

# Azure deployment guide

## Sitecore Horizon 9.3

*How to deploy and configure Horizon*

# Deploy and configure Horizon 9.3 on Azure

This guide tells you how to deploy Sitecore Horizon to Microsoft Azure<sup>®</sup>.

## Prerequisites

Before installing Sitecore Horizon, you must have:

- Sitecore Experience Platform (XP) 9.3 or later version installed on your Azure subscription. Refer to the following walkthrough topic when you install Sitecore XP: [Walkthrough: Deploying a new Sitecore environment to the Microsoft Azure App service](#).

### Note

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

- Sitecore Azure Toolkit 2.4.0 and all its prerequisites installed on your computer. See [Getting started with the Sitecore Azure Toolkit](#) for help.
- Sitecore Identity installed.
- Google has announced plans to change how Chrome interprets the [SameSite cookie attribute](#), with the change taking effect on February 4th. After the change takes effect, Horizon will only work if the Sitecore authoring host service and the Sitecore identity service are under the same domain as the Sitecore Content Management instance. To prepare for this change, we recommend that you assign custom domains to the Sitecore authoring host and Sitecore identity services. The custom domains must be subdomains of the Sitecore Content Management instance. For example, if your Sitecore CM instance has the custom domain `https://cmdomain.com`, then Sitecore identity must have the custom domain `identity.cmdomain.com` and Horizon authoring must have the custom domain `authoringHost.cmdomain.com`.

## Get Sitecore Horizon installation files

To get the Horizon installation files:

1. Download the Horizon Azure installation .zip file (`Horizon.Azure.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 [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 installation files that you downloaded and unzipped previously. From the `\Horizon.Azure` folder, upload the `\arms` and `\wdps` folders to the container that you created in the storage account.
5. Find the Sitecore Bootloader module, `Sitecore.Cloud.Integration.Bootload.wdp.zip`, in the Sitecore Azure Toolkit folder. 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.

## Install 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 generate a self-signed certificate:

1. 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.

To install Horizon on Azure:

4. 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
<code>authoringHostMsDeployPackageUrl</code>	The URL of the <code>AuthoringHost.{version}.scwdp.zip</code> package.
<code>featureIntegrationMsDeployPackageUrl</code>	The URL of the <code>Horizon.FeatureIntegration.{version}.scwdp.zip</code> package.
<code>featureIntegrationInsightsMsDeployPackageUrl</code>	The URL of the <code>Horizon.FeatureIntegration.Insights.{version}.scwdp.zip</code> package.  <b>Note</b> Remove this token from <code>parameters.json</code> if your Sitecore instance uses the XM topology.
<code>sitecoreIdentityIntegrationMsDeployPackageUrl</code>	The URL of the <code>SitecoreIdentityIntegration.{version}.scwdp.zip</code> package.

Parameter name	Information to enter
<code>bootloaderMsDeployPackageUrl</code>	The URL of the <code>Sitecore.Cloud.Integration.Bootload.wdp.zip</code> package.
<code>sqlServerLogin</code> and <code>sqlServerPassword</code>	The Azure SQL server permissions used for the Sitecore XP installation.
<code>authCertificateBlob</code>	The content of the generated certificate from the BLOB file.
<code>authCertificatePassword</code>	The certificate password.
<code>deploymentId</code>	You can derive the deployment ID from the service names in the resource group. 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 <code>sitecore-xp0-sql</code> then the deployment ID is <code>sitecore-xp0</code> .
<code>sitecoreIdentityServerClientSecret</code>	To derive this value, you need to access the file system of the Sitecore Identity server named <code>your deployment id-si</code> . In the file system, open <code>\Config\production\Sitecore.IdentityServer.Host.xml</code> . Copy the value of the <code>ClientSecret1</code> parameter to the <code>sitecoreIdentityServerClientSecret</code> parameter.

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

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

6. In Windows, launch PowerShell as an administrator.

7. In a PowerShell command line, run the `DeployHorizon.ps1` cmdlet.
8. On the Azure Portal website, restart the following services:

- Sitecore Identity Server WebApp named `your deployment id-si`
- Sitecore Authoring Host WebApp named `your deployment id-ah`.

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

## Update style sheet

Before you can use the alignment and indentation features in the rich text editor in Horizon, you must define a number of styles in your style sheet. To enable alignment and indentation in the rich text editor:

- Add the following lines to your default stylesheet:

```
.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;
}
```

## Update openIdConnect.xml

If you change the `ClientSecret1` parameter in the `\Config\production\Sitecore.IdentityServer.Host.xml` file for the Sitecore IdentityServer service after you deploy Horizon, you must update the value of the `ClientSecret1` parameter in the `\Config\openIdConnect.xml` file for the Horizon service.