sitecore®

Sitecore Azure 7.2

Getting Started with Sitecore Azure

An Overview for Sitecore Administrators

Table of Contents

Chapter 1

Getting Started with Sitecore Azure

This document introduces Sitecore Azure. Sitecore administrators should read this document before installing or using Sitecore Azure.

Beginning with an overview of the underlying Windows Azure and Sitecore Azure concepts and terminology, this document describes procedures for accomplishing specific tasks with Sitecore Azure, answers common questions about Sitecore Azure, and provides information about known issues and troubleshooting techniques.

This document contains the following sections:

- Overview

- Using Sitecore Azure

- Using Sitecore DMS with Windows Azure

- Answers to Common Questions

- Known Issues, Considerations, and Troubleshooting

1.1 Overview

Sitecore Azure provides a set of software and services to simplify the deployment of your web solutions to the Windows Azure cloud. You can use Sitecore Azure with the Sitecore CMS to host Sitecore solutions in the Windows Azure cloud.

Windows Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed datacenters. Sitecore Azure is built around Windows Azure cloud services (PaaS) and includes automatic deployment of web servers as well as Microsoft SQL databases. You can read more about Windows Azure here: http://www.windowsazure.com.

You can acquire Windows Azure services from Microsoft and manage deployment of your own Web Applications, or you can acquire services through Sitecore and gain special licensing benefits, such as free bursting for increased loads. Sitecore Azure also provides an integrated user interface to manage your cloud delivery infrastructure.

With Sitecore Azure, you can deploy from an on-premise content editing server to any number of Sitecore Azure cloud services hosted at the various Windows Azure locations. Sitecore Azure provides automation and a browser-based user interface to manage your Windows Azure deployments.

Sitecore Azure provides the following benefits:

- Sitecore Azure provides for extreme scalability, stability and simplicity.

- Sitecore Azure provides a highly available infrastructure through Windows Azure's SLA (99.95%) up time. http://www.windowsazure.com/en-us/support/legal/sla/. You must have two or more instances deployed in the same Availability Set to qualify for this level of external connectivity.

- Sitecore Azure significantly reduces the costs and amount of time required to deploy your solution to a new server.

- Sitecore Azure significantly reduces the costs and amount of time required to provide geographically distributed load balancing and redundancy between geographies for performance, and disaster preparedness.

- Sitecore Azure reduces the cost and time required to adjust server capacity to meet expected load.

- Sitecore Azure automates most of the process of migrating a website to Windows Azure.

1.1.1 Options for Sitecore Azure 7.2 Deployment

Sitecore Azure 7.2 supports content editing as well as content delivery from the cloud. You can decide on the best combination of on-premise or cloud-based editing and delivery environments, called "Editing Farm" and "Delivery Farm". A "farm" is a set of web roles connected to a single SQL Azure instance.

Microsoft provides a local Windows Azure Emulator that can be used for initial setup and testing. Instructions for installing and using the emulator are included in this document.

Some example deployment topologies are shown here for consideration.

The following example shows an installation that uses an on-premise Sitecore installation for content editing and then publishes to several Sitecore content delivery farms in multiple Microsoft Azure locations.
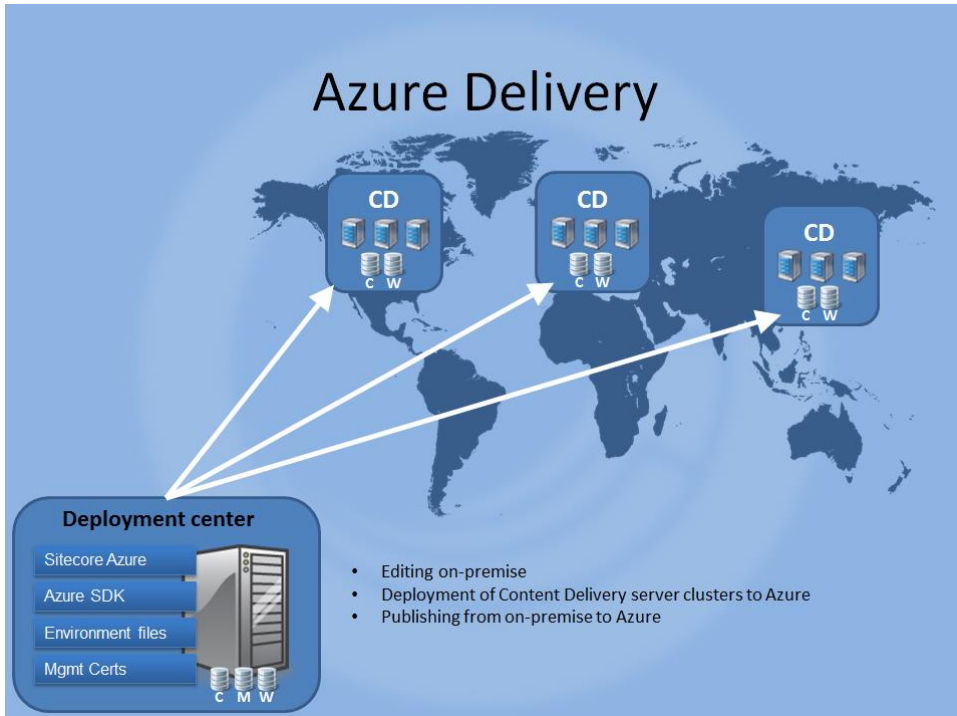
This example shows a Sitecore Azure installation that runs in live mode. Both content editing and content delivery run in the same farm. No publishing is required. Editing changes made in the content editing environment are immediately available to the public.
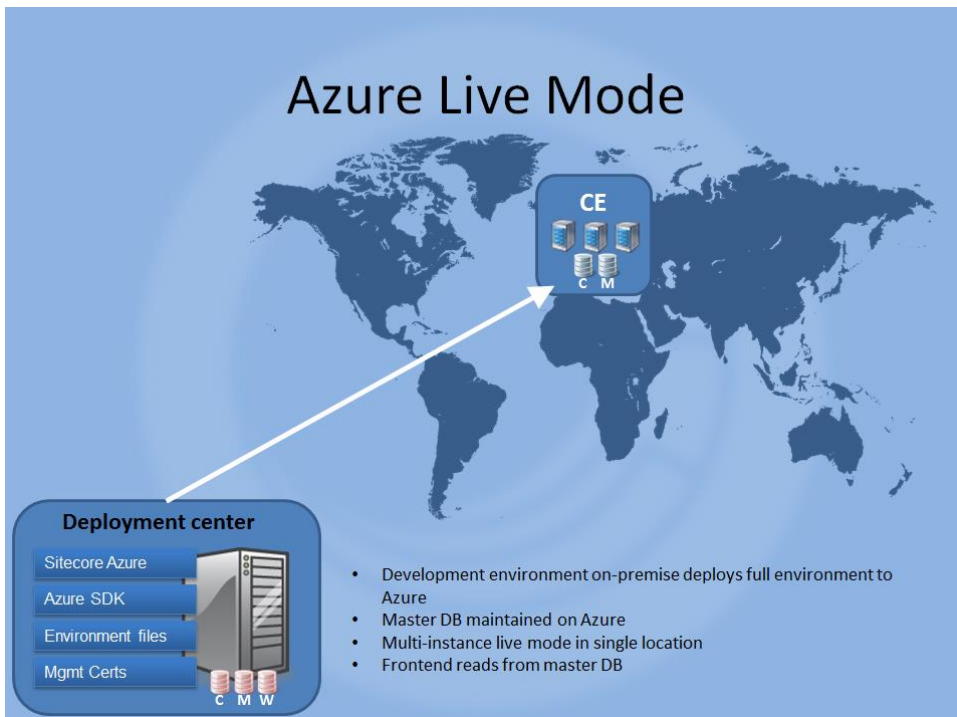
This example shows a deployment topology in which both content editing and content delivery are running in the Microsoft Azure cloud.
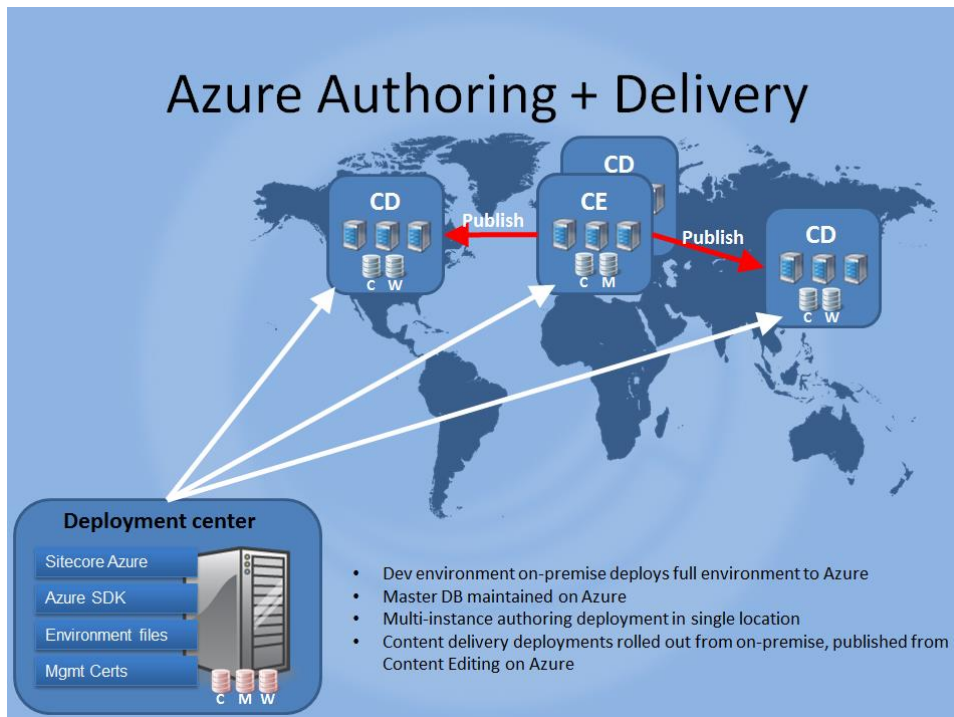
1.1.2   Prerequisites for Sitecore Azure Installation

You must have the following requirements in place before installing Sitecore Azure:

- Microsoft Windows — 64-bit.

- Microsoft .NET Framework v 4.0 or higher.

- Sitecore CMS 7 requires .NET Framework v 4.5.

- A Microsoft Account (formerly Windows Live ID) with a valid Azure subscription (30 days free trials are available, but you must provide contact and credit card information — http://www.windowsazure.com/en-us/pricing/free-trial/).

- Sitecore CMS 7.1 or 7.2.

- Microsoft SQL Server 2008 R2, Service Pack 1 or newer.

- Install the Windows Azure SDK version 2.2 in your on-premise content editing environment (Web Server).

- Download and install the following tools from Microsoft — http://www.microsoft.com/en-us/download/details.aspx?id=40893

    o WindowsAzureAuthoringTools-x64.msi

    o WindowsAzureEmulator-x64.exe

    o WindowsAzureLibsForNet-x64.msi

    o WindowsAzureStorageTools.msi

- Download and install the following tools from Microsoft -- http://www.microsoft.com/en-us/download/details.aspx?id=38797:

    o Microsoft SQL Server 2012 Shared Management Objects

    o Microsoft System CLR Types for Microsoft SQL Server 2012

Sitecore Azure requires that you install these tools on your on-premise content editing environment (Web Server).

- Download and install the following tools from Microsoft — http://www.microsoft.com/en-us/download/confirmation.aspx?id=29065:
    - o SQLSysClrTypes.msi (X64 Package)
    - o Shared Management Objects.msi (X64 Package)
- You must request a Sitecore Azure environment file from Sitecore.

    For more information on how to do this, see the section *How to Install Sitecore Azure*.

Important
You must configure your web server's firewall rules to allow Sitecore Azure traffic. For more information see the section *How to Configure Firewall Rules*.

1.1.3   Using the Windows Azure Local Emulator

The Windows Azure local emulator provides a local environment for development purposes. By using the local emulator you can test your deployment for faster deployment and without incurring any cost.

For more information about the Windows Azure Storage Emulator, see the MSDN documentation at http://msdn.microsoft.com/en-us/library/windowsazure/hh403989.aspx.

Note
To run the Windows Azure Local Emulator you must have administrative privileges on the computer and run the `Runner.bat` file as an administrator user.

1.1.4   Using a Microsoft Azure Subscription vs. a Sitecore-Managed Azure Subscription

When you purchase a Sitecore Azure subscription, you receive the benefits of Sitecore's automated features for managing your Azure installations. You will also receive licensing benefits such as bursting and prioritized support. Your local Sitecore office can provide more details about the licensing benefits.

If you already have an existing Microsoft Azure subscription that you want to use for your Sitecore installation, you can provide that information to Sitecore and continue with your installation.

If you are using a Microsoft Azure subscription, you will still get many of the benefits of using Sitecore's Azure tools.

1.1.5   How to Install Sitecore Azure

To install Sitecore Azure, you need a Sitecore CMS license and at least one working Sitecore installation. If you do not already have that, you can contact Sitecore in your region to obtain the software and licensing. To contact Sitecore, see http://www.sitecore.net/contact.aspx.

Sitecore environment files are configuration files that contain information about your Windows Azure subscription. You can have more than one environment file so that you can support different kinds of environments, for example, you may have separate development, testing and production environments. Sitecore Azure allows you to manage all of these environments from within the same Sitecore installation. The instructions in this section tell you how to obtain these environment files.

Download the Sitecore Azure installation package from the SDN web site.

To install Sitecore Azure:

1. Log in to the desktop of your internal Content Editing environment as an administrator.
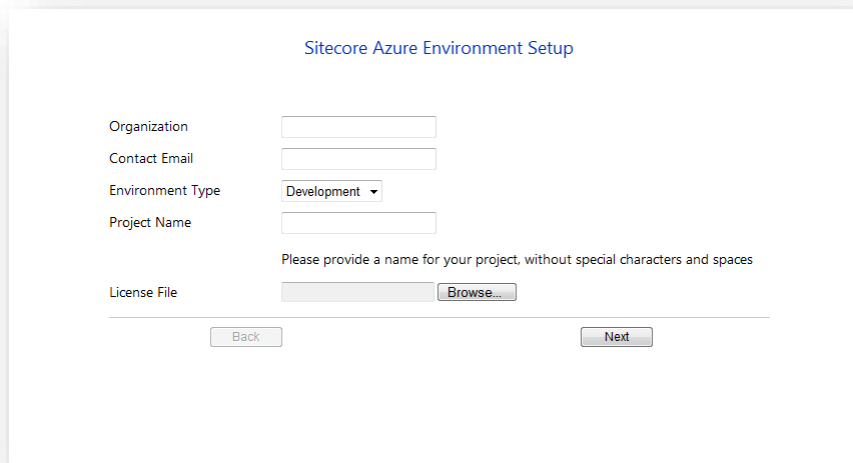
2. In the Sitecore desktop, click **Sitecore**, **Development Tools**, **Installation Wizard**, and then use the Installation Wizard to install the *Sitecore Azure* package. You should overwrite any existing components when prompted.

3. In the Sitecore desktop, click **Sitecore**, and then click **Sitecore Azure**. Sitecore Azure prompts you with the local emulator environment represented by a single grey dot.
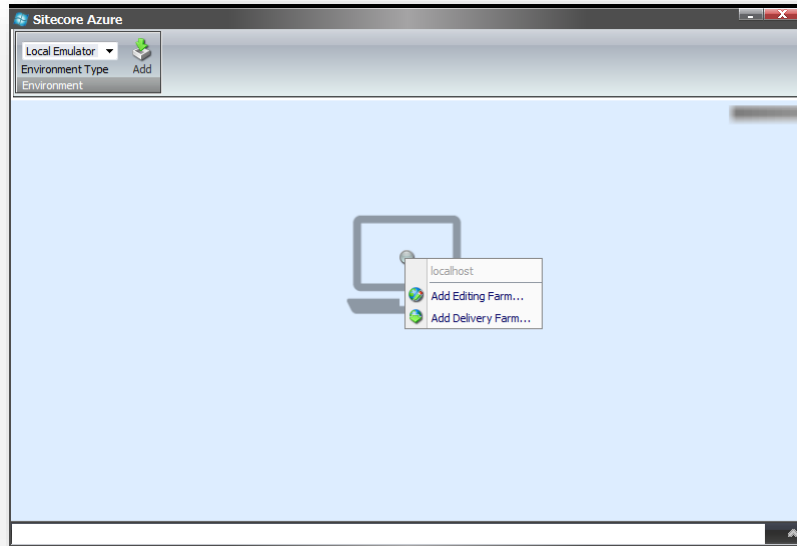
4. To deploy to the Windows Azure cloud, you must upload the Sitecore Azure environment file.

   If this is a first time installation and you have not already obtained a Sitecore environment file; click on the link shown in the dialog box to complete a form and submit your information to Sitecore. You will receive an environment file within 48 hours.
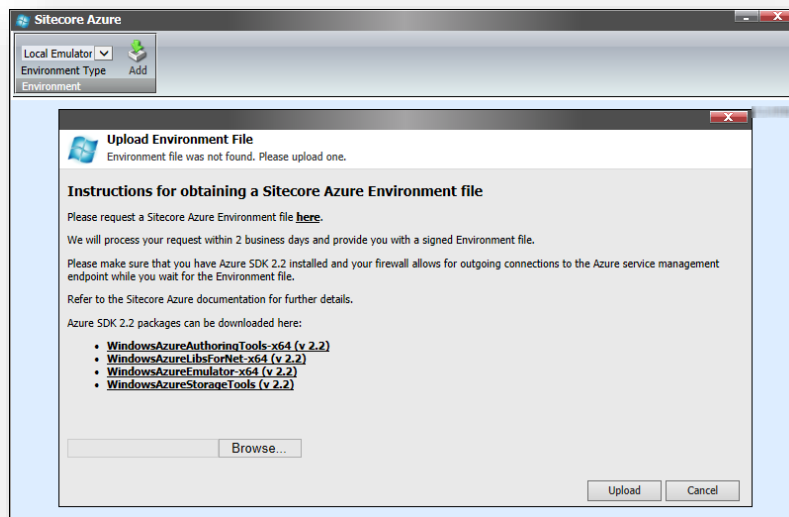
This Sitecore form asks you for basic contact information, a project name, and to upload your Sitecore license file.

5. Once you have obtained your environment file, you can continue with the process in the **Upload Environment File** dialog box, click **Choose File …** and locate the Environment file you obtained from Sitecore. Once you have selected the file, click **Upload** to continue. Your environment file name will be `Environment.xml`.

6. At this point, you may choose to set up a test environment using the Windows Azure Emulator or select a test or production environment. Use the **Environment** dropdown list in the **Environment** group of the ribbon to make a selection.

7. If you choose to deploy directly to a Windows Azure location, either **Development**, **Test**, **or Production** in the **Environment Type** drop-down list in the **Environment** tab of the ribbon. These options may vary depending on which type of Microsoft Azure subscription you have. Then click on your selected location to add a farm. You may add either an Editing Farm or Delivery Farm, depending on your planned deployment strategy.

Note that Sitecore Azure only supports one Editing Farm per environment.
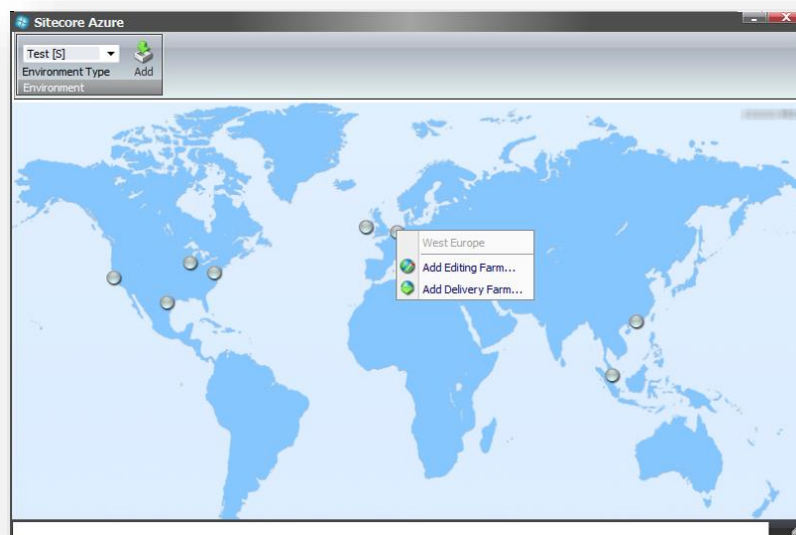
8. If no credentials have yet been established, you will be prompted to set up certificate access. There are two options available:

o **Option A** – 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.

o **Option B** – this is the option for advanced users. You can choose to use an existing certificate or generate a self-certificate. Upload the `.pfx` for Sitecore. Once you have selected either a `.publishsettings` or `.pfx` file, you must provide a password for a `.pfx` file. Click **Upload**.

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.
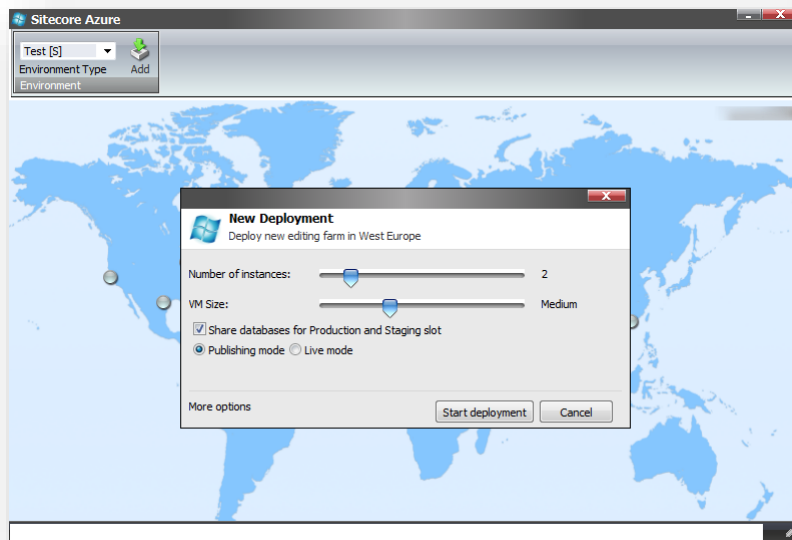
9. For each new deployment you can set the number of instances, the size of the virtual machine, database options, and publishing options. The default settings are shown in the screenshot.

10. To change these settings based on your deployment strategy, click **Start deployment**. The deployment procedure will take some time, depending on the location and the amount of data that is transferred.

For more information, see the section *Considerations for Creating a Sitecore Azure Farm*.

While the Editing Farm is being created, you can begin an operation to create a Delivery Farm in another location. Once your Editing and Delivery Farms are created (shown as a green circle on the map), you can then, click on a location and expand the Editing or Delivery menus. Use the Swap command to move a location from Staging to Production.

1.2 Using Sitecore Azure

Sitecore Azure provides a map overlaid with icons that represent available Windows 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.

- **Green Circle**: Started. You can access this content delivery farm over the internet.

- **Red Circle**: Suspended. You can start this content delivery farm.

- **Blue Circle**: Farm created but not deployed.

- **Hourglass**: Busy. Sitecore Azure is attempting to start, suspend, delete, or otherwise administer the location, or the current operation for the location is queued.

When you click an icon to select a location, Sitecore Azure displays a status dialog box for that location. Ribbon commands operate on the selected location. You can open multiple Sitecore Azure windows in the Sitecore desktop to access the status of tasks in multiple locations.
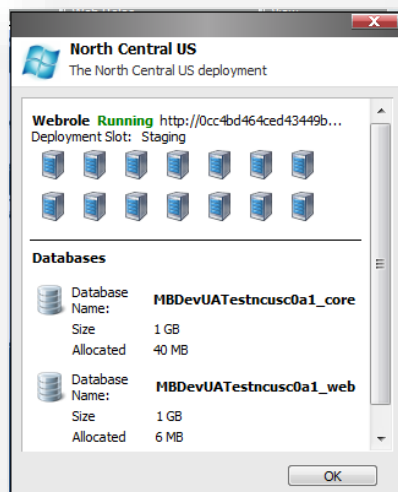
Warning

Certain operations in the Sitecore Azure user interface, such as, adding or removing an SQL Azure publishing target database, updating configuration, and so on, 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.

Note

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

1.2.1 Considerations for Creating a Sitecore Azure Farm

For information about how Sitecore Azure assembles a solution to deploy, see the section *How Does Sitecore Azure Assemble a Solution to Deploy?*

During Sitecore Azure installation you created one or more farms. Here are some additional considerations:

1. To avoid the need to publish after deployment, before creating a Sitecore Azure farm, publish to the default publishing target named Internet (the Web database).

2. To minimize deployment time, remove unnecessary files from the document root of the content editing environment.

3. You can change settings for existing farms or add additional delivery farms at any time. In **Sitecore Azure**, click a location circle to add a new farm, following the same steps as during your original installation. Click a green circle to change settings for an existing, running farm. Expand the **Editing** or **Delivery** menu to display your choices for editing and managing your farms.

4. In the **New Deployment** dialog box, make selections for Number of Instances — at least 2 instances are recommended to ensure high availability — and Virtual Machine Size. Medium is the default but we recommend scaling up to a larger Virtual Machine Size before scaling out and adding more farms since it usually implies a simpler scaling strategy. Small can be used for Evaluation and Test deployments. Microsoft Azure does not currently support changing role size after a deployment is made. The recommended approach is to make a new deployment to the staging slot with the larger Virtual Machine size and swap when the deployment is up and running.
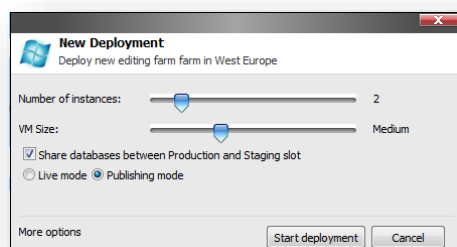
5. The default database selection is to share databases between Production and Staging. You can also select between a controlled publishing environment and live mode. Live mode causes changes in the content editing farm to be immediately available on the frontend of the web site. This does not automatically transfer content to Delivery Farms.

   For more information about live mode, see Sitecore CMS 7.0 documentation.

For information about changing the number of instances in a location, see the section *Change the Number of Instances in an Environment*.

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

1.2.2 Publish to a Sitecore Azure Content Delivery Environment

To publish to a Sitecore Azure content delivery environment:

- Publish to the publishing target database associated with the Windows 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. Note that an on-

premise solution that runs on live mode will not automatically have changes reflected in your Sitecore Azure deployments in the cloud.

1.2.3 Deploy Files to an Environment

To deploy files to a Sitecore Azure content delivery environment:

- In **Sitecore Azure**, click the icon that represents the location. Expand the **Editing** or **Delivery** menu and select the **Upgrade Files** option.

When this operation is invoked, the solution is packaged and uploaded with the new files included, then added to the farm as a whole without any downtime if there is more than one instance. This operation works with files only, no database or item changes will be applied.

1.2.4 Change the Number of Instances in an Environment

To change the number of Sitecore instances in a Sitecore Azure content delivery environment:

1. In **Sitecore Azure**, click the icon that represents the location. Expand the **Editing** or **Delivery** menu and select the **Scale** option.

2. Select the number of instances 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 roles instances, there is no down time when you upgrade the files and you get the 99.9% availability as stated in Microsoft's Service Level Agreement.

1.2.5 Suspend or Start the Instances in an Environment

To suspend the Sitecore content delivery instances (Windows Azure web roles) in an environment:

- In **Sitecore Azure**, click the icon that represents the location. Expand the **Editing** or **Delivery** menu and select the **Suspend** option.

To start the instances in an environment:

- In **Sitecore Azure**, click the icon that represents the location. Expand the **Editing** or **Delivery** menu and select the **Start** option.

Note
Suspending a role does not delete the instance and costs will still be incurred while the role is suspended.

### 1.2.6 Remove Instances from an Environment

To remove all instances from an environment:

1. In **Sitecore Azure**, click the icon that represents the location. Expand the **Editing** or **Delivery** menu and select the **Suspend** option.

2. When the **Suspended** operation is complete, expand the **Editing** or **Delivery** menu and select the **Delete** option.

1.3 Using Sitecore DMS with Windows Azure

Implementing a Sitecore site using Sitecore DMS introduces new concerns about scalability. Sitecore Azure handles this scalability well if certain guidelines are followed. Scaling for DMS brings up considerations in three different areas: computing resources, server cache for session state data, and database resources. The following diagram illustrates these three concerns.

- To add computing (CPU) resources add additional instances to your webrole.

- To add server cache for session state data add additional webroles to your editing and content delivery farms. This is the most important consideration for Sitecore DMS scalability because Sitecore DMS maintains a high volume of data in the session state cache for each web site user.

- To add database resources create additional content delivery farms. Each farm has its own database server. Sitecore Azure can support up to 99 farms per location.
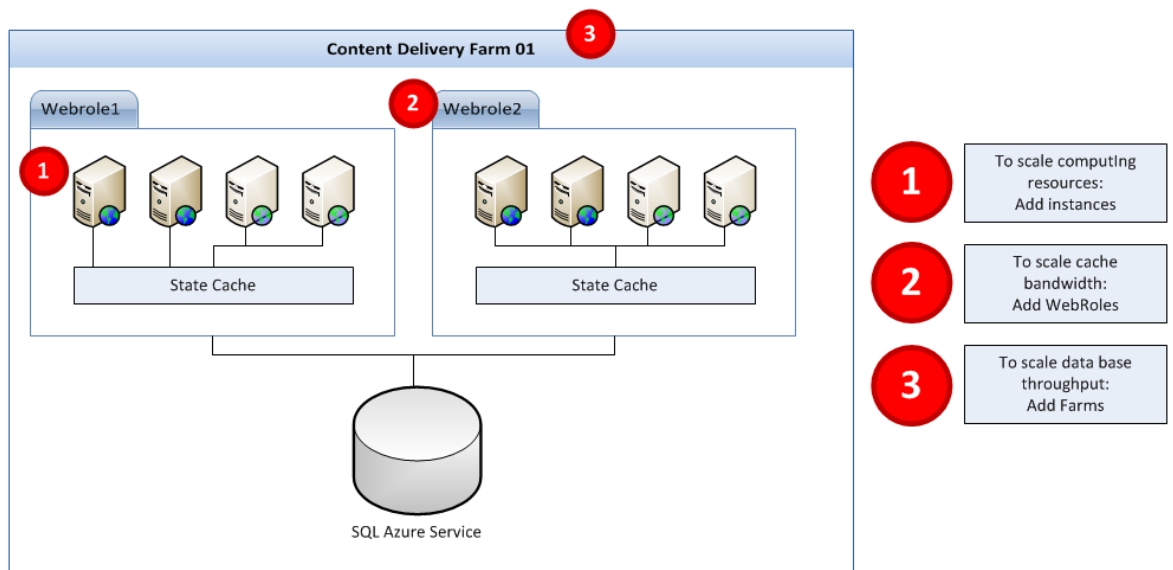
1.4 Answers to Common Questions

This section provides answers to some common questions.

### 1.4.1 How Do Sitecore Azure Content Delivery Environments Clear Caches?

Sitecore Azure uses the same mechanism as on premise Sitecore solutions to clear caches. After completing a publishing operation, the Sitecore Azure Content Editing environment writes a timestamp to the publishing target database. Sitecore Azure content delivery environments check the timestamp on each request, and clear caches accordingly.

1.4.2 Where does Sitecore Azure Store Configuration?

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

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

Under the `/sitecore/system/Modules/Azure` item there are one or more items which represent installed environments. A few levels down is `the Azure Deployment` item which represents settings for deployment. You can access this through the Content Editor or by using the **More Options** button in the **New Deployment** dialog box.
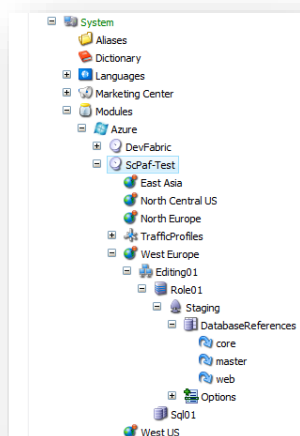
There are several fields which contain configuration information for deployment, and can be changed manually for customization.

- Under each `Azure Deployment` item there is a `Database_Reference` item which contains database projects for deployment to SQL Azure. By default there are `core` and `web` databases for the Delivery Farm and `core`, `master` and `web` databases for the Editing Farm. Additionally, if Sitecore DMS is enabled there is an `analytics` database for both the Delivery Farm and the Editing Farm. If you need to deploy another database to SQL Azure you can create another database item.

- During the "Preparing Environment" operation each SQL Azure database server and storage will be created if it does not already exist, and will be used during the deployment process.

1.4.3 How Does Sitecore Azure Assemble a Solution to Deploy?

To assemble the document root of a solution to deploy, Sitecore Azure performs the following tasks, not necessarily in this order.

- Sitecore Azure copies the document root subfolder of your on-premise Sitecore installation to `$(datafolder)\Data\AzurePackages\(xx)Azure\SitecoreWebSite` for a Windows Azure deployment and to `$(datafolder)\Data\AzurePackages\(xx)DevFabric\SitecoreWebSite` for a local emulator deployment.

- Sitecore Azure removes some unnecessary files and subfolders from the `Sitecore/Website` folder.

- Sitecore Azure copies 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 to the `$(datafolder)\AzurePackages\(xx)Azure\SitecoreWebSite` subfolder.

- Sitecore Azure prepares `Web.config` by including files from the `Include` folder as in a normal installation, does an XSLT transformation based on the XSLT patch in the **Custom Web Config Patch** field of the Azure Deployment item. To decrease the size of `web.config` by the end of preparation, it is divided into several include files which are stored in the `App_Config/Include` subfolder.

- Sitecore Azure edits the `connectionStrings` element of `Web.config`. It removes unnecessary connection strings and adds new ones based on databases which are stored in the **Connection Strings Patch** field of the Azure Deployment item. Custom connection strings are supported.

- Sitecore Azure copies license information from the local installation to the content delivery package. If you need a different license file in the content delivery installation, add this file into the `\App_Data\AzureOverrideFiles` subfolder (the default path specified in the **Override Sources Folder** field of the Azure Deployment item).

- Sitecore Azure modifies the `Web.config` file in the content delivery package to configure or eliminate databases, archives, agents, sites, watchers, and other features.

1.4.4 How Do I Use My Own Domain Name?

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

CNAME records must always be mapped to a specific subdomain, you cannot map to the root domain itself or do wildcard mappings. You can find an article that describes this procedure in more detail at http://www.windowsazure.com/en-us/develop/net/common-tasks/custom-dns.

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

1.4.5 How to Encrypt Your Package

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

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

1.4.6 How to Enable SSL in a Sitecore Azure Project

Basic information about using SSL with Microsoft Windows Azure can be found at:
http://www.windowsazure.com/en-us/develop/net/common-tasks/enable-ssl.

Sitecore settings that control SSL are found in the **Service Configuration** and **Service Definition** fields of the Deployment item. After your initial installation edit the following sections in these fields to enable SSL. For more information, see the section *Where does Sitecore Azure Store Configuration?*

1.4.7 How to Enable Remote Desktop Access to Webrole Instances

You can find information about enabling Remote Desktop Access to Webrole Instances at https://www.windowsazure.com/en-us/manage/services/cloud-services/how-to-configure-a-cloud-service/#remoteaccess.

Remote desktop is best enabled after a successful deployment through the Windows Azure management portal. It will create certificates if needed and you can specify or change access.

This allows for enabling RDP for Sitecore customer service or Microsoft technical support.

1.4.8 Deployment Naming Convention

The deployment name must be globally unique and may be used for multiple purposes in an Azure implementation. The deployment name will be 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 `myprojectDeveWeCd01Role01Sc561_201304181010` would indicate 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.

1.4.9 How to Configure Firewall Rules

When configuring firewall rules, allow access for the following common servers and ports:

For Microsoft datacenters:

- https://management.database.windows.net:8443

- https://management.core.windows.net (443)

- https://blob.core.windows.net (443)

In addition, depending on which deployment location you are using, port 1433 needs to be opened on the following hosts:

- **North Europe** — data.db3-1.database.windows.net [94.245.114.93]

- **West Europe** — data.am2-1.database.windows.net [94.245.97.22]

- **East Asia** — data.hkn1-1.database.windows.net [111.221.69.22]

- **South East Asia** — data.sgp1-1.database.windows.net [207.46.63.13]

- **South Central US** — data.sn1-1.database.windows.net [65.55.74.144]

- **North Central US** — data.ch1-1.database.windows.net [207.46.203.22]

- **East-US** – data.bl2-1.database.windows.net [168.62.47.75

- **West-US** -- data.by1-1.database.windows.net [168.62.15.75]

- **Japan West** – data os1-1.database.windows.net [23.98.56.75]

- **Japan East** – data.kw1-1.database.windows.net [23.98.57.44]

Note
Locations and IP addresses may change and new locations may be added from time to time, contact Microsoft technical support if you are having trouble with these addresses.

Note
Some locations may be closed for new deployments.

1.4.10   How to Implement a Custom WebRole Entry Point

By default Sitecore Azure extends the RoleEntryPoint class to add functionality to configure the diagnostics by using the Windows Azure Storage Account when the role instance is initialized (on start event).This class is named WebRole and placed in the `\bin\WebRole.dll` assembly of the Sitecore website.

To implement a custom WebRole entry point:

1. Implement a class that extends the default `WebRole.WebRole` class from the `WebRole.dll` assembly. This assembly is found in the `\Website\bin` folder after Sitecore Azure installation.

2. Override the needed 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 WebRole assembly name in the **Service Definition** field of the Deployment item:
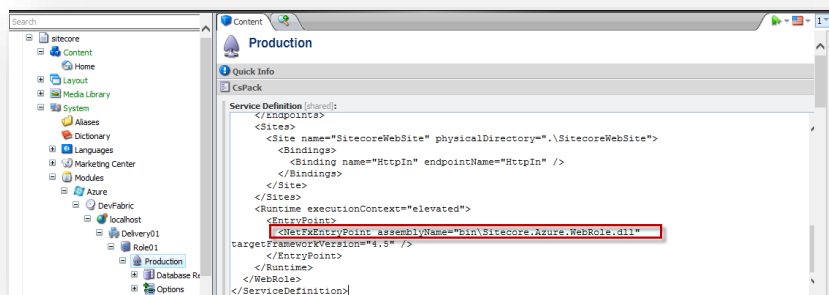
5. Create a new deployment or upgrade an existing one by using Sitecore Azure to apply the changes.

1.5 Known Issues, Considerations, and Troubleshooting

This section describes known issues with Windows Azure, SQL Azure, and Sitecore Azure.

1.5.1 ASP.NET Considerations on Windows Azure

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

- For ASP.NET session management the `DistributedCacheSessionStateStoreProvider` is now the default choice. This is an out of process state provider and considerations in code for such should be taken.

- Before moving to the cloud, consider any integration within the solution to internal applications.

- Use configuration files rather than Windows IIS Manager.

- You cannot deploy individual files to Windows Azure. You must deploy the entire document root.

- Windows Azure uses local storage for custom files and folders that are large in size, such as disk caches. Use appropriate API's for I/O tasks. Sitecore Azure is preconfigured to provide a default local storage called SitecoreWebSite.

- It is recommended to 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.

- Windows Azure instances are volatile by design. Design your architecture according to this premise.

1.5.2 SQL Azure Considerations

Consider the following when using SQL Azure with Sitecore.

- When accessing a SQL Azure database from a remote location, 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).

  For information about SQL Azure security, see http://msdn.microsoft.com/en-us/library/ff394108.aspx.

- SQL Azure licensing limits the maximum database size to 150GB.

  For information about SQL Azure sizing and licensing, see http://msdn.microsoft.com/en-us/library/ee336245.aspx#dcasl.

- 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.

1.5.3   Sitecore Azure Considerations

When you use 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).

- Sitecore Azure does not support file media.

1.5.4   Troubleshooting Sitecore Azure

General troubleshooting steps for Sitecore Azure include:

- Restart ASP.NET.

- Clear the browser cache.

- Investigate the Sitecore logs in the Content Editing environment for errors.

- Remove the instances from the environment and deploy again.

- After deployment, 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 Windows 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.