service* section in the I*nstallation Guide for Data Exchange Tenant Web Service* on the Sitecore download page.
2. In the folder where you extracted the SFMCBDE-TS container deployment package, navigate to the `TenantService.SFMCBDE\k8s\<version>` folder, for example, `TenantService.SFMCBDE\k8s\ltsc2019`. Copy the content of the `overrides` subfolder to the Sitecore Experience Platform (SXP) container deployment package folder `k8s\<version>`

`\overrides\xp1`. If some of the files already exist in the target `overrides` folder, update them with the new content.

3. In the SXP container deployment package, in each of the `overrides\xp1` and `overrides\xp1\secrets` folders, locate the `kustomization.yaml` file. In each of the kustomization.yaml files, in the `images:` section, update the `newName` and `newTag` parameters with the values for the `cm`, `cd`, `id`, `xdbcollection`, `xdbsearch`, `xdbsearchworker`, `xdbautomationworker`, and `tenantservice` images you pushed to the Azure Registry.
4. In the `overrides\xp1\secrets` folder, update the `sitecore-tenant-connection-string.txt` file with the client ID and client secret for your Sitecore Identity Server.

> **NOTE**
> After you create the tenant in Sitecore, you can update the tenant service endpoint ID in the `TENANT_SERVICE_CONFIG_EXECUTE_URL` variable in the `patch-ts.yaml` file. If you do this you must then redeploy the tenant service.

5.3. 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 SFMCBDE 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.

6. 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**.

7. Upgrade the SFMCBDE connector to version 7.0 in Docker

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

7.1. Requirements

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

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

7.2. Upgrade process

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

To do so, you must:

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

7.3. Build new Docker images

To upgrade SFMCBDE you must build new Docker images. To do so:

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 SFMCBDE container deployment package for 7.0 package from the Sitecore download page. Extract it to your local workstation with the folder structure intact.
3. Perform step 2 to 6 from the *Prepare the installation files* section in Add SFMCBDE to Sitecore in Docker without the tenant service.

4. 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.
5. Perform step 2 to 10 from the *Build the Docker images* section in Add SFMCBDE to Sitecore in Docker without the tenant service.
6. Go to the previous Sitecore Container Deployment 10.1 folder, copy the databases from the `mssql-data` folder and paste them into the Sitecore Container Deployment 10.2 folder.
7. Open the Windows console, go to Sitecore Container Deployment 10.2 folder and run the following commands:

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

7.4. Build the mssql-upgrade image

To upgrade a Sitecore solution that has SFMCBDE 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 - BDE Upgrade resources package and extract it to your local machine.
2. From the upgrade resources, copy the `SFMC - BDE 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 - BDE 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 SFMCBDE 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 2.

4. To build the mssql-upgrade image with SFMCBDE 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.

7.5. 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
   ```

8. Upgrade the SFMCBDE connector to version 7.0 in Kubernetes

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

8.1. Requirements

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

- Back up the mssql databases.

8.2. Build and push the mssql-upgrade image

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

1. Build the images for SFMCBDE as explained in the *Build the mssql-upgrade image* section in Upgrade the SFMCBDE 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-sfmcbde-xp1-cm:<imageVersionTag> $registry/sitecore-sfmcbde-xp1-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-sfmcbde-xp1-cm:<newTag>
```

8.3. Perform upgrade process

To upgrade to SFMCBDE 7.0 on Sitecore 10.2:

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. 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` attributes of 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 SFMCBDE 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-bde-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 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 SFMCBDE items from the database since SFMCBDE 7.0 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.