Sitecore Azure 1.0.5

Sitecore Azure Troubleshooting Guide

A Guide for Sitecore Administrators

Considerations and Troubleshooting

This section describes Consideration for implementing solutions with Windows Azure, SQL Azure, and Sitecore Azure and troubleshooting steps.

ASP.NET Considerations on Windows Azure

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

- Eliminate use of ASP.NET session or consider `TableStorageSessionStateProvider` for session management.

- Remote access to the Windows Azure environment is extremely limited. You cannot attach a debugger to the Windows Azure environment. You cannot directly access the file system of a Windows Azure environment, including log files.

- 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 an individual file to Windows Azure. You must deploy the entire document root.

SQL Azure Considerations

Consider the following when using SQL Azure with applications such as Sitecore.

- When accessing an 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 management environment). For more information about SQL Azure security, see http://msdn.microsoft.com/en-us/library/ff394108.aspx.

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

- SQL Azure requires SQL Server 2008R2 or later SQL Management Studio.

Sitecore Azure Considerations

When you use Sitecore Azure:

- You cannot use the automated installer to install Sitecore on Windows Azure. You can use the automated installer to create a local Sitecore instance, and then use Sitecore Azure to replicate that environment to the cloud.

- Sitecore Azure *only* supports SQL Azure databases.

- Sitecore Azure does not support file media.

- Windows Azure uses certificates to exchange files securely over HTTP. Sitecore Azure subscriptions are restricted to specific certificates and licenses. SQL Azure uses IP address restrictions for database security. Work with Sitecore Azure services if you plan to move your content management instance from one machine or location to another.

- Sitecore has not tested modules including the Online Marketing Suite (OMS) on Windows Azure.

- Sitecore originally developed Sitecore Azure for Sitecore CMS 6.3 targeting the 3.5 version of the .NET framework. Sitecore Azure now supports .Net 4.0. In order to enable .NET 4.0 you should switch AppPool of your site to .NET 4.0 mode, run Sitecore Azure and update your environment.

- Because Sitecore Azure does not deploy the `/sitecore` directory, you should configure the `ErrorPage`, `ItemNotFoundUrl`, `LayoutNotFoundUrl`, `LinkItemNotFoundUrl`, `NoAccessUrl`, `NoLicenseUrl`, and `SystemBlockedUrl` settings in the `web.config` file to specify files that exist outside of the `/sitecore` directory.

Troubleshooting Sitecore Azure

Here are some general troubleshooting guidelines for Sitecore Azure. See the section *Troubleshooting Sitecore Azure Installations* for steps to resolve specific problems:

- Restart ASP.NET.

- Clear the browser cache.

- Investigate the Sitecore logs in the content management environment for errors.

- Suspend and restart the instances at a location.

- Remove the instances from the environment and deploy again.

- After initiating a deployment, copy the `$dataFolder/WebRole` subdirectory to a temporary location, configure that subdirectory 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

Setting the `mode` attribute to `Off` can expose detailed error information in the Sitecore 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.

Troubleshooting Sitecore Azure Installations

1. When you try to run the Sitecore Azure application it cannot be started and hangs in the "Connecting to the cloud" state. Check to see that the `<Path to Website>\sitecore\shell\Applications\Azure\certificate.cer` file was created during Sitecore Azure installation. If the file does not exist, run the `MakeCert.bat` file to create it. `MakeCert.bat` must be run with Administrator privilege.

Sitecore Azure application in stuck in "Connecting to the cloud" state

Symptoms

When you try to run Sitecore Azure application it cannot be started and hangs in the "Connecting to the cloud" state

Cause 1: Certificate for Sitecore Azure was not created

In this case you will find the following error in your Sitecore log file: "Certificate was not found. Please run `MakeCert.bat` from `Website\sitecore\shell\Applications\Azure` folder".

Resolution:

Check if the file `certificate.cer` exists in the folder `<Path to Website>\sitecore\shell\`

`Applications\Azure`. If the file is absent, run `MakeCert.bat`. This script will create and install the certificate to the machine certificate storage.

The `MakeCert.bat` script should be run as Administrator.

Cause 2: Certificate was not installed in the system or the necessary permissions were not granted

Error "Certificate is not valid for SubscriptionID : < Subscription id>" will appear in the Sitecore log.

Resolution:

Grant the account that is used to run the Application pool the required security rights to the certificate:

Run **Microsoft Management Console**.

In the **File** menu, click **Add/Remove Snap-in.**

In the **Available snap-ins** list, select **Certificate** and click **Add**.

In the **Certificates snap-in** dialog box select the **Computer Account** check box:
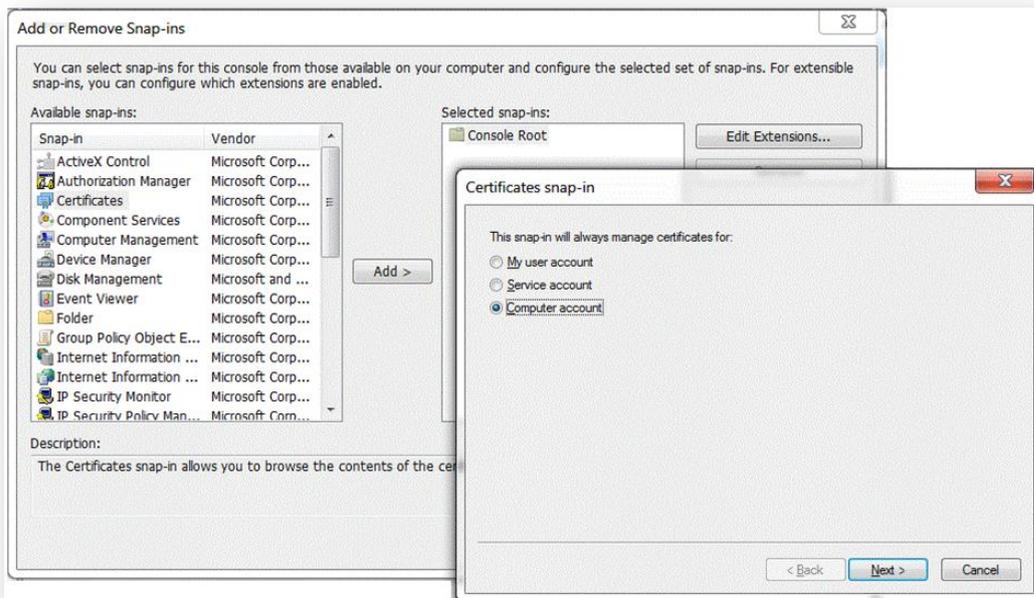
In the Select Computer dialog box, select the computer that you want to manage.
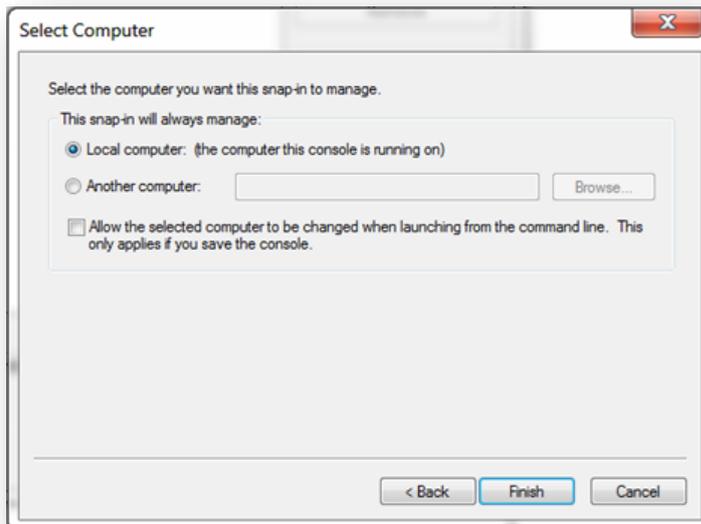
If you cannot see a certificate with the name "Windows Azure Authentication Certificate"- navigate to `<Path to Website>\sitecore\shell\Applications\Azure` and run `MakeCert.bat` as Administrator.

In Microsoft Management Console, select Windows Azure Authentication Certificate, right click it, and select All tasks — Manage Private keys.
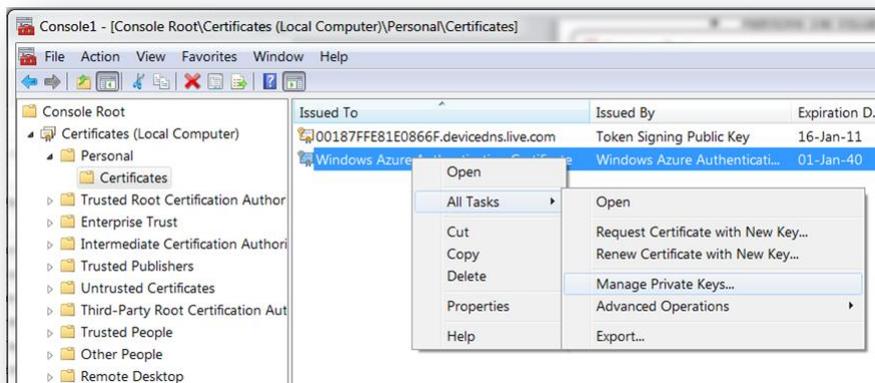
Add the account that is used to run the corresponding Application pool and grant it Read permission:
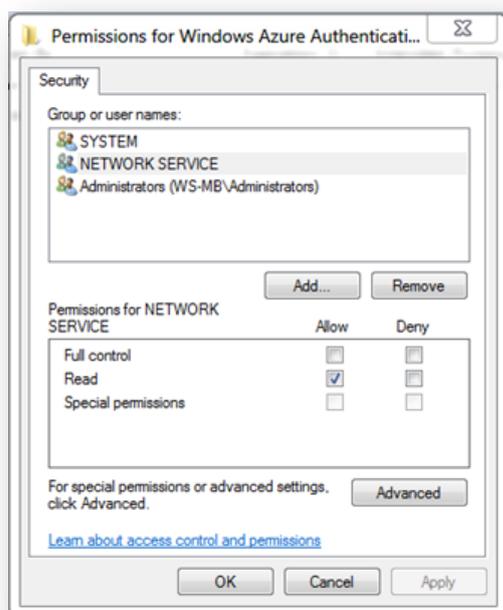
Cause 3: Wrong environment information

The applied environment information was generated for another installation. Search for the error message "Unable to connect to SubscriptionID : <SubscriptionID>" in the Sitecore log file. Check if the Subscription ID is the correct one.

In the Sitecore Azure User Interface if you see an error message that states that the certificate is not valid this is also as a symptom for this same issue.

Resolution:

Clean the environment info. Follow these steps:
1. Make sure that the Sitecore Azure application is closed
2. Login to the Sitecore desktop.
3. Open the Content Editor.
4. Navigate to the item `/sitecore/system/Settings/Azure`.
5. Delete the contents of the Environment field. Save the Item
6. Run the Sitecore Azure application. You will be asked to provide an Environment File.
7. Upload your Environment file.

Note
In this example the NETWORK SERVICE account is used as an example, this may vary on different configurations. You can learn more about Application Pool Identities at http://learn.iis.net/page.aspx/624/application-pool-identities/.

Cause 4: Certificate was not uploaded to Windows Azure

Error "Certificate is not valid for SubscriptionID : < Subscription id>" will appear in the Sitecore log.

Resolution:

Upload the certificate, following these steps:
1. Login to `https://windows.azure.com/`.
2. Click "Hosted services, Storage accounts & CDN" from the left menu.
3. Select "Management Certificates".
4. Press "Add Certificate" in the ribbon.
5. Choose the relevant subscription and upload certificate.

Files from the temp folder are not deployed

Files that are located in the `\website\temp` folder are not deployed to Windows Azure. One example of a file that is located in this folder is the `dictionary.dat` file which is used to translate content language.

The `\website\temp` folder is added to the Exclude Directory list in the **/sitecore/system/Settings/Azure/EnviromentTypes/ContentDelivery** item. You can delete **"temp;"** value from the **WebsiteExcludeDirectories** field to allow this folder to be deployed to Microsoft Windows Azure.