

Data Exchange Framework 7.0 Tenant Service Container Deployment Guide

A guide to deploying the Data Exchange Framework Tenant Service to Docker and
Azure Kubernetes Service

November 25, 2021

Table of Contents

1. Introduction to Tenant Service for containers	3
2. Prepare the installation	4
2.1. Requirements	4
3. Add the Tenant Service module to Sitecore in Docker	5
3.1. Prepare the installation files	5
3.2. Deploy the Tenant Service container	6
3.3. Deploy a plugin to the Tenant Service container	7
4. Add Tenant Service module to Sitecore in Azure Kubernetes Service	8
4.1. Prepare files and folders for deployment	8
4.2. Deploy the AKS container	9

1. Introduction to Tenant Service for containers

Sitecore Data Exchange Framework Tenant Service (TS) enables you to trigger data synchronization processes with multiple systems.

This guide shows you how to deploy TS to Sitecore container installations for Docker and Azure Kubernetes Service.

For more information about TS, see the [Sitecore developer documentation](#).

2. Prepare the installation

This section explains what you need to prepare for deploying the Sitecore Data Exchange Framework Tenant Service (TS) to Sitecore containers for Docker and Azure Kubernetes Service (AKS).

2.1. Requirements

Before you deploy TS to Docker or AKS, the following requirements must be met:

- Docker Desktop must be 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 Downloads 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 Downloads site](#).
- You must have a build of the DEF Docker images on your local machine. For instructions on how to build the Data Exchange Framework images, see the *Data Exchange Framework 7.0 Container Deployment Guide* on the [Sitecore Downloads site](#), and follow the instructions on how to build the Docker images.
- If you have already installed the Data Exchange Framework on Docker, you must have the DEF `docker-compose.override.yml` deployed in the Sitecore container deployment folder.
- If the installation is done on Kubernetes, you must have completed the preparation steps up to the section *Prepare the configuration files for deployment* in the *Data Exchange Framework 7.0 container deployment guide*, available on the [Sitecore Downloads site](#).

3. Add the Tenant Service module to Sitecore in Docker

To add Sitecore Data Exchange Framework (DEF) Tenant Service (TS) module in Docker, you must:

- [Prepare the installation files](#)
- [Deploy the Tenant Service container](#)

3.1. Prepare the installation files

To prepare the files you need for the installation:

1. Download the DEF-TF 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 DEF-TS container deployment package. Locate the folder for the Windows version and topology you are using, for example, `def-ts\compose\ltsc2019\xp1`.
3. Copy the `tenant-service.override.yml` file to the Sitecore container deployment `compose\<version>\<topology>` folder (the same location as the `docker.compose.yml` file).
4. In the folder, open the `.env-example` file in an editor. The following shows an example of the file's content:

```
#The host for the Tenant service website, for example, <topology-to-deploy>ts.localhost
TENANT_SERVICE_HOST=<...>

#The version tag for the Tenant service image
MODULE_VERSION=<...>

#The Tenant Id after it is created in Sitecore
SITECORE_TENANT_ID=<...>

#The Client Id of Sitecore Identity Server
SITECORE_CLIENT_ID=<...>

#The Client Secret of Sitecore Identity Server
SITECORE_CLIENT_SECRET=<...>
```

5. Copy all the variables in the file to the clipboard.
6. In the Sitecore container deployment folder on your local machine, open the `.env` file in an editor, and paste in the variables from the DEF-TS `.env-example` file.
7. Save the `.env` file.

3.2. Deploy the Tenant Service container

When you have prepared the installation files, you must deploy the DEF TS container.

To deploy a TS container:

1. Add HTTPS/TLS certificates for the tenant service. For detailed instructions, see the *Installation Guide for Developer Workstation with Containers* available on the [Sitecore Downloads](#) site.
2. In the Windows console, go to the folder containing the `tenant-service.override.yml` file. Run this command:

```
docker-compose -f docker-compose.yml -f docker-compose.override.yml -f tenant-service.override.yml up --detach
```

3. To test if the tenant service is running, open a browser, and navigate to `https://<topology>ts.localhost`.
4. To update your search indexes, browse to your Sitecore URL, for example, `https://xp1cm.localhost/`. Open the control panel, and click **Populate Solr Managed Schema**. After Sitecore has populated the Solr Schema, click **Indexing Manager**, select the index to rebuild, and then click **Rebuild**.
5. In the Sitecore content management container, create a DEF tenant, and upgrade the tenant.
6. In the DEF tenant, create a DEF tenant service endpoint. Enable the DEF tenant and the DEF tenant service endpoint.
7. Copy the DEF tenant ID from the DEF tenant, and in the `.env` file, paste it in as the value for the `SITECORE_TENANT_ID` variable.
8. Open a PowerShell window with administrator rights, and navigate to your local deployment folder.
9. To update all the containers, run the following command:

```
Docker-compose -f docker-compose.yml -f docker-compose.override.yml -f tenant-service.override.yml up -detach
```

10. To test if the tenant is available, open a browser, and browse to `https://<topology>.localhost/api/tenant`.

NOTE

Some modifications to Sitecore deployments, such as adding connection strings or changing the web configuration files, require you to use configuration transforms to change the configuration files. For information on how to apply configuration transforms, see the Sitecore .

3.3. Deploy a plugin to the Tenant Service container

If you want to use TS in a connector module such as, for example, Sitecore Connect for Salesforce CRM, you must deploy a plugin for that connector.

To deploy a plugin to the TS container:

1. In the Window console, get the name of the TS container by running this command:

```
docker-compose ps
```

2. Stop the TS container by running this command:

```
docker stop <container name>
```

3. Add the plugin by running this command:

```
docker cp <source path> <container name>:"c:/inetpub/wwwroot"
```

NOTE

Replace <source path> with the path to the folder containing the plugin.

4. Start the TS container again by running this command:

```
docker start <container name>
```

4. Add Tenant Service module to Sitecore in Azure Kubernetes Service

This section explains how to add Data Exchange Framework (DEF) Tenant Service (TS) in Azure Kubernetes Service (AKS).

NOTE

For AKS deployments, the preferred URLs for TS connection strings are:

```
host=http://cm.default.svc.cluster.local;  
auth_endpoint=http://id.default.svc.cluster.local/
```

To add TS to AKS, you must:

- [Prepare files and folders for deployment.](#)
- [Deploy the containers using *kubectl* commands.](#)

4.1. Prepare files and folders for deployment

To prepare files and folders in your Kubernetes installation for deployment:

1. Download the Sitecore Experience Platform (SXP) container deployment package from the [Sitecore download page](#). Extract the package and save it, with the folder structure intact, in a temporary folder on your local machine.
2. In the DEF-TS container deployment folder, copy the `def-ts\k8s\ltsc2019\overrides\<topology>` folder to the clipboard.
3. In the Sitecore Experience Platform (SXP) container deployment package, navigate to the `k8s\<version>\overrides` folder, and paste in the `def-ts\k8s\ltsc2019\overrides\<topology>` folder (from the clipboard) as `ts-<topology>` (at the same level as the `def topology` folder).
4. In the SXP container deployment package, in the `overrides\ts-<topology>\ingress-nginx` folder, locate the `kustomization.yaml` file. Make sure that parameters in the `bases:` section are overlays with the Sitecore SXP `ingress-nginx` folder for your installation, for example, `../../../../../xpl/ingress-nginx`.
5. In the SXP container deployment package, in each of the `overrides\ts-<topology>` and `overrides\ts-<topology>\secrets` folders, locate the `kustomization.yaml` file. In each file, make sure that parameters in the `bases` section are overlays with the Sitecore DEF `overrides` folder respectively for your installation, for example, `../xpl`.

NOTE

Parameters in the `bases:` section contain the placement of the original Sitecore container deployment files or Sitecore DEF deployment files that the `kustomization.yaml` files override.

6. In the `overrides\ts-<topology>\kustomization.yaml` file, in the `images:` section, make sure the `newName` and `newTag` parameters have the appropriate values for the TS images.
7. In the `overrides\ts-<topology>\secrets` folder, open the `tenant-service-sitecore-connection.txt` file and update the connection string details. The file contains an example of how the connection string should look.
8. In the `overrides\ts-<topology>\secrets\tls\global-ts` folder, add the TLS certificates `tsl.crt` and `tls.key` for the `ts.globalhost` host. To add HTTP/TLS certificates for the tenant service, see the *Installation Guide for Production Environment with Kubernetes* guide.

4.2. Deploy the AKS container

Before you deploy the containers to Kubernetes, you must 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 container and the necessary Kubernetes components:

1. Open the Windows console, and navigate to the `k8s\<version>` folder.
2. Deploy the secrets. Use this command:

```
kubectl apply -k ./overrides/ts-<topology>/secrets/
```

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

```
kubectl apply -k ./overrides/ts-<topology>/ingress-nginx/
```

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

```
kubectl apply -k ./<topology>/external/
```

5. Wait for all containers to have the status *Ok/Running*. You can check the status with this command:

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

6. Run the `init` folder. Use this command:

```
kubectl apply -k ./<topology>/init/
```

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

```
kubectl get pods
```

8. To create persistent volumes, run this command:

```
kubectl apply -f ./<topology>/volumes/azurefile
```

9. Run the Sitecore containers with the SFCRM changes. Use this command:

```
kubectl apply -k ./overrides/ts-<topology>/
```

10. Wait for all containers to have the status *Ok/Running*. You can check the status with the `kubectl get pods` command. Obtain the external IP address by using this command:

```
kubectl get service -l app=nginx-ingress
```

11. Update the local host file with the external IP address. For information on how to do this, see the *Installation Guide for Production Environment for Kubernetes*, which is available on the [Sitecore download page](#).

12. To check that the service is running, open a browser, and browse to `https:\
\ts.globalhost`.

NOTE

If you get a **Your connection isn't private** response, click **Advanced**, then click **Continue to ts.globalhost (unsafe)** to continue.