SITECORE®

Azure Deployment Guide

Sitecore Horizon 10.2

How to deploy and configure Horizon in Microsoft Azure

Table of Contents

Deploy and configure Horizon 10.2 on Azure

This guide tells you how to deploy Sitecore Horizon to Microsoft Azure®.

Prerequisites

Before deploying Sitecore Horizon, you must:

- Install Azure Powershell

- Deploy Sitecore Experience Platform (XP) 10.2 to your Azure subscription. Refer to the following topic when you deploy Sitecore XP: Deploying a new Sitecore environment to the Microsoft Azure App service.

    Note
    Horizon supports the XP, XP Single, XM, and XM Single topologies.

- Install Sitecore Azure Toolkit 2.5.1 and all its prerequisites on your computer. See Getting started with the Sitecore Azure Toolkit for help.

- Install Sitecore Identity.

SameSite cookies and custom domains

The companies that create commonly-used browsers such as Chrome and Safari are changing how the browsers interpret the SameSite cookie attribute. After the change takes effect, Horizon only works if the Sitecore Horizon host service, the Sitecore identity service, and the Sitecore Content Management (CM) service belong to the same site. (For more information about SameSite cookies, read this article.) We recommend that you assign custom domains to the Sitecore Horizon host and Sitecore identity services.

The following are examples of valid configuration:

Example 1
Sitecore CM: `https://cm.mycompany.com`
Sitecore identity: `https://si.mycompany.com`
Horizon: `https://horizon.mycompany.com`

Example 2
Sitecore CM: `https://cmdomain.com`
Sitecore identity: `https://si.cmdomain.com`
Horizon: `Horizon.cmdomain.com`.

If you use custom domains for the services, then after you deploy Horizon on Azure, you must also complete the steps described in the Configure services for your custom domain section of this document.

Prerequisites to integrate Content Hub with Horizon

To work with Content Hub media in Horizon, you must:

1. Deploy and configure Sitecore Content Hub and install Sitecore Connect™ for Content Hub. (Compatible versions are listed here.)

2. Enable content Hub integration for Horizon during Horizon deployment (see Deploy Horizon on Azure).

3. Configure the CORS settings for Sitecore DAM.

Prerequisites to integrate Sitecore Experience Accelerator (SXA) with Horizon

To work with SXA in Horizon, you must:

1. Deploy and configure the appropriate version of the Sitecore Experience Accelerator module.

2. Enable SXA module integration for Horizon during Horizon deployment (see Deploy Horizon on Azure).

Get Sitecore Horizon deployment files

To get the Horizon deployment files:

1. Download the Horizon Azure deployment .zip file (`Sitecore Horizon for Azure 10.2.0.zip`).

2. Unpack the .zip file.

Upload Horizon installation components to Azure Storage

Before you can deploy Horizon on Azure, you must upload the Azure Resource Manager (ARM) templates and Web Deploy packages (WDPs) containing the application code and resources to a Microsoft Azure storage account.

To host ARMs and WDPs in a Microsoft Azure storage account:

1. Create a Microsoft Azure storage account.

2. In the Microsoft Azure storage account menu, click **Blob Service**, **Containers**, and then add a container.

3. Use Azure Storage Explorer to connect to your Microsoft Azure storage account.

4. In Azure Storage Explorer, navigate to the deployment files that you downloaded and unzipped previously. From the `\Sitecore Horizon for Azure 10.2.0` folder, upload the `\arms` and `\wdps` folders to the container that you created in the storage account.

5. In the Sitecore Azure Toolkit folder, find the Sitecore Bootloader module, `Sitecore.Cloud.Integration.Bootload.wdp.zip`. Upload it to the container you created in the storage account.

6. In Azure Storage Explorer, for the Sitecore Bootloader module, and each ARM template and WDP package, create a Shared Access Signature (SAS) token, and make a note of the value in the URL string field.

   The SAS token allows temporary access to the WDPs during the deployment process. When you create the SAS token, ensure that you allow enough time between the *Start* time and the *Expiry* time to guarantee access during the scheduled deployment. Make a note of the URLs because you will need them for deployment later.

Deploy Horizon on Azure

Horizon on Azure uses the `HTTPS` protocol by default. To set up `HTTPS` on Azure App Service, you must provide your own SSL certificate signed by a certificate authority or generate a self-signed certificate for a development environment.

To deploy Horizon on Azure:

1. To generate a self-signed certificate, in Windows, launch PowerShell as an administrator.

2. In a PowerShell command line, run the `generateCertificate.ps1` cmdlet.

3. When prompted, enter a password for the generated certificate.

The script generates two files in the current folder: a certificate file, and a BLOB file that contains the content of the certificate file in base64 format.

4. To deploy Horizon on Azure, in the `parameters.json` file, insert the URLs of the WDP packages that you uploaded to your Azure Storage account, and update parameters as follows:

Parameter name	Information to enter
`horizonHostMsDeployPackageUrl`	The URL of the `HorizonHost.{version}.scwdp.zip` package.
`featureIntegrationMsDeployPackageUrl`	The URL of the `Sitecore Horizon {topology} Integration {version}.scwdp.zip` package. (`Topology` is either `XP` or `XM`, depending on the Sitecore topology type deployed.)
`sitecoreIdentityIntegrationMsDeployPackageUrl`	The URL of the `SitecoreIdentityIntegration.{version}.scwdp.zip` package.
`bootloaderMsDeployPackageUrl`	The URL of the `Sitecore.Cloud.Integration.Bootload.wdp.zip` package.
`authCertificateBlob`	The content of the generated certificate from the BLOB file.
`authCertificatePassword`	The certificate password.
`deploymentId`	The Sitecore deployment ID. Each service in the resource group is named with the deployment ID followed by a suffix that represents the service. For example, when the SQL service is named *sitecore-xp0-sql* then the Sitecore deployment ID is *sitecore-xp0*.

Parameter name	Information to enter
enableContentHub	Set to `true` to enable ContentHub integration with Horizon on the deployed Sitecore instance, or `false` to disable the integration. If set to `true`, it enables the `Sitecore/Plugins/Filters/ContentHub` node in the `sitecorehost.xml` file, in the Horizon site physical path.
enableSXA	Set to `true` to enable SXA integration with Horizon on the deployed Sitecore instance, or `false` to disable the integration.
solrCorePrefix	The prefix of the Sitecore Solr core index for Horizon. Make sure it is aligned with the prefix for the sitecore_master_index from your current deployment.

5. In the `DeployHorizon.ps1` script, set the following parameters:

Parameter name	Information to enter
ResourceGroupName	The name of the Azure resource group that Sitecore is deployed to.
LicenseXmlPath	The Sitecore license file location including the file name. For example, `C:\Project\Files\license.xml`
Location	The name of the Azure datacenter where you want the resources deployed. The Azure datacenter compatibility table lists the Microsoft Azure datacenters that you can deploy Sitecore XP to.
ParametersPath	The path to the `parameters.json` file.
SitecoreAzureToolkit	The location of the `/Sitecore Azure Toolkit` folder.
SubscriptionName	The name of the Azure Subscription you want to deploy to.
TemplateUri	The URL of the relevant Horizon ARM template that you uploaded to the storage account. (The ARM folder contains templates for XP, XP Single, XM, and XM Single topologies–enter the URL for the template for the relevant topology.)

6. In Windows, launch the PowerShell command line.

7. Run the `DeployHorizon.ps1` cmdlet.

8. Log in to Microsoft Azure when you are required to, soon after the script starts.

9. After the script is successfully finished, on the Azure Portal website, restart the following service:

   o Sitecore Identity Server WebApp named `your deployment id-si`

   Now you can access the Horizon website at: `https://your_deployment_id-hrz.azurewebsites.net.`

Update the style sheet

Before you can use the alignment and indentation features in the rich text editor in Horizon, you must define some styles in your style sheet.

To enable alignment and indentation in the rich text editor:

- Add the following lines to your default style sheet:

```
.rte-indent-1 {
  padding-left: 40px;
}
.rte-indent-2 {
  padding-left: 80px;
}
.rte-indent-3 {
  padding-left: 120px;
}
.rte-indent-4 {
  padding-left: 160px;
}
.rte-indent-5 {
  padding-left: 200px;
}
.rte-indent-6 {
  padding-left: 240px;
}
.rte-indent-7 {
  padding-left: 280px;
}
.rte-indent-8 {
  padding-left: 320px;
}

.rte-align-center {
  text-align: center;
}
.rte-align-justify {
  text-align: justify;
}
.rte-align-right {
  text-align: right;
}
```

Install and enable Content Hub after installing Horizon

If you did not install Sitecore Content Hub before you installed Horizon, you can enable Sitecore Content Hub integration for Horizon after you install both Horizon and Content Hub.

To enable Content Hub integration after installing Horizon:

1. Make sure you have installed and configured Sitecore Content Hub and Sitecore Connect™ for Content Hub.

2. Open the `sitecorehost.xml` file, which is in the root folder of the Sitecore Horizon site physical path.

3. In the Plugins section, uncomment the `<ContentHub>+ContentHub</ContentHub>` line.

4. Delete the line `<ContentHub>-ContentHub</ContentHub>` if it exists. The result looks like this:

```
<Plugins>
  <Filters>
    <ExplicitEnablement>-ExplicitEnablement</ExplicitEnablement>

    <!-- Uncomment the line below to enable ContentHub plugin -->
    <ContentHub>+ContentHub</ContentHub>
  </Filters>
</Plugins>
```

Configure the CORS settings for Sitecore DAM

If you have enabled integration with Content Hub, you must add the Horizon site URL to the CORS settings for your Sitecore DAM™ instance.

To update the CORS settings for your Sitecore DAM instance:

- In the Sitecore Connect™ for Content Hub documentation, follow the steps in the CORS configuration section, but add the Sitecore Horizon site URL instead of the Sitecore instance URL.

Install and enable SXA after installing Horizon

If you did not install Sitecore Experience Accelerator (SXA) before you installed Horizon, you can enable SXA integration with Horizon after you install both Horizon and the SXA module.

To enable SXA integration after installing Horizon:

1. Make sure you have installed and configured the Sitecore Experience Accelerator module and all its prerequisites.

2. Open the `sitecorehost.xml` file, which is in the root folder of the Sitecore Horizon site physical path.

3. In the Plugins section, uncomment the `<ExperienceAccelerator>+SXA</ExperienceAccelerator>` line.

4. Delete the `<ExperienceAccelerator>-SXA</ExperienceAccelerator>` line if it exists.

   The result looks like this:

```
<Plugins>
  <Filters>
    <ExplicitEnablement>-ExplicitEnablement</ExplicitEnablement>

    <!-- Uncomment the line below to enable SXA plugin -->
    <ExperienceAccelerator>+SXA</ExperienceAccelerator>
  </Filters>
</Plugins>
```

5. Restart the Horizon application service.

Now you can work with SXA pages and controls in the Horizon application.

Configure services for your custom domain

If you have followed the recommendation to assign a custom domain to your Sitecore Horizon host service and Sitecore identity service, you must complete the configuration of the services with the custom domain addresses.

To configure services for your custom domain:

1. Access the file system of the Sitecore Content Management service named either `your deployment id-cm` service or `your deployment id-single` service (the name depends on your topology).

   o Open `wwwroot\App_Config\Modules\Horizon\Sitecore.Horizon. Integration.config` and replace the value of the `Horizon.ClientHost` setting with your Horizon custom domain address.

2. Access the file system of the Sitecore Horizon service named `your deployment id-hrz` service.

   o Open `wwwroot\Config\Host.xml` and replace the value within the `<HostBaseUrl> </HostBaseUrl>` tags with your Horizon custom domain address.

   o Open `wwwroot\Config\Host.xml` and replace the value within the `<SitecoreHost> </SitecoreHost>` tags with your CM custom domain address.

   o Open `wwwroot\Config\OpenIdConnect.xml` and replace the value within *two sets* of `<Authority> </Authority>` tags with your Sitecore Identity custom domain address.

      Note
      You must update both sets of `<Authority> </Authority>` tags.

3. Access the file system of the Sitecore Identity server named `your deployment id-si` service.

   o Open `wwwroot\Config\production\Sitecore.IdentityServer. AuthoringHost.xml` and replace the value within the `<HorizonHttpsCorsOrigin> </HorizonHttpsCorsOrigin>` tags with your Horizon custom domain address.

4. Restart all three services.

Prepare the Solr search engine

For the search feature to work in Horizon, after you deploy Horizon, you must populate the Solr managed schema and build the Horizon index.

To prepare the Solr search engine for Horizon:

1. Log into Sitecore.

2. On the Sitecore Launchpad, click **Control Panel**.

3. In the **Indexing** section, click **Populate Solr Managed Schema**.

4. In the **Schema Populate** dialog box, ensure that the **sitecore_horizon_index** and the **sitecore_master_index** checkboxes are selected.

5. Click **Populate**.

9

6. When the **Schema Populate** dialog box indicates that the indexes have been populated successfully, click **Close**.

7. On the **Control Panel**, in the **Indexing** section, click **Indexing Manager**.

8. In the **Indexing Manager** dialog box, select the **sitecore_horizon_index** and the **sitecore_master_index** checkboxes and then click **Rebuild**.

9. When the search indexes have been rebuilt, click **Close**.

Troubleshooting Horizon Search

If Horizon Search does not work, repeat the steps above, except after step 6, restart the CM instance in your resource group before continuing to step 7.