



Sitecore AI 3.0 Installation Guide for Kubernetes

Sitecore AI Auto-Personalization Standard

January 20, 2022



Table of Contents

1. Introduction	3
2. Prerequisites	4
2.1. Download required packages	4
2.2. Prepare Sitecore AI files	4
2.3. Prepare your Sitecore XP Kubernetes deployment	4
3. Build custom images for Sitecore AI	6
4. Configure the custom images	9
5. Deploy the Sitecore platform with Sitecore AI to a Kubernetes cluster	10

1. Introduction

This guide explains how to deploy the Sitecore AI Auto-personalization Standard module with Sitecore Experience Platform installation using Kubernetes.

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

NOTE

For instructions to deploy Sitecore AI Auto-personalization Standard on a developer workstation using Docker, refer to the [Sitecore AI Installation Guide for Docker Containers](#).

2. Prerequisites

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

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

The Sitecore AI for Kubernetes specification files are designed to be deployed using the Kubernetes Kubectl CLI.

To prepare Kubernetes specification files:

1. Extract the `SitecoreAIStandardContainerDeployment.{version}.{revision}.zip` archive, and copy the content of the `k8s\1tsc2019\xp1\` folder to your *working directory*.
2. Extract the `SitecoreContainerDeployment.10.*.*.zip` archive, and copy the content of `k8s\1tsc2019\xp1` folder to the same *working directory*.

2.3. Prepare your Sitecore XP Kubernetes deployment

To prepare your Sitecore XP Kubernetes deployment:

1. [Download](#) the *Sitecore XP Installation Guide for Production Environment with Kubernetes*, and familiarize yourself with its contents.

2. Ensure that you meet the system requirements specified in the *Sitecore XP Installation Guide for Production Environment with Kubernetes*.
3. In the *XP Installation Guide for Production Environment with Kubernetes*, go to the **Prerequisites** section, and follow the instructions to prepare your Sitecore XP Kubernetes deployment.

3. Build custom images for Sitecore AI

The following instructions assume you have completed all the steps in the "Prerequisites" section of the *Sitecore XP Installation Guide for Production Environment with Kubernetes*.

To prepare the Sitecore XP images to work with Sitecore AI Auto-personalization extension, you must install the Sitecore AI for Auto-personalization asset image on top of the following Sitecore XP1 topology roles:

- Content management (CM)
- Content Delivery (CD)
- Cortexprocessingworker
- Xdbsearchworker
- Xdbcollection
- Xdbrefdata
- Cortexreporting

To build the custom images for the XP1 topology roles:

1. In your Docker installation, create a folder named `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.	xp1
SITECORE_VERSION	The version of the base Sitecore images.	Copy the new <code>Tag</code> value from any <code>kustomization.yaml</code> file from the <code><working directory>/xp1</code> . For example: <code>10.1-ltsc2019</code> , where <code>10.1</code> is the Sitecore version, and <code>ltsc2019</code> 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 <code>ltsc-2019: 3.0.0-ltcs2019</code>

Variable name	Description	Value
SITECOREAI_INTEGRATION_ASSET_IMAGE	The name of the base Sitecore AI asset image.	scr.sitecore.com/sxp/modules/sitecore-ai-xp1-assets
SITECORE_DOCKER_REGISTRY	The Sitecore container registry.	scr.sitecore.com/sxp/

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

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

4. In the `build` folder, create a sub-folder for each XP1 role, and then create the required docker files:

- 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 XDBC_COLLECTION_BASE_IMAGE
ARG SITECOREAI_INTEGRATION_ASSET_IMAGE
FROM ${SITECOREAI_INTEGRATION_ASSET_IMAGE} as sitecoreai_integration
FROM ${XDBC_COLLECTION_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\
```

5. To build custom images that include the Sitecore AI files, in the `build` folder, run the `docker-compose build` command.
6. Push the newly created custom images to a private container registry ready for the deployment process to consume them later.

NOTE

To find out the names and tags of the custom images to push, open the `build/docker-compose.yaml` file, and for each role, look at the value of the `image` field.

4. Configure the custom images

You must configure the custom images in the Sitecore AI Kubernetes configuration files.

To configure the images:

1. In your *working directory*, open the `overlays\Sitecore.AI.Standard\kustomization.yaml` file, and change the values of the `newname` and `newTag` properties to the corresponding name and tag values for the `CM`, `CD`, `cortexprocessingworker`, `xdbsearchworker`, `xdbcollection`, `xdbrefdata`, and `cortexreporting` images that you built and pushed to the private container registry as part of the [previous step](#).
2. In your *working directory*, open the `overlays\Sitecore.AI.Standard\patch-cm.yaml` file, scroll to the end of the file, and enter values for the following variables located under the `env` tag:

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>

3. In your *working directory*, open the `overlays\Sitecore.AI.Standard\patch-cd.yaml` file, and enter the same values as in [step 2](#) above.

5. Deploy the Sitecore platform with Sitecore AI to a Kubernetes cluster

To deploy Sitecore XP containers with Sitecore AI Auto-personalization Standard to a Kubernetes cluster:

1. In the *Installation Guide for Production Environments with Kubernetes for Sitecore XP*, follow the instructions from Chapter 2 - Deploying Sitecore XP to the Azure Kubernetes Service and, when you reach the step to **Deploy the Sitecore pods**, perform the following instead.

NOTE

In the XP installation guide for Kubernetes, references to *root* folder correspond to the `working directory/xp1` you created.

2. To deploy the Sitecore pods, from your *working directory*, run the following command:

```
kubectl apply -k ./overlays/Sitecore.AI.Standard
```

3. To check the status of the pods, run the following command:

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

4. To wait until the status of all the pods is `Running/OK`, run this command:

```
kubectl wait --for=condition=Available deployments --all --timeout=1800s
```

5. Go to the *Installation Guide for Production Environments with Kubernetes Sitecore XP*, and in the **Chapter 2** section, perform the step to **Update the local host file**, and all remaining steps to resume the Sitecore Kubernetes deployment process.

Refer to the Sitecore documentation website for information on how to [configure Sitecore AI auto-personalization](#) [configure Sitecore AI auto-personalization](#).

NOTE

In a XP1 topology configuration, you create interactions by visiting the content delivery (CD) website.