# Introduction

This document gives a walkthrough how to deploy the VENUS-C Generic Worker (GW) on the Windows Azure platform. Screenshots are used for illustration purposes in order to ease the explanation of concepts and how-tos. The complete deployment depending on the internet speed will take between 30-45 minutes.

This document organized as follows:

* Prerequisites to Deploy GW on Azure
* Deploying GW
* Post Deploy Steps
* Useful Tools

The prerequisites part, explains why certificates are needed to deploy the GW and how to create them. The GW specific settings and configuration are explained in the Deploying GW section. After the deployment completes, final steps need to be followed which are explained in the Post Deploy Steps section. After the first 3 sections, the GW service will be ready to be used. The 4th section in the document, the Tools section, is highlighting some of the tools which can ease the tasks explained in the first 3 sections. It is strongly recommended to read the document from beginning to the end at least once and use the tools after understanding to concepts.

**Prerequisites to Deploy GW on Azure**

## Create Certificates

In general, at least two different certificates are needed for the GenericWorker. One for managing the deployment (e.g. scaling) and one for securing the endpoints (SSL / STS). Two ways will be described for generating these certificates

#### SSL / STS

At least one certificate is needed for securing the endpoints like JobManagement endpoint for submitting jobs or Scaling endpoint for increasing or decreasing the amount of virtual machines used by the GenericWorker.

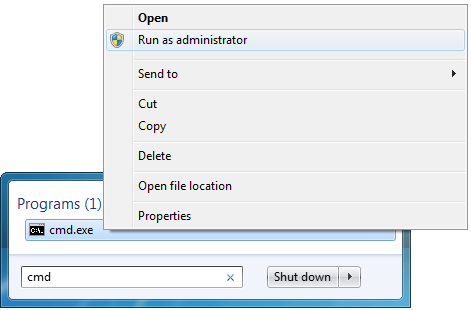
For the first try, the same certificate for SSL and STS can be used. So basically this certificate is used when calling the endpoints using https and for encrypting / signing security tokens issued by the STS.

There are several possibilities to create self-signed certificates. This document will show examples for using the makecert tool which is available on many Windows 7 machines and for using OpenSSL.

##### **makecert**

* Start a command prompt as administrator.

Windows 7 this can be done as shown by searching “cmd” in the windows start menu, right-clicking “cmd.exe” and then selecting “Run as administrator”.



* Go to the directory where makecert.exe is located. Usually this is a path like this: “C:\Program Files (x86)\Microsoft SDKs\Windows\v7.0A\Bin”
* Enter the following line in the command prompt:

makecert -r -pe -n "CN=MyAzureSTS" -a sha1 -len 2048 -sky exchange -sr localmachine -ss My -sv "C:\Certificates\_Test\MyAzureSTS.pvk" "C:\Certificates\_Test\MyAzureSTS.cer"

You can define a CN name for your certificate on your own, this is not fixed to “MyAzureSTS”. You are also free to change the directory and / or names of the files storing your public and private key (\*.cer / \*.pvk).

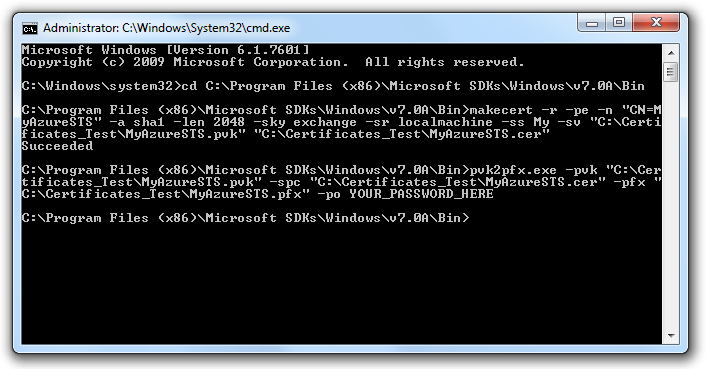
* You will be prompted for a password protecting the private key of your certificate.
* After executing this command, you should get the message “Succeeded” in the command prompt and there should be the two files “AzureSTS.pvk” and “AzureSTS.cer” in the directory you have specified.
* You now have a certificate, but in order to use it within Azure the public and private key have to be packaged into a single container file.

This can be done by entering the following command:

pvk2pfx.exe -pvk "C:\Certificates\_Test\MyAzureSTS.pvk" -spc "C:\Certificates\_Test\MyAzureSTS.cer" -pfx "C:\Certificates\_Test\MyAzureSTS.pfx" -po YOUR\_PASSWORD\_HERE

It is very important to enter a password for the pfx-file again, otherwise you will not be able to use the certificate in Azure. You will also be prompted for the password you entered when executing the makecert command.

Note that you have to adopt the filenames and paths if you changed them in the first command.



##### **openssl**

* Start a command prompt as administrator.

This can be done by searching “cmd” in the windows start menu, right-clicking “cmd.exe” and then selecting “Run as administrator”.

* Go to the directory where OpenSSL has been installed. For this example the path is like this: “C:\Program Files\OpenSSL-Win64\bin”
* Enter the following line in the command prompt:

openssl req -x509 -nodes -days 365 -subj /CN=MyAzureSTS -newkey rsa:2048 -keyout C:\Certificates\_Test\MyAzureSTS.pem -out C:\Certificates\_Test\MyAzureSTS.pem

You can define a CN name for your certificate on your own, this is not fixed to “MyAzureSTS”. You are also free to change the directory and / or name of the file storing your public and private key (\*.pem).

* After executing this command, you should be able to find the file “MyAzureSTS.pem” in the directory you have specified.
* You now have a certificate, but in order to use it within Azure the public and private key have to be packaged into a single container file.

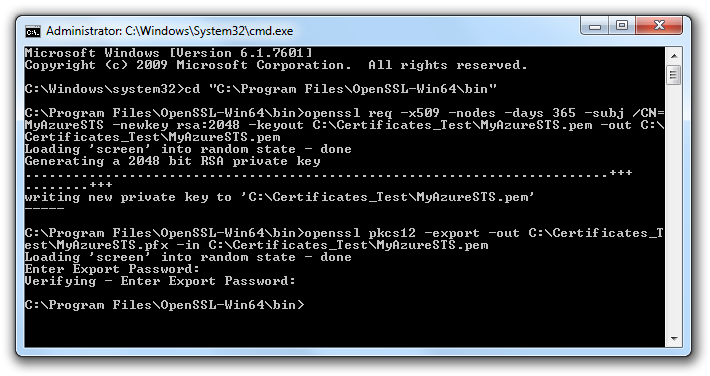
This can be done by entering the following command:

openssl pkcs12 -export -out C:\Certificates\_Test\MyAzureSTS.pfx -in C:\Certificates\_Test\MyAzureSTS.pem

You will be asked for a password for protecting your certificate. It is very important to enter a password for the pfx-file, otherwise you will not be able to use the certificate in Azure.

* Install the certificate in your local personal store by double-clicking it.

Note that when using the commands above the keys in pem file are not encrypted and visible to everyone.



#### Azure Management

Another certificate is needed for managing your deployment which is basically increasing or decreasing the number of virtual machines (VMs). Azure uses public / private key authorization for managing the deployment. This means, a public key is installed in Azure and everyone who knows the corresponding private key is able to manage the deployment. Therefore it is highly recommended to use different certificates for securing the endpoints and doing the management.

Please use the same ways described before to create a management certificate.

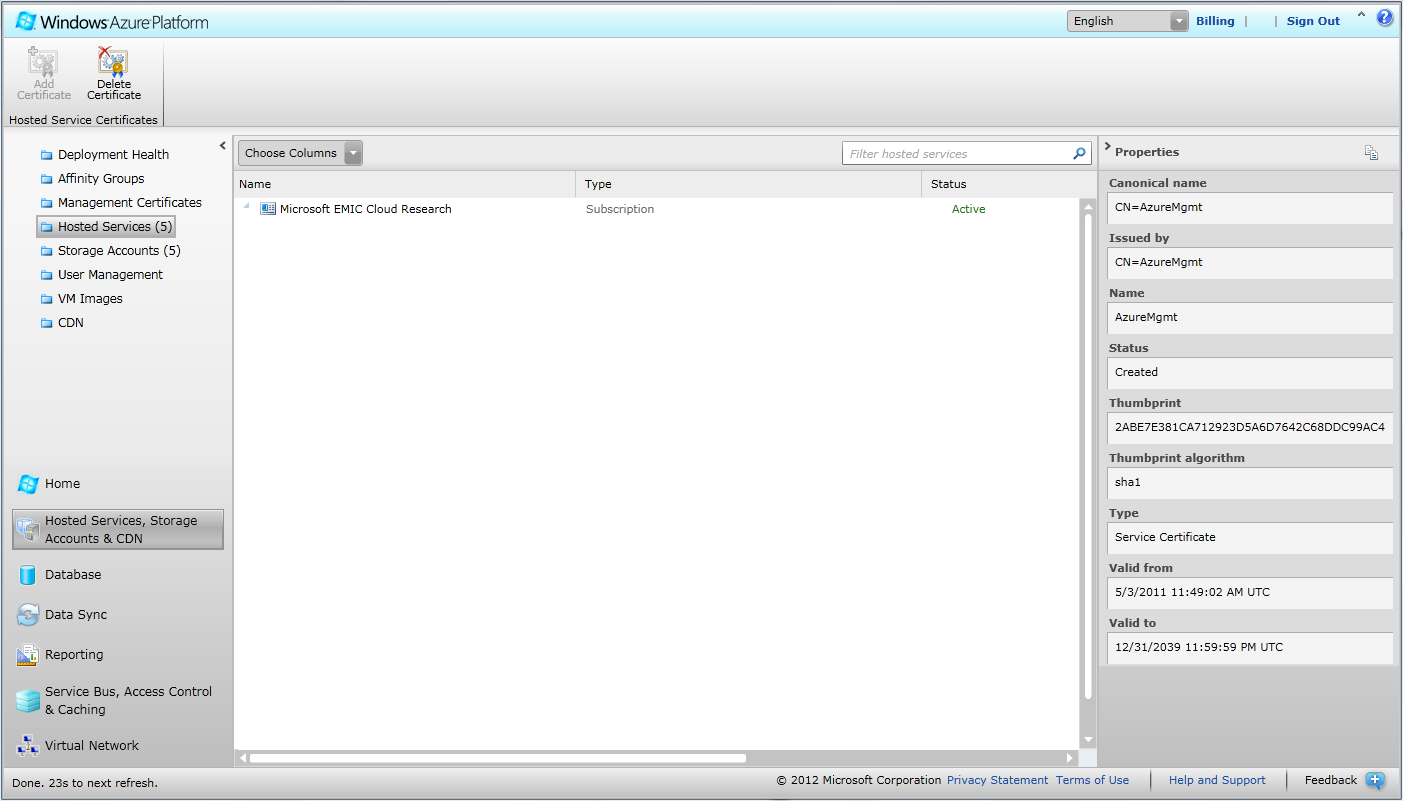
If you use OpenSSL, for the managing certificate another small step is required. The certificate is available in pem and pfx format. Both formats include public and private key. To be able to use this certificate as a managing certificate, the public key is needed in a single file.

The public key can be extracted by the following command:

openssl x509 -outform der -in C:\Certificates\_Test\MyAzureSTS.pem -out C:\Certificates\_Test\MyAzureSTS.cer

## Create Hosted Service

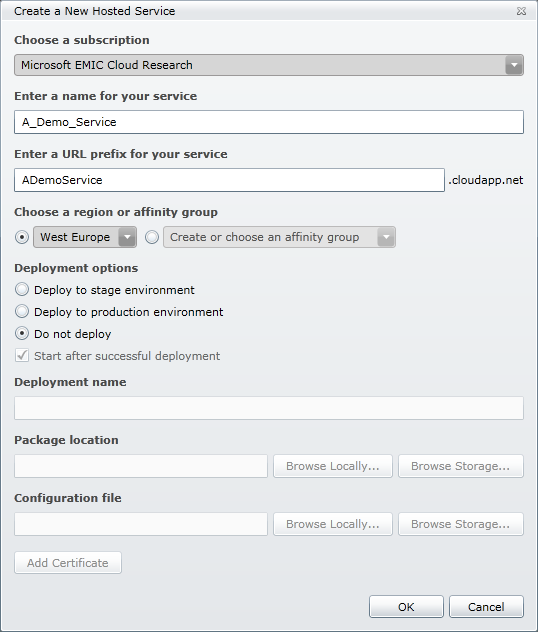
It is now time to start the Azure management portal. Browse to <https://windows.azure.com> and login with the LiveID provided for the MOSA. The management portal looks like this:



* Browse to “Hosted Services, Storage Accounts & CDN” which can be found in the lower left corner.
* Click on “Hosted Services” in the upper left area.
* The panel in the middle should now show a single with your subscription. If you already created some hosted services or deployments before, they are also shown in a tree under your subscription.
* Select your subscription and click on “New Hosted Service”.
* Enter a name and a URL prefix for your service and choose a region. As there some other steps to do before being able to deploy, select the “Do not deploy” option.

By clicking “Ok” your hosted service will be created.

As soon as your hosted service is available, you can continue with the next steps.

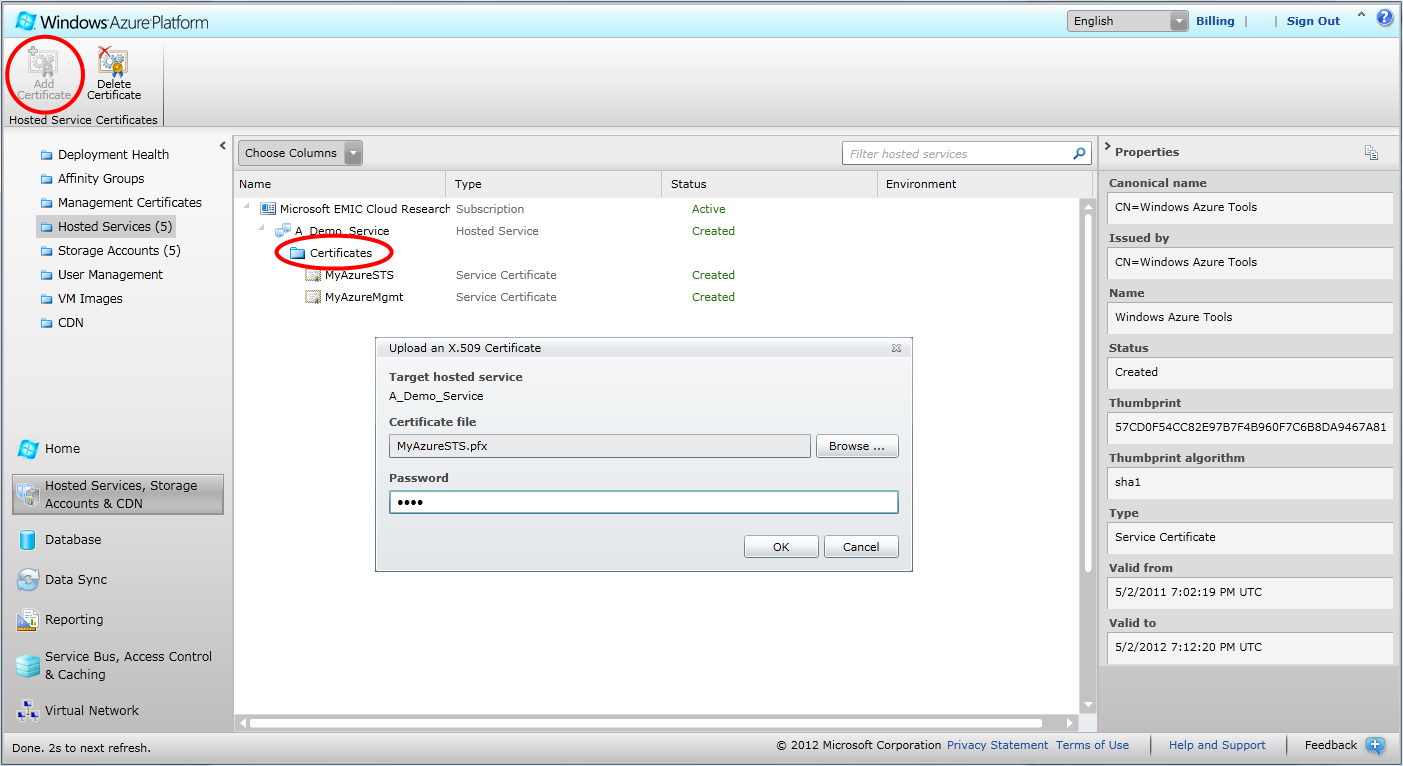


## Upload Certificates

Certificates used by a deployment have to be uploaded to the hosted service first. All certificates belonging to the hosted service will automatically be installed in the certificate store of the VMs.

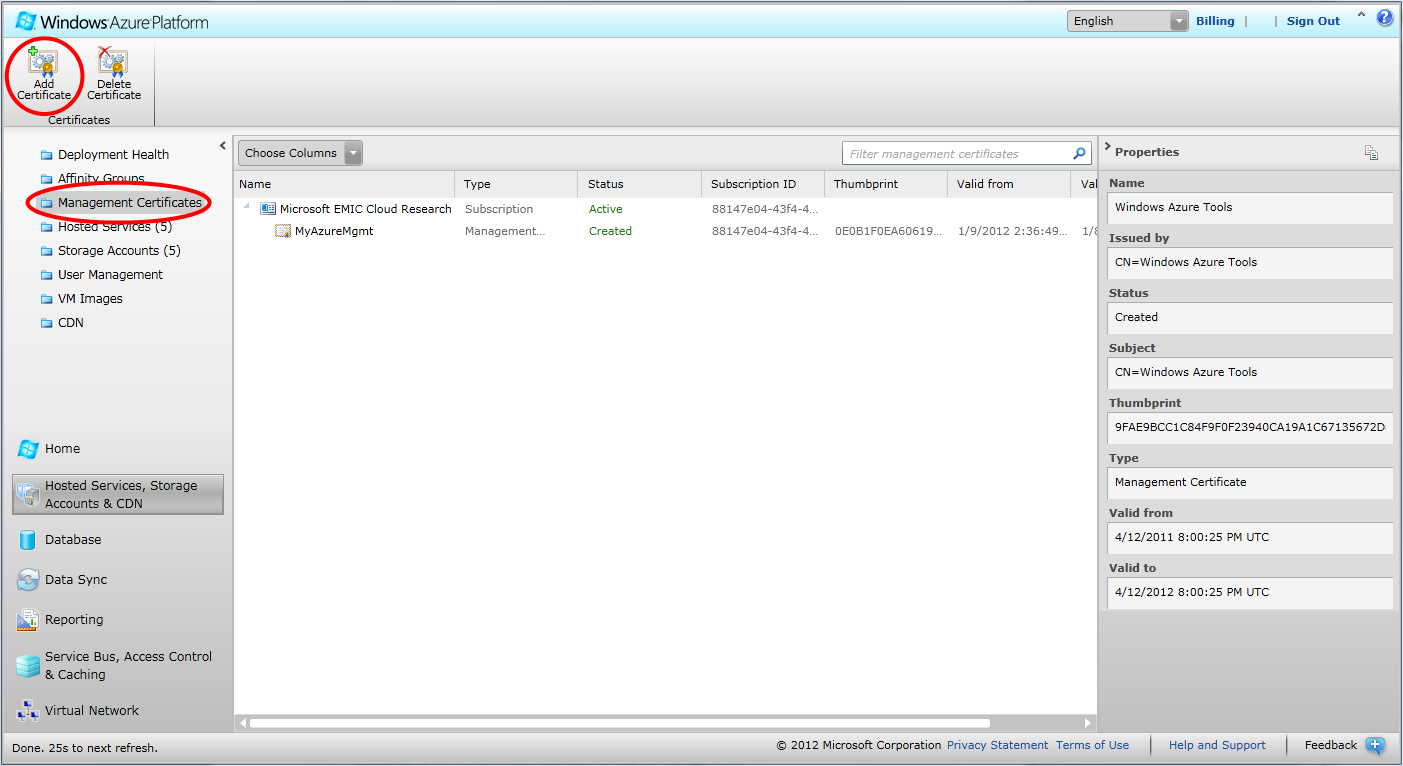
The following steps need to be done for the files containing public and private key (\*.pfx):

* Select the folder “Certificates” under your hosted service and click on “Add Certificate”
* Select the certificate you want to use for SSL / STS, enter the password and click “Ok”.
* Do the same for the certificate you want to use for management.



In addition the public of the management certificate (\*.cer) needs to be uploaded so that Azure can determine if someone doing a management call is authorized for this action.

* Click on “Management Certificates” in the upper left area.
* Click on “Add Certificate”
* Select the public key of the management certificate you uploaded to the hosted service before.



As the private key of the management certificate is available to the deployment (= GenericWorker), the deployment will be able to make management calls like increasing or decreasing the number of VMs.

# Deploy GenericWorker

## Download Files

There is a precompiled and packaged version of the GenericWorker available. It can be downloaded from <http://resources.venus-c.eu/gw/cloud>

Both files, “Cloud.cspkg” and “ServiceConfiguration.cscfg”, are needed for creating a deployment.

## Configuration

The file “ServiceConfiguration.cscfg” contains all necessary settings for the GenericWorker. It is necessary that these settings are completed / adopted by you. As the ServiceConfiguration.cscfg is a XML file a simple editor is sufficient for editing the settings.

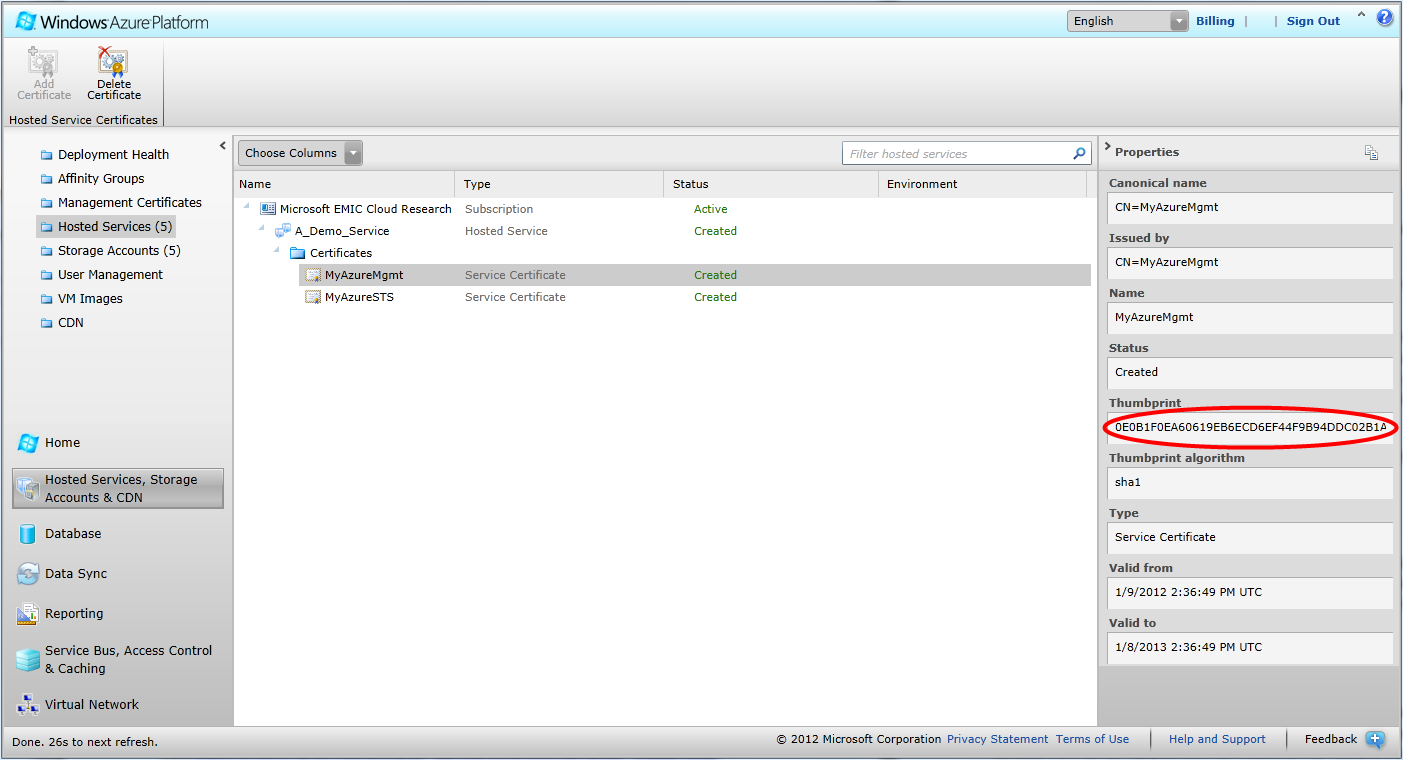
#### Certificate thumbprints

The thumbprints of the certificates uploaded to the hosted service are required in the ServiceConfiguration. There are two sections where the thumbprints have to be completed.

##### **Certificates area**

The certificates belonging to the thumbprints in this section are installed in the certificate store of every VM.

The thumbprint of a certificate can be found in the right panel when clicking on a certificate of a hosted service.



Following the examples for creating the certificates in this tutorial, the thumbprint of “MyAzureSTS” has to be used for the settings “Microsoft.EMIC.Cloud.SSLCert” and “Microsoft.EMIC.Cloud.STSCert”. The example value for these settings like "<YOUR CERTIFICATE THUMBPRINT UPPERCASE>E04FB18B3317F79D5D70B1B6FF9A4C1D45630B01" has to be replaced completely by the thumbprint only.

The thumbprint of the management certificate is used for the setting “Microsoft.EMIC.Cloud.MgmtCert”.

There is one additional setting called “Microsoft.WindowsAzure.Plugins.RemoteAccess.PasswordEncryption”. This setting is required for enabling RDP access to VMs. As this is only for advanced users and normally not needed, the thumbprint of any certificate of the hosted service can be used. If you leave this setting as it is, an error will be raised by Azure during the deployment process.

##### **Settings area**

While the certificates area just specifies which certificates are installed on a VM, the settings area defines which certificate is used by the different services of the GenericWorker. This is necessary because there can be many different certificates on the VMs and the GenericWorker exactly needs to know which of them should be used for what purpose.

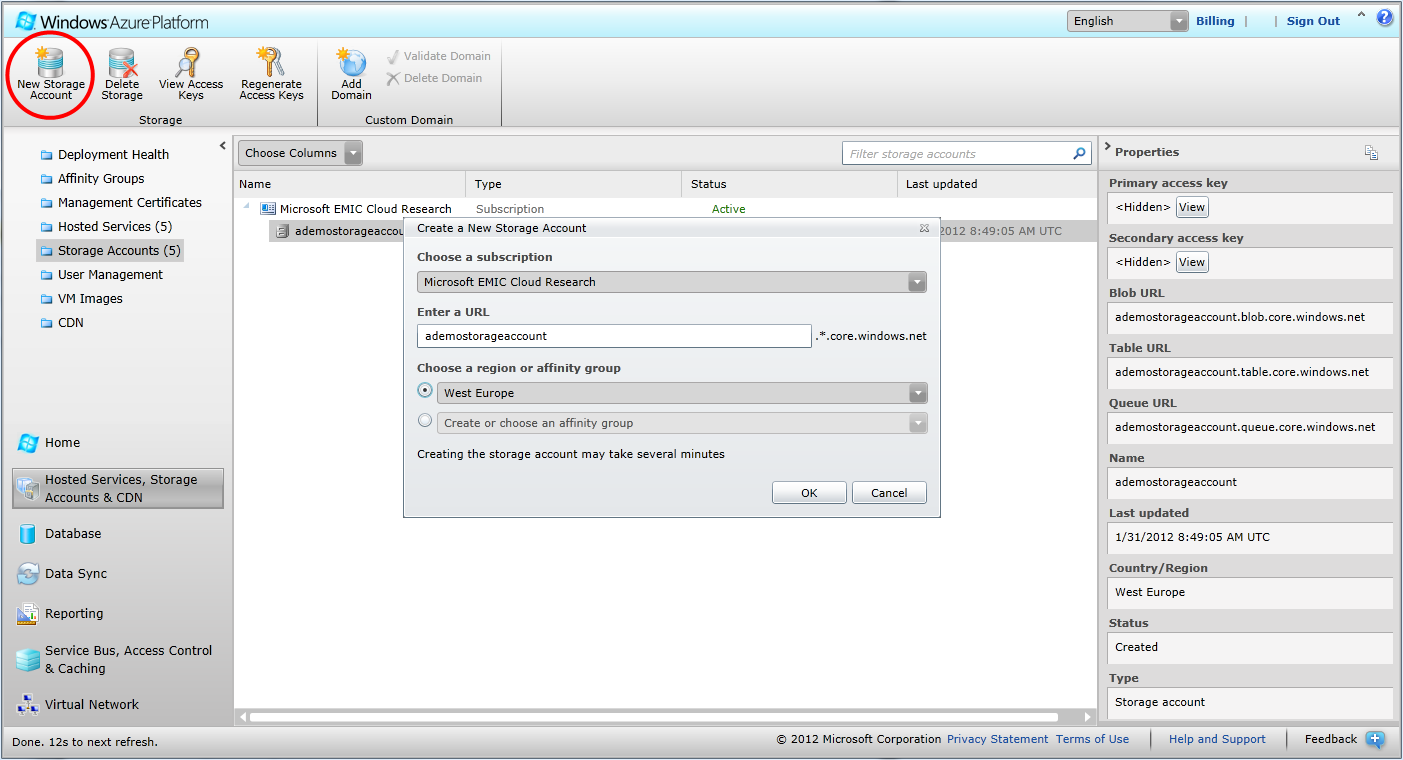
There are two settings for telling the GenericWorker which certificates should be used. These are “Microsoft.EMIC.Cloud.STS.Certificate.Thumbprint” and “Microsoft.EMIC.Cloud.Azure.MgmtCertThumbprint”.

The values of these settings have to be changed in the same way as for the certificates area and with the same thumbprints of “MyAzureSTS” and “MyAzureMgmt”.

#### Connection string

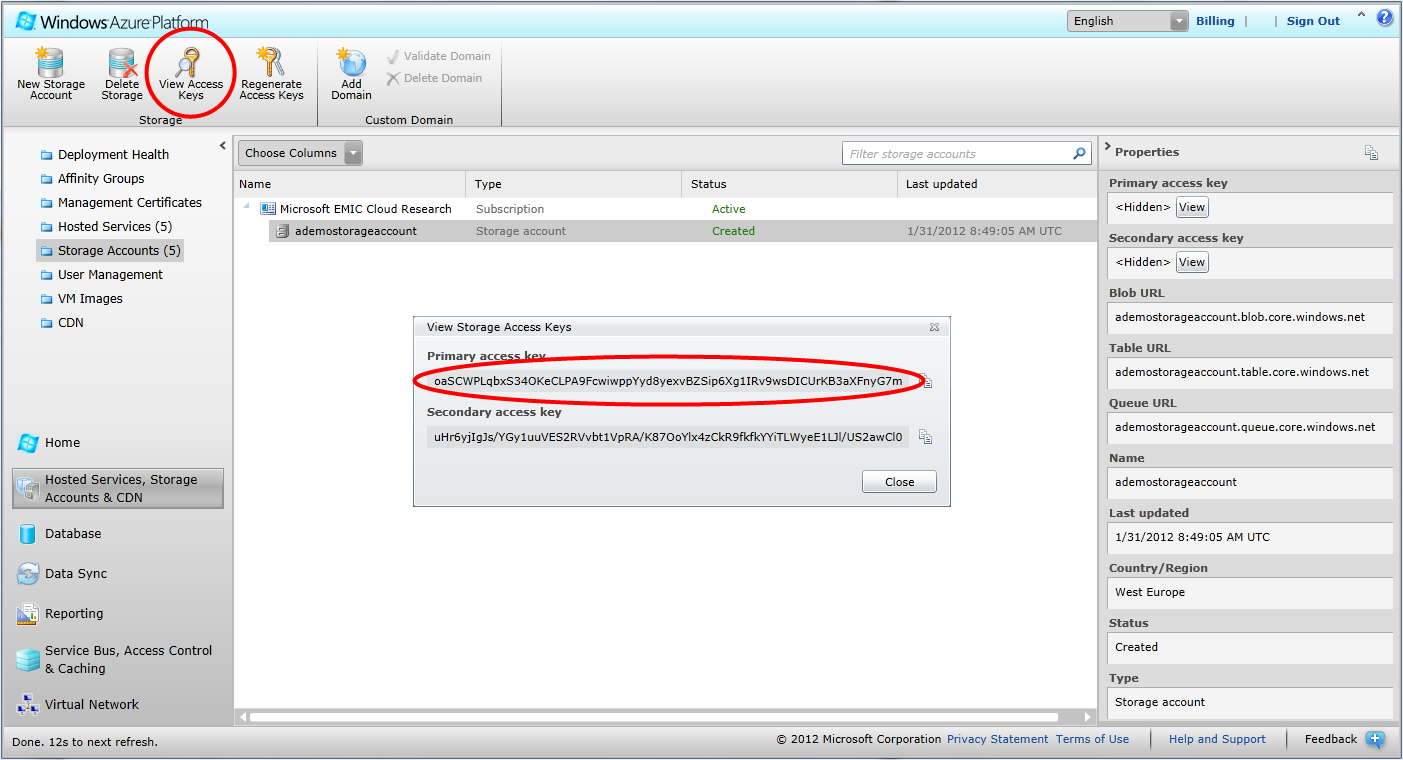
All settings containing the keyword “ConnectionString” have to be modified to represent a valid connectionstring for Azure Storage Services. If you do not have a storage account yet, you can create one by the following steps:

* Click on “Storage Accounts” in the upper left area.
* Click on “New Storage Account”
* Select a region, enter a name for this storage account and click “Ok”.



A connectionstring for Azure has the following format: “DefaultEndpointsProtocol=https;AccountName=<YOUR\_ACCOUNT\_NAME>;AccountKey=<YOUR\_ACCOUNT\_KEY>”

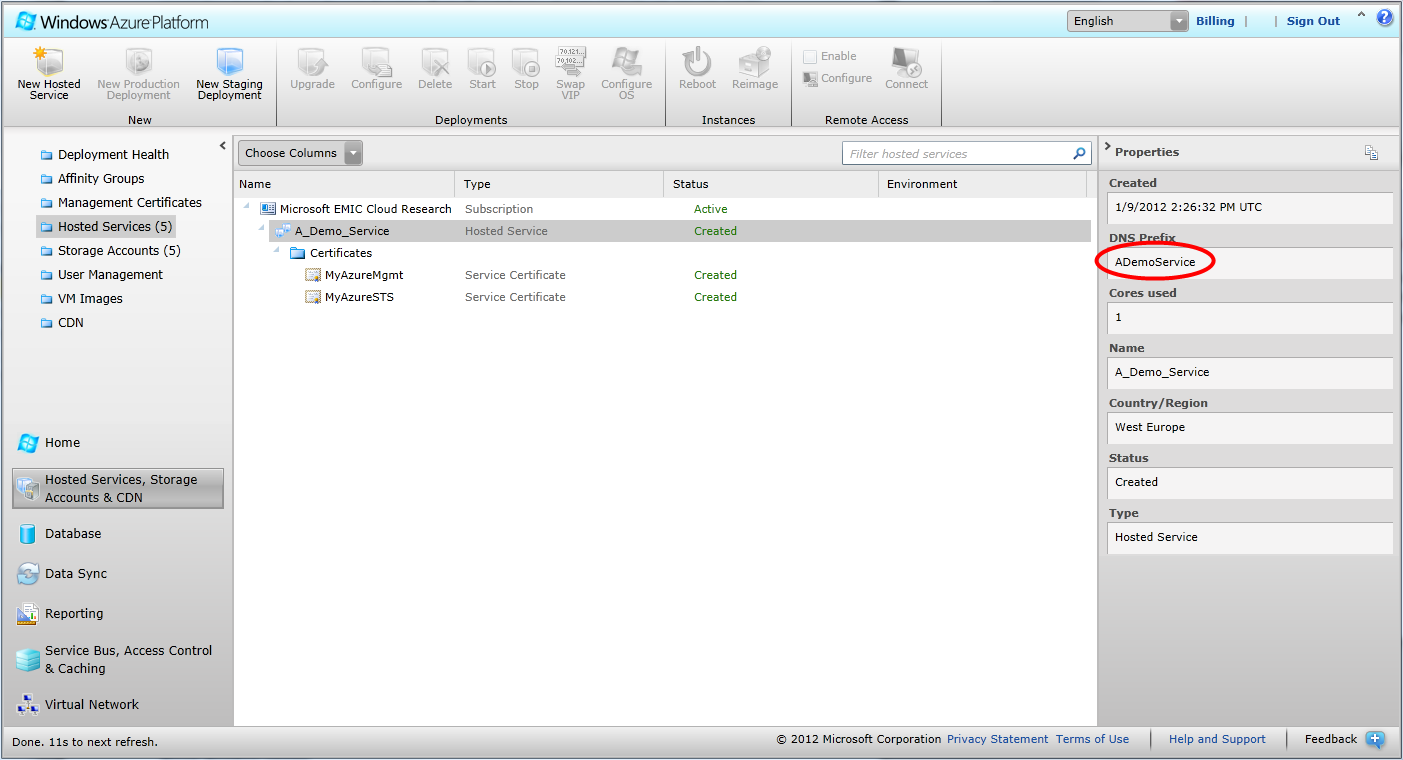
The AccountName value has to be name entered when creating the storage account. The AccountKey can be retrieved by selecting the storage account and clicking on “View access keys”. It does not matter if the primary or secondary key is used as there is no difference between them.



#### URL

There are various settings ending with “URL”. These settings need to contain valid URLs representing the published GenericWorker endpoints. The URLs are basically used by the SecurityTokenService of the GenericWorker so that this component knows, for which endpoints it is allowed to generate SecurityTokens.

The ending of the URLs can be left as it is but the DNS name must be changed. The setting “Microsoft.EMIC.Cloud.STS.URL” for example has the value “http://<YOUR SERVICE HOST>my.genericworker.net/STS/UsernamePassword.svc”. This is changed to the value “http://ADemoService.cloudapp.net/STS/UsernamePassword.svc”. The DNS name of a deployment is always “<DNS Prefix>.cloudapp.net” and the DNS Prefix can be found on the right panel when selecting a hosted service.

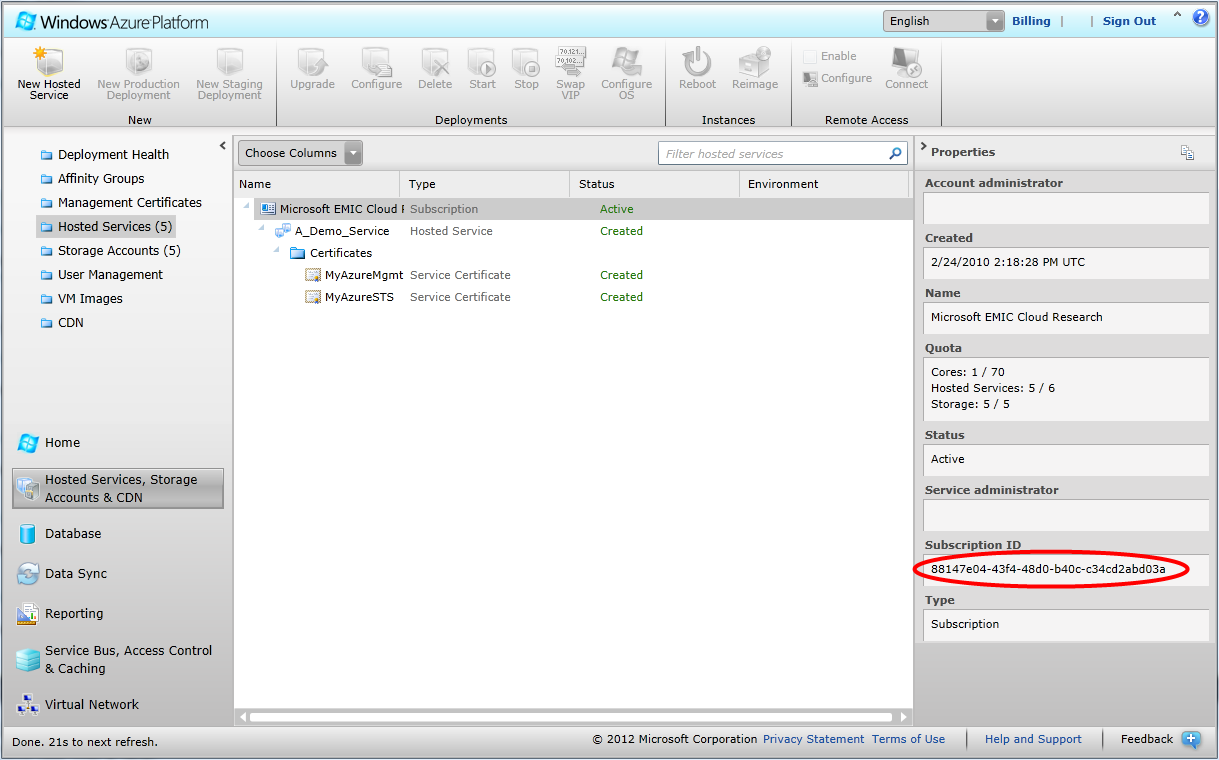


#### Scaling / Subscription

To enable scaling support it is necessary to adopt another two settings in addition to the management thumbprint.

The first setting is “Microsoft.EMIC.Cloud.Azure.ServiceName” and the value must exactly be the DNS Prefix of the hosted service.

The second setting is “Microsoft.EMIC.Cloud.Azure.SubscriptionId”. The Subscription ID can be found on the right panel when selecting a subscription on the Hosted Services page.

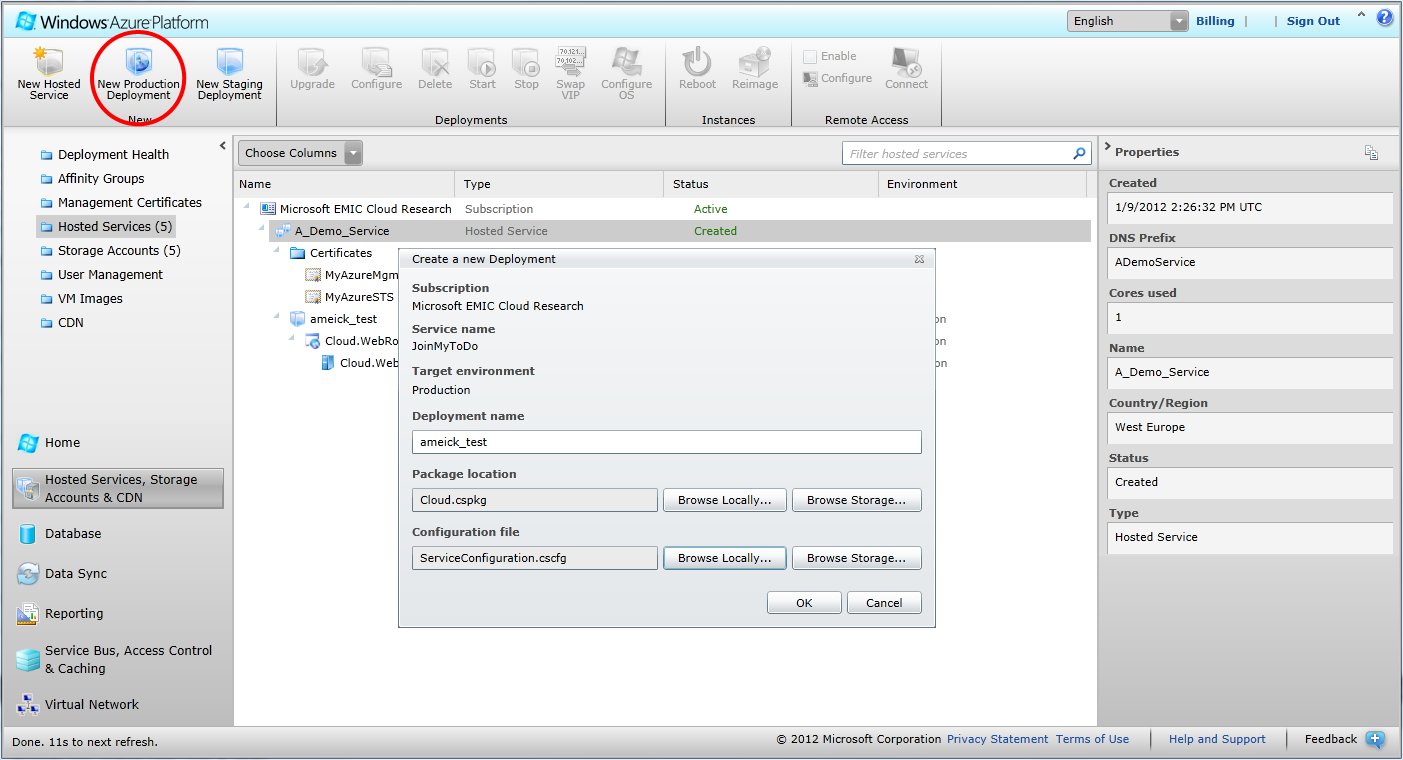


## Create Production Deployment

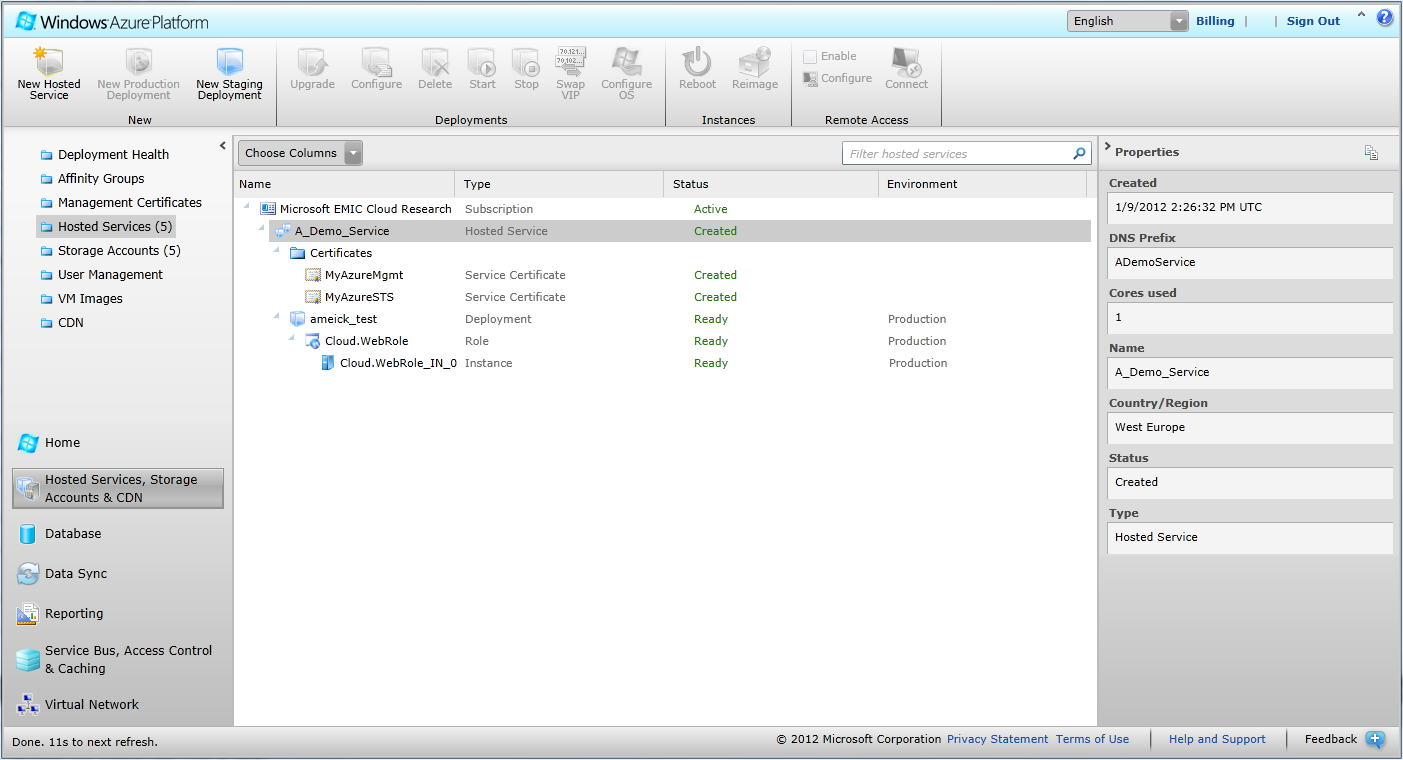
As the configuration is now completed, the deployment of the GenericWorker can be created.

To do this the following steps are required:

* Click on “Hosted Services” in the upper left area.
* Select the hosted service which has been created before in this tutorial.
* Click on “New Production Deployment”.
* Enter a name for your deployment. This name can be any name, it is not used afterwards.
* Select the previously downloaded file “Cloud.cspkg”.
* Select the adopted ServiceConfiguration.
* Click “Ok” to start the creation of the deployment.



It may take some minutes to finish to process of creation, but finally the status of the new deployment should be “ready”. If this is not the case there is most probably a mistake in ServiceConfiguration. A double check of all steps mentioned in this tutorial is then required.



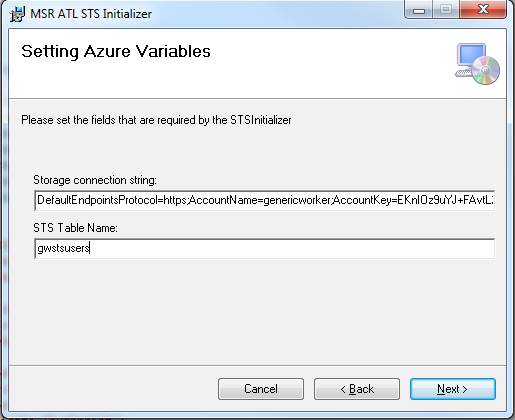
# Post Deploy Steps

## Setup of Users and Passwords

If the sample SecurityTokenService (STS) or the simple username / password security is used, users and passwords must be setup in a special table. This table is called “gwstsusers” and is used by the GenericWorker to lookup users and their corresponding passwords and rights.

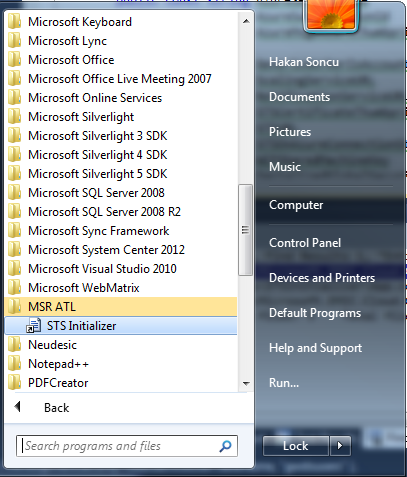
To setup users there is a small tool called “STSInitializer” available. This tool can be found in the Installers folder.

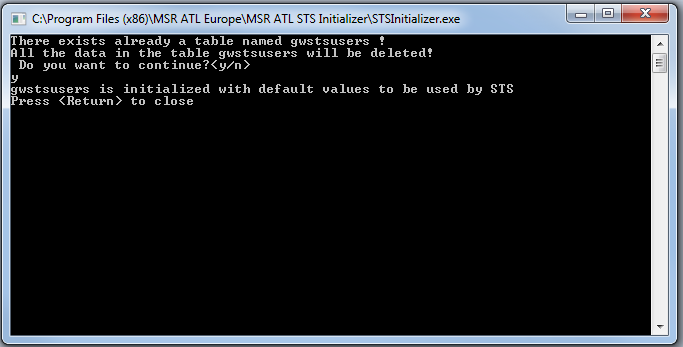
Here are the steps to install and run the STSInitalizer:



Please set the connection string of your storage account as shown in the picture. Do not change the default table name which is gwstsusers.

After installation, you can find the program in the start menu and run it.





# Useful Tools

## Service Configuration Modifier

Instead of changing the service configuration file by using copy/paste which is error-prone, a simple tool named service configuration modifier has been provided together with the GW. This tool can be found in the Installers folder as well.

By using this tool, service configuration file can be modified easily according to your settings. The settings modified by the help of this tool are:

* Instance Count
* Data Connection String
* STS Thumbprint
* Management Thumbprint
* Deployment URL
* Remote Account User Name
* Encrypted Password for Remote Account
* Allow Insecure Access
* Enable Accounting
* Service Name
* Subscription Id