WorkshopPLUS

Microsoft Azure Infrastructure as a Service (IaaS)

Introduction to Microsoft Azure Virtual Machines

Student Lab Manual

V2.1, November 30, 2015

Information in this document is subject to change without notice. The example companies, organizations, products, people, and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarked, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, MS, Windows, Windows NT, MSDN, Active Directory, BizTalk, SQL Server, SharePoint, Outlook, PowerPoint, FrontPage, Visual Basic, Visual C++, Visual J++, Visual InterDev, Visual SourceSafe, Visual C#, Visual J#,  and Visual Studio are either registered trademarks or trademarks of Microsoft Corporation in the U.S.A. and/or other countries.

Other product and company names herein may be the trademarks of their respective owners.

Contents

[Introduction to Microsoft Azure Virtual Machines 4](#_Toc419958144)

[Prerequisites 5](#_Toc419958145)

[Task 1 – Create a Storage Account 5](#_Toc419958146)

[Task 2 – Create a Virtual Network 6](#_Toc419958147)

[Task 3: Creating Virtual Machines for IIS 9](#_Toc419958148)

[Creating IIS Virtual Machines 9](#_Toc419958149)

[Task 4 – Create an IaaS SQL Server Virtual Machine 21](#_Toc419958150)

[Task 5: Configuring the IIS Virtual Machines 25](#_Toc419958151)

[Task 6: Configuring the SQL Server 2012 Instance 28](#_Toc419958152)

[Configuring SQL Server 2012 Instance 29](#_Toc419958153)

[Installing the AdventureWorks Database 32](#_Toc419958154)

[Task 7: Deploying a Simple MVC4 Application 36](#_Toc419958155)

[Verification 39](#_Toc419958156)

[Appendix A – Troubleshooting Tips 40](#_Toc419958157)

[Appendix B – Implementing DNS Name Resolution for the SQL Server Machine 41](#_Toc419958158)

[Option 1 – Put the SQL Server IaaS machine in the same Cloud Service as the IIS machines 41](#_Toc419958159)

[Option 2 – Create an IaaS AD/DNS Server and put all of the machines in the same Windows domain 41](#_Toc419958160)

[Option 3 – Leave the SQL Server IaaS machine in a separate Cloud Service and use the FQDN of the SQL Server IaaS machine 42](#_Toc419958161)

[Option 4 – Create an IaaS server in Azure (same virtual network) and set it up as the DNS Server 42](#_Toc419958162)

[Option 5 – Create a Site-To-Site Network and use an external DNS Server 43](#_Toc419958163)

# Introduction to Microsoft Azure Virtual Machines

In this lab, you will create 3 virtual machines. Two of the virtual machines will be IIS servers that will be load balanced and the other machine with have SQL Server 2012 installed to serve as the application database. These machines will be connected via a Microsoft Azure Virtual Network.

You'll learn:

* How to create an Azure Storage Account
* How to directly connect load balanced IaaS IIS web servers to a SQL Server running in a virtual machine through a simple virtual network
* Configure a SQL Server virtual machine
* Deploy the sample Web application to the IaaS IIS virtual machines

## Prerequisites

The following is required to complete this hands-on lab:

* [Microsoft Azure PowerShell](http://www.microsoft.com/windowsazure/sdk/)
* Install the SQL Server PowerShell extensions (link)
* A Microsoft Azure subscription

## Task 1 – Create a Storage Account

We will need a storage account in order to have a place to put our Virtual Machine vhd blob files. Although while creating a virtual machine, you have the option to create a storage account, you would not be able to name the storage account. Creating the storage account up front gives us more flexible options.

If you already have a storage account, feel free to skip Task 1 and move ahead to Task 2.

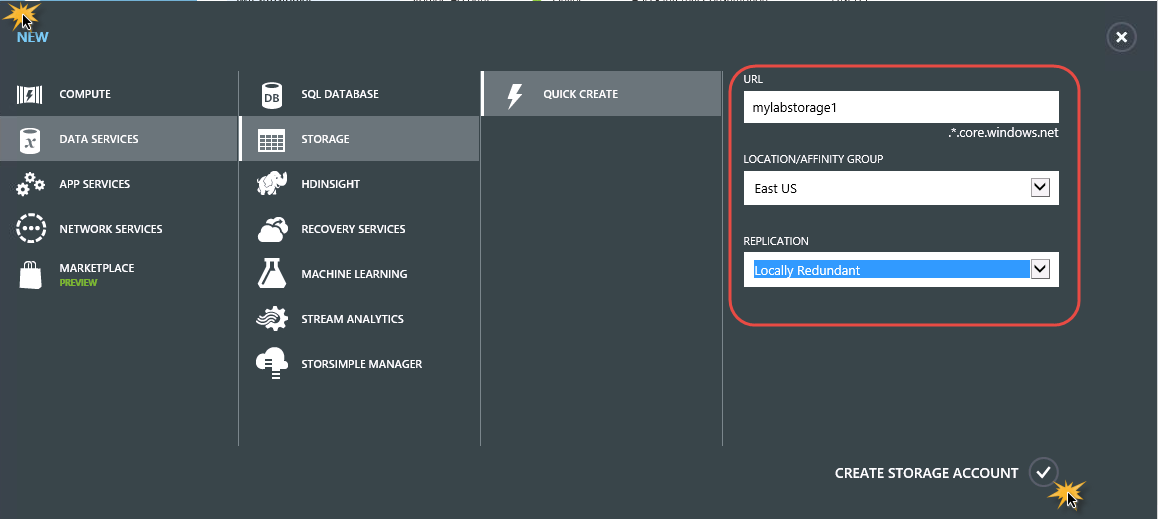
1. From within the Azure Portal, click on the **+New** menu item.



New button

1. Select **Data Services | Storage | Quick Create**. Give the storage account a unique name and then choose the data center nearest to your location. Make sure to choose **Locally Redundant** replication (we do not need geo-replication for this lab exercise). Select the **Create Storage Account** check button.

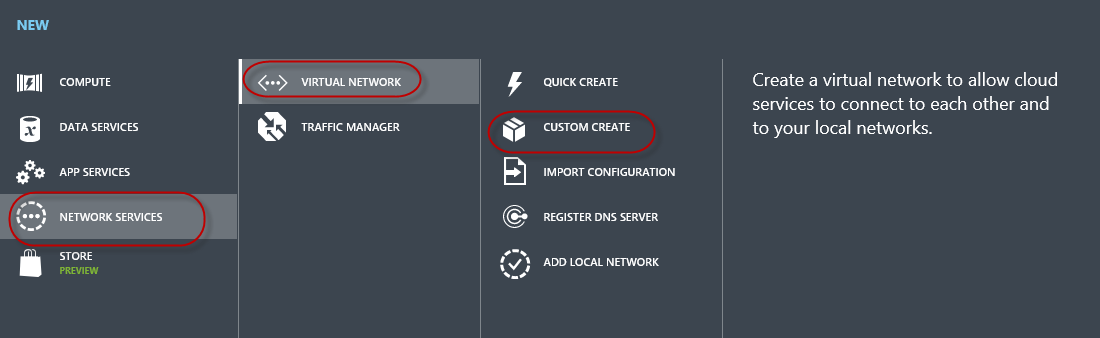
Wait until the creation of the storage account has completed (by viewing the progress in the Azure Portal Storage dashboard).



Create a Storage Account

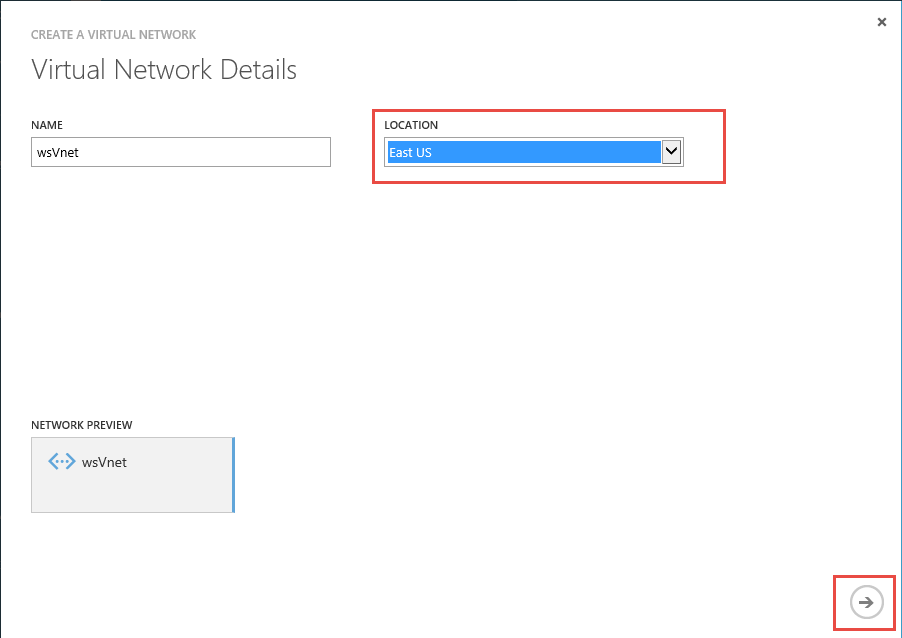
## Task 2 – Create a Virtual Network

1. Within the portal window, click ****New** button** (lower left corner of the screen), select ****Network Services** | **Virtual Network**** and then click ****Custom Create****.



Virtual Network custom create

1. Set a **Name** for the virtual network (for example wsVnet) and the desired data center/region to place the virtual network in. Click the **Next** arrow button to continue.

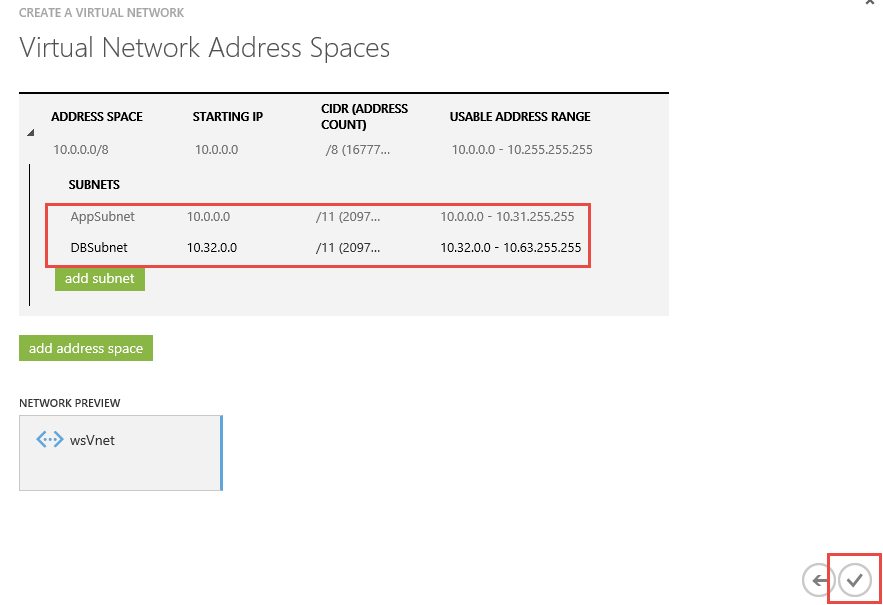


Creating a new virtual network

1. On the *DNS and VPN Connectivity* page, leave the default values and click the **Next** arrow button.
2. On the *Virtual Network Address Spaces* page, leave the default settings.  We are not connecting this virtual network with any other external network, but it would be good to name our subnets.
   1. Rename Subnet-1 to AppSubnet – this is where we will place our IIS machines.
   2. Click the ‘add subnet’ button.
   3. Rename Subnet-2 to DBSubnet.
   4. Click the **Finish** check button.

It is important to note here that if you had clicked on the ‘add subnet’ button before renaming the first subnet, the first subnet edit functionality would be disabled. You can’t go back and change a subnet.

Wait until Azure has completed creating the network. You can view the progress of the network creation in the dashboard window.



Virtual network address settings

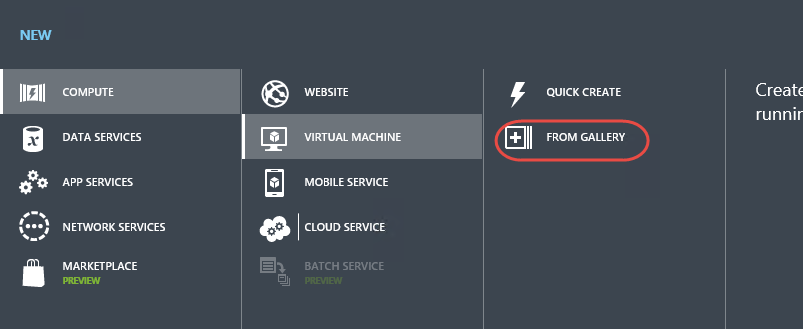
## Task 3: Creating Virtual Machines for IIS

In this task, you will learn how to create a Virtual Machine in Microsoft Azure. Then, you will configure the machine for Internet Information Server, adding roles to use later on in this lab.

### Creating IIS Virtual Machines

In this task, you will provision a Virtual Machine and configure the Load Balancing to host an MVC4 application.

1. In the menu located at the bottom left of the portal window, select **New | Compute | Virtual Machine | From Gallery** to start creating a new virtual machine.

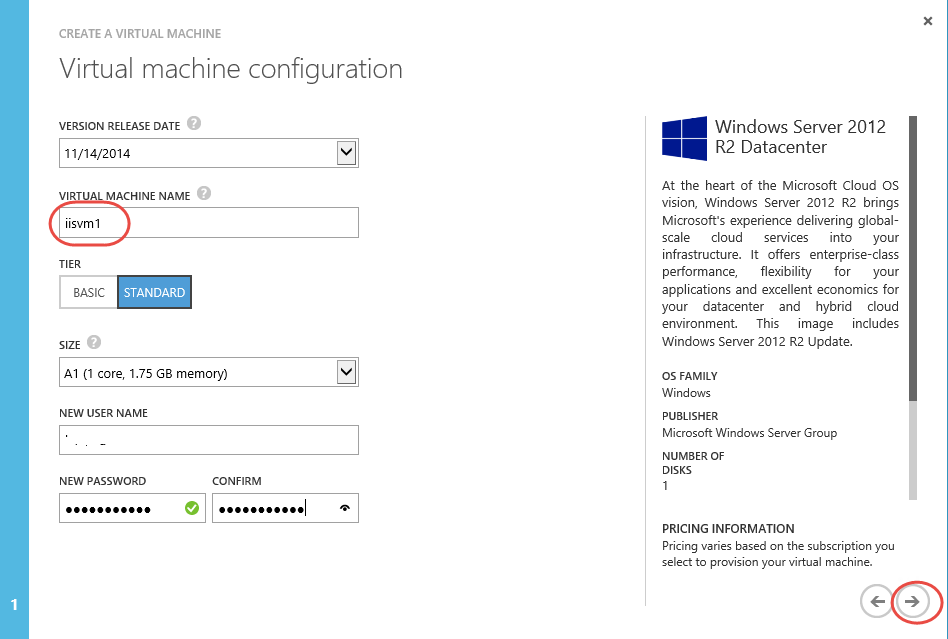


Creating a new Virtual Machine

1. In the *Choose an image* page, click **Microsoft |Windows Server** from the menu on the left menu and then select the **Windows Server 2012 R2 Datacenter** OS image from the list. Click the **Next** arrow to continue.
2. In the *Virtual Machine Configuration* page, leave the version release date by default (latest). Enter the Virtual Machine Name (i.e. "iisvm1"), provide a user name for the *New User Name* field and a password for the *New Password* and *Confirm Password* fields (this is the RDP username/password). This password needs to contain three of these - lower case characters, uppercase characters, numbers and special characters. Make sure you remember your password! Lastly, set the Virtual Machine *Size* to **A1** and click the **Next** arrow to continue.

NOTE: Currently, Azure VMs come in two tiers. The **Standard** tier has VM sizes A0-A4 (Extra Small to Extra Large), and the “Memory Intensive Instances” tier has sizes A5-A11. The **Basic** tier will be similar machine configurations to the "Standard" A0-A4, but the new "Basic" VMs won’t include load balancing or auto-scaling options.

These new "Basic" VMs might be of interest to developers who are using Azure VMs for Dev/Test workloads and don’t need the load balancing and auto-scaling features. These basic VMs will give developers equivalent compute capabilities at a lower cost if load balancing and auto-scaling are not needed.



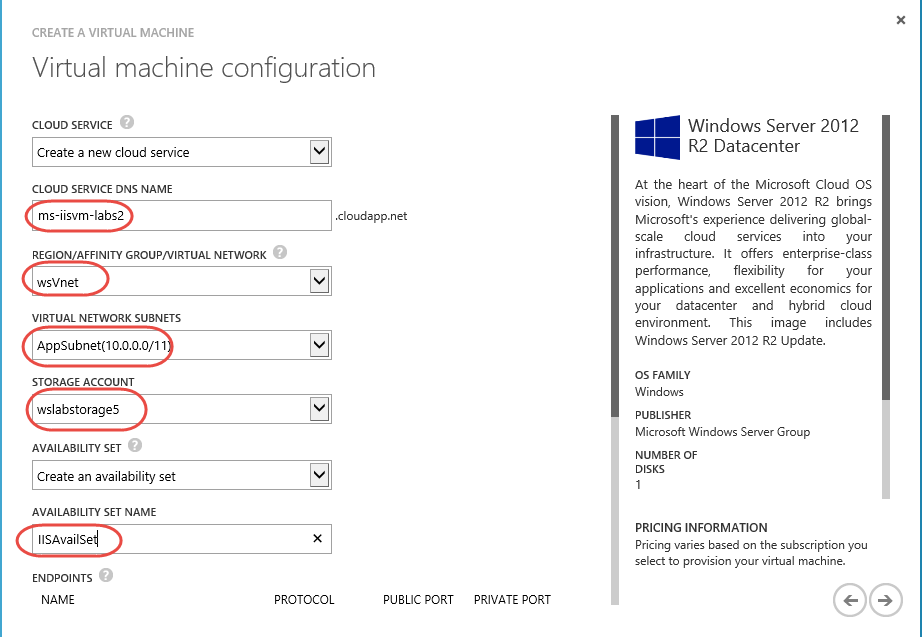
Creating a Virtual Machine - Configuration

**NOTE:** It is suggested to use secure passwords for admin users, as Microsoft Azure virtual machines could be accessible from the Internet knowing just their DNS.

You can also read this document on the Microsoft Security website that will help you select a secure password: <http://www.microsoft.com/security/online-privacy/passwords-create.aspx>

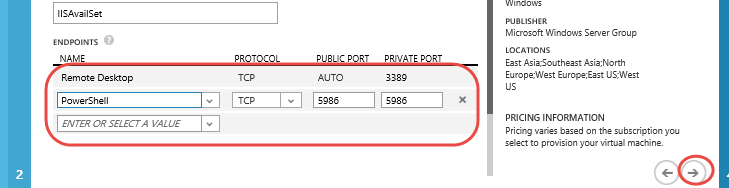
1. In the *Virtual Machine Mode* page, select **Create a New Cloud Service** and do the following.

* Enter a unique DNS Name
* Select a currently existing storage account or leave the default value *Use Automatically Generated Storage Account*.
* For the *Region/Affinity Group/Virtual Network***,** select the **virtual network** you created in the previous task. You can leave the subnet value as it is (it should be AppSubnet).
* In this scenario, we will want to create an Availability Set which assures that if the Azure host environment needs to reboot our IaaS VM, it will not reboot all machines at the same time.
* Select the **Create an availability set**, give the set a name.



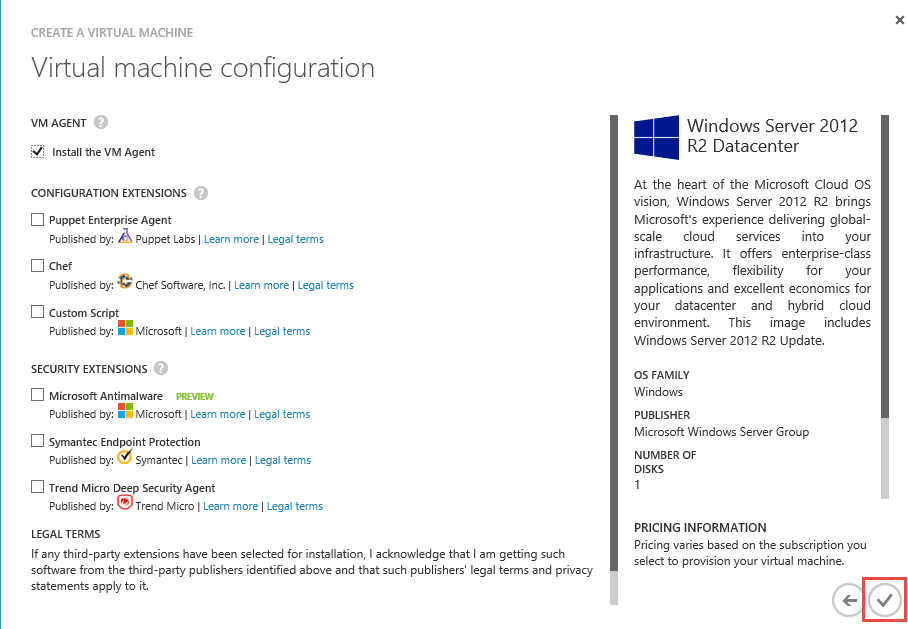
Creating a Virtual Machine - Virtual Machine Mode

1. Scroll the virtual machine configuration wizard window down until you see the port settings. Leave the default remote desktop and remote PowerShell port as they are and select the **Next** arrow.

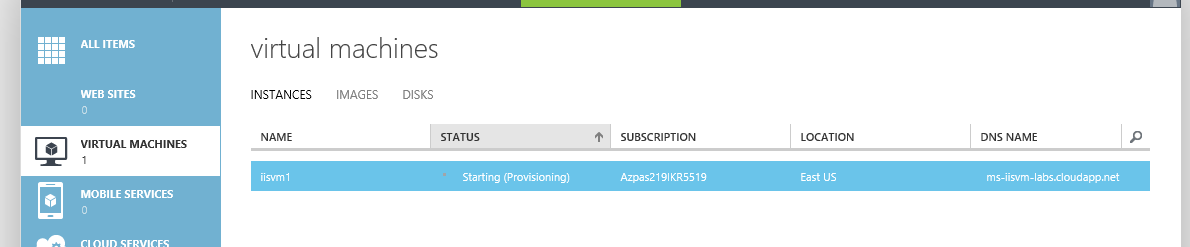


Creating a Virtual Machine - Virtual Machine Mode – default port settings

1. In the *Virtual Machine Configuration*page, leave the default values and click the **Finish** check button to create a new virtual machine.



1. In the Virtual Machines section, you will see the virtual machine you created with a *Starting (provisioning)* status. When Azure has completed the VM creation process, the status will change to ‘***Running’***.

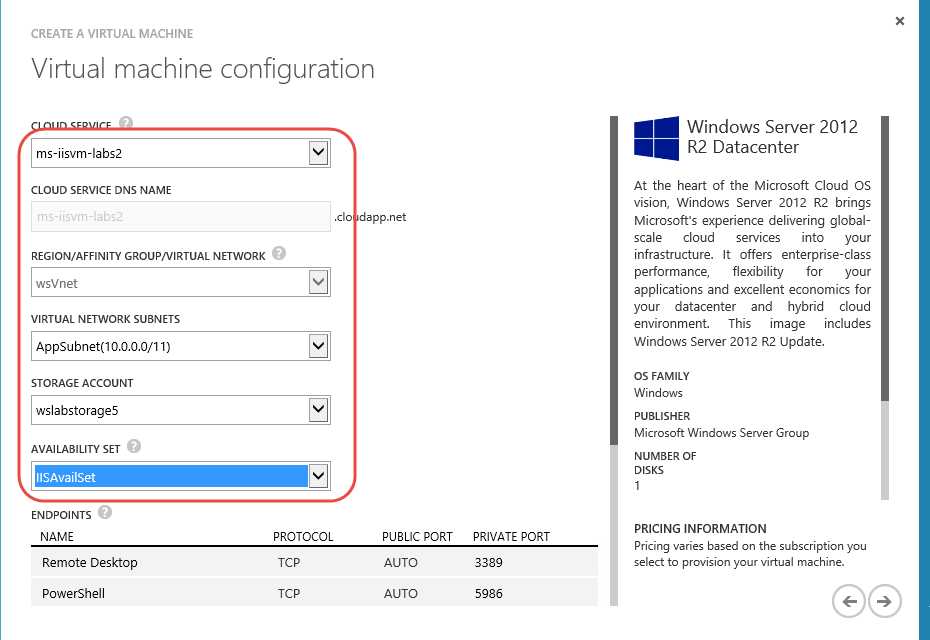


Creating Virtual Machine for IIS Web Farm

**NOTE:** It will take from 8 to 10 minutes for the Virtual Machine to complete the provisioning process. We are going to be creating a second machine and placing it into the same Cloud Service as this first VM. In order to create the second machine, you need to wait 3 – 5 minutes to assure the Cloud Service has already been created and that the first VM is in the process of being created. This means you don’t have to wait until the first VM has been completely created in order to start an additional machine creation but you do have to give Azure some time to get the process started.

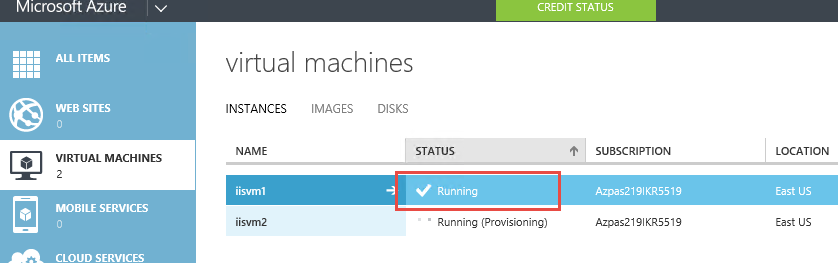
1. After waiting 3 to 5 minutes, add the second Virtual Machine to be used as the second IIS Server. In the portal, select **New | Compute | Virtual Machine | From Gallery**.
2. In the Choose an imagepage, click **Microsoft | Windows Server** from the menu on the left and then select the **Windows Server 2012 R2 Datacenter,** (whatever the latest image date is) OS image from the list. Click the arrow to continue.
3. In the Virtual Machine Configuration page, choose the latest release date. Enter the Virtual Machine Name (i.e. "**iisvm2**"), a User Name, a Password and the Size. Click the right arrow to continue.
4. In the Virtual Machine Configuration page, select the Cloud Service drop down list and select the cloud service DNS name of the previously created VM (this is the key to making this lab work).  This will put the new VM inside the same Cloud Service as the first VM.

NOTE: If you do not see the name of the Cloud Service you previously created, you have not waited long enough for Azure to complete the process of creating the first VM. In this case, completely close the wizard window and wait a few minutes. You will need to start back over at step 10.  
  
Select the storage account you used for the first VM you created or leave the default value *Use Automatically Generated Storage Account* and then select the Availability Set you created for the previous VM creation. Click the **Next** arrow button to continue. NOTE: Generally speaking, you would normally want to put both virtual machines in the same storage account, although it is not required.



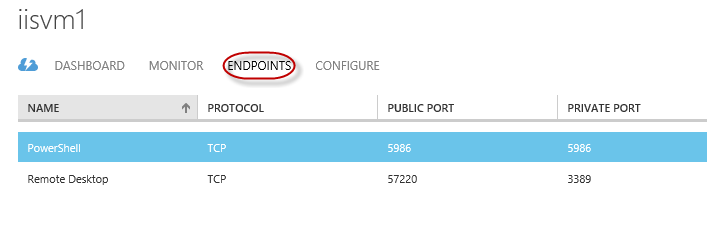
Creating a Virtual Machine - Virtual Machine Mode

1. In the Virtual Machine Configuration page, leave the default values and click the **Next** arrow button to create a new virtual machine.
2. You can check the Virtual Machine status from the Virtual Machines section within the portal.   
     
   Note: It will take from 8 to 10 minutes for the virtual machine to complete the provisioning process but you can move to the next step in the lab now.
3. From the Virtual Machine dashboard, check to make sure the first VM has completed the creation process and has a status of **Running**.



Virtual Machine Status of Running

1. Add a new endpoint to public port 80 in the first VM you created first (iisvm1). To do this, click on the first Virtual Machine Name (iisvm1) to go to the Dashboard page and then click **Endpoints**. Click **Add** button on the bottom pane of the portal window.



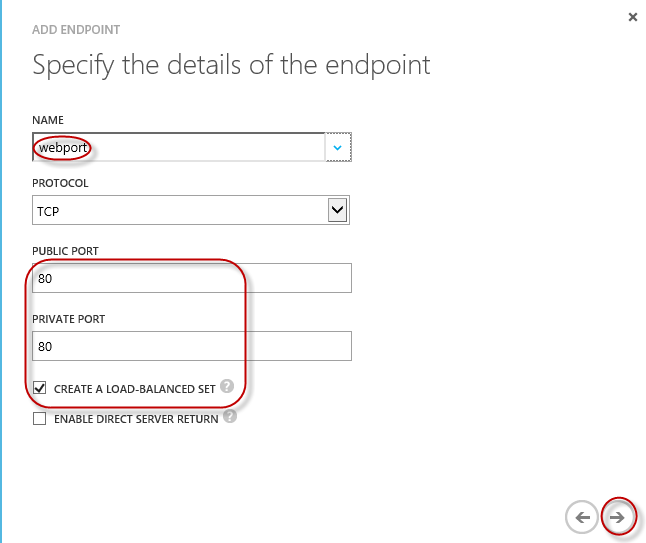
Selecting Add Endpoint in the dashboard

1. Make sure that **Add a Stand-Alone Endpoint** option is selected and then click the right arrow button to continue.



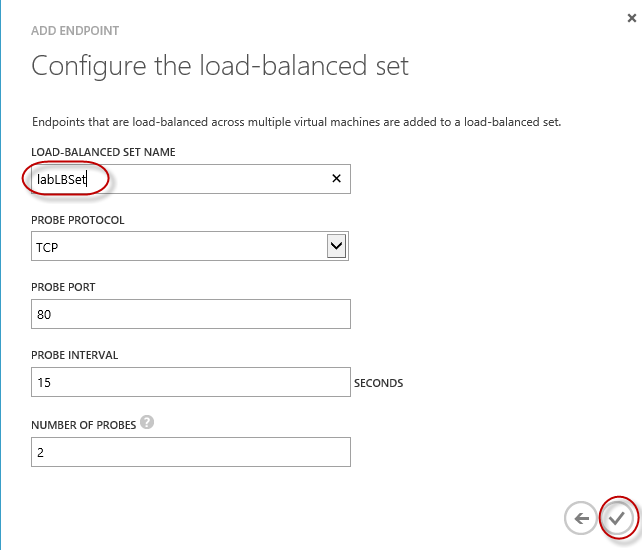
Adding a new Endpoint

1. In the Specify endpoint details page, set the Name to ***webport***, the *Protocol* to **TCP** and the *Public Port* and *Private Port* to **80**. Click the checkbox for **Create A Load-Balanced Set**. Click the **Next** arrow button.



Creating a Load Balanced Set

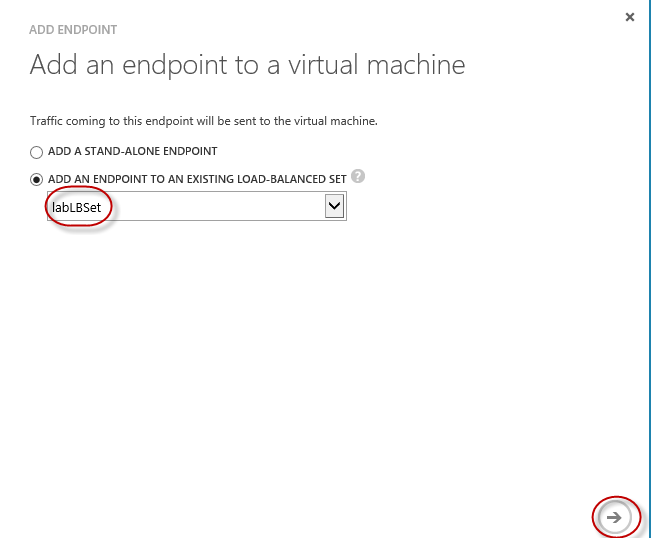
1. On the load balanced set configuration page, enter the name you want the load balanced set to have and leave the other settings as they are.  Click on the **Finish** button.



Creating a Load Balanced Set

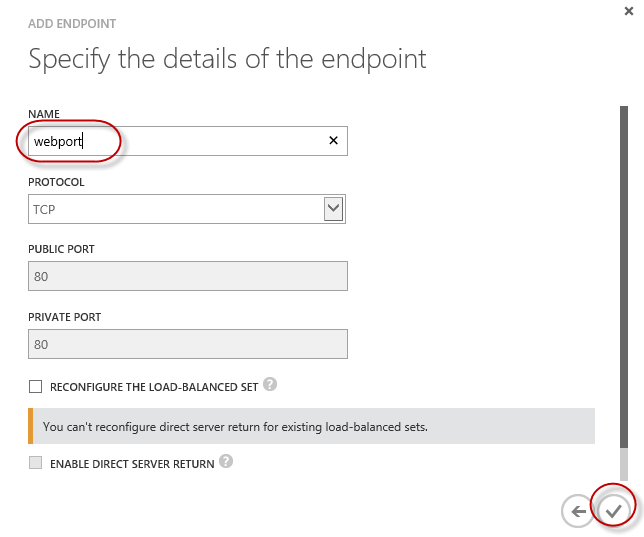
**Note**: It will take some time to create the new endpoint and load balanced set.

1. Now, create a new Endpoint in the second Virtual Machine in order to enable Load Balancing between both virtual machines. To do this, click the **Virtual Machines** main menu item and then select the second virtual machine you created (ex. iisvm2). Then, click the **Endpoints** menu item at the top of the virtual machine dashboard.
2. Click **Add**, and then select **Add Endpoint to an existing Load-Balanced Set** option. Select the load balanced set you created for the first Virtual Machine from the drop down list and then click the right arrow to continue.



Load Balance Traffic On an Existing Load Balanced Set

1. In the *New Endpoint Details* page, set the Name to ***webport***. Click the check button to create the endpoint. The ‘name’ actually is not relevant, it doesn’t not need to be the same name as the endpoint we added for the other VM.



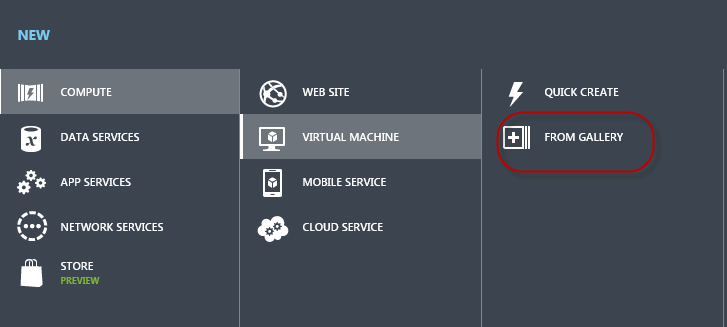
Joining a load balanced set

**NOTE:** It can take a few minutes to create a new endpoint.

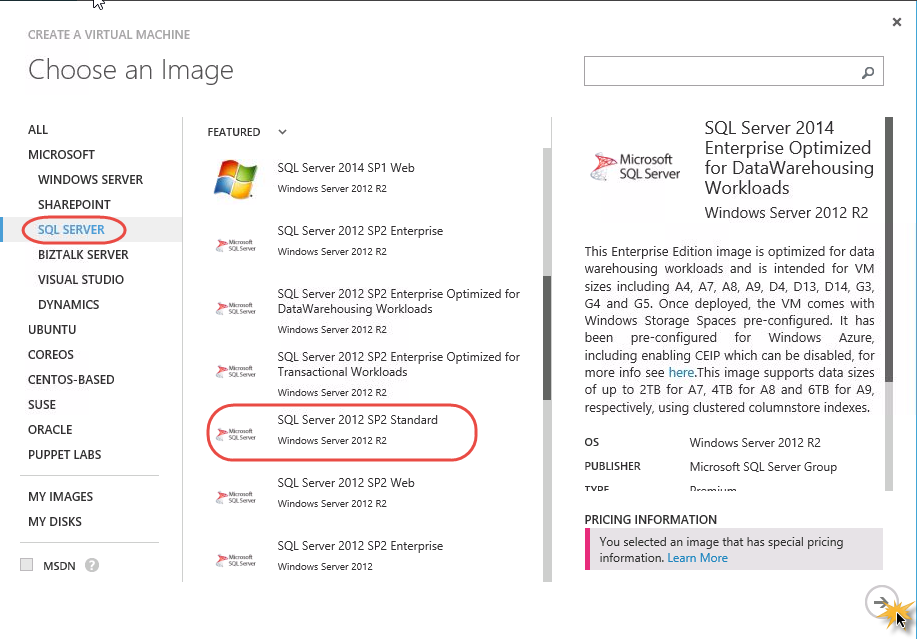
## Task 4 – Create an IaaS SQL Server Virtual Machine

In this step, you will create a new virtual machine using the Microsoft Azure Portal that will serve as your database server.

1. Click the **New** button in the lower left panel of the portal window and select **Compute | Virtual Machine | From Gallery** option.

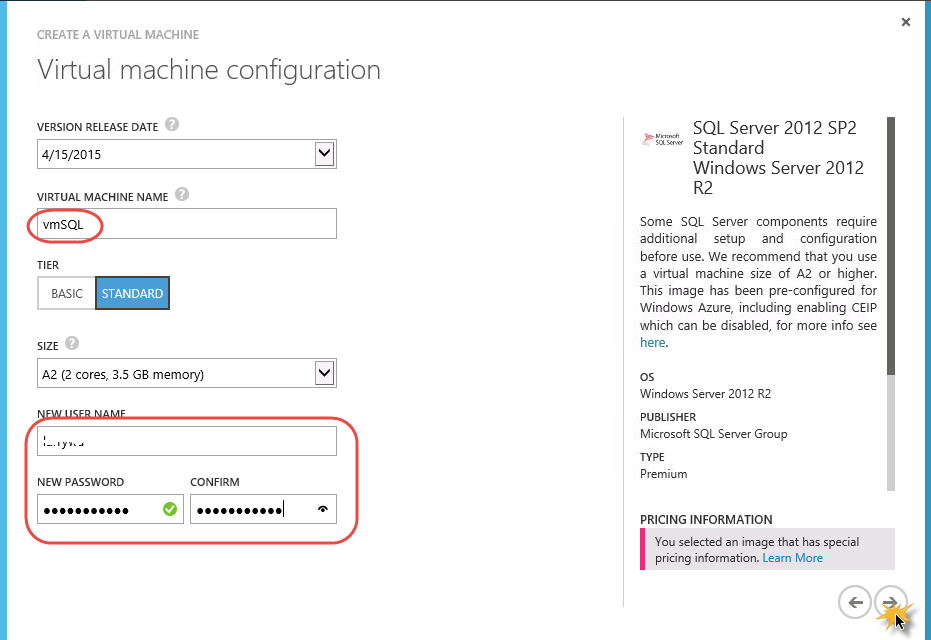
Creating a New Virtual Machine

1. In the Virtual machine operating system selection page, click **Microsoft | SQL Server** on the left menu and select the **SQL Server 2012 SP2 Standard on Windows Server 2012 R2** image from the list. Click the **Next** arrow to continue.



Selecting SQL Server

1. In the *Virtual machine configuration* page, leave the Version Release Date drop down as it is and then enter a Virtual Machine name (ex. *vmSQL)*.  Set the machine size to **Small** and enter a remote desktop username and password.  Click on the **Next** arrow button.



Virtual Machine Configuration

1. In the Virtual Machine Configurationpage, choose the create a **new** Cloud Service and set the Cloud Service DNS Name field to a unique name other than the name of the virtual machine.

Question: Why have we chosen a new Cloud Service name? This is just to prove a point that if you have multiple Cloud Services within a virtual network, the virtual machines in these Cloud Services should be able to reference each other by IP address. If we were to NOT be using a virtual network, all machines would be required to be in the same Cloud Service in order to communicate through IP addresses.

1. Select the Virtual Network that was previously created and then select **DBSubnet** for the subnet to put the database in. You can choose to use an existing storage account or create a new one. Leave the settings for the endpoints as is and leave the Availability Set as is (None) and click the arrow button to continue.   
     
   NOTE: Although we used an availability set with the IIS VMs, since this machine is a totally different configuration, putting it in an availability set does not serve any purpose. In a production situation, you would be expected to have at least two SQL IaaS machines with enabled features such as AlwaysOn.

Question: Why did we choose to put the database server in a separate subnet? Answer: Although we could have put the database server in the same subnet, we would like to have the ability to assure that our database server IP address doesn’t change in case the virtual machine is shut down and then restarted. By having only this one machine alone in its own subnet assures that when the machine is restarted it will be given the same IP address by Microsoft Azure virtual networking. Azure will assign IP addresses to machines in a subnet in the order in which the machines are restarted.

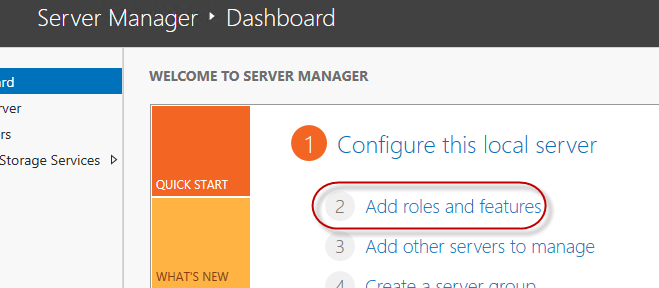
1. On the final wizard page, leave the settings as they are and select the **Check** button. In the Virtual Machines section of the portal, you will see the virtual machine you created displaying a *provisioning* status. Wait until it changes to *Running* in order to continue.

**NOTE:** It will take from 8 to 10 minutes for the virtual machine to complete the provisioning process.

## Task 5: Configuring the IIS Virtual Machines

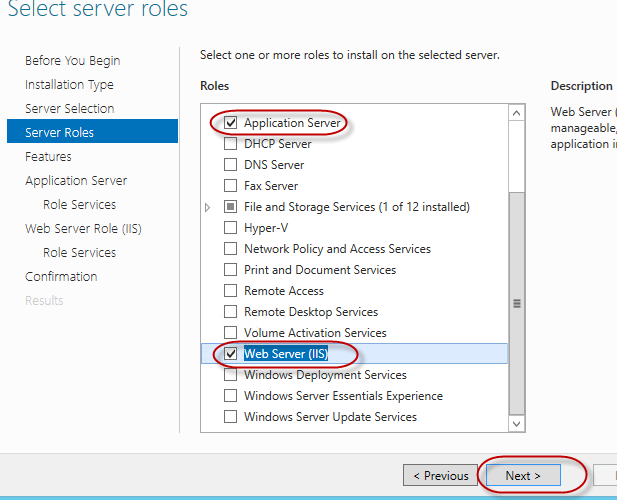
Next, you will configure the IIS Virtual Machines by adding the necessary roles to deploy the MVC application.

1. In the Azure Portal, click **Virtual Machines** on the main menu.
2. You will see a list with your existing Virtual Machines. Select the first VM (ex. iisvm1) you created and click the **Connect** button in the bottom menu bar.
3. You will be asked to download the remote desktop settings file. Click **Open** and log on using the remote desktop credentials you defined when creating the virtual machine. You will also be prompted to accept the certificate. Select Yes.
4. Wait until the Server Manager appears and then select the **Add Roles and Features** link in the dashboard.



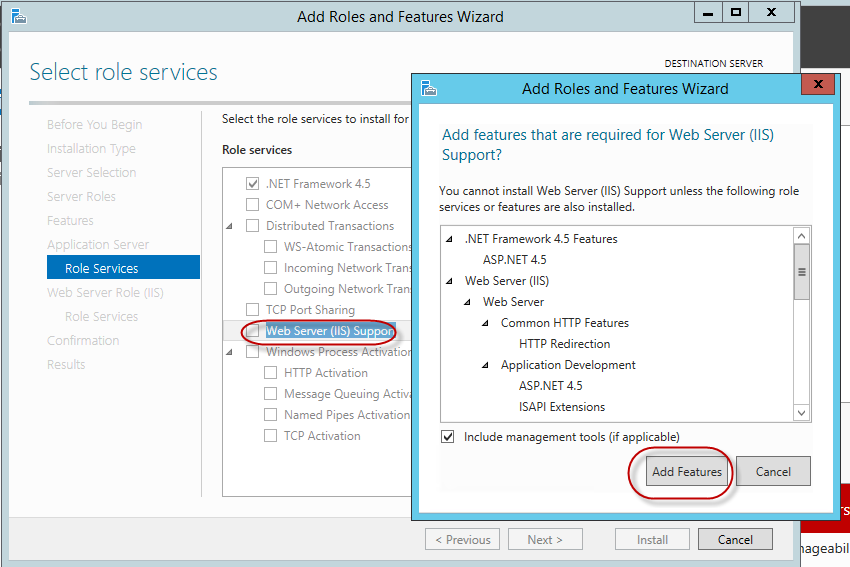
Adding roles and features

1. Keep clicking the Next button until you get to the Select Server Roles window. In this window, you need to select **Application Server** and **Web Server (IIS)**. Click the **Add Features** button and then select the **Next** button.



Adding Server Roles

1. Click the **Next** button.
2. Click the **Next** button.
3. In the Select Role Services page, select **Web Server (IIS) Support**. It will prompt a dialog warning about Required Role Services. Click **Add Features** button to install them and then click **Next.**

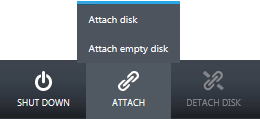


Adding Server Roles

1. Click the **Next** button.
2. Click the **Next** button.
3. Click the **Install** button.
4. Once the installation has completed, you can click the **Close** button.
5. Close the Remote Desktop Connection.
6. Repeat this task on the second Virtual Machine to install IIS. If you used the proposed name earlier in this lab, the second Virtual Machine should be named *iisvm2*.

## Task 6: Configuring the SQL Server 2012 Instance

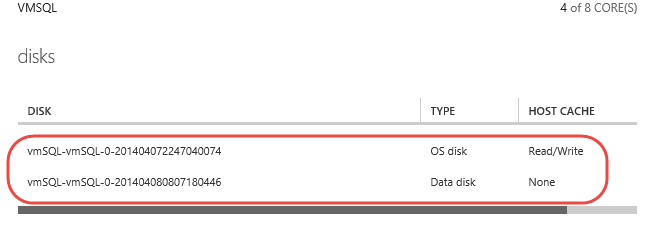
1. Now, you will create and attach an empty data disk to store the SQL Server logs and data files. To do this, in the Virtual Machines section, select the SQL Server virtual machine you previously created.
2. In the virtual machine's Dashboard, click **Attach** in the menu at the bottom of the page and select **Attach Empty Disk**.



Attach Empty Disk

1. In the Attach Empty Disk page, set the Size to **10** GB and leave the other settings as they are. Select the **Check** button to create the disk.
2. Wait until the attach disk process finishes.
3. You will see two disks for the virtual machine listed in the dashboard for your SQL VM: one for the operating system and other one for data and logs.

**NOTE:** It might take a few minutes until the data disk appears in the virtual machine's dashboard within the Microsoft Azure Portal.



OS and Data disk

### Configuring SQL Server 2012 Instance

In this step, you will set up SQL Server instance and database to be used by the web application.

1. In the Microsoft Azure Management Portal, click the Virtual Machines menu item from the main menu.
2. Select your SQL Server virtual machine from the virtual machines list and click **Connect** to connect using Remote Desktop Connection.

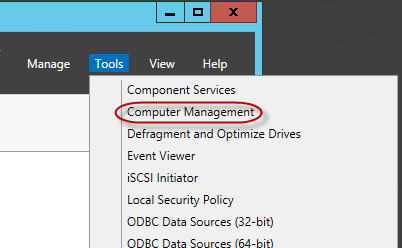
**NOTE:** use the credentials that you inserted when creating the virtual machine in the previous task.

1. In the virtual machine, open Server Manager (if it is not already opened) by clicking on the icon in the lower left hand corner of the window.   
     
   NOTE: Server Manager should open by itself the first time you log into the machine.



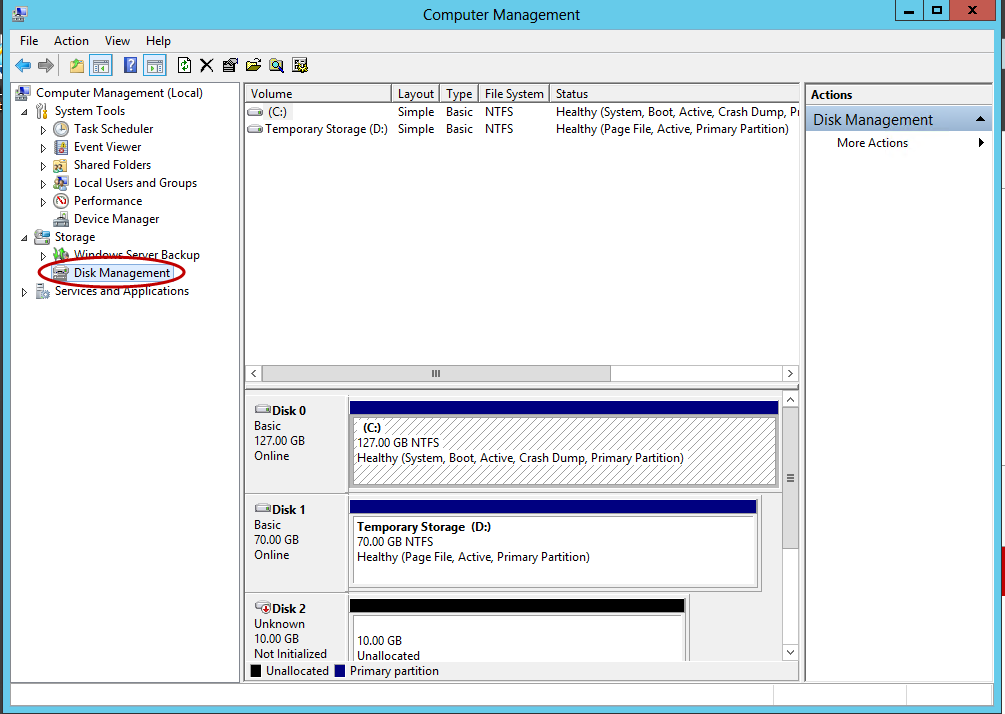
Server Manager Icon

1. Once the server manager has opened, click on the **Tools** menu item and then **Computer Management**.



Selecting Computer Management

1. Expand **Storage** node and select **Disk Management** option.

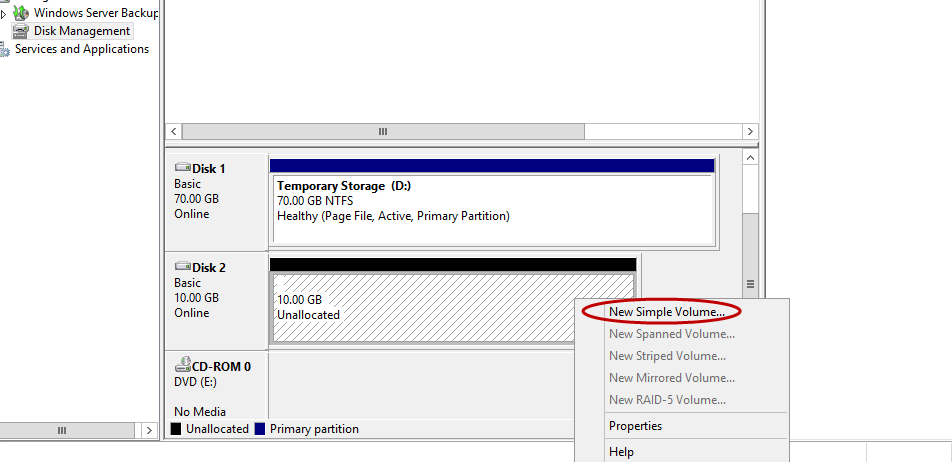


Disk Management

1. After selecting Disk Management, an Initialize Disk dialog will be displayed. Leave the default values and click **OK**.

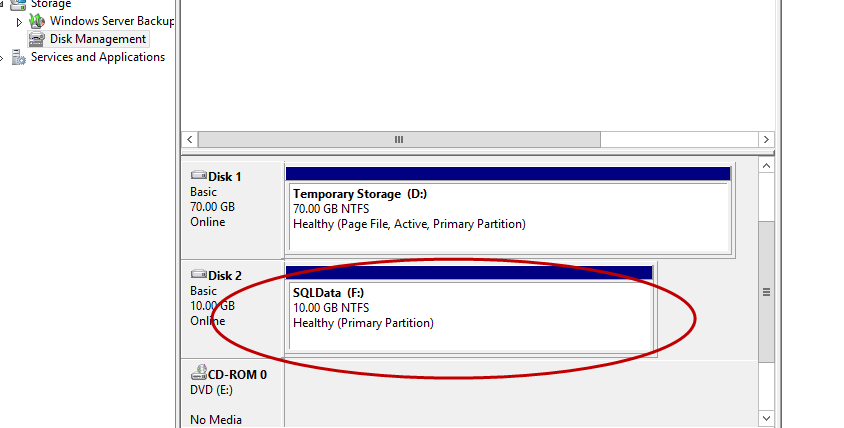
**NOTE**: If the Initialize Disk dialog is not displayed when selecting Disk Management, locate the disk you created using the **Attach Empty Disk** feature from the Microsoft Azure Management Portal, right-click the first disk and select **Initialize Disk**. Leave the default values and click **OK**.

1. Right-click the disk unallocated space and select **New Simple Volume**.



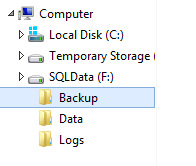
Disk Management

1. Follow the New Simple Volume Wizard. When asked for the Volume Label use ***SQLData***.   
     
   NOTE: Make sure you leave the drive letter as F:
2. Wait until the process for the disk is completed. Note that you will be prompted with a dialog box that asks you if you want to format the disk (select OK or Yes) and then you may be prompted by another dialog box saying it is not possible to format the disk. Ignore the second dialog box.
3. The Disk Management list of available disks should now show the **SQLData** disk like in the following figure (close the disk management windows after this step).



Disk Management

1. Using Windows Explorer, go to your new **F:\** drive and create the directories **F:\Data**, **F:\Logs** and **F:\Backups**.



### Installing the AdventureWorks Database

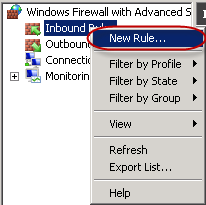
In this step, you will add the AdventureWorks database that will be used by the sample application.

1. In the hands-on lab material (on the hosted lab HOST machine), browse to the **C:\AzureIaaSWS\M3-VirtualMachines\Labs\IntroToWindowsAzureVirtualMachines\Source\Assets** directory. If you are not using a Microsoft hosted lab, you will need to browse to wherever you extracted the lab material to on your hard drive.
2. Copy the entire **Database** directory over to the SQL Server IaaS virtual machine and drop it on the C:\ drive. You can do this by selecting the Database directory on the source drive, select CTRL+C and then doing a CTRL+V onto the C:\ Drive of the database VM.
3. On the SQL IaaS machine, browse to **C:\Database**.
4. Right click the **AdventureWorks2012\_Database.zip** file and extract the contents to **F:\Data**.
5. Browse to the **C:\Database\Scripts** directory.
6. Right click on the **SQLPSX.msi** file and select **Install**. Step through the install process, accepting the defaults.
7. Right click on the **InstallDB.cmd** file and select **Run as Administrator**. The DBSetup.ps1 PowerShell script will be executed and will:
   1. Attach the AdventureWorks2012 database.
   2. Set the database server instances to mixed mode.
   3. Add a user login, CloudShop and sets user mappings.
8. Open the **SQL Server Management Studio** by going back to the Start menu (tile page) and then typing SQL Server Management Studio. Click on the icon that will appear on the left hand side.



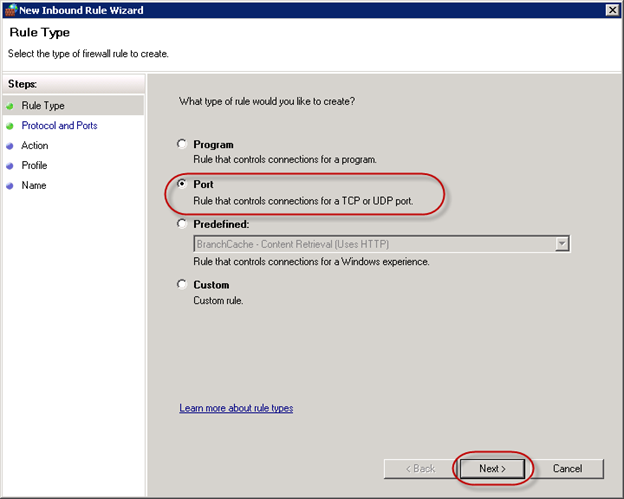
Opening SQL Server Management Studio

1. Connect to the SQL Server 2012 default instance using your Windows Account.
2. Expand the Databases node to make sure the **AdventureWorks2012 database** has been attached.
3. Close SQL Server Management Studio.
4. In order to allow the MVC4 application access the SQL Server database you will need to add an **Inbound Rule** for the SQL Server requests in the **Windows Firewall**. To do this, go to the Start page (tile Window) and open the Control Panel.
5. Choose Windows Firewall from the Control Panel icons and then choose **Advanced Settings**.
6. Select **Inbound Rules** node, right-click it and select **New Rule**.



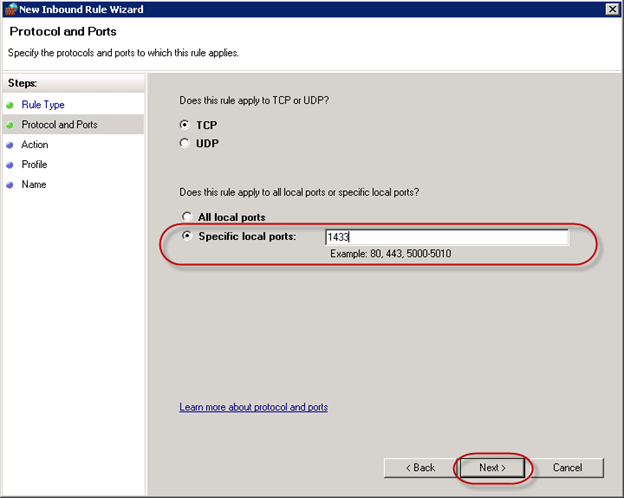
Creating an Inbound Rule

1. In the New Inbound Rule Wizard, select **Port** as **Rule Type** and click **Next**.



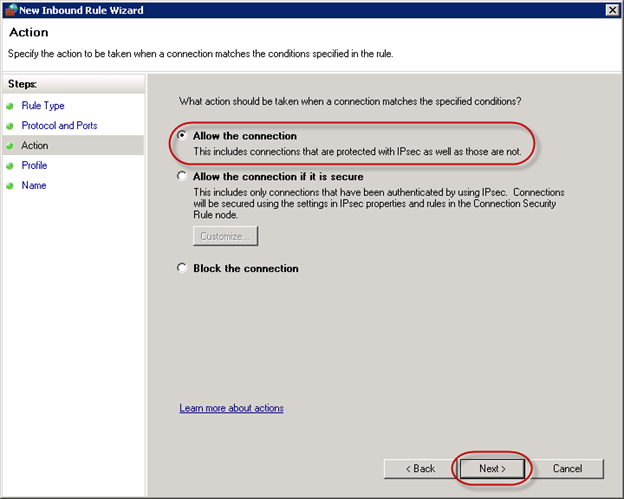
Inbound Rule's Type

1. In Protocols and Ports step, select **Specific local ports** and set its value to ***1433***. Click **Next** to continue.



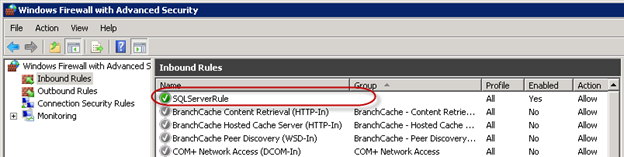
Inbound Rule's Local Port

1. In the Action step, make sure **Allow the connection** option is selected and click **Next**.



Inbound Rule's Action

1. In the Profile step, leave the default values and click **Next**.
2. Finally, set the Inbound Rule's **Name** to SQLServerRule and click **Finish**.



New Inbound Rule

1. Close the Windows Firewall with Advanced Security window.
2. Close the remote desktop session for your SQL IaaS machine.

## Task 7: Deploying a Simple MVC4 Application

In this step, you will learn how to deploy a simple ASP.NET MVC4 application in the IIS of the Azure Virtual Machine you have previously configured.

**NOTE:** To make this solution highly available, you need to configure the SQL Servers in an availability set and set up SQL Server Mirroring between the instances.

#### Deploying the MVC4 Application

1. In the Microsoft Azure Management Portal, click Virtual Machines on the left menu.
2. Remote desktop into the first IIS virtual machine you created in the earlier task by clicking **Connect** from the menu button at the bottom of the portal window. If you used the proposed name, this Virtual Machine's should be named **iisvm1**.
3. You will be prompted to download the remote desktop client. Click **Open** and log on using the remote desktop credentials you defined when creating the virtual machine.
4. Open the **C:\inetpub\wwwroot** folder and copy the file **CloudShop.zip** located in the hands-on lab material **.\Source\Assets\CloudShop** folder. To do this, copy CloudShop.zip (Ctrl + C) and paste it (Ctrl + V) in the virtual machine's wwwroot folder. Extract all files to **C:\inetpub\wwwroot\CloudShop** folder.
5. Open with Notepad, the **Web.config** file located in **C:\inetpub\wwwroot\CloudShop**. Replace the connection strings placeholder with the internal IP address of your SQL Server (by default, is the virtual machine's IP address which you can obtain by looking in the virtual networks dashboard). DO NOT INCLUDE THE BRACKETS AT THE BEGINNING AND END OF THE Data Source setting.

XML

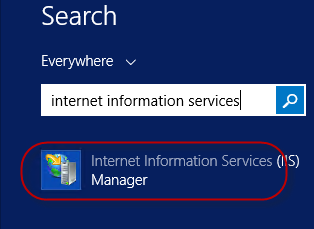
**<connectionStrings>**

**<add name="AdventureWorksEntities" connectionString="metadata=res://\*/Models.AdventureWorks.csdl|res://\*/Models.AdventureWorks.ssdl|res://\*/Models.AdventureWorks.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source=[ENTER YOUR SQL MACHINE INTERNAL IP ADDRESS];initial catalog=AdventureWorks2012;Uid=CloudShop;Password=Azure$123;multipleactiveresultsets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />**

**<add name="DefaultConnection" connectionString="Data Source=[ENTER YOUR SQL MACHINE INTERNAL IP ADDRESS];initial catalog=AdventureWorks2012;Uid=CloudShop;Password=Azure$123;MultipleActiveResultSets=True" providerName="System.Data.SqlClient" />**

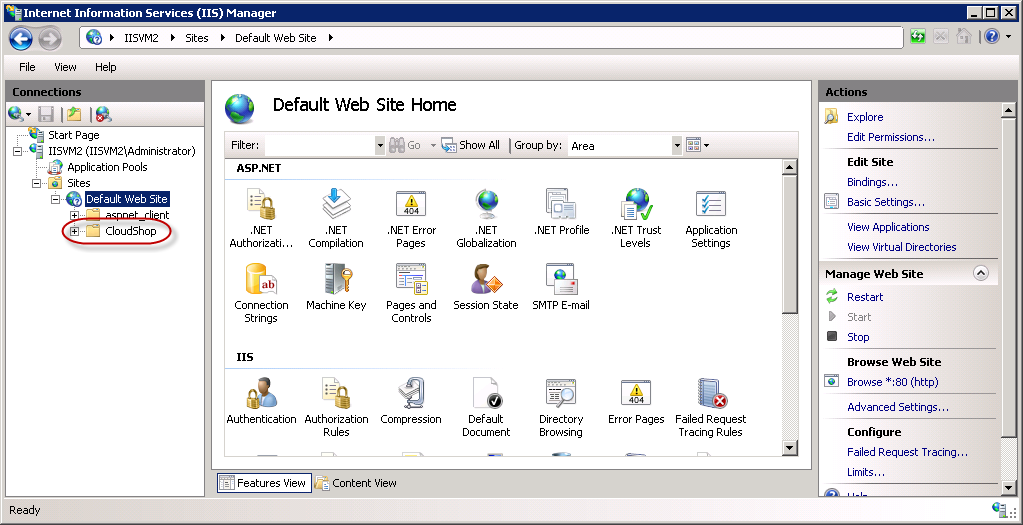
**</connectionStrings>**

1. Open the **Internet Information Services (IIS) Manager** by going to the Start page of the Windows Server and then typing in ‘Internet Information Services’. Click on the icon.



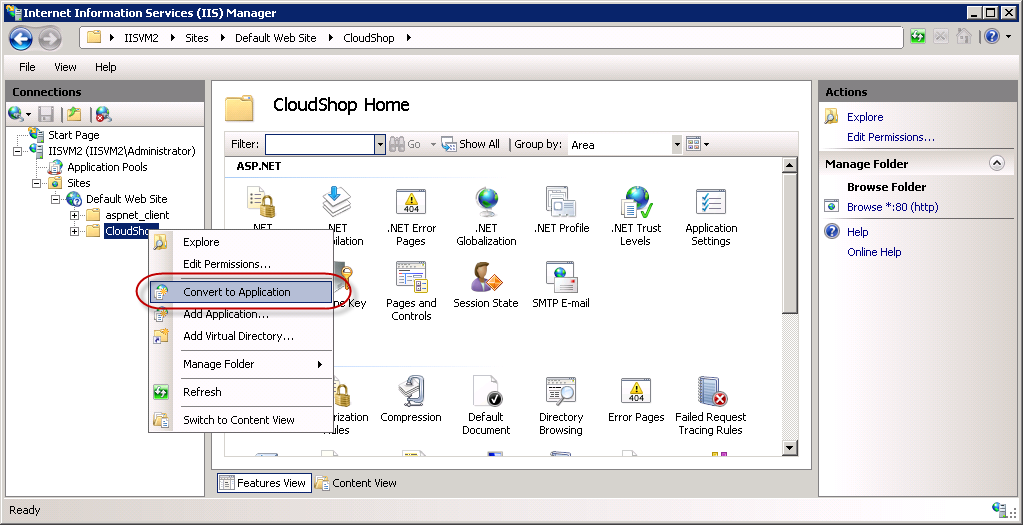
Starting Internet Information Services Manager

1. In the *Connections* pane, expand Default Web Site within your IIS Server's node. You will see the CloudShop folder you copied in the wwwroot folder.



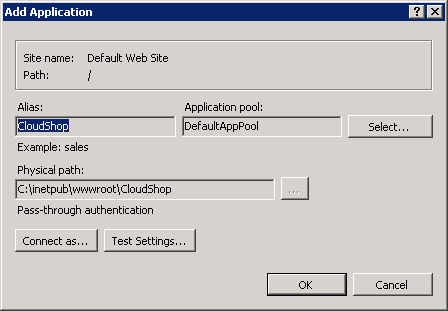
IIS Manager

1. Right-click CloudShop folder and select **Convert to Application**.



IIS Manager - Convert to Application

1. In the **Add Application** dialog, click **OK**.



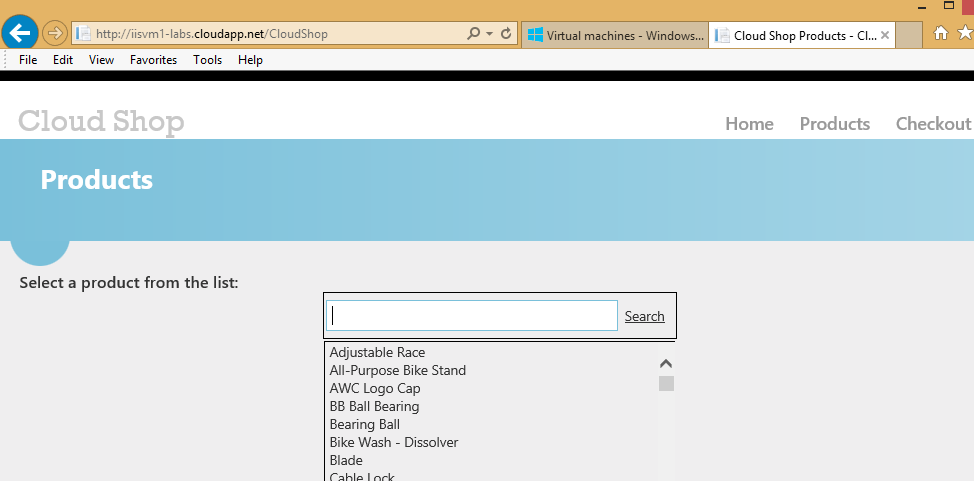
Add Application dialog

1. Close the Internet Information Server (IIS) Manager window.
2. Close the Remote Desktop Connection.
3. Repeat this task in the second IIS virtual machine you created in the earlier tasks. If you used the proposed name, this Virtual Machine should be named **iisvm2**.

### Verification

In this task, you will test the Cloud Shop MVC4 application you deployed in the previous task.

1. In your local machine, open Internet Explorer.
2. Go to http://**[YOUR-SERVICE-NAME].**cloudapp.net/CloudShop. The Service Name is the one you used when creating the IIS Virtual Machines (you can also check it in the Azure Portal, within Virtual Machine's dashboard).



MVC4 Application running in the Web Farm

1. In the *Search* box, type ***Classic*** and click Search. It will show all the products that have a product name that match the search criteria.

# Appendix A – Troubleshooting Tips

Although this lab has been tested many times, there are still places where students might mistakenly enter the wrong information while working on configuring the lab virtual machines. Here is a list of a few things that may help you with certain issues.

**Scenario 1** – You receive an error when you test your web Cloud app that states that the CloudShop user cannot log in. This means that the call actually went to the database but there is an authentication problem.

1. Remote desktop into the SQL Server machine and open SSMS to make sure the database has been attached.
2. Remote desktop into the SQL Server machine and open SSMS. Right click on the SQL Server (machine name) properties and click on the Security list item. Make sure Mixed Mode is enabled.
3. Remote desktop into the SQL Server machine and make sure there is an inbound firewall rule for port 1433 enabled.

**Scenario 2** – You believe you have everything setup correctly and you go to a separate client machine and enter http://<yourcloudservice>.cloudapp.net/CloudShop and receive (this could be a variety of http error numbers) and http error. This could mean there is a problem with one or both of the web servers and you don’t know which one because the load balancer is in control of sending the request.

1. Remote desktop into iisvm1. Using the Server Manager console, select Local Server and then select IE Security. Set the radio buttons in the dialog box that appears to OFF on both selections.
2. Open IE and type in <http://localhost/CloudShop>. This will test the app from just this web server.
3. Most http errors mean that you may not have your web.config file setup correctly, or the site is not setup correctly. Try these things:
   1. Open Internet Information Services Manager and make sure CloudShop is setup as an Application.
   2. Open the web.config file to make sure that you have entered the SQL Server private IP address correctly.

# Appendix B – Implementing DNS Name Resolution for the SQL Server Machine

In most cases, you aren’t going to want to depend on the IP address of the SQL Server machine in your connection string. As of version 2.2 of the Azure SDK, if someone were to shut down the SQL Server IaaS machine and then restart it, depending on what subnet you have the virtual machine it, it could be assigned a new IP address, thereby requiring you to change the connection string information in your web applications configuration file. So, what are the options of resolving the SQL Server IaaS machine by name?

## Option 1 – Put the SQL Server IaaS machine in the same Cloud Service as the IIS machines

By putting the SQL Server IaaS server in the same Cloud Service as the IIS machines, you can change your connection string to just use the SQL Server name, such as ‘vmSQL’. There is nothing wrong with putting all machines in the same Cloud Service although most developer generally would like different types of machines to be in different Cloud Services.

In this lab, we used a separate Cloud Service AND we chose to put the SQL Server IaaS machine in its own subnet. If this is the only machine in this subnet, if the machine is shut down and then is restarted, it will be given its original IP address.

## Option 2 – Create an IaaS AD/DNS Server and put all of the machines in the same Windows domain

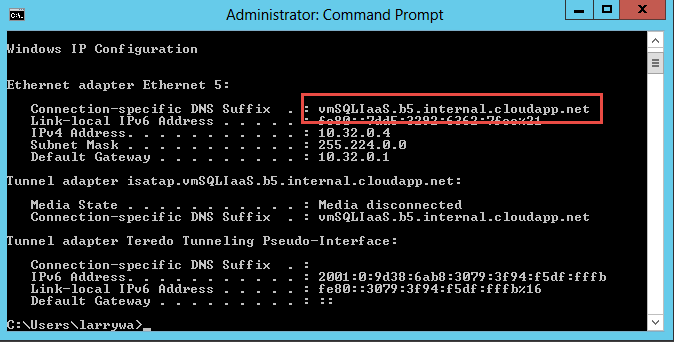
By joining the SQL Server IaaS server and the IaaS machines in the Windows AD Domain (you do not have to have them in the same Cloud Service, but they need to be in the same virtual network), the DNS on the AD server will allow you to resolve all machines by name.

Once the AD/DNS IaaS server has been created, retrieve the internal IP address of the machine and add a DNS server name and IP address to your network settings. You will then need to restart the IIS and SQL machine so they will pick up the new DNS settings.

## Option 3 – Leave the SQL Server IaaS machine in a separate Cloud Service and use the FQDN of the SQL Server IaaS machine

It is possible to use the fully qualified domain name (FQDN) of the SQL Server IaaS machine in your connection string. To get the FQDN, you will need to remote desktop in to the SQL VM and run ipconfig from a command prompt. Looking at the sample screenshot below you can see only part of the FQDN:

<cloudservicename>.<ZoneId>.internal.cloudapp.net



The actual FQDN would be:  
<vmname>.<cloudservice>.<ZoneId>.internal.cloudapp.net

vmSQL.vmSQLIaaS.b5.internal.cloudapp.net

The issue with this approach is that currently, the ZoneId can change if the machine is stopped and restarted, resulting in the FQDN changing, thereby breaking your connection string.

## Option 4 – Create an IaaS server in Azure (same virtual network) and set it up as the DNS Server

When creating your virtual network (you can also do this after the virtual network has been created), you can add a new Windows Server to your virtual network and install DNS on it. By adding a new forward lookup zone to your DNS, you can put in the SQL Servers IP address and name.

If your IIS machines are added to the same zone in your DNS server, you can resolve by name.

The issue with this approach is still that if the SQL Server IaaS machines IP address changes that will break the DNS zone setting. You could put the SQL Server IaaS machine in its own subnet though to alleviate the IP address change problem.

Also, with your DNS server IaaS machine in Azure, depending on which subnet this machine is in, if you shut it down and restarted it, this machine also could have a different IP address. You would need to put this machine in its own subnet to alleviate this problem.

## Option 5 – Create a Site-To-Site Network and use an external DNS Server

For this option, the site-to-site network using an on-premises VPN device and an on-premises DNS server could be implemented. The on-premises DNS server IP address can be entered into your virtual network configuration information and then, as in Option 4, you could enter your Azure SQL Server IaaS machines IP address into a forward lookup zone.

Although you still have the challenge of making sure the IP address of the database server does not change, on-premises you could assign a static IP address to your DNS server to assure that you have no IP address changes on the DNS machine.