Azure Deployment Guide

Sitecore Horizon 10.0.1

How to deploy and configure Horizon in Microsoft Azure

Deploy and configure Horizon 10.0.1 on Azure

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

Prerequisites

Before deploying Sitecore Horizon, you must:

- Deploy Sitecore Experience Platform (XP) 10.0.1 to your Azure subscription. Refer to the following topic when you deploy Sitecore XP: <u>Deploying a new Sitecore environment to the Microsoft Azure App service.</u>

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

- Install Sitecore Azure Toolkit 2.5.0 and all its prerequisites on your computer. See <u>Getting started with the Sitecore Azure Toolkit</u> 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 <u>SameSite cookie attribute</u>. After the change takes effect, Horizon will only work 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 <u>this article</u>.) 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 <u>Sitecore Content Hub 4.0</u> and install <u>Sitecore Connect™ for Content Hub 4.0</u>.

2. Enable content Hub integration for Horizon during Horizon deploy (in Deploy Horizon on Azure section).

3. Configure the CORS settings for Sitecore DAM.

Get Sitecore Horizon deployment files

To get the Horizon deployment files:

1. Download the Horizon Azure deployment .zip file (`Sitecore Horizon for Azure 10.0.1.zip`).

2. Unpack the .zip file.

Upload Horizon ARM templates, WDP packages, and the Sitecore Bootloader module 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 <u>storage account</u>.

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

3. Use <u>Azure Storage Explorer</u> 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 10.0.1` 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. In order 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 `Horizon.FeatureIntegration {version}.scwdp.zip` package.
`featureIntegrationInsightsMsDeployPackageUrl`	The URL of the `Horizon.FeatureIntegration.Insights {version}.scwdp.zip` package. **Note** Remove this token from `parameters.json` if your Sitecore instance uses the XM topology.
`featureIntegrationZenithMsDeployPackageUrl`	The URL of the `Sitecore Zenith Integration {version}.scwdp.zip` package.
`sitecoreIdentityIntegrationMsDeployPackageUrl`	The URL of the `SitecoreIdentityIntegration.{version}.scwdp.zip` package.
`bootloaderMsDeployPackageUrl`	The URL of the `Sitecore.Cloud.Integration.Bootload.wdp.zip` package.
`sqlServerLogin` and `sqlServerPassword`	The Azure SQL server permissions used for the Sitecore XP deployment.
`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.

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 <u>Azure datacenter compatibility table</u> 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:

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

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

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

5. In the Sitecore Horizon site physical path, open the `sitecore\Sitecore.Horizon.ContentHub.Dam.Plugin\Config\Plugin.xml` file.

6. Enter the URL of the Sitecore Content Management site in the `PlatformUrl` parameter.

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:

- Follow the steps in the 3.2.1 Configure CORS section of the <u>Sitecore Connect™ for Content Hub 4.0 documentation</u>. In step 4, add the Sitecore Horizon site URL instead of the Sitecore instance URL.

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\sitecore\Sitecore.Horizon.Authoring.Plugin\Config\Host.xml` and replace the value within the `<HostBaseUrl> </HostBaseUrl>` tags with your Horizon custom domain address.

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

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

    Note
    Ensure that you update both sets of `<Authority> </Authority>` tags.

    o Open `wwwroot\sitecore\Sitecore.Horizon.ContentHub.Dam.Plugin\Config\Plugin.xml` and replace the value within the `<PlatformUrl> </PlatformUrl>` tags with your CM custom domain address.

o Open `wwwroot\sitecore\Sitecore.Zenith\Config\Sitecore.Zenith.Proxy.xml` and replace the value within the `<DataServiceEndpoint> </DataServiceEndpoint>` tags with your CM custom domain address.

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.