Sitecore AI 3.0 Installation Guide with Docker Containers

Sitecore AI Auto-Personalization Standard

January 20, 2022

Table of Contents

1. Introduction

This guide explains how to deploy the Sitecore AI Auto-personalization Standard module with Sitecore Experience Platform on a developer workstation using containers.

This guide is for deploying Sitecore AI Auto-personalization containers to a Sitecore XP0 or XP1 deployment topology using Docker.

> **NOTE**
>
> For instructions on how to deploy Sitecore AI Auto-personalization Standard using Kubernetes, refer to Sitecore AI Installation Guide for Kubernetes available on the Download site.

2. Prerequisites

You must complete the following prerequisite steps to prepare for deploying Sitecore XP with the Sitecore AI Auto-personalization module to a developer workstation with containers.

> **NOTE**
> These instructions assume that you have signed up for Sitecore AI – Automated Personalization, and have the provided tokens, configuration service URL, and assigned tenant IDs at hand.

2.1. Download required packages

Download the following packages from the Sitecore Downloads page:

- Sitecore Experience Platform 10.1.1 Container Deployment Package
- Sitecore AI container package

2.2. Prepare Sitecore AI files

To prepare Sitecore AI files:

1. Extract the `SitecoreAIStandardContainerDeployment.{version}.{revision}.zip` archive to any location on your workstation. In the archive, the `compose/deploy/<XP_Topology>` folder contains a `docker-compose.override.yml` file that you use for deploying the images that are created during the build stage.
2. Extract the `SitecoreContainerDeployment.10.*.*.zip` archive, and, in the folder where you extracted it, locate the `compose/<yourbaseOS>/<XP_Topology>` folder. In the instructions that follow, this folder is referred to as your *deploy* folder.

2.3. Prepare your Sitecore XP Docker deployment

To prepare your Sitecore XP docker deployment:

1. Download the Sitecore XP *Installation Guide for a Developer Workstation with Containers*, and familiarize yourself with its contents.

2. Make sure you meet the system requirements specified in the *Sitecore XP Installation Guide for a Developer Workstation with Containers*.
3. Make sure that your system meets all software and hardware requirements and follow the instructions to complete all steps in the **Preparation** section.

3. Build custom images for Sitecore AI

The following instructions assume you have completed all the steps in the **Prerequisites** section in Sitecore XP *Installation Guide for a Developer Workstation with Containers*.

In order to prepare the Sitecore XP images to work with the Sitecore AI Auto-personalization extension, you must install the Sitecore AI asset image on top of the affected Sitecore XP roles in the configuration topology.

To build the custom images:

> **NOTE**
> The following instructions apply for building images for XP0 topology and XP1 topology roles.

1. In your Docker installation, create a folder and name it *build*.
2. In the *build* folder, create an `.env` file, copy and paste the following variables into it, and then set values:

```
TOPOLOGY=
SITECORE_DOCKER_REGISTRY=
SITECOREAI_INTEGRATION_ASSET_IMAGE=
SITECORE_AI_VERSION=
SITECORE_VERSION=
```

The following table describes possible values for each environment variable:

Variable name	Description	Value
`TOPOLOGY`	The Sitecore XP deployment topology you need to build.	`xp0` `xp1`
`SITECORE_DOCKER_REGISTRY`	The Sitecore container registry.	`scr.sitecore.com/sxp/`
SITECOREAI_INTEGRATION_ASSET_IMAGE	The name of the base Sitecore AI asset image.	`scr.sitecore.com/sxp/modules/sitecore-ai-xp1-assets`
`SITECORE_VERSION`	The version of the base Sitecore images.	Copy this value from the Sitecore XP `.env` file for your topology. For example: `10.1-ltsc2019`, where *10.1* is the Sitecore version, and `ltsc2019` is the base OS version.
`SITECORE_AI_VERSION`	The version tag of the base Sitecore AI asset image.	Use one of the following, based on the OS version you are using: • `2009: 3.0.0-20H2` • `ltsc2019: 3.0.0-ltsc2019`

3. Copy the `SitecoreAIStandardContainerDeployment/compose/build/<xp0 or xp1>/docker-compose.yml` file into your `build` folder.

   Your `build` folder now contains the `docker-compose.yml` and `.env` files that are required to build a custom image of each role included in the XP1 topology.

4. Complete this step if you are using the XP0 topology.

   > **NOTE**
   >
   > If you are deploying the XP1 topology, skip this step, and go to Step 5 instead.

   In the *build* folder, create a sub-folder for each role in the XP0 topology, with the required docker files as follows:

   - Create a sub-folder named `cm`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

     ```
     # escape=`
     ARG CM_BASE_IMAGE
     ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
     FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
     FROM ${CM_BASE_IMAGE} AS base
     COPY --from=sitecoreai_integration C:\module\cm\content\ C:\inetpub\wwwroot
     ```

   - Create a sub-folder named `xconnect`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

     ```
     # escape=`
     ARG XCONNECT_BASE_IMAGE
     ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
     FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
     FROM ${XCONNECT_BASE_IMAGE} AS base
     COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\inetpub\wwwroot
     ```

   - Create a sub-folder named `cortexprocessingworker`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

     ```
     # escape=`
     ARG CORTEXPROCESSING_BASE_IMAGE
     ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
     FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
     FROM ${CORTEXPROCESSING_BASE_IMAGE} AS base
     COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\service\
     ```

   - Create a sub-folder named `xdbsearchworker`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

     ```
     # escape=`
     ARG XDBSEARCHWORKER_BASE_IMAGE
     ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
     FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
     FROM ${XDBSEARCHWORKER_BASE_IMAGE} AS base
     COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\service\
     ```

5. If you are using the XP0 topology, skip this step and go to step 6 instead. If you are using the XP1 topology, in the *build* folder, create a sub-folder for each role in the XP1 topology, with the required docker files as follows:

    - Create a sub-folder named `cm`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

    ```
    # escape=`
    ARG CM_BASE_IMAGE
    ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
    FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
    FROM ${CM_BASE_IMAGE} AS base
    COPY --from=sitecoreai_integration C:\module\cm\content\ C:\inetpub\wwwroot
    ```

    - Create a sub-folder named `cd`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

    ```
    # escape=`
    ARG CD_BASE_IMAGE
    ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
    FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
    FROM ${CD_BASE_IMAGE} AS base
    COPY --from=sitecoreai_integration C:\module\cm\content\ C:\inetpub\wwwroot
    ```

    - Create a sub-folder named `cortexprocessingworker`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

    ```
    # escape=`
    ARG CORTEXPROCESSINGWORKER_BASE_IMAGE
    ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
    FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
    FROM ${CORTEXPROCESSINGWORKER_BASE_IMAGE} AS base
    COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\service\
    ```

    - Create a sub-folder named `xdbsearchworker`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

    ```
    # escape=`
    ARG XDBSEARCHWORKER_BASE_IMAGE
    ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
    FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
    FROM ${XDBSEARCHWORKER_BASE_IMAGE} AS base
    COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\service\
    ```

    - Create a sub-folder named `xdbcollection`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

    ```
    # escape=`
    ARG XDBCOLLECTION_BASE_IMAGE
    ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
    FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
    FROM ${XDBCOLLECTION_BASE_IMAGE} AS base
    COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\inetpub\wwwroot\
    ```

    - Create a sub-folder named `xdbrefdata`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

```
# escape=`
ARG XDBREFDATA_BASE_IMAGE
ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
FROM ${XDBREFDATA_BASE_IMAGE} AS base
COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\inetpub\wwwroot\
```

- Create a sub-folder named `cortexreporting`. In that sub-folder, create a new file named `dockerfile`, and copy and paste the following content into the file:

```
# escape=`
ARG CORTEXREPORTING_BASE_IMAGE
ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
FROM ${CORTEXREPORTING_BASE_IMAGE} AS base
COPY --from=sitecoreai_integration C:\module\xconnect\content\ C:\inetpub\wwwroot\
```

6. To build the required custom images that includes the Sitecore AI files, in the `build` folder, run the `docker-compose build` command.

4. Deploy Sitecore XP and Sitecore AI to a developer workstation using Docker containers

To deploy Sitecore XP with Sitecore AI Auto-personalization Standard to a developer workstation using Docker containers:

1. In the *Installation Guide for a Developer Workstation with Containers for Sitecore XP 10.1.0*, follow the instructions in Chapter 2 - Deploying a Sitecore XP workstation up to and including the step where you **Open the docker-compose.yaml file** to review its content (after you installed the self-signed root authority certificate).
2. In your `deploy` folder, verify that all variable values have been correctly populated in the `.env` file, and add the following additional Sitecore AI variables:

> **NOTE**
>
> For an example, you can refer to the `env-example` file that is included in the following location: `SitecoreAiContainerDeployment/compose/deploy/<xp0 or xp1>/`

Name	Value
`TENANTID`	<The unique tenant ID provided to you>
`ML_CONFIG_BASEURL`	<The Config service url provided to you>
`ML_CONFIG_TOKEN`	<The unique config token provided to you>
`TOPOLOGY`	`xp0` or `xp1`
`SITECORE_AI_VERSION`	The value that matches the `SITECORE_AI_VERSION` value in the `.env` file used to build images.

3. In your *deploy* folder, create a sub-folder named `override`, and then, based on your deployment topology, copy the `SitecoreAiContainerDeployment/compose/deploy/<xp0 or xp1>/docker-compose.override.yml` file into it.
4. Run the following command:

```
docker-compose -f .\docker-compose.yml -f .\override\docker-compose.override.yml up -d
```

5. Go to the *Installation Guide for a Developer Workstation with Containers for Sitecore XP 10.1.0*, and in the **Deploy a workstation** section, perform the step *to check the Docker container status* and, then resume the deployment process and clean up instructions.

Refer to the Sitecore documentation website for information on how to work with Sitecore AI auto-personalization.

NOTE

In the XP0 deployment topology, you visit the CM (content management) website to generate interactions. In the XP1 deployment topology, you generate interactions by visiting the CD (content delivery) website.