Sitecore Azure 7.5

Getting Started with Sitecore Azure

An overview for Sitecore administrators

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

Table of Contents

Chapter 1 Getting Started with Sitecore Azure 3
1.1 Introduction 4
1.1.1 Options for Sitecore Azure 7.5 Deployment 4
1.1.2 Using the Microsoft Azure Local Emulator 8
1.1.3 Using a Microsoft Azure Subscription Versus a Sitecore-Managed Azure Subscription 8
1.2 Prerequisites 9
1.3 Install Sitecore Azure 10
1.3.1 Installing the Sitecore Azure Installation Package 10
1.3.2 Deploying to the Microsoft Azure Cloud 12
1.4 Administer Sitecore Azure 17
1.4.1 Publishing to a Sitecore Azure Content Delivery Environment 18
1.4.2 Deploying Files to an Environment 18
1.4.3 Changing the Number of Instances in an Environment 18
1.4.4 Moving a location from staging to production 18
1.4.5 Suspending or Starting the Instances in an Environment 18
1.4.6 Removing Instances from an Environment 19
1.4.7 Changing the default Azure SQL Database Service Tier 19
1.4.8 Enable batch upload 20
1.5 Using Sitecore Experience Platform 7.5 with Microsoft Azure 21
1.5.1 Scalability Options for the Sitecore Experience Database 21
1.6 Getting Started with Sitecore Azure 22
1.6.1 Connect to the xDB Cloud from Sitecore Azure 22
1.6.2 Connect to your own xDB from Sitecore Azure 22
1.7 Frequently Asked Questions 24
1.7.1 How Do Sitecore Azure Content Delivery Environments Clear Caches? 24
1.7.2 Where does Sitecore Azure Store Configuration? 24
1.7.3 How Sitecore Azure Assembles a Solution to Deploy? 25
1.7.4 How Do I Use My Own Domain Name? 25
1.7.5 How to Encrypt Your Package 26
1.7.6 How to Enable SSL in a Sitecore Azure Project 26
1.7.7 How to Enable Remote Desktop Access to Webrole Instances 26
1.7.8 Deployment Naming Convention 26
1.7.9 How to Configure Firewall Rules 26
1.7.10 How to Implement a Custom WebRole Entry Point 27
1.8 Known Issues, Considerations, and Troubleshooting 29
1.8.1 Valid Certificate for This Environment Is Not Found 29
1.8.2 ASP.NET Considerations on Microsoft Azure 29
1.8.3 SQL Azure Considerations 29
1.8.4 Sitecore Azure Considerations 30
1.8.5 Troubleshooting Sitecore Azure 30
1.8.6 Error – TooManyRequests 30

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

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 introduction of the underlying Microsoft Azure and Sitecore Azure concepts and terminology, this document describes procedures for installing and performing 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:

- Introduction
- Prerequisites
- Installing Sitecore Azure
- Administer Sitecore Azure
- Using Sitecore Analytics with Microsoft Azure
- Frequently Asked Questions
- Known Issues, Considerations, and Troubleshooting

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.1 Introduction

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

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

You can acquire Microsoft 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 the on-premises content editing server to any number of Sitecore Azure cloud services hosted at the various Microsoft Azure locations. Sitecore Azure provides automation and a browser-based user interface to manage your Microsoft Azure deployments.

With Sitecore Azure you get the following benefits:

- Sitecore Azure provides for extreme scalability, stability and simplicity.
- Sitecore Azure provides a highly available infrastructure through Microsoft 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 Microsoft Azure.

1.1.1 Options for Sitecore Azure 7.5 Deployment

Sitecore Azure 7.5 supports content editing as well as content delivery from the cloud. You can decide on the best combination of on-premises 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.

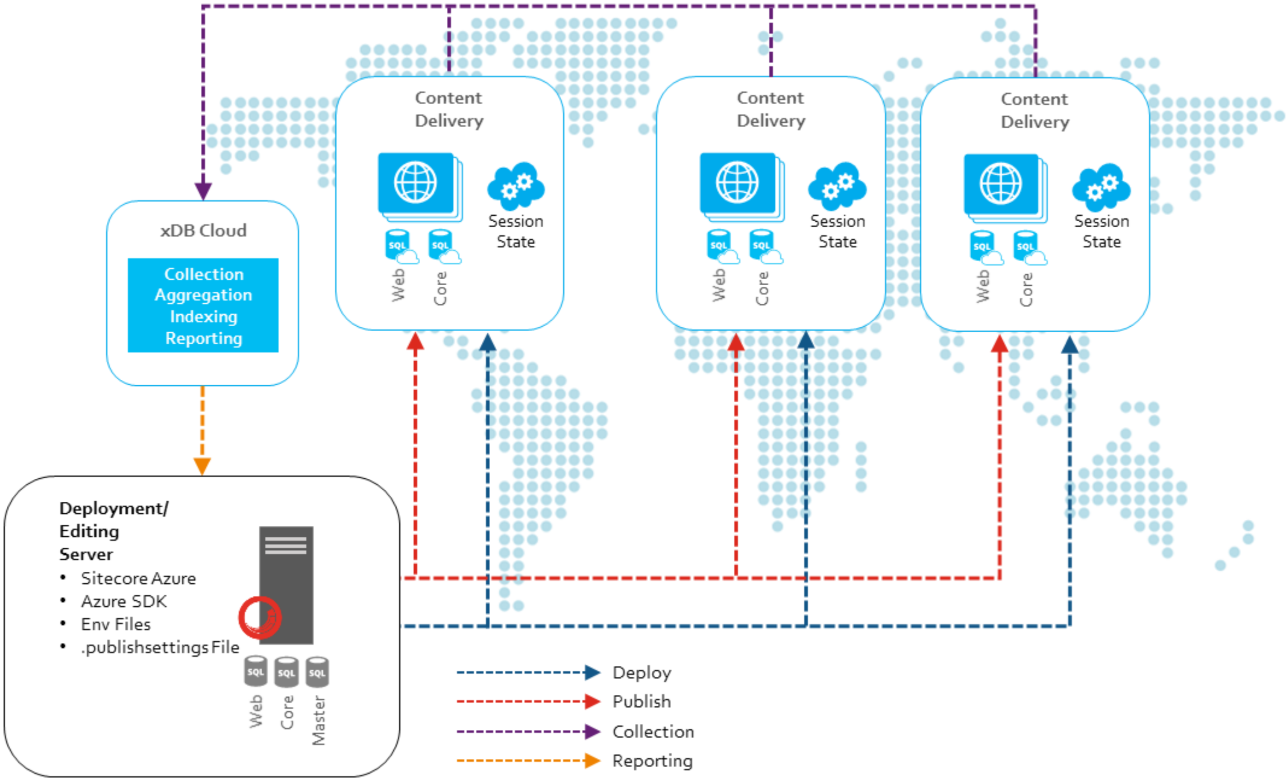
Here are some example deployment topologies for consideration:

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

Sitecore Azure delivery

This example shows an installation that uses an on-premises Sitecore installation for content editing and then publishes to several Sitecore content delivery farms in multiple Microsoft Azure locations.

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

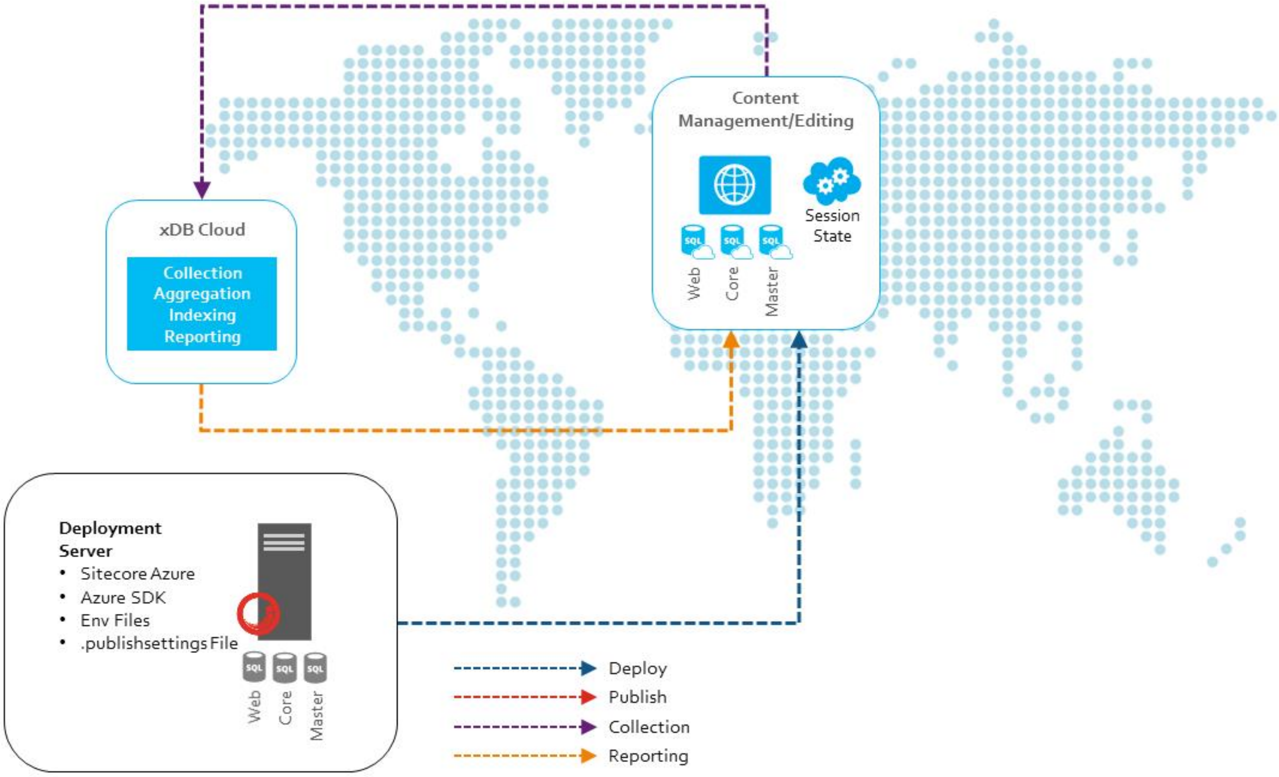


Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

Sitecore Azure live

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 on the content editing server are immediately available to the public.

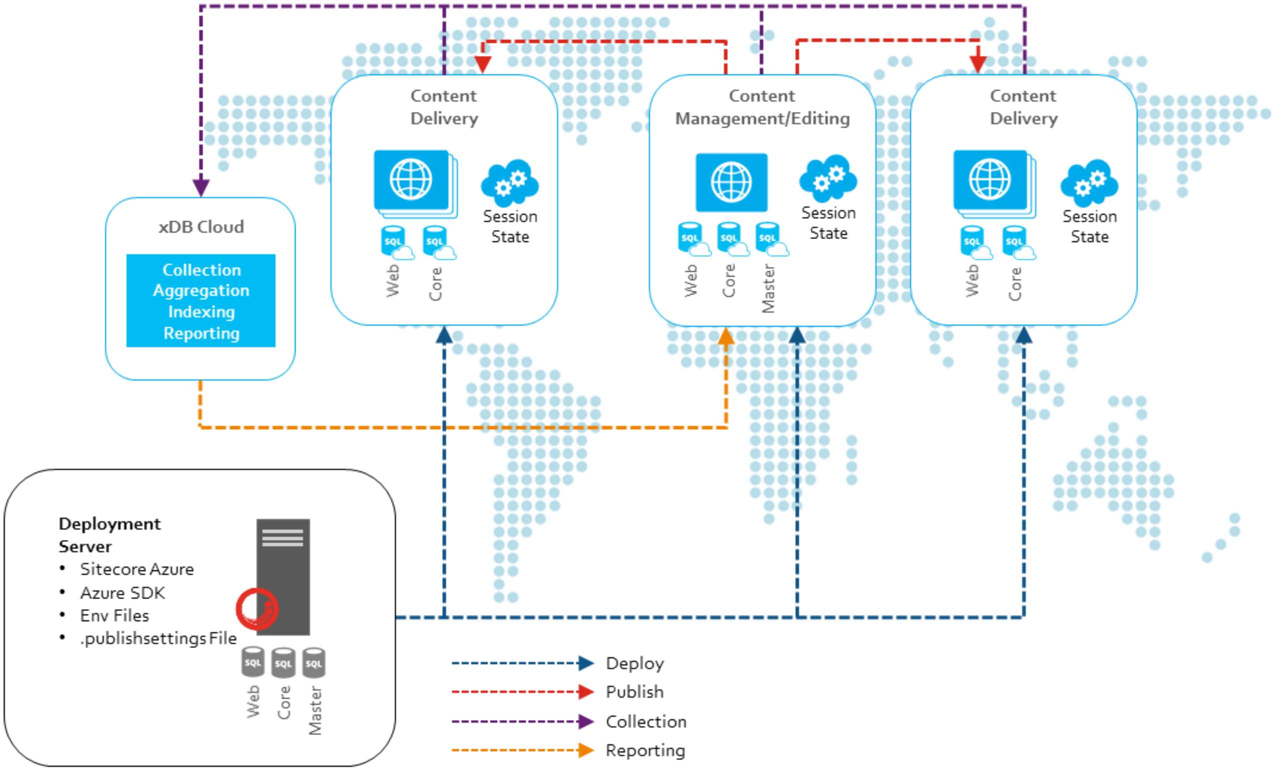
- The on-premises development environment deploys the full environment to Microsoft Azure.
- The Master database is maintained on Microsoft Azure.
- Multi-instance live mode deployment in a single farm.
- The frontend reads from the Master database.



Sitecore Azure authoring and delivery

This example shows how both the content editing and content delivery environments are running in the Microsoft Azure cloud.

- The on-premises development environment deploys the full environment to Microsoft Azure.
- The master database is maintained on Microsoft Azure.
- Multi-instance authoring deployment in a single farm.
- The content delivery deployments are rolled out from the on-premises server and published from the content editing server on Microsoft Azure.

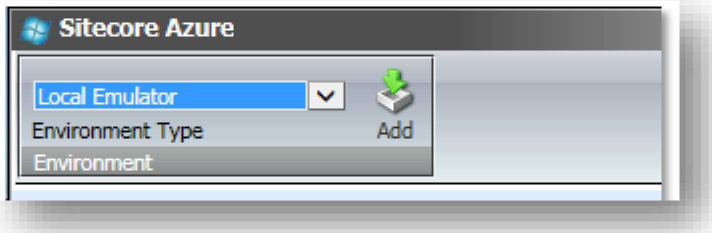


Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.1.2 Using the Microsoft Azure Local Emulator

Sitecore Azure allows you to deploy to the Microsoft Azure local emulator that is included with the Azure SDK, enabling you to test your deployment without incurring any cost.

To test deploy to the local emulator, open Sitecore Azure and in the ribbon, in the **Environment Type** group, select **Local Emulator**.



Note
To run the Microsoft Azure Local Emulator you must have administrative rights to the computer, and when prompted in the notification field at the bottom of the dialog, download and run the `Runner.bat` file.

1.1.3 Using a Microsoft Azure Subscription Versus 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 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. You still get the benefits of using Sitecore Azure tools, for example, the automated features for managing your Azure installations and the licensing benefits such as bursting and prioritized support.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.2 Prerequisites

You must have the following requirements in place on the on-premises content editing server (Web Server) before you can install Sitecore Azure:

- Sitecore CMS 7.5.
- 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/, however, contact and credit card information is required.
- Microsoft SQL Server 2008 R2, Service Pack 1 or newer. In addition, ensure that you have:
    - ENU\x64\SharedManagementObjects.msi
    - ENU\x64\SQLSysClrTypes.msi

  To download and install, go to: http://www.microsoft.com/en-us/download/details.aspx?id=35580
- Windows Azure SDK version 2.4 and the following tools:
    - MicrosoftAzureAuthoringTools-x64.msi
    - MicrosoftAzureLibsForNet-x64.msi
    - MicrosoftAzureStorageTools.msi
    - MicrosoftAzureComputeEmulator-x64.exe
    - WindowsAzureStorageEmulator.msi

  To download and install, go to: http://www.microsoft.com/en-us/download/details.aspx?id=43709

  For more information about the Microsoft Azure Compute Emulator see *Run an Azure Application in the Compute Emulator* at http://msdn.microsoft.com/en-us/library/azure/hh403990.aspx.

  For more information about the Windows Azure Storage Emulator see *Using the Azure Storage Emulator for Development and Testing* at http://msdn.microsoft.com/library/azure/hh403989.aspx.
- Configure your web server’s firewall rules to allow Sitecore Azure traffic. For more information see the section *How to Configure Firewall Rules.*

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.3 Install Sitecore Azure

To install Sitecore Azure, you need CMS 7.5 running on your on-premises content editing server, the Sitecore Azure installation package that you can download from the SDN site. To deploy Sitecore to Azure, you also need a Sitecore Azure Environment file.

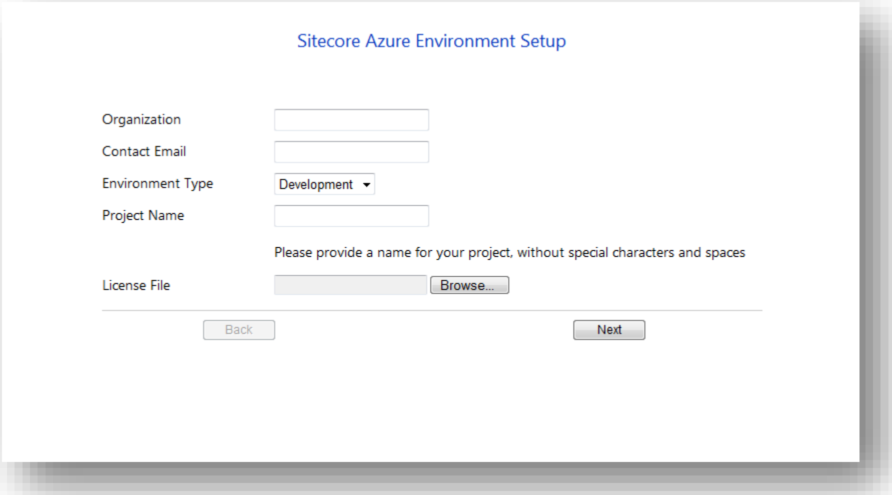
Sitecore environment files are configuration files that contain information about your Microsoft Azure subscription. You can have more than one environment file so that you can support different kinds of environments, for example, you may have separate development, testing and production environments.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

If you do not already have a Sitecore Azure Environment file, you should submit a request for the file before you start to install Sitecore Azure. To receive the file can take up to 48 hours.

Note
While you wait for the environment file, you can install the Sitecore installation package on your Sitecore installation and use the local emulator to you can test your deployment without incurring any costs.

To request a Sitecore Azure Environment file:

- Go to http://sitecoreenvironments.trafficmanager.net/ and enter the required information, such as basic contact information, a project name, and your Sitecore license file.



Sitecore Azure Environment Setup

Organization

Contact Email

Environment Type: Development

Project Name

Please provide a name for your project, without special characters and spaces

License File

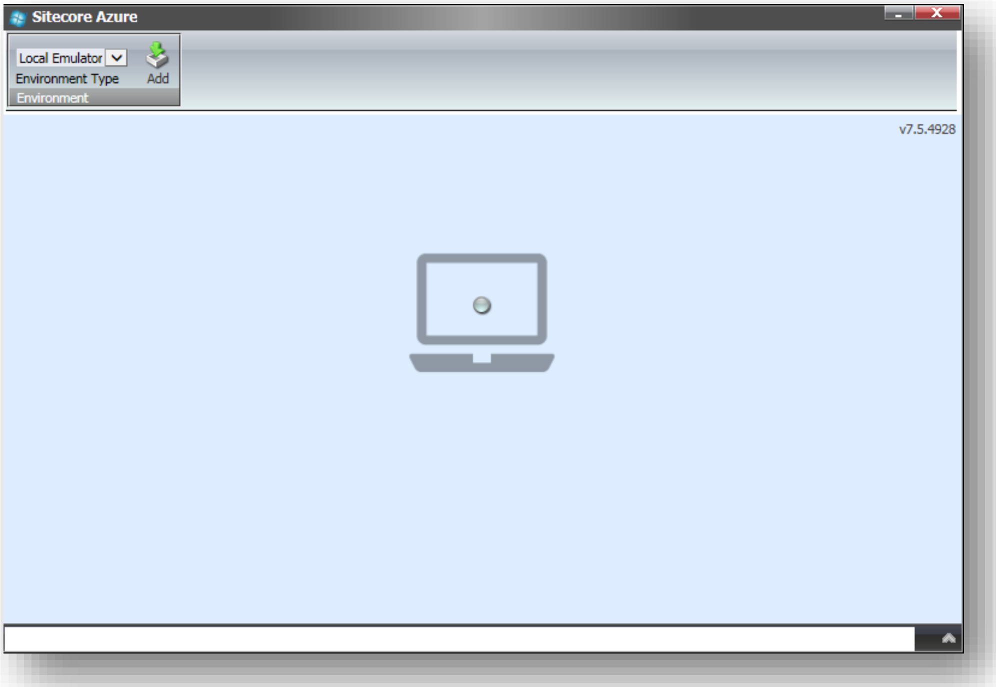
1.3.1 Installing the Sitecore Azure Installation Package

To install the Sitecore Azure installation package:

1. Log in to your on-premises content editing server as an administrator and open the Sitecore Desktop.
2. Click the **Sitecore Start** button, **Development Tools**, **Installation Wizard** and follow the steps of the **Installation Wizard** to install the *Sitecore Azure* package. When prompted, you should overwrite any existing components.
3. When the installation has completed, click the **Sitecore Start** button, and then click **Sitecore Azure**. Sitecore Azure opens with the local emulator environment represented by a single grey

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

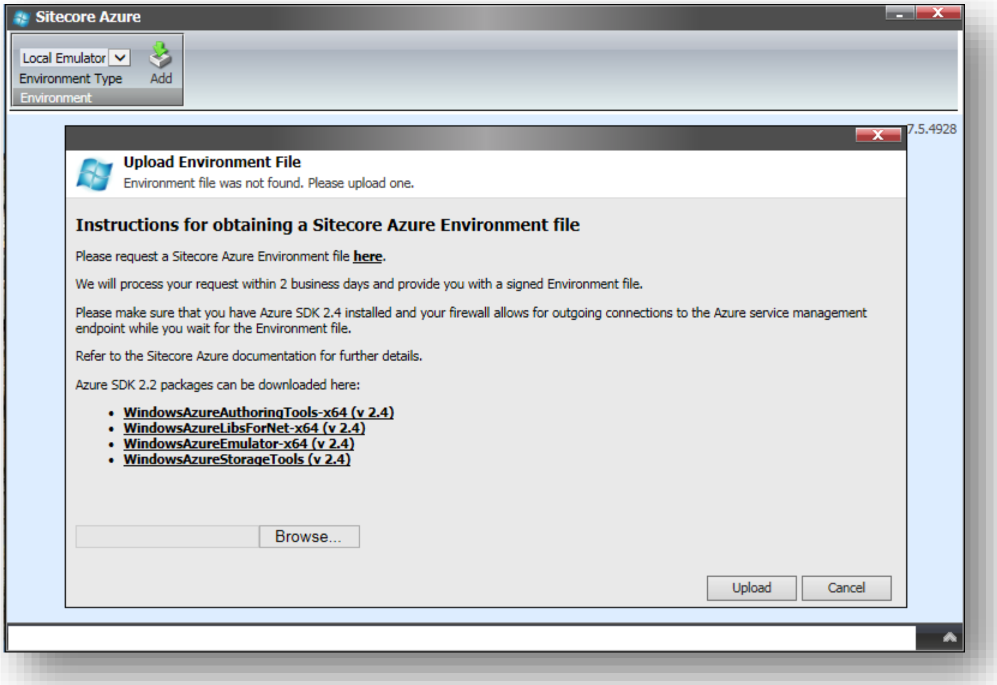
dot.



4. To upload your Sitecore Azure environment file, click **Add**.
5. In the **Upload Environment File** dialog box, click **Browse** and locate the environment file you have received from Sitecore and click **Upload**. Your environment file name will be

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

`Environment.xml`.



Upload Environment File

Environment file was not found. Please upload one.

Instructions for obtaining a Sitecore Azure Environment file

Please request a Sitecore Azure Environment file **here**.

We will process your request within 2 business days and provide you with a signed Environment file.

Please make sure that you have Azure SDK 2.4 installed and your firewall allows for outgoing connections to the Azure service management endpoint while you wait for the Environment file.

Refer to the Sitecore Azure documentation for further details.

Azure SDK 2.2 packages can be downloaded here:

- **WindowsAzureAuthoringTools-x64 (v 2.4)**
- **WindowsAzureLibsForNet-x64 (v 2.4)**
- **WindowsAzureEmulator-x64 (v 2.4)**
- **WindowsAzureStorageTools (v 2.4)**

6. At this point, you can choose to set up a test environment using the Microsoft Azure Emulator or select a test or production environment.

1.3.2 Deploying to the Microsoft Azure Cloud

The environment type options that you can deploy to may vary depending on which type of Microsoft Azure subscription you have.

Before you deploy, consider the following:

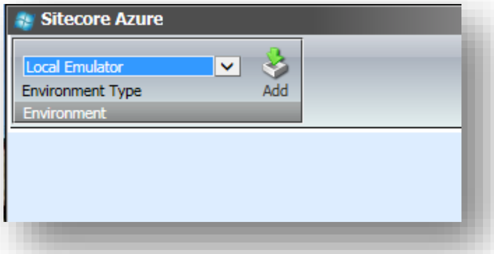
- To avoid the need to publish after deployment, before creating a Sitecore Azure farm, publish to the default publishing target named Internet (the Web database).
- To minimize deployment time, remove unnecessary files from the document root of the content editing environment.

Note
For information about how Sitecore Azure assembles a solution to deploy, see the section *How Sitecore Assembles a Solution to Deploy.*

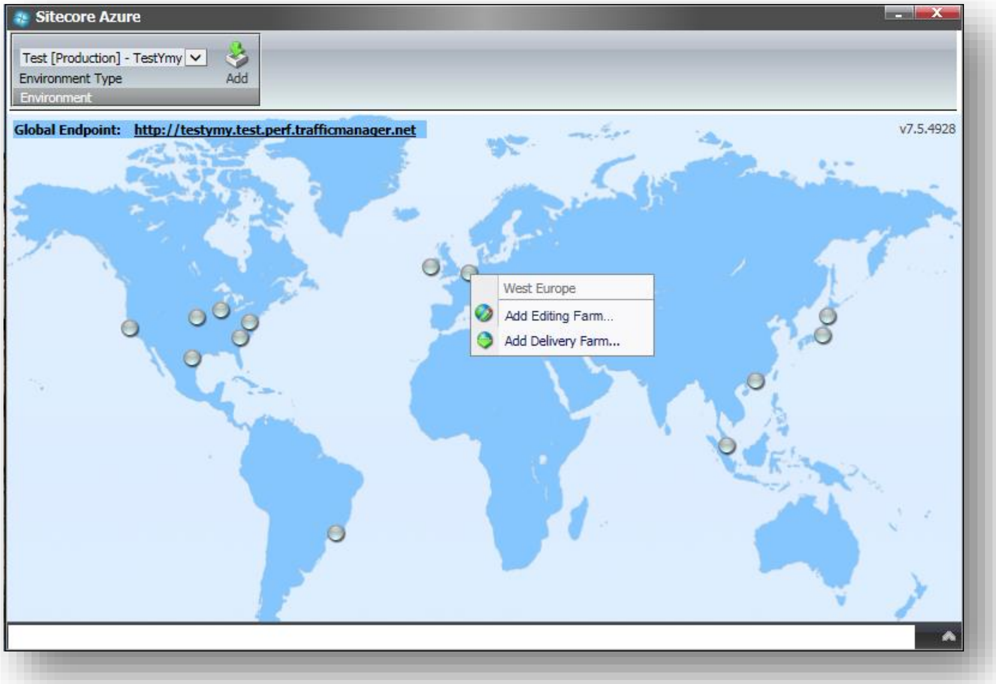
To add an editing farm or a delivery farm to a Microsoft Azure hosting location:

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1. Open Sitecore Azure and in the ribbon, in the **Environment Type** group, click the environment that you want to deploy to.



2. On the map, click the relevant Microsoft Azure hosting location and in the menu that appears, click the farm that you want to create.



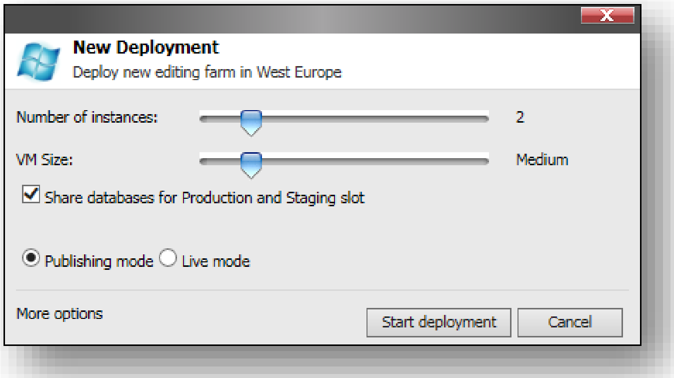
3. In the **New Deployment** dialog box, specify the relevant settings for:

   - **Number of instances** – specify the number of instances you want to deploy. To ensure high availability, you should choose at least two instances.
     For information about changing the number of instances in a location, see the section *Changing 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. You can use a small size for evaluation and test deployments.

Note
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.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

- **Shared database settings -** the default database selection is to share databases between Production and Staging. If you have separate databases for staging and production, you should clear this check box.
- **Publishing options** – when you create an editing farm you can select **Publishing mode** to use a controlled publishing environment. If you select **Live mode,** the changes in the content editing farm will be available on your website immediately. In live mode, the content is not automatically transferred to the delivery farms. For more information about live mode, see the manual *Content Reference for Sitecore CMS 6.0 or later*.



New Deployment

Deploy new editing farm in West Europe

Number of instances: 2

VM Size: Medium

☑ Share databases for Production and Staging slot

● Publishing mode ○ Live mode

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

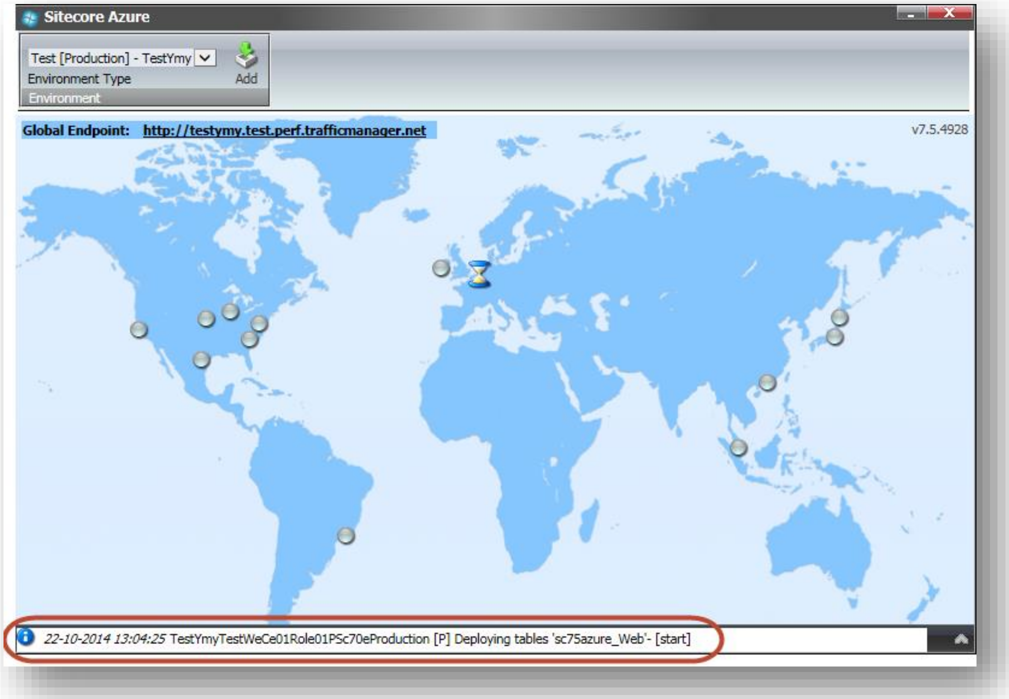
4. Click **Start deployment**.

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.

You can follow the progress of the deployment in the info bar at the bottom of the **Sitecore Azure** dialog box. During the deployment process of an Editing farm, you are able to create a Delivery

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

farm on another location



Global Endpoint: http://testymy.test.perf.trafficmanager.net

```
22-10-2014 13:04:25 TestYmyTestWeCe01Role01PSc70eProduction [P] Deploying tables 'sc75azure_Web'- [start]
```

Important

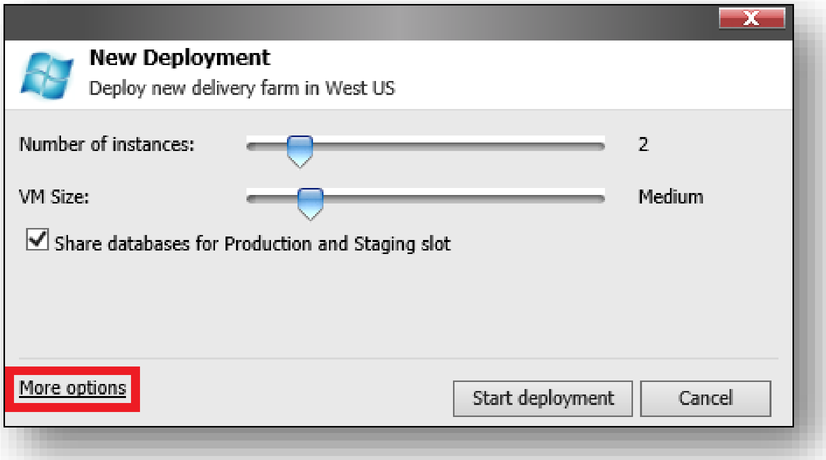
If you do not have a valid subscription for Sitecore xDB Cloud, you must disable the Analytics feature before deployment starts.

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

To change the `Analytics.Enabled` setting:

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

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. *Copyright © 2001-2015 Sitecore. All rights reserved.*



New Deployment

Deploy new delivery farm in West US

Number of instances: 2

VM Size: Medium

☑ Share databases for Production and Staging slot

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

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:msxsl="urn:schemas-microsoft-com:xslt" exclude-result-prefixes="msxsl">
  <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.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.4 Administer Sitecore Azure

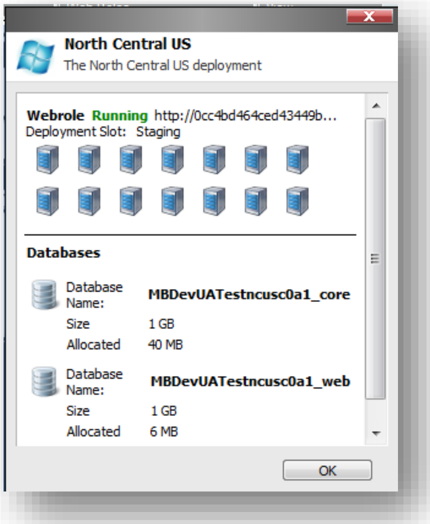
Sitecore Azure provides a map overlaid with icons that represent available Microsoft Azure hosting locations. Each icon and its color indicate the status of the deployment in that environment:

- **Gray Circle**: Unused. You can create a content delivery 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. 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 administer the location, or the current operation for the location is queued.

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

To perform an action on an existing farm:

1. In Sitecore Azure, click a location circle and in the drop-down menu, click the relevant editing or delivery farm.
2. In the menu, select the action that you want to perform on the farm, for example, click **Properties** to see a status of the selected location. You can open multiple Sitecore Azure windows in the Sitecore desktop to access the status of tasks in multiple locations.



North Central US
The North Central US deployment

Webrole Running http://0cc4bd464ced43449b...
Deployment Slot: Staging

Databases

Database Name: **MBDevUATestncusc0a1_core**
Size 1 GB
Allocated 40 MB

Database Name: **MBDevUATestncusc0a1_web**
Size 1 GB
Allocated 6 MB

The following sections describe how you can administer the active farms in your environment.

Warning
Some 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.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.4.1 Publishing 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 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. Note that an on-premises solution that runs on live mode will not automatically have changes reflected in your Sitecore Azure deployments in the cloud.

1.4.2 Deploying 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 click **Upgrade Files**.

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.

1.4.3 Changing 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 click **Scale**.
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.

1.4.4 Moving a location from staging to production

To move a location from Staging to Production:

- In **Sitecore Azure**, click the icon for the relevant location, expand the farm that you want to move and click Swap.

1.4.5 Suspending or Starting the Instances in an Environment

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

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

  Important
  When you suspend a role it is not deleted the instance and costs will still be incurred while the role is suspended. If you do no longer use the instance, you should remove it from the cloud entirely.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

To start the instances in an environment:

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

1.4.6 Removing 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 click **Delete**.

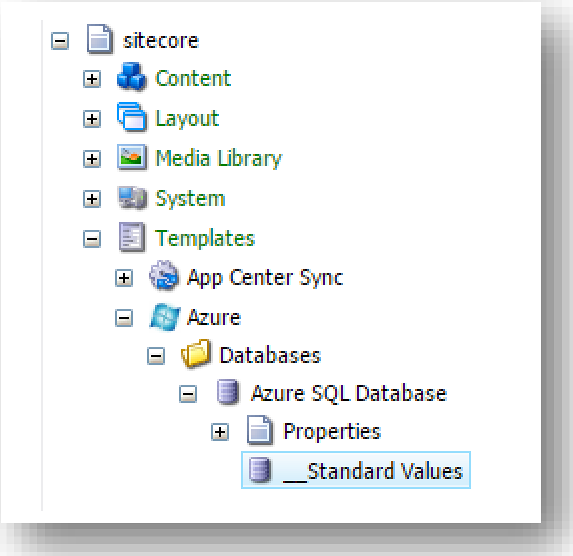
1.4.7 Changing the default Azure SQL Database Service Tier

The default service tier is 250 GB – Standard S2 edition, but you can change this on the template before deployment.

Note
You can find an overview of the supported predefined Azure SQL Database Service Tiers at
http://msdn.microsoft.com/en-us/library/azure/dn741336.aspx

To change the default Azure SQL Database Service Tier:

1. In the Content Editor, navigate to the item /sitecore/Templates/Azure/Databases/Azure SQL Database/__Standard Values



2. In the **Edition** field, specify the service tier and in **Size** field, specify the maximum database size. This sets all databases (Core, Master and Web) to the same size and service tier.

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

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.4.8 Enable batch upload

To enable 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.

For additional information, go to: http://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlbulkcopy.batchsize(v=vs.110).aspx

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.5 Using Sitecore Experience Platform 7.5 with Microsoft Azure

Using the Sitecore Experience Database with Microsoft AzureThe Sitecore xDB 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, and makes the data available to marketers to manage the customer experience in real time.

In Sitecore 7.5, you can choose to install Sitecore as an on-premises solution, as a hybrid solution (partly on the premises and partly in the cloud), or as an entirely in-the-cloud solution.

For information about installing and configuring the Sitecore xDB, see the respective documentation at http://sdn.sitecore.net/Reference/Sitecore%207/Sitecore%207,-d-,5%20Documentation.aspx

For information about installing and configuring Sitecore xDB Cloud Edition, see the *xDB Configuration Guide* at http://sdn.sitecore.net/Reference/Sitecore%207/xDB%20Cloud%20Quick%20Start%20Guide.aspx

1.5.1 Scalability Options for the Sitecore Experience Database

The Sitecore Experience Database (xDB) is a flexible and scalable architecture that is capable of handling very large amounts of customer data. You can scale both up and down from small single server, low traffic solutions to larger solutions consisting of hundreds of servers designed to handle high volumes of traffic.

For an overview of xDB scalability options, see the manual *xDB Overview and Architecture* at http://sdn.sitecore.net/sdn5/reference/sitecore%207/xdb%20overview%20and%20architecture.aspx

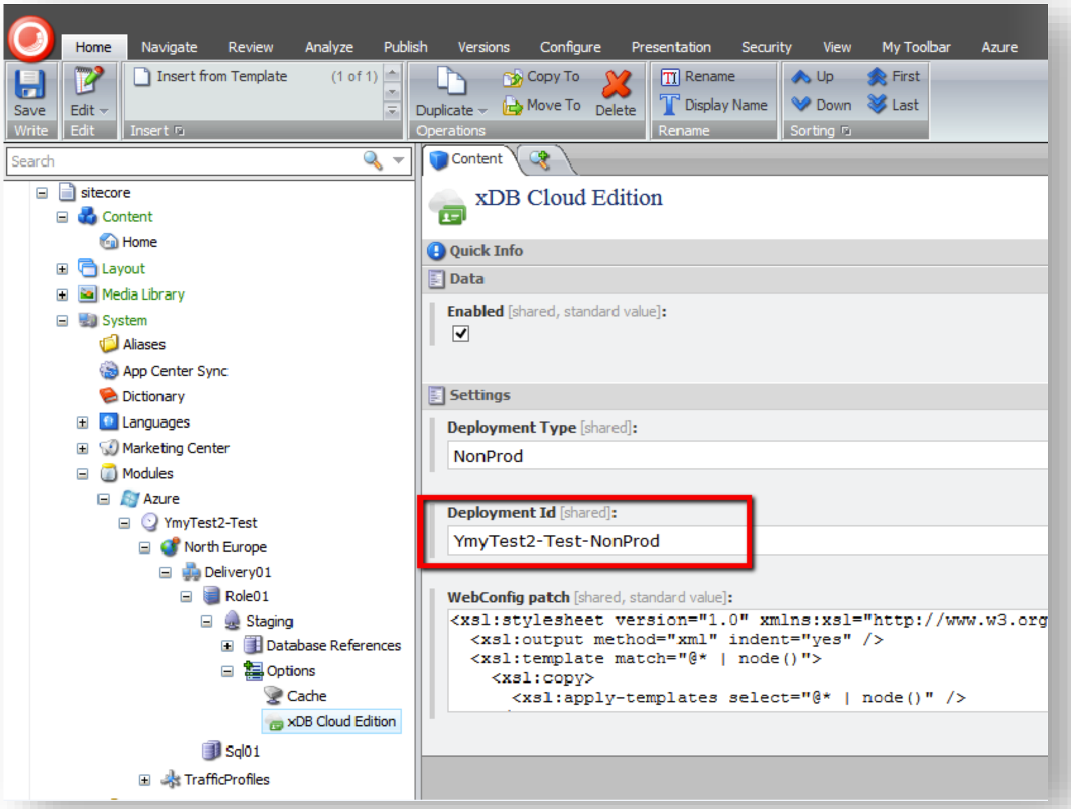
1.6 Getting Started with Sitecore Azure

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 an individually deployed xDB.

1.6.1 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 same deployment ID and license ID to connect to xDB. If you need a different xDB deployment ID go to **xDB Cloud Edition**, and change the deployment ID in the **Deployment ID** field.

Deployment ID



Enabled [shared, standard value]:
☑

Deployment Type [shared]:
NonProd

Deployment Id [shared]:
YmyTest2-Test-NonProd

WebConfig patch [shared, standard value]:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org
  <xsl:output method="xml" indent="yes" />
  <xsl:template match="@* | node()">
    <xsl:copy>
      <xsl:apply-templates select="@* | node()" />
```

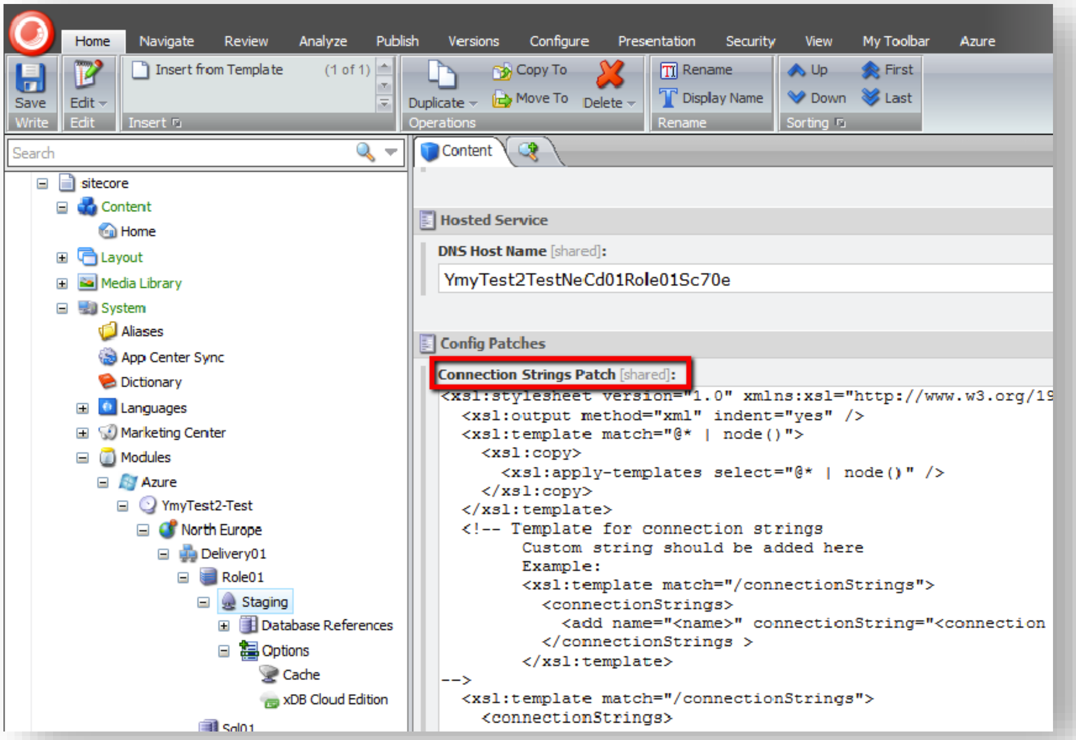
1.6.2 Connect to your own xDB from Sitecore Azure

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).

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

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



DNS Host Name [shared]:

YmyTest2TestNeCd01Role01Sc70e

Connection Strings Patch [shared]:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/19
  <xsl:output method="xml" indent="yes" />
  <xsl:template match="@* | node()">
    <xsl:copy>
      <xsl:apply-templates select="@* | node()" />
    </xsl:copy>
  </xsl:template>
  <!-- Template for connection strings
        Custom string should be added here
        Example:
        <xsl:template match="/connectionStrings">
          <connectionStrings>
            <add name="<name>" connectionString="<connection
          </connectionStrings >
        </xsl:template>
-->
  <xsl:template match="/connectionStrings">
    <connectionStrings>
```

1.7 Frequently Asked Questions

This section provides answers to some of the most common questions.

1.7.1 How Do Sitecore Azure Content Delivery Environments Clear Caches?

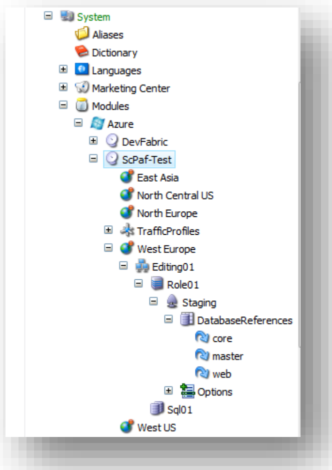
Sitecore Azure uses the same mechanism as on premise Sitecore solutions to clear caches. After completing a publishing operation, the Sitecore Azure content editing server writes a timestamp to the publishing target database. Sitecore Azure content delivery server check the timestamp on each request,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

1.7.2 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 that 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.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.7.3 How Sitecore Azure Assembles 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-premises Sitecore installation to `$(datafolder)\Data\AzurePackages\(xx)Azure\SitecoreWebSite` for a Microsoft 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.

1.7.4 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 Microsoft 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 Microsoft 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 Microsoft Azure may change the underlying VIP.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.7.5 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" />
```

1.7.6 How to Enable SSL in a Sitecore Azure Project

Basic information about using SSL with Microsoft 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?*

1.7.7 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 Microsoft Azure management portal. It will create certificates if needed and you can specify or change access.

This allows for enabling Remote Desktop Protocol (RDP) for Sitecore customer service or Microsoft technical support.

1.7.8 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.

1.7.9 How to Configure Firewall Rules

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

For more information about the Microsoft datacenters, go to:

- 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]

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

- **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]
- **Central US**
- **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]
- **East US 2** -
- **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]
- **Brazil South -**

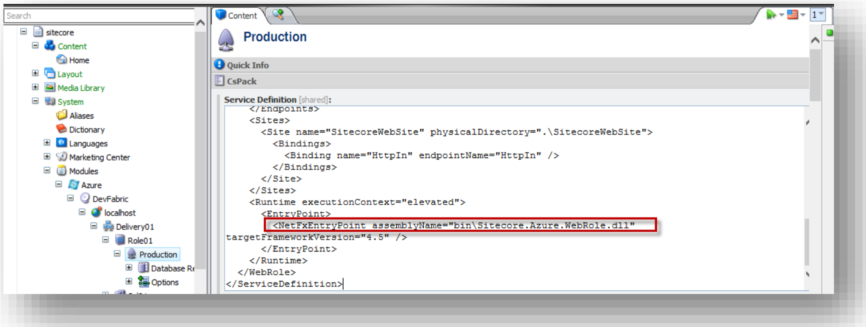
Note
Locations and IP addresses may change and new locations may be added from time to time. In addition, some locations may be closed for new deployments. If you have trouble with these addresses, contact Microsoft technical support.

1.7.10 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 Microsoft 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:



Service Definition [shared]:

```
    </Endpoints>
    <Sites>
      <Site name="SitecoreWebSite" physicalDirectory=".\SitecoreWebSite">
        <Bindings>
          <Binding name="HttpIn" endpointName="HttpIn" />
        </Bindings>
      </Site>
    </Sites>
    <Runtime executionContext="elevated">
      <EntryPoint>
        <NetFxEntryPoint assemblyName="bin\Sitecore.Azure.WebRole.dll"
targetFrameworkVersion="4.5" />
      </EntryPoint>
    </Runtime>
  </WebRole>
</ServiceDefinition>
```

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

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

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

1.8 Known Issues, Considerations, and Troubleshooting

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

1.8.1 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:

- **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.
- **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**.

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.

1.8.2 ASP.NET Considerations on Microsoft Azure

Consider the following for any ASP.NET application such as Sitecore on Microsoft 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 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 API's 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.

1.8.3 SQL Azure Considerations

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

- 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).

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.

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.

1.8.4 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.

1.8.5 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 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.

1.8.6 Error – TooManyRequests

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 that is located in \website\AppConfig\Include and then adjust the Azure.CheckAsyncOperationStatusIntervalInMS value.

Sitecore® is a registered trademark. All other brand and product names are the property of their respective holders. The contents of this document are the property of Sitecore. Copyright © 2001-2015 Sitecore. All rights reserved.