

Sitecore Azure 8.0

All the official Sitecore documentation.



sitecore[®]
Own the experience[™]

Getting started with Sitecore Azure

Sitecore Azure provides an integrated user interface to manage your cloud delivery infrastructure and simplify the deployment of your website to the Microsoft Azure cloud. You can use Sitecore Azure with Sitecore to host your website in the Microsoft Azure cloud.

Sitecore Azure is built around Microsoft Azure cloud services (PaaS), and includes automatic deployment of web servers as well as SQL databases. You can also deploy from the on-premise content editing server to any number of Sitecore Azure cloud services that are hosted at various Microsoft Azure locations globally. This means continuous delivery because you can deploy code and Sitecore items to the Sitecore Azure cloud on demand.

[Microsoft Azure](#) is a cloud-computing platform and an infrastructure for building, deploying, and managing applications and services through a global network of data centers managed by Microsoft.

If you already have an existing Microsoft Azure subscription, you can continue to use it for your Sitecore installation by providing your subscription information, and you still gain the benefits of using Sitecore Azure tools.

Sitecore Azure provides:

- Extreme scalability, simplicity, and stability.
- A highly available infrastructure with 99.95% uptime with Microsoft Azure's service licensing agreement (SLA): <http://www.windowsazure.com/en-us/support/legal/sla>.

Note

You must have two or more instances deployed in the same availability set to qualify for guaranteed 99.95% of external connectivity.

- Significantly reduced costs and time required to deploy your solution to a new server.
- Greater savings on the cost and time required to provide geographically distributed load balancing.
- Reduced redundancy between geographies for performance, as well as disaster preparedness.
- Reduced cost and time required to adjust server capacity to meet the expected load.
- Automation of most of the process of migrating a website to Microsoft Azure.

Deployment options for Sitecore Azure 8.0

Sitecore Azure 8.0 supports content editing as well content delivery from the cloud. You can choose the best combination of on-premise or cloud-based editing and delivery environments. An editing environment is called an editing farm and a delivery environment is called a delivery farm. A farm is a set of web roles connected to a single SQL Azure instance.

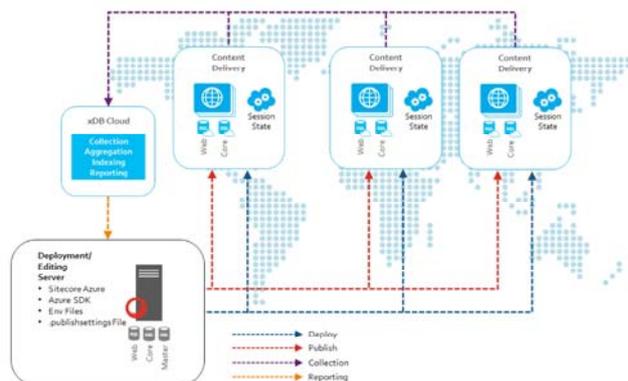
The following sections describe three examples of possible deployment topologies.

Example 1: Sitecore Azure delivery

This example illustrates an on-premise Sitecore installation for content editing that publishes to several Sitecore content delivery farms in multiple Microsoft Azure locations across the globe.

In this type of installation:

- Content editing is carried out on the on-premise server.
- Content delivery server clusters are deployed to Microsoft Azure.
- The on-premise server is published to Microsoft Azure.

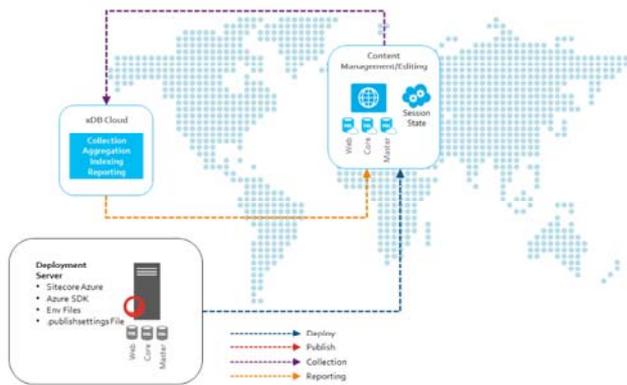


Example 2: Sitecore Azure live

This example illustrates a Sitecore installation that runs in live mode. Both content editing and content delivery run on the same farm, so no publishing is required. Editing changes that are made on the content editing server are immediately available to the public.

In this type of installation:

- The on-premise development environment deploys the full environment to Microsoft Azure. This includes both the Content editing and Content delivery environments.
- The Master database is maintained by Microsoft Azure.
- You can have a multi-instance live-mode deployment within a single farm.
- The front end reads from the Master database.

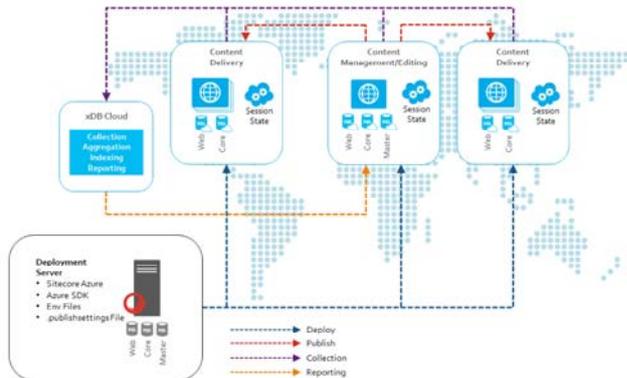


Example 3: Sitecore Azure authoring and delivery

This example illustrates how both the content editing and content delivery environments run in the Microsoft Azure cloud.

In this type of installation:

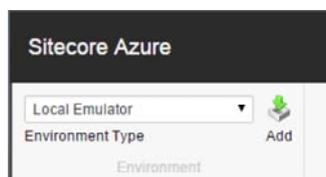
- The on-premise development environment deploys the full environment to Microsoft Azure. This includes both the Content editing and Content delivery environments.
- The Master database is maintained by Microsoft Azure.
- You can have multi-instance authoring deployment within a single Microsoft Azure location which can publish content to all of the other locations in Microsoft Azure.
- Content delivery deployments are rolled out from the on-premise server and published directly from Microsoft Azure's content editing server.



Using Microsoft Azure's local emulator

One of the features of Sitecore Azure is that you can deploy directly to the Microsoft Azure local emulator that is included with the Azure SDK. This means that you can test your deployment without incurring any cost.

You can test deploy to the local emulator, in Sitecore Azure, on the ribbon, in the Environment Type group, by selecting Local Emulator from the drop-down list.



Note

To run the Microsoft Azure local emulator, you must have administrative rights to the computer. You also need to download and run the `Runner.bat` file when prompted in the notification field at the bottom of the dialog.

Use Sitecore xDB or Sitecore xDB Cloud Edition with Microsoft Azure

The [Sitecore Experience Database \(xDB\)](#) and [Sitecore xDB Cloud Edition](#) collects all online and offline customer interactions from all channel sources in a real-time big data repository. Sitecore xDB connects interaction data to create a comprehensive, unified view of each individual customer, enabling marketers to manage the customer experience in real time.

With a Sitecore 8.0 installation, you can set up Sitecore as an on-premise solution, as a hybrid solution (partly on the premises and partly in the cloud), or as an entirely in-the-cloud solution.

Scalability options

Sitecore xDB is a [scalable architecture](#) that can handle large amounts of customer data. You can scale up or down from small single-server, low traffic solutions to larger solutions consisting of hundreds of servers designed to handle high volumes of traffic.

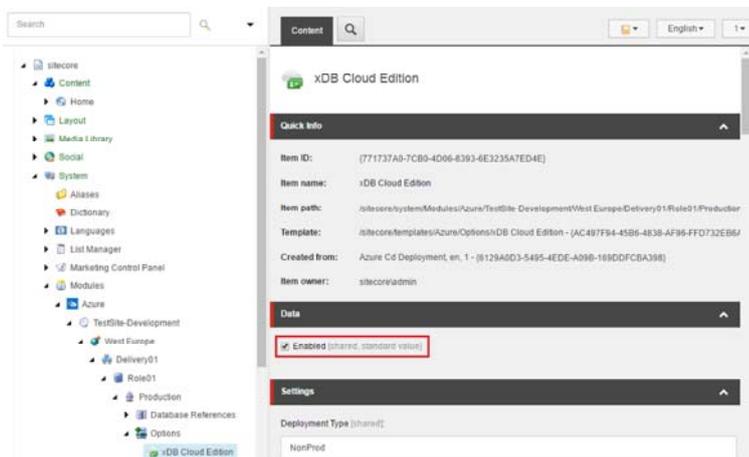
Send feedback about the documentation to docsite@sitecore.net.

Connect to xDB Cloud 2.0 from Sitecore Azure

Configure xDB Cloud 2.0 the way you want it with open tenancy for Azure, increased privacy, and 24/7 support.

To configure your deployment for xDB Cloud 2.0,

1. In the content tree, go to System/Modules/Azure/< your environment>/<your deployment location>/< your farm>/< your web role>/<your azure deployment>/Options/xDB Cloud Edition.
2. In the Data section, disable xDB Cloud by clearing the checkbox.

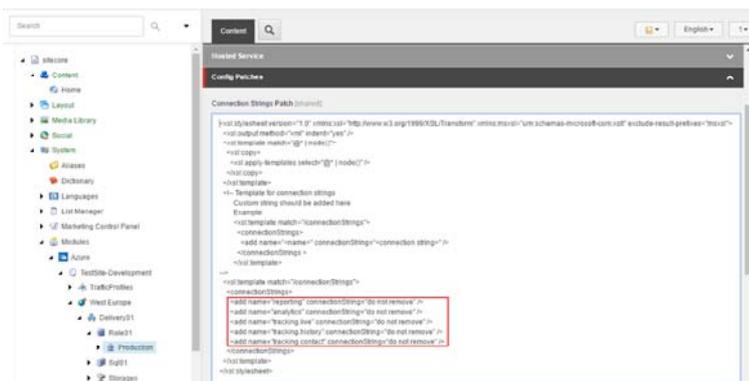


1. To configure your deployment, work through [Connect to xDB Cloud 2.0](#)

Important

Not all steps in [Connect to xDB Cloud 2.0](#) apply to Azure users. Take note of the following:

- Skip the section *Prepare a scaled on premise Sitecore environment*, it does not apply.
- In the section, *Prepare a standalone Sitecore instance*, you must also disable the following configuration file by adding `.disable` to the extension `/App_Config/Include/ExperienceAnalytics/Sitecore.ExperienceAnalytics.Reduce.config`
- In the section, *Prepare a standalone Sitecore instance*, you must enable the following configuration files by removing `.disable` from the extension `/App_Config/Include/Sitecore.Xdb.Remote.Client.MarketingAssets.config.disabled` and `/App_Config/Include/Sitecore.Xdb.Remote.Client.MarketingAssets.config.disabled`
- In the section, *Configure the xDB Cloud client and connection to the xDB Cloud service*, add all of the `connectionString` changes to your deployment in the Connection Strings Patch field, instead of the `ConnectionStrings.config` file. This replaces your local MongoDB database connection strings on your deployments.



When you complete the rest of the steps in [Connect to xDB Cloud 2.0](#), your deployment server should be ready to deploy Sitecore with xDB Cloud 2.0 enabled for Azure.

Send feedback about the documentation to docsite@sitecore.net.

Install Sitecore Azure

Before you can install Sitecore Azure, you must have Sitecore CMS 8.0 installed on your on-premise content editing server, and you must download the Sitecore Azure installation package from the [Sitecore Developer Portal](#). If you want to have separate environments for development, testing, and production, you can do this because you have more than one environment file. Sitecore Azure lets you manage all of these environments from within the same Sitecore installation.

This topic outlines how to:

[Prepare your system requirements](#)

[Deploy Sitecore to Azure](#)

[Install the Sitecore Azure installation package](#)

[Deploy to the Microsoft Azure Cloud](#)

[Disable the Analytics feature](#)

Prepare your system requirements

Before you install Sitecore Azure, you must have the following system requirements in place on the on-premise content editing server (Web Server):

- Sitecore XP 8.0 rev. 141212 (8.0 Initial Release) or later.
- Sitecore xDB Cloud Subscription.
- Microsoft Windows 64-bit.
- Microsoft .NET Framework v 4.5 or higher.
- A Microsoft account (formerly Windows Live ID) with a valid Azure subscription. A thirty-day free trial is available at: <http://www.windowsazure.com/en-us/pricing/free-trial/>.
- Download and install the following components of the [Microsoft® SQL Server® 2012 Feature Pack](#):
 - Microsoft® System Clear Types for Microsoft SQL Server 2012
 - Microsoft® SQL Server® 2012 Shared Management Objects
- Download and install [Microsoft Azure SDK for .NET version 2.5.1](#)
- Allow outgoing connections to the Microsoft Azure data centers on ports: 80 (HTTP), 443 (HTTPS), 8443 and 1433 (SQL). You can also [download the full list of IP ranges for Microsoft Azure data centers](#).

Note

To use Analytics in the Sitecore Cloud, you first need to [configure the Sitecore xDB Cloud client](#).

Deploy Sitecore to Azure

To deploy Sitecore to Azure, you must have a Sitecore Azure environment file. Sitecore Azure environment files are configuration files that contain information about your Microsoft Azure subscription.

Important

If you do not have a Sitecore Azure environment file, submit a request for it *before* you install Sitecore Azure. It can take up to 48 hours to receive the file. However, while you wait for the environment file, you can install the Sitecore Azure package on your Sitecore installation and use the local emulator to test your deployment without incurring any costs.

To request a Sitecore Azure Environment file, in the Sitecore Azure Environment Setup dialog box, enter your contact information, a project name, and your Sitecore license file.

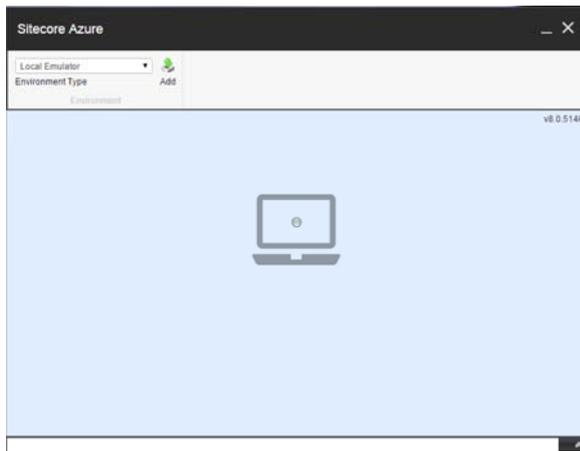
If you are in China then request a Sitecore Azure Environment file [here](#).

Install the Sitecore Azure installation package

You can download the Sitecore Azure installation package from dev.sitecore.net.

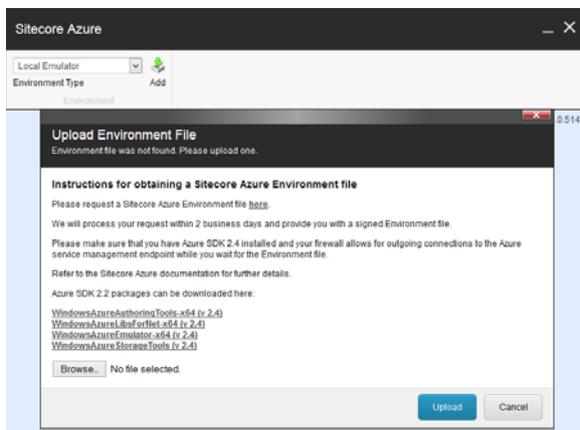
To install the Sitecore Azure installation package:

1. Log in to your on-premise content editing server as an administrator and open the Sitecore Desktop.
2. On your Sitecore desktop, click Sitecore Start, Development Tools and use the Installation Wizard to install the *Sitecore Azure* package. When prompted, overwrite the existing components.
3. When the installation completes, on the Sitecore desktop, click Sitecore Start, Sitecore Azure. The *Local Emulator* environment is represented by a single grey dot.



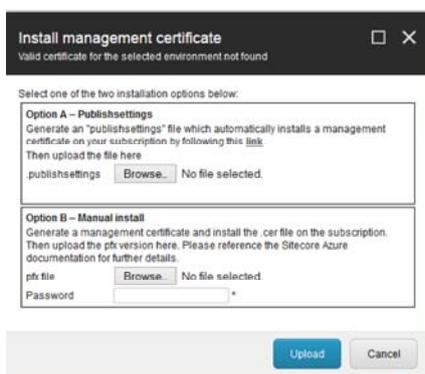
4. To upload your *Sitecore Azure* environment file, click Add.
5. In the Upload Environment File dialog box, click Browse and locate the `Environment.xml` file, and then click Upload.

At this point, you can set up a test environment with the *Microsoft Azure Emulator* or select a test or production environment.



Note

If you are using your own environment file, you will receive a prompt to upload a management certificate the next time you launch Sitecore Azure. Option A is the simplest way to do this.

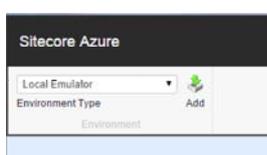


Deploy to the Microsoft Azure Cloud

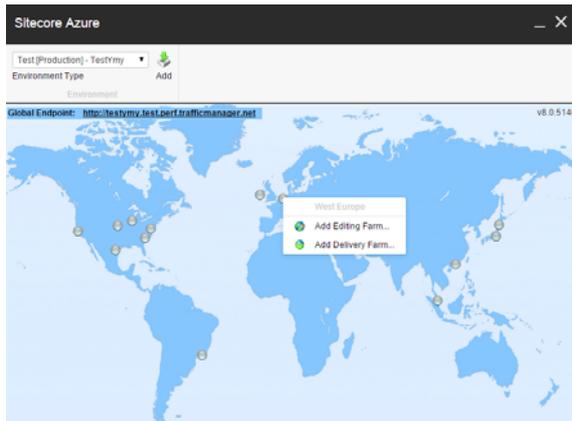
The environment types that you can deploy to depend on the type of Microsoft Azure subscription you have, and there are many ways that you can set up [Sitecore to assemble a solution to deploy](#). You can avoid publishing after deployment by going to `web.config`, `<setting name="DefaultPublishingTargets" value="" />`, and setting the default publishing target to `Internet` (the Web database) if you have not already created a Sitecore Azure farm. You can minimize deployment time by removing unnecessary files from the document root of the content editing environment.

To add and deploy a new editing farm or a delivery farm to a Microsoft Azure hosting location:

1. In Sitecore Azure, on the ribbon, in the Environment Type group, choose the environment type you want to deploy to from the drop down box and click Add.



On the map, click the relevant Microsoft Azure hosting location, and add the farm type that you want to create. Sitecore Azure supports only one Editing farm and multiple Delivery farms.



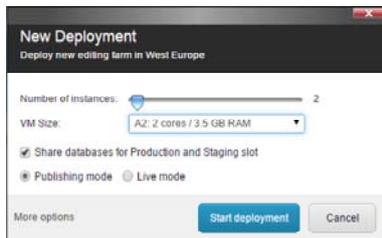
2. In the New Deployment dialog box, specify the settings for instances and web roles:
 - Number of instances – Specify the number of instances you want to deploy. To ensure high availability, you should choose at least two instances. You can also [change the number of instances in an environment](#).
 - VM Size – Specify the size of the virtual machine. The default size is *Medium*, but if you add multiple farms, you should scale up to a larger size. Use a small size for evaluation and test deployments.

Note

Microsoft Azure does not currently support changing the web role size after a deployment is made. You should make a new deployment to the staging slot with the larger Virtual Machine size and swap when the deployment is up and running.

- Share databases for Production and Staging slot – The default database selection is to share databases between Production and Staging. If you have separate databases for production and staging, you should clear this check box.
- Publishing mode – Select this mode to use a controlled publishing environment when you create an editing farm.
- Live mode – Select this mode to make the changes in the content editing farm available on your website immediately. In [live mode](#), the content is not automatically transferred to the delivery farms.

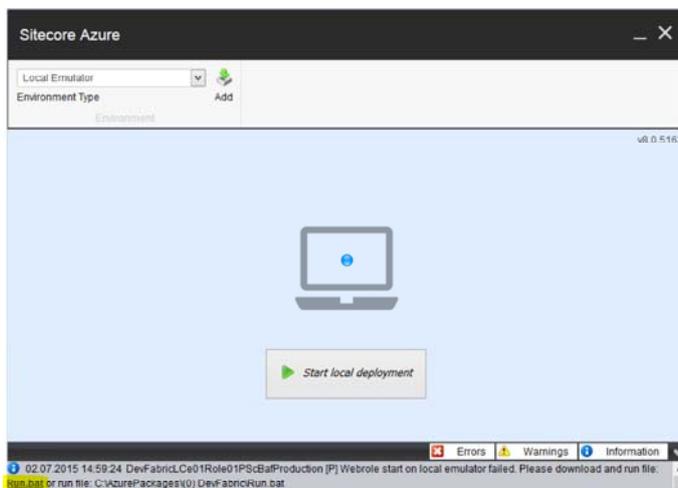
For additional configuration options, click More options to open the farm that you are creating in the Content Editor.



3. Click Start deployment.

Note

The deployment processing time takes can vary greatly between environments, depending on the size of the solution, network conditions, and other factors. Expect a minimum of 15 minutes. Creating storage accounts usually takes a long time so the deployment time will decrease after the first deployment to a single location. This is because deployments in the same location share storage accounts.



1. Make sure you are logged in as an administrator. Click Run.bat.

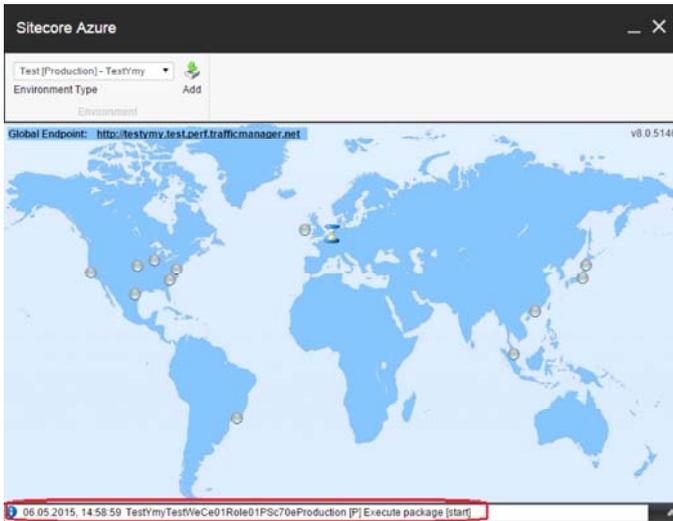
You can follow the progress of the deployment in the info bar at the bottom of the Sitecore Azure dialog box.

Important

If the error *bat loader could not be found* occurs, rename `RunDevFabricDownloader` to `batdownloader` (`Website/sitecore/shell/Applications/Azure`).

Azure deploys your site to a localhost using the next port that is available, (not port 80). In `Data/AzurePackages/(0)DevFabric`, you can see the files that are being deployed and the folder hierarchy. Note that with each deployment the number (0) increases incrementally.

During the deployment process of an editing farm, you can create a delivery farm on another location.

**Disable the Analytics feature**

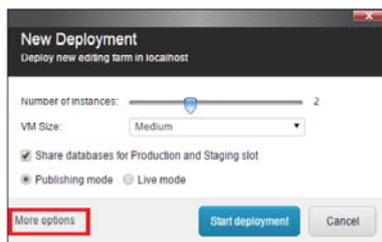
To disable Analytics, go to the Config patches section of Sitecore Azure. In the `Sitecore.Analytics.config` file you can change the Analytics setting.

Important

If you do not have a valid subscription for Sitecore xDB Cloud, you must disable the Analytics feature before you start deployment.

To change the `Analytics.Enabled` setting:

1. Add an *Editing* or a *Delivery* farm.
2. In the New Deployment dialog, click More Options. The relevant Azure Deployment item appears in the Content Editor.



3. In the Config patches section, in the Custom Web Config Patch field, add the following setting patch.

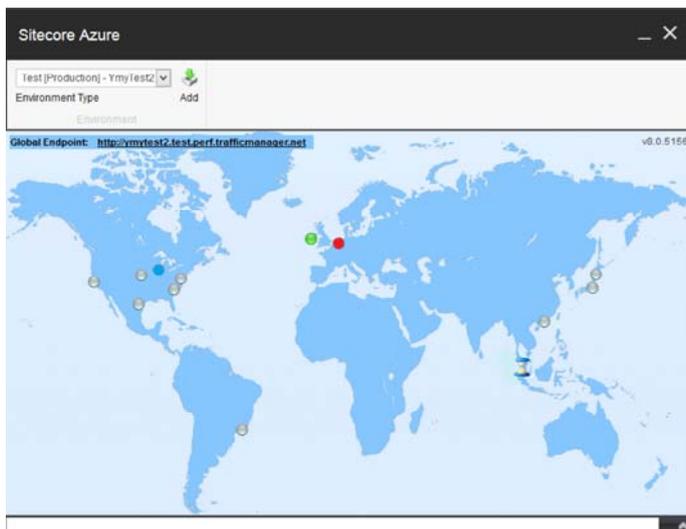
```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclu
<xsl:output method="xml" indent="yes" />
<xsl:template match="@* | node()">
  <xsl:copy>
    <xsl:apply-templates select="@* | node()" />
  </xsl:copy>
</xsl:template>
  <xsl:template match="sitecore/settings/setting[@name='Analytics.Enabled']/@value">
    <xsl:attribute name="value">false</xsl:attribute>
  </xsl:template>
</xsl:stylesheet>
```

4. Save the item and continue the deployment process.

Send feedback about the documentation to docsite@sitecore.net.

Manage your Sitecore Azure service

Sitecore Azure provides a map overlaid with icons that represent available Microsoft Azure hosting locations.

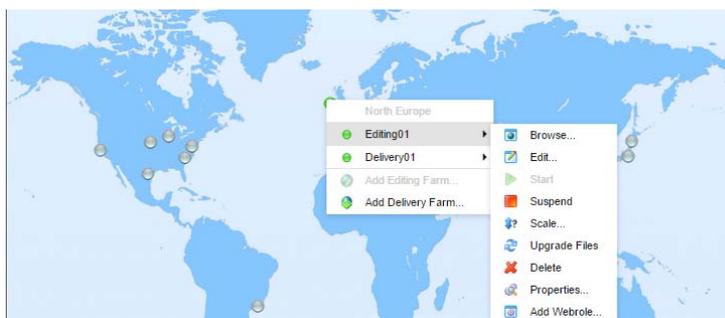


Each icon and its color indicate the status of the deployment in that environment:

- Gray Circle: Unused. You can create a content delivery farm or a content editing farm.
- Green Circle: Started. You can access the content delivery or content editing farm over the internet.
- Red Circle: Suspended. The farm will no longer accept any traffic. However, you can start the farm again.
- Blue Circle: A farm has been created but it is not yet deployed.
- Hourglass: Busy. Sitecore Azure is attempting to start, suspend, delete, or otherwise manage the location, or the current operation for the location is queued.

Note

Sometimes the Sitecore Azure user interface can seem unresponsive because of long-running remote service calls.



You can click a location circle and select Editing or Delivery, and then select the action that you want to perform on the editing or delivery farm. You can also open multiple Sitecore Azure windows in the Sitecore desktop to access the status of tasks in multiple locations.

The following table describes the different actions you can carry out on a location in either the Editing or Delivery menus.

Action	Description
Browse	Review your website in the cloud.
Start /Suspend	Start or suspend the Sitecore content delivery instances (Microsoft Azure web roles) Important When you suspend a role it is not deleted. The instance and costs are still incurred while the role is suspended. If you no longer use the instance, wait for the Suspend operation to complete then click Delete to remove it from the cloud entirely.
Scale	Change the number of instances in a Sitecore Azure content delivery environment to configure at this location. Note Microsoft recommends that you create at least two web role instances in each location. When there are two or more web role instances, there is no downtime when you upgrade the files and you get the 99.9% availability as stated in Microsoft's Service Level Agreement.
Upgrade Files	Deploy and upgrade files to a Sitecore Azure content delivery environment. The new files are added to the farm as a whole without any downtime if there is more than one instance. Note This operation only works with files, no database or item changes are applied.
Delete	Remove all instances from an environment.

Note

You must first suspend Sitecore Azure content delivery instances before you can delete them.

Properties See a status of the selected location.

Add Webrole Extend the number of webroles.

Warning

Some operations in the Sitecore Azure user interface, such as adding or removing an SQL Azure publishing target database and updating configuration cause ASP.NET to restart. These operations can therefore interfere with other users and processes on the system. Deployment operations will also cause an ASP.NET restart during the final stage. If you attempt to run multiple deployments concurrently, Sitecore will wait until all deployments are finished to trigger the ASP.NET restart and for the changes in all deployments to be available.

Publish to a Sitecore Azure content delivery environment

Publish to a Sitecore Azure content delivery environment with the publishing target database associated with the Microsoft Azure location.

Note

If you have chosen a live mode deployment for your Sitecore Azure farm, you do not need to run a publishing operation. Attempting to run a publishing operation when your site is configured for live mode may generate runtime errors. Live mode only applies to Content Editing farms. An on-premise solution that runs on live mode will not automatically reflect changes in your Sitecore Azure deployments in the cloud.

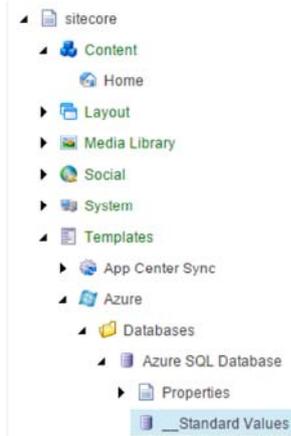
Move a location from staging to production

You can move a location from Staging to Production by clicking the location, expanding the farm that you want to move, and then clicking Swap.

Change the default Azure SQL database service tier

The default service tier is 250 GB – Standard S2 edition, but you can change this on the template before deployment. The size of the databases is set to maximum by default.

1. To change the [default Azure SQL Database Service Tier](#), in the Content Editor, navigate to `sitecore/Templates/Azure/Databases/Azure SQL Database/_Standard Values`.



2. In the Tier field, specify the service tier. This sets all databases (Core, Master, and Web), to the same size and service tier.

Note

After deployment, you must make any further changes (including assigning different sizes and service tiers for the databases) in the Microsoft Azure Portal.

Enable batch upload

To [enable a batch upload](#), open the `Sitecore.Azure.config` file of your Sitecore installation located at `website/AppConfig/Include` and change the `Azure.BulkCopyBatchSize` value to the relevant batch size. The value must be greater than zero.

Send feedback about the documentation to docsite@sitecore.net.

Troubleshooting Sitecore Azure

A useful reference for some of the most common troubleshooting queries about using Microsoft Azure, SQL Azure, and Sitecore Azure.

- [Assemble a solution for deployment with Sitecore Azure](#)
- [A valid certificate for this environment is not found](#)
- [Avoid a lengthy path exception during deployment](#)
- [Considerations for ASP.NET applications for Microsoft Azure](#)
- [Considerations for using SQL Azure with Sitecore](#)

- [Deployment naming conventions](#)
- [Dynamic Link Libraries \(DLL\) not copied over](#)
- [Encrypt packages](#)
- [Enable remote desktop access to web role instances](#)
- [Enable SSL in a Sitecore Azure project](#)
- [File media support by Sitecore Azure](#)
- [How Sitecore Azure content delivery environments clear caches](#)
- [How to configure a Sitecore Delivery environment for multiple sites](#)
- [Implement a custom web role entry point](#)
- [Sitecore Azure fails to deploy](#)
- [Support for Automatic deployment to SQL Azure databases by Sitecore Azure](#)
- [TooManyRequests error](#)
- [Use your own domain name](#)
- [Where Sitecore Azure stores configuration information](#)

Assemble a solution for deployment with Sitecore Azure

Sitecore Azure assembles the document root of a solution for deployment by:

- Copying the document root subfolder of your on-premise Sitecore installation to `$(datafolder)\AzurePackages\{xx}Azure\SitecoreWebSite` for a Microsoft Azure deployment and to `$(datafolder)\AzurePackages\{xx}DevFabric\SitecoreWebSite` for a local emulator deployment.
- Removing unnecessary files and subfolders from the `SitecoreWebsite` folder.
- Copying the contents of the specified subfolder in the Override Sources Folder field of the Azure Deployment item to the document root of the content editing environment, the `$(datafolder)\AzurePackages\{xx}Azure\SitecoreWebSite` subfolder.
- Preparing the `web.config` file by including files from the `Include` folder and the Sitecore section configuration source, which is `App_Config\Sitecore.config` by default. It also prepares the `web.config` file by carrying out an Extensible Stylesheet Language Transformation (XSLT) based on the XSLT patch in the Custom Web Config Patch field of the Azure deployment item.
- Decreasing the size of the `web.config` file by the end of preparation, the Sitecore section of the `web.config` file is stored in the Sitecore section configuration source file, which is `App_Config\Sitecore.config` by default. Editing the `connectionStrings` element of the `web.config` file. The editing process removes unnecessary connection strings and adds new ones based on the databases that are stored in the Connection Strings Patch field of the Azure deployment item. Custom connection strings are supported.
- Copying license information from the local installation to the content delivery package. If you need a different license file in the content delivery installation, add it to the `\App_Data\AzureOverrideFiles` subfolder (the default path specified in the Override Sources Folder field of the Azure deployment item).
- Modifying the configuration files in the content delivery package to configure or delete databases, archives, agents, sites, watchers, and other features.

A valid certificate for this environment is not found

If the correct credentials have not been established, you will be prompted to set up certificate access.

In the Install Management Certificate dialog, there are two installation options:

- Basic installation – this is the fast and simple method offered by Microsoft. Click the [link](#) to get a `.publishsetting` file from Microsoft. Follow the instructions on the Microsoft Portal site, using your Windows Live account. To get the file, save the `.publishsetting` file locally, then upload it using the Sitecore Azure Install Management Certificate dialog.
- Advanced installation – this is the option for advanced users. You can use an existing certificate or generate a self-certificate. Upload the `.pfx` file for Sitecore. Once you have selected either a `.publishsettings` or `.pfx` file, you must provide a password for a `.pfx` file. Click Upload.

Note

If you do not already have a certificate from a certificate authority installed on your web server, you can create a self-generated certificate for development and testing purposes.

Avoid a lengthy path exception during deployment

As a part of assembling a solution for deployment, the document root subfolder of your on-premise Sitecore installation (the Deployment server) is copied to a specific folder. During this process some paths to Sitecore documents that are copied to a specific folder can become too long. To avoid this, you can tune the path to the target folder by changing the value of the Build folder field of the appropriate Environment item. See [Assemble a solution for deployment with Sitecore Azure](#) for more details.

Note

There is a configuration item for each environment under `/Sitecore/system/Modules/Azure`. During assembly, the Build folder for the configuration item is filed and specifies the target folder to copy the Sitecore documents to.

Considerations for ASP.NET applications for Microsoft Azure

Consider the following for any ASP.NET application, such as Sitecore, on Microsoft Azure.

- The default choice for ASP.NET session management is `RedisSessionStateProvider`. This is an out-of-process state provider, therefore factor this in to your code.
- Before moving to the cloud, consider any internal applications integration within the solution.
- Use configuration files rather than Windows IIS Manager.
- You cannot deploy individual files to Microsoft Azure. You must deploy the entire document root.
- Microsoft Azure uses local storage for custom files and folders that are large in size, such as disk caches. Use appropriate APIs for I/O tasks. Sitecore Azure is preconfigured to provide a default local storage called `SitecoreWebSite`.
- You should use local storage for *all* custom I/O tasks that do not require persistence of data. Consider using the storage account for persisting data, but be aware that this approach can introduce a bottleneck in high performance environments.
- Microsoft Azure instances are volatile by design. Design your architecture according to this premise.

Considerations for using SQL Azure with Sitecore

When you use SQL Azure with Sitecore, you should consider the following:

- When accessing a SQL Azure database from a remote location, the performance depends on network conditions and other factors.
- Use an [encrypted connection](#) over port 1433 when accessing SQL Azure remotely, (outbound access from the content editing environment).
- The maximum database size in SQL Azure depends on [the service tier](#) that you choose.

- If SQL Server 2008 R2 is installed it must be with SP1.
- SQL Azure uses IP address restrictions for database security.
- If SQL Server is installed on a separate server, an appropriate Microsoft SQL Server Shared Management Objects package must be installed on the server where the Sitecore CMS is deployed.

Deployment naming conventions

The deployment name must be globally unique and can be used for multiple purposes in an Azure implementation. The deployment name is assigned using the following pattern:

```
[projectname][environmenttype][location][Farm Type][XX]Role[XX]Sc[first 3 chars of subscription id][UTCTimeStamp]
```

For example, a deployment name of `myprojectDevWeCd01Role01Sc561201304181010` indicates a project with the name of *myproject* running as a development environment, using the *West Europe* hosting location and the deployment is a *content delivery* farm.

Dynamic Link Libraries (DLL) not copied over

If you notice that some Dynamic Link Libraries (DLLs), were not automatically copied over, you must manually add the following DLLs to the bin folder of your website. Sitecore Azure will automatically include them in the next deployment.

System.Web.Mvc.dll 3.0.0.0

System.Web.Helpers.dll 1.0.0.0

System.Web.WebPages.dll 1.0.0.0

System.Web.WebPages.Deployment.dll 1.0.0.0

System.Web.WebPages.Razor.dll 1.0.0.0

Microsoft.Web.Infrastructure.dll 1.0.0.0

Encrypt packages

The package for Sitecore Azure is encrypted by default. If you want to disable encryption, in the `/App_Config/Include/Sitecore.Azure.config` file, set `Azure.Package.NoEncryptPackage` to true. For example:

```
<setting name="Azure.Package.NoEncryptPackage" value="true"/>
```

Enable remote desktop access to web role instances

After you have successfully deployed your Sitecore instances to Azure, you can enable [Remote desktop through the Microsoft Azure management portal](#). Remote desktop gives you access to the web role virtual machines.

Enable SSL in a Sitecore Azure project

You can find Sitecore settings that control SSL in the Service Configuration and Service Definition fields of the deployment item. Before you start deploying, add the appropriate information in these fields to enable SSL. For more information, see the section Where Sitecore Azure stores configuration information. You can find basic information about using SSL with Microsoft Azure on [Microsoft's website](#).

File media support by Sitecore Azure

Sitecore Azure does not support file media.

How Sitecore Azure content delivery environments clear caches

Sitecore Azure uses the same process as on-premise Sitecore solutions to clear caches. After completing a publishing operation, the Sitecore Azure content editing server writes a time stamp to the publishing target database. The Sitecore Azure content delivery server checks the time stamp on each request, and clears the caches accordingly.

How to configure a Sitecore Delivery environment for multiple sites

Before you can [configure a Sitecore Azure content delivery environment for multiple sites](#), you must configure a Web Role for Multiple Sites. To use the Sites section in the service definition file to configure a Web Role for multiple sites, refer to the [WebRole Schema documentation](#) on MSDN.

Note

The service definition configuration is stored in the Service Definition field of the Azure Deployment item. To apply the service definition changes you must redeploy.

Test a Sitecore Azure content delivery environment for multiple sites on your local machine:

1. In Sitecore Azure, on the map, click the deployment that is running. In the Farm menu select Properties.
2. In the Properties dialog box, copy the value of your Windows Azure host name from the Webrole section.
3. In the Windows host file, map your hostname (hostname and hostHeader attributes value) to your Windows Azure host name.

Implement a custom web role entry point

By default, Sitecore Azure extends the `Microsoft.WindowsAzure.ServiceRuntime.RoleEntryPoint` class to add functionality to configure the diagnostics by using the Microsoft Azure Storage Account when the role instance is initialized (on start). [The RoleEntryPoint class](#) is named `Sitecore.Azure.WebRole` and it is located in the `\bin\Sitecore.Azure.WebRole.dll` assembly of the Sitecore website.

To implement a custom web role entry point:

1. Implement a class that extends the default `Sitecore.Azure.WebRole.WebRole` class from the `Sitecore.Azure.WebRole.dll` assembly. You can find this assembly in the `\Website\bin` folder after the installation of Sitecore Azure completes.
2. Override the methods that execute when a role instance is initialized (`OnStart`), run (`Run`), and stopped (`OnStop`), in your class.
3. Compile your code into a new assembly and put it in the `\Website\bin` folder of your solution.
4. Update the assembly name in the Service Definition field of the deployment item:

```
<EntryPoint>
<NetFxEntryPoint assemblyName="bin\Sitecore.Azure.WebRole.dll" targetFrameworkVersion="4.5" />
</EntryPoint>
```

5. Create a new deployment, or upgrade an existing one, by using Sitecore Azure to apply the changes.

Sitecore Azure fails to deploy

- Restart ASP.NET.
- Clear the browser cache.
- Investigate the Sitecore logs in the Sitecore Deployment server environment for errors.
- Remove the instances from the environment and deploy again.
- After deploying, configure the `$(dataFolder)\AzurePackages\ (xx)Azure\SitecoreWebSite` subfolder as the document root of an IIS web site, and diagnose that site.
- Temporarily set the `mode` attribute of the `/configuration/system.web/customErrors` element in the `web.config` file to `Off`.

Warning

If you set the `mode` attribute to `off`, it can expose detailed error information in the Microsoft Azure environment. Consider other techniques to protect the application while `mode` is `Off`. Remember to set `mode` to `On` or `RemoteOnly` after diagnosis, and remember to deploy that change.

Support for Automatic deployment to SQL Azure databases by Sitecore Azure

Sitecore Azure only supports automatic deployment to SQL Azure databases. You can change the configuration settings to deploy to databases on Azure virtual machines (IaaS).

TooManyRequests error

During a deployment, if you get a `TooManyRequests` error, you can increase the Get Operation Status interval.

To change the Get Operation Status interval, open the `Sitecore.Azure.config` file in the `Include` folder (`\website\AppConfig\`) and then adjust the `Azure.CheckAsyncOperationStatusIntervalInMS` value.

Use your own domain name

To use your own domain name you must create a CNAME record with your internet provider. Using the tool provided by your internet provider, create an entry that maps the desired alias (such as "www") for your domain to the host name that was assigned to it when your Microsoft Azure application was created (such as "sitecoreassignedpolicyname.trafficmanager.net"). Then use the tool supplied by your internet provider to forward the root domain to the subdomain that you have mapped to your Microsoft Azure host name.

[CNAME records must always be mapped to a specific subdomain](#); you cannot map to the root domain itself or map a wildcard address.

Important

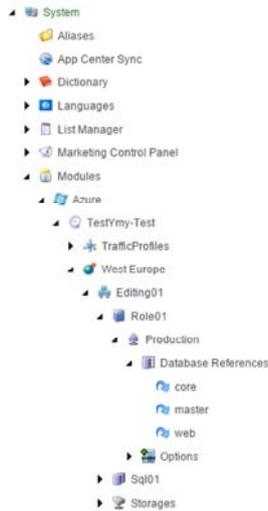
Never map directly to a Virtual IP address (VIP) because in certain instances Microsoft Azure may change the underlying VIP.

Where Sitecore Azure stores configuration information

Sitecore Azure stores configuration information in a number of locations, with basic configuration details stored in the `/App_Config/Include/Sitecore.Azure.config` file.

Sitecore manages configuration information using the `Azure` item (`Sitecore/System/Modules/`) and its descendants.

The installed environments and the relevant deployment items contain configuration settings that are under the `/sitecore/system/Modules/Azure` item. A few levels down is the `Azure Deployment` item (`Editing01`), which contains configuration settings for deployment. You can access this through the Content Editor or by clicking More Options in the New Deployment dialog box.



You can manually customize several fields that contain configuration information for deployment.

Under each Azure deployment item, the `Database References` folder contains database projects for deployment to SQL Azure. By default, the delivery farm contains a core and web database, and the editing farm contains a core, master, and web database. If you need to deploy another database to SQL Azure, you can create another database item.

While the environment is being prepared, the SQL Azure database servers and storage are created, if they do not already exist, and will be used during the deployment process.

Send feedback about the documentation to docsite@sitecore.net.

Connect to xDB Cloud from Sitecore Azure

Note

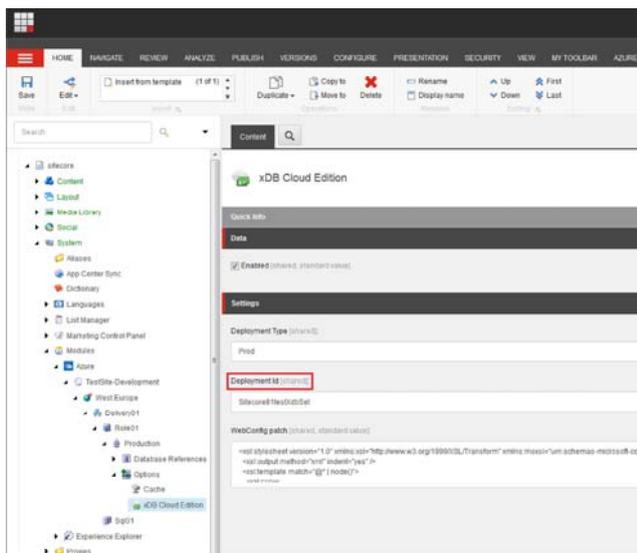
This topic has moved, use [this link](#) for your bookmark instead.

This topic describes how to connect to xDB Cloud 1.0, to connect xDB Cloud 2.0 see: [Configure xDB Cloud open tenancy for Sitecore Azure](#).

Sitecore Azure deploys content delivery (CD) and content management (CM) server farms without xDB. Sitecore Azure also deploys a session state service which is not based on MongoDB, but does exactly the same job across the server farms. The CDs and CMs can connect with either Sitecore Experience Database (xDB) Cloud or your own individually deployed xDB.

Connect to the xDB Cloud from Sitecore Azure

Sitecore Azure automatically connects with xDB Cloud. If you are running Sitecore Azure on a local computer, and the xDB Cloud connection is also set up on this computer, the xDB cloud settings are sent with the Azure web role deployment. The xDB cloud settings use the deployment ID of a newly created xDB set. If you want to use a different xDB deployment ID go to xDB Cloud Edition, and change the deployment ID in the Deployment ID field.

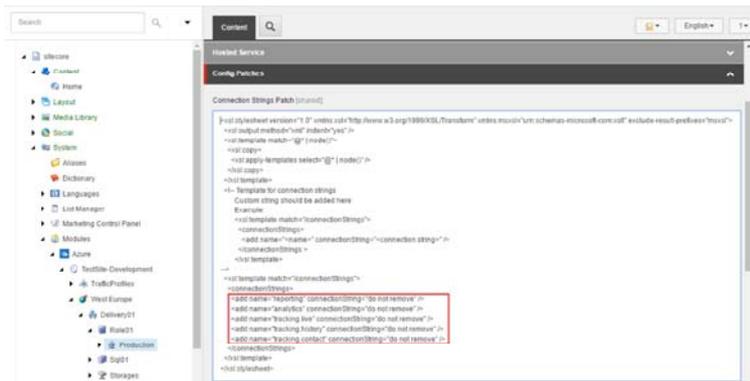


Connect to your own xDB from Sitecore Azure

To connect to your own xDB, you must edit the connection strings.

1. Ensure that the xDB Cloud configuration files are disabled on your local machine, (they are disabled by default).

2. In System, Modules, Staging, Config Patches, go to the Connection Strings Patch field to edit the connection strings.



Send feedback about the documentation to docsite@sitecore.net.

Cloud configuration options

Note

This topic has moved, remember to [update your bookmark](#).

You can deploy Sitecore partly or entirely in the cloud. This topic outlines some of the advantages and disadvantages of using the cloud and how you should approach making your first cloud deployment.

You can choose to install the Sitecore Experience Database (xDB) as an entirely on-premise solution, a partly on-premise solution, partly in the cloud (hybrid) or entirely in the cloud.

Sitecore xDB Cloud Edition is a service that enables you to run Sitecore xDB entirely in the cloud. This includes the following:

- Sitecore application servers for processing, aggregation, and reporting.
- Microsoft SQL Server reporting database
- MongoDB collection database

The following table outlines how the different versions of Sitecore can be combined with the xDB on-premise and xDB Cloud Editions:

Sitecore versions	xDB on-premise	xDB Cloud Edition
Sitecore CEP 7.2 (on-premise)	No	No
Sitecore XP 7.5 (on-premise)	Yes	Yes
Sitecore XP 7.5 (deployed using Sitecore Azure 7.5)	No	Yes
Sitecore XP 8.0 (on-premise)	Yes	Yes
Sitecore XP 8.0 (deployed using Sitecore Azure 8.0)	No	Yes
Sitecore XP 8.1 (on-premise)	Yes	Yes
Sitecore XP 8.1 (deployed using Sitecore Azure 8.1)	No	Yes

There are several other cloud offerings currently available to choose from, such as Rackspace, and Amazon Web Services (AWS). Each of these solutions have their own set of advantages and disadvantages.

Benefits of using Sitecore xDB Cloud Edition:

- Powerful, scalable, and fully managed – saves the cost of acquiring, configuring, and maintaining the infrastructure to support xDB on the premises.
- Easy to connect to, simple to set up and use.
- Low cost.
- Globally available at several Azure locations.

For more information on using the Sitecore xDB Cloud Edition, contact your Sitecore partner.

Send feedback about the documentation to docsite@sitecore.net.

Configure a Sitecore xDB Cloud connection

Note

This topic has moved, remember to [update your bookmark](#).

If you signed up for xDB Cloud *before* July 21st 2016, but are only starting to use it now, you must follow the instructions in the [xDB Cloud 2.0 configuration guide](#) instead.

To run Sitecore services in the cloud, you connect to Azure via different servers. Sitecore xDB Cloud Edition is a Sitecore service that can run Sitecore xDB in the cloud.

If you are upgrading from Sitecore Experience Platform (Sitecore XP) version 7.5 to 8.0, you must first upgrade your Sitecore Experience Database (xDB) set before you configure the client. To configure Sitecore XP, see the [Sitecore xDB configuration topics](#). If you are running Sitecore XP 8.0, go to <http://dev.sitecore.net> or <http://support.sitecore.net> for further support.

Note

Sitecore Support Service will continue to support existing customers who are currently using the 7.5 version of Sitecore. However new subscriptions for 7.5 will not be supported.

Configure the Sitecore xDB Cloud client

Before you configure the Sitecore xDB Cloud client, you must:

- [Install the latest compatible Sitecore xDB Cloud client package](#) on your Sitecore solution.
- Configure the Sitecore xDB client, specifically the Content Management server (CM) and the Content Delivery server (CD), as per the table below:

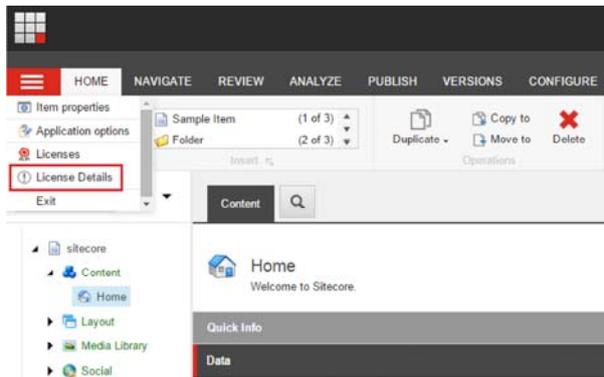
Important

Sitecore provides the Processing and Reporting servers by default (indicated by *), so you do not need to configure them.

Configuration file	Folder	Server	Enable	Disable
Sitecore.Cloud.Xdb.config (disabled by default)	App_Config/Include/xDBCloud	CM	✓	
		CD	✓	
		Processing*		✓
		Reporting*		✓
Sitecore.ContentSearch.Cloud.Index.Analytics.config (disabled by default)	App_Config/Include/xDBCloud	CM	✓	
		CD	✓	
		Processing*		✓
		Reporting*		✓
Sitecore.ContentSearch.Cloud.DefaultIndexConfiguration.config (disabled by default)	App_Config/Include/xDBCloud	CM	✓	
		CD	✓	
		Processing*		✓
		Reporting*		✓

To enable the xDB Cloud service on your Sitecore instance:

1. Make sure you are running Sitecore XP version 8.0 or later. To check which version you are running:
 - Go to `Website\sitecore\shell\sitecore.version.xml`
 - Go to the Content Editor, System menu, and right-click License Details.



2. In the folder that contains your Sitecore installation, open the *XdbCloud* folder (Website/App_Config/Include/XdbCloud).
3. In the text editor, open the *Sitecore.Cloud.Xdb.config.disabled* configuration file.
4. Change the *Sitecore.Cloud.Xdb.DeploymentID* setting to a value that uniquely identifies your xDB instance in conjunction with your license ID, (this can be any alpha-numeric string).
5. Go to the *Sitecore.Cloud.Xdb* file, in the setting `<setting name="Sitecore.Cloud.Xdb.DeploymentType" value="Prod" />` ensure the value is *Prod*.

Important

The setting *Sitecore.Cloud.Xdb.DeploymentType* is deprecated and is removed from the *Sitecore.Xdb.Cloud* file in future releases.

6. Set the region to `<setting name="Sitecore.Cloud.Xdb.DeploymentRegion" value="Your preferred data center" />`

For example:

```
<setting name="Sitecore.Cloud.Xdb.DeploymentRegion" value="North Europe" />
```

For the supported values of the [compatible data center locations](#), see the xDB Cloud Service compatibility tables.

Important

After you configure Sitecore to use xDB Cloud and *before* you run Sitecore for the first time, you must [update to the newest Sitecore Experience Platform client](#) that includes the location setting, and specify your preferred deployment region for your region setting. After you have specified the region, for example "North Europe" Sitecore will always automatically look for that location.

The region setting is available with the [Sitecore xDB Cloud update support package](#)

7. Enable the following configuration files by removing the *.disabled* extension from the name:
 - *Sitecore.Cloud.Xdb.config.disabled*
 - *Sitecore.ContentSearch.Cloud.Index.Analytics.config.disable*
 - *Sitecore.ContentSearch.Cloud.DefaultIndexConfiguration.config.disabled*
8. To ensure that the connection was successful, when your installation has finished, open the log files and search for *xDB Cloud*.
 - If the installation was successful, the log files contain the following message: *INFO xDB Cloud – Initialization completed successfully*.
 - If the installation was not successful, the log files contain an error reference code and a message outlining what went wrong. Email support.sitecore.net and remember to quote the error reference code.

Note

Most customers purchase Sitecore products and services through a certified Sitecore partner. Because Sitecore does not directly offer customized development, design, and implementation services, you should continue to use your Sitecore partner for help with integration. If you do not currently have a certified Sitecore partner, contact your local Sitecore office who will direct you to a Sitecore partner in your area.

Firewall configuration

To update and configure your Sitecore xDB Cloud connection, you must also configure your firewall. Connect to the Discovery Service through *discovery.xdb.cloud.sitecore.net*. Sitecore xDB Cloud connects through multiple separate endpoints, for example, an endpoint for MongoDB, an endpoint for the reporting service, and an endpoint for the discovery service.

Additional endpoints may appear as the service expands. These services run on highly available pools of compute hosts. By design, the hosts are not set with static IPs because hosts are swapped in and out of the pool on a regular basis to ensure maximum availability. This means that switching IPs is expected behavior – Microsoft has a [helpful list of IP ranges](#) that are currently in use and updated regularly.

All Sitecore cloud services are run in Azure Data Centers and all hosts use the wildcard domain address: **.cloud.sitecore.net*

Important

You also need to [open specific ports for Mongo communication](#). The ports vary from customer to customer, therefore you should contact support.sitecore.net to obtain your personalized Mongo ports.

Send feedback about the documentation to docsite@sitecore.net.