Microsoft Azure - Starter Kits for Partners

Hands on Lab

Application Servers Scenario

With HA & Auto Scaling

(Azure Resource Manager Mode)

Last Update: May 2016





**MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.**

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

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, trademarks, 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.

The descriptions of other companies’ products in this document, if any, are provided only as a convenience to you. Any such references should not be considered an endorsement or support by Microsoft. Microsoft cannot guarantee their accuracy, and the products may change over time. Also, the descriptions are intended as brief highlights to aid understanding, rather than as thorough coverage. For authoritative descriptions of these products, please consult their respective manufacturers.

© 2016 Microsoft Corporation. All rights reserved. Any use or distribution of these materials without express authorization of Microsoft Corp. is strictly prohibited.

Microsoft and Windows are either registered trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Contents

[Overview 4](#_Toc450913793)

[Objectives 4](#_Toc450913794)

[Prerequisites 4](#_Toc450913795)

[Exercises 5](#_Toc450913796)

[Infrastructure Provisioning 5](#_Toc450913797)

[Exercise 1: Creating SQL Server, Domain Controllers and Virtual Network (New Portal/ARM) 5](#_Toc450913798)

[Exercise 2: Deploying High Available IIS Farm with Scale Set and ARM 5](#_Toc450913799)

[Exercise 3: Deploying a Simple MVC4 Application - Manually 20](#_Toc450913800)

[Summary 27](#_Toc450913801)

## Overview

Using Microsoft Azure, as your Infrastructure as a Service (IaaS) platform, will enable you to create and manage your infrastructure quickly, provisioning and accessing any host ubiquitously. Grow your business through the cloud-based infrastructure, reducing the costs of licensing, provisioning and backup.

In this hands-on Lab, you will learn how to deploy a simple ASP.NET MVC Web application to a Highly Available IIS Web and SQL Server Farm hosted in Microsoft Azure with Azure Resource Manager Model.

**Estimated time** to complete this lab: **180 minutes**.

**Audience**: IT Pro, Architect, Application Owners and Developers

### Objectives

In this hands-on lab, you will learn how to:

* Learn how to deploy SQL Server Always ON and IIS Farm in Azure Resource Manager Model, which will include
  + Create a Virtual Network with and Subnets
  + Create a Web Farm (IIS) using Microsoft Azure Management Portal
  + Configure Load Balancing for Web Farm
  + Create Domain Controller Virtual Machines
  + Create Virtual Machines with SQL Server
    - Enable Full-Text Search feature to be consumed by the MVC Application
    - Configure SQL Server to use Always On
  + Deploy a Simple MVC4 Application that consumes SQL Server Features
  + Automate provisioning Azure Resource Manager
  + Configure Auto Scaling with Scate Set

### Prerequisites

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

* A Microsoft Azure subscription - [sign up for a free trial](http://aka.ms/WATK-FreeTrial)
* [Azure PowerShell 0.7.4  or higher](https://azure.microsoft.com/en-us/documentation/articles/powershell-install-configure/)

## Exercises

## Infrastructure Provisioning

### Exercise 1: Creating SQL Server, Domain Controllers and Virtual Network (New Portal/ARM)

#### Deployment through Azure Resource Manager Model

1. Before deploying the solution through ARM, read to understand technical capabilities. [Azure ARM: SQL Server High-Availability and Multi-Datacenter Disaster Recovery with Internal Load Balancers (ILB)](https://blogs.msdn.microsoft.com/igorpag/2016/01/26/azure-arm-sql-server-high-availability-and-multi-datacenter-disaster-recovery-with-internal-load-balancers-ilb/)
2. Below you will find the **step by step for** deploying the solution through the ARM Template SQL Server AlwaysOn
3. <https://blogs.technet.microsoft.com/dataplatforminsider/2014/08/25/sql-server-alwayson-offering-in-microsoft-azure-portal-gallery/>

#### Listener Configuration for AlwaysOn Availability Groups

This topic shows you how to configure a listener for an AlwaysOn Availability Group. Your Availability Group can contain replicas that are on-premises only, Azure only, or span both on-premises and Azure for hybrid configurations. Azure replicas can reside within the same region or across multiple regions using multiple virtual networks (VNets).

The steps below assume you have already configured an availability group but have not configured a listener. Note the following limitations on the availability group listener in Azure:

**Step by step:** <http://msdn.microsoft.com/en-us/library/azure/dn425027.aspx>

### Exercise 2: Deploying High Available IIS Farm with Scale Set and ARM

VM Scale Sets allow you to deploy and manage a group of identical virtual machines as a set. Advantages include:

* **Autoscale** - simply change the instance count and Scale Sets will increase or decrease the number of VMs evenly across update and fault domains.
* **Performance** - deploying multiple VMs results in a single call to the fabric, allowing inherent performance optimizations.
* **Customization** - built on Azure IaaS, Scale Sets support all Windows and Linux VMs including custom images and extensions.
* **Ease of management** - building on the simple declarative modelling introduced with Azure Resource Manager, Scale Sets are the simplest way to manage sets of identical VMs. Focus on compute at scale without managing scaling of storage accounts and NICs.

**Before you start:** read [Virtual Machine Scale Sets Overview](https://azure.microsoft.com/en-us/documentation/articles/virtual-machine-scale-sets-overview/) and how its integrate with ARM:

**Deploy using the ARM Template below.**

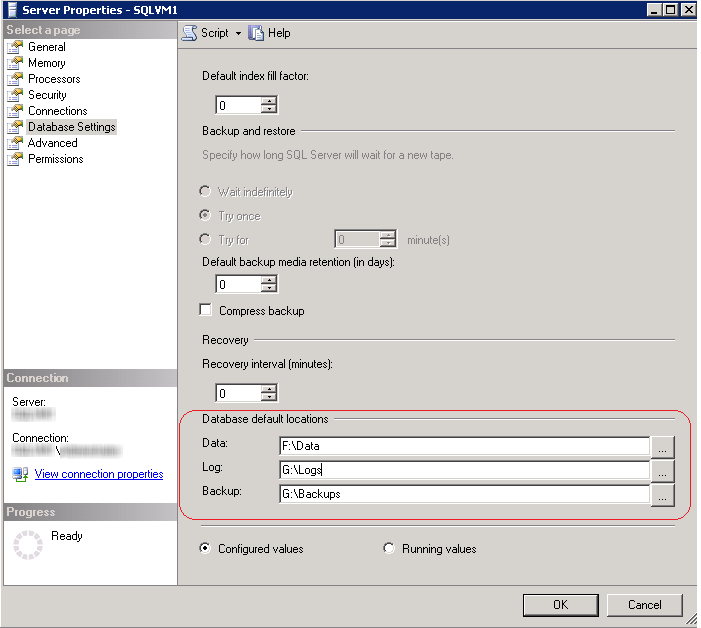
This is an example which creates a VM scale set of VMs running IIS web server, and uses a load balancer to balance the load that each VM receives. It also uses the HTTP protocol to ping a specific URL on each VM. look at the Microsoft.Network/loadBalancers resource type and the networkProfile and extensionProfile in the virtualMachineScaleSet

<https://github.com/gbowerman/azure-myriad/blob/master/vmss-win-iis-vnet-storage-lb.json>

#### Configure AdventureWorksDatabase

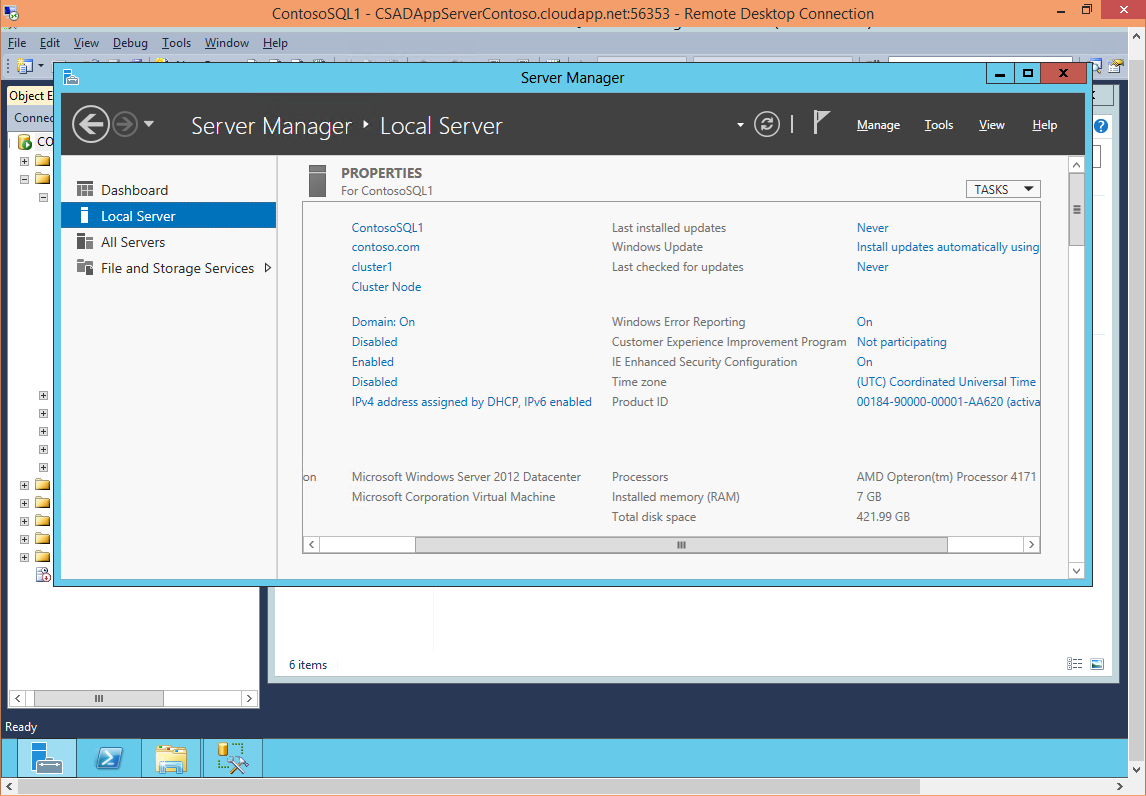
In this task, you will configure SQL Server 2012. You will create the database that will be used by the MVC4 application and add Full-Text Search capabilities to it. Additionally, you will create a SQL Server user for the MVC4 website.

1. If you have not logged out of the remote desktop session for the VM, do so now.
2. Launch the RDP file for the VM and log in as BUILTIN\AzureAdmin.
3. Open Windows Explorer and create the following folders: **F:\Data, G:\Logs** and **G:\Backups**.
4. Open the SQL Server Management Studio from **Start Screen | SQL Server Management Studio**.
5. Connect to the SQL Server 2012 default instance using your Windows Account.
6. Now, you will update the database's default locations in order to split the DATA from the LOGS. To do this, right click on your SQL Server instance and select **Properties**.
7. Select **Database Settings** from the left side pane.
8. Locate the **Database default locations** section and update the default values to point to the disks you attached in the previous task and then click "Ok".

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/Setting-Database-Default-Locations.png?raw=true)

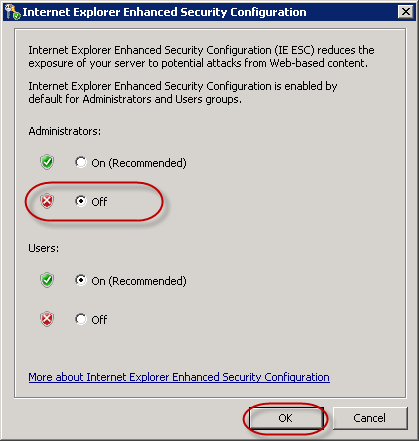
Setting Database Default Locations

1. Restart SQL Server. In the **Object Explorer**, right-click on the server node and select **Restart**.
2. In order to enable downloads from Internet Explorer you will need to update **Internet Explorer Enhanced Security Configuration**. In the Azure Virtual Machine, open **Server Manager** from **Start | Administrative Tools | Server Manager**.
3. In the **Server Manager**, click **Configure IE ESC** within **Security Information** section.



Configuring IE ESC

1. In the **Internet explorer Enhanced Security** configuration, turn **off** the enhanced security for **Administrators** and click **OK**.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/internet-explorer-enhanced-security2.png?raw=true)

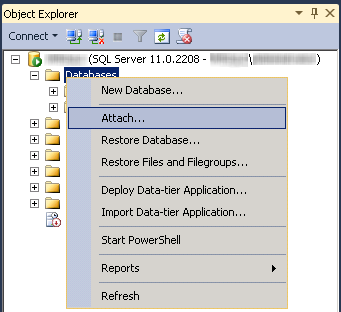
Internet Explorer Enhanced Security

**Note:** Modifying **Internet Explorer Enhanced Security** configurations is not good practice and is only for the purpose of this particular lab. The correct approach should be to download the files locally and then copy them to a shared folder or directly to the Virtual Machine.

1. This lab uses the **AdventureWorks2012** database. Open an **Internet Explorer** browser and go to <http://msftdbprodsamples.codeplex.com/> to download the **SQL Server 2012** sample databases. Once on the page click on **AdventureWorks Databases – 2008, 2008R2 and 2012** and then download Adventure Works 2012 Data File. Download the file to F:\Data.

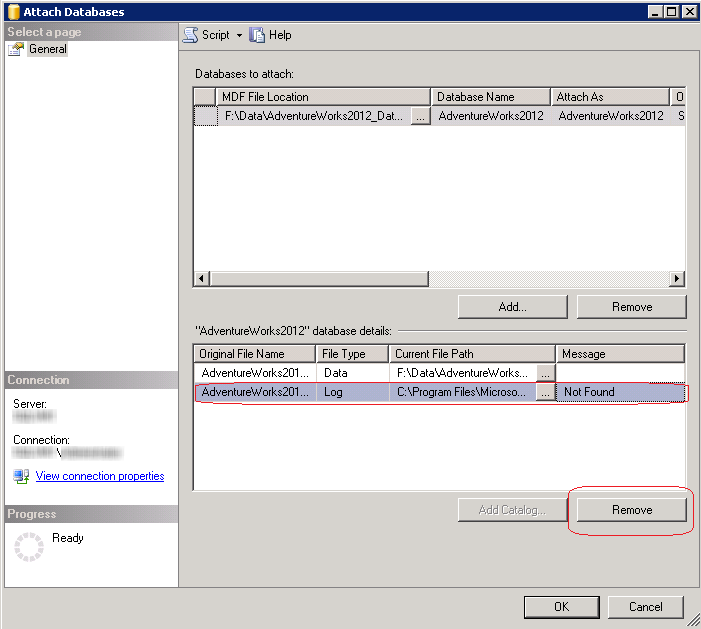
**Note:** The **AdventureWorks2012** database can also be downloaded as a .zip file. If you choose this format, right-click the file to open its properties window and then click **Unblock**. Then, extract the database to F:\Data.

1. Add the **AdventureWorks2012** sample database to your SQL Server. To do this, in the **SQL Server Management Studio**, locate your SQL Server instance node and expand it. Right click the **Databases** folder and select **Attach**.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/attaching-adventureworks-database-menu.png?raw=true)

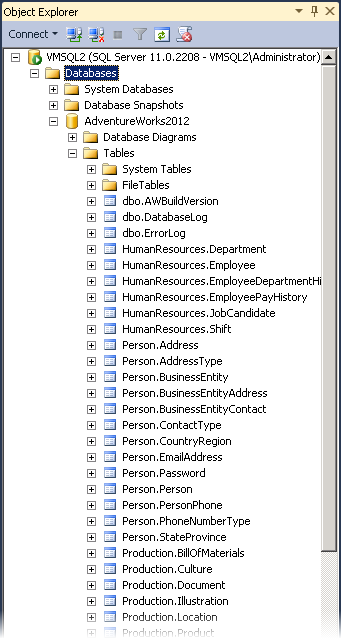
Attaching the database

1. In the **Attach Databases** dialog, press **Add**. Browse to the data disk and select the Adventure Works 2012 data file.
2. Select the **AdventureWorks2012** Log entry and click **Remove**.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/removing-adventureworks-log-entry.png?raw=true)

Removing AdventureWorks2012 Log entry

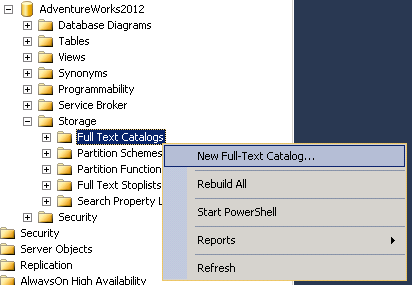
1. Press **OK** to add the database.
2. In the **Databases** folder, locate the new **AdventureWorks2012** database and explore its tables.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/adventureworks-sample-database.png?raw=true)

AdventureWorks Sample Database

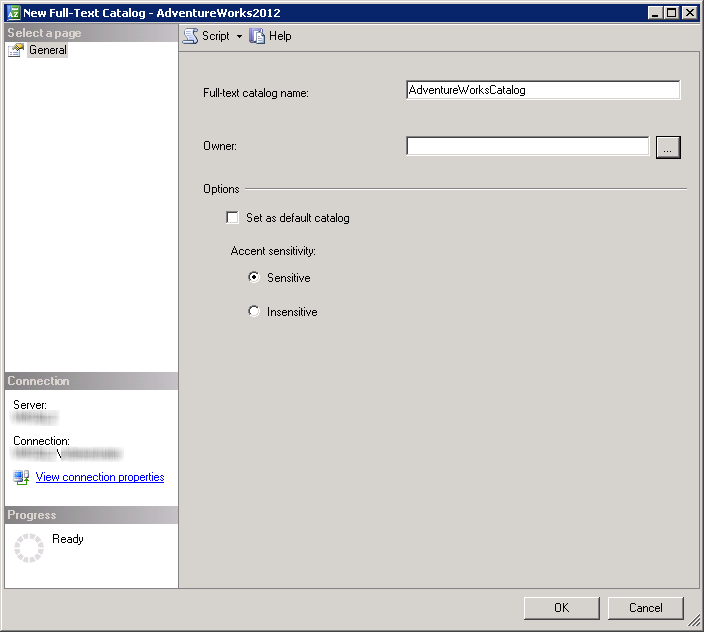
1. Expand **Storage** node within **AdventureWorks2012** database, right-click **Full Text Catalogs** folder and select **New Full-Text Catalog**.

**Note:** You are creating a Full Text Catalog for the database that will be used later by the MVC application.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/create-new-full-text-catalog2.png?raw=true)

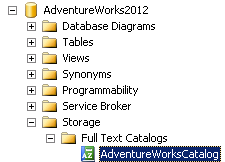
Create New Full-Text Catalog

1. In the New Full-Text Catalog dialog, set the **Name** value to AdventureWorksCatalog and press **OK**.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/create-new-full-text-catalog3.png?raw=true)

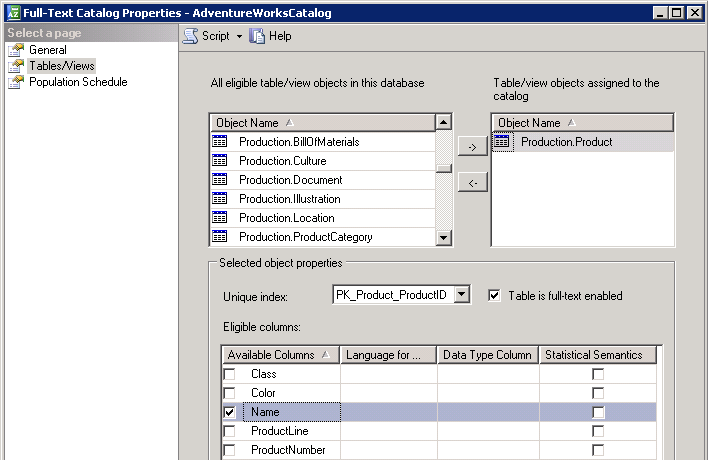
Create New Full-Text Catalog

1. Check that the Full-Text Catalog you created appears in the **Full-Text Catalogs** folder.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/create-new-full-text-catalog5.png?raw=true)

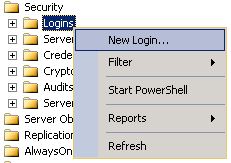
Create New Full-Text Catalog

1. Right-click **AdventureWorksCatalog** and select **Properties**. In the **Full-Text Catalog Properties** dialog, switch to **Tables/Views** page.
2. Add the **Production.Product** table to the **Table/View objects assigned to the Catalog** list. Then, check the Name column and click **OK**.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/create-new-full-text-catalog4.png?raw=true)

Create New Full-Text Catalog

1. Add a new user for the MVC4 application you will deploy in the following exercise. To do this, expand **Security** folder within the SQL Server instance. Right-click **Logins** folder and select **New Login**.

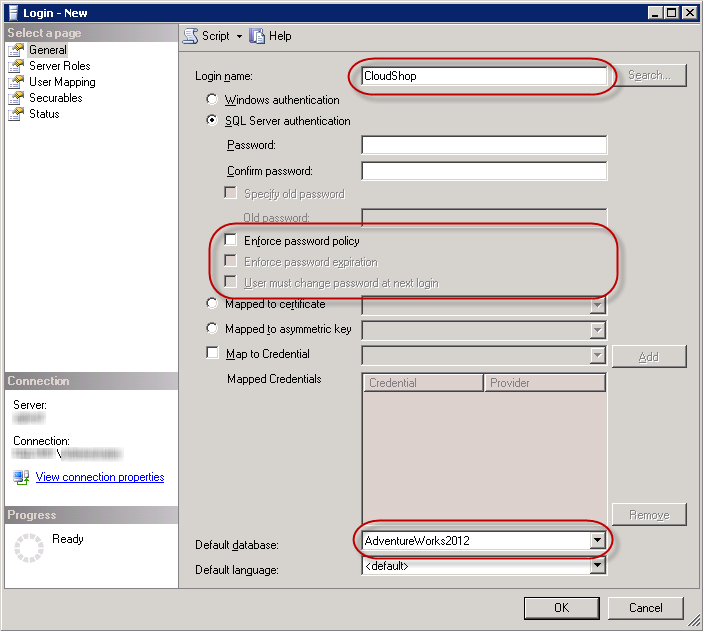
[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/creating-a-new-login2.png?raw=true)

Creating a New Login

1. In the **General** section, set the **Login name** to CloudShop. Select **SQL Server authentication** option and set the **Password** to Azure$123.

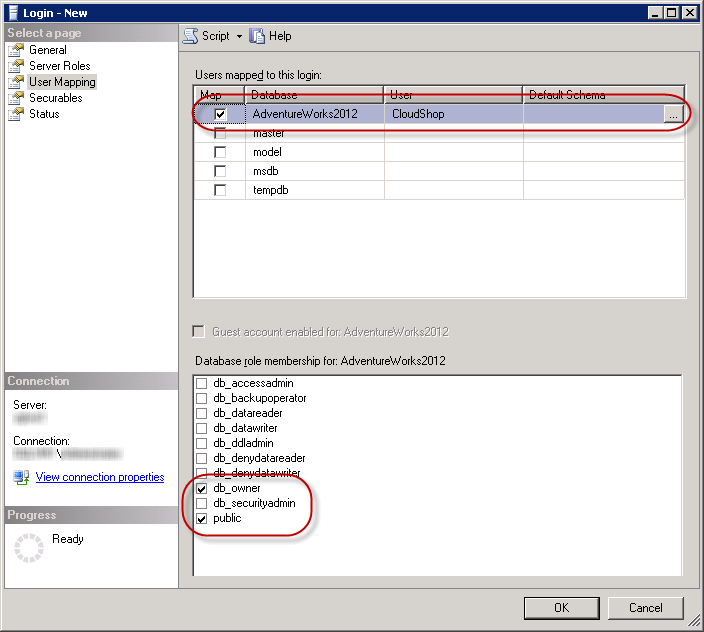
**Note:** If you enter a different username or password than those suggested in this step, do not forget in the next exercise to update the web.config file of the MVC4 application to match those values.

1. Unselect **Enforce password policy** checkbox to avoid having to change the password the first time you log on, and set the **Default database** to AdventureWorks2012.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/Creating-a-New-Login.png?raw=true)

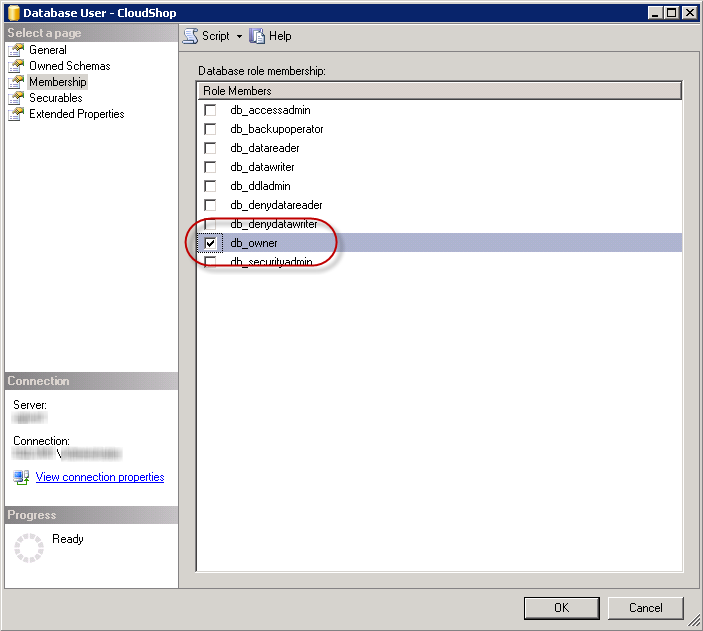
Creating a New Login

1. Click **User Mapping** on the left pane. Select the map checkbox in the AdventureWorks2012 database row and click **OK**.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/mapping-new-user-database-2.png?raw=true)

Mapping the new User to the AdventureWorks Database

1. Expand **AdventureWorks2012** database within **Databases** folder. In the **Users** folder under **Security**, double-click **CloudShop** user.
2. Select the **Membership** page, and select the db\_owner role checkbox for the **CloudShop** user and click **OK**.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/Adding-Database-role-membership-to-CloudShop-user.png?raw=true)

Adding Database role membership to CloudShop user

**Note:** The application you will deploy in the next exercise uses Universal Providers to manage sessions. The first time the application runs, the provider will create the Sessions table within the database. For that reason, you are assigning a db\_owner role to the CloudShop user. Once you run the application for the first time, you can remove this role as these permissions will not be needed.

1. Now, enable **Mixed Mode Authentication** to the SQL Server instance. To do this, in the **SQL Server Management Studio**, right-click the server instance and click **Properties**.
2. Click **Security** in the left side pane and then select **SQL Server and Windows Authentication mode** under **Server Authentication** section. Click **OK** to save changes.
3. Restart the SQL Server instance. To do this, right-click the SQL Server instance and click **Restart**.
4. Close the **SQL Server Management Studio**.
5. Repeat the steps below on SQL Server Contoso 2 –
   1. Steps 1 through 12,
   2. Steps 24 through 28.
   3. You don´t need to create the AdventureWorks on SQL Server Contoso 2, you will create it via a Full Back up in the next Steps

### Exercise 3: Deploying a Simple MVC4 Application - Manually

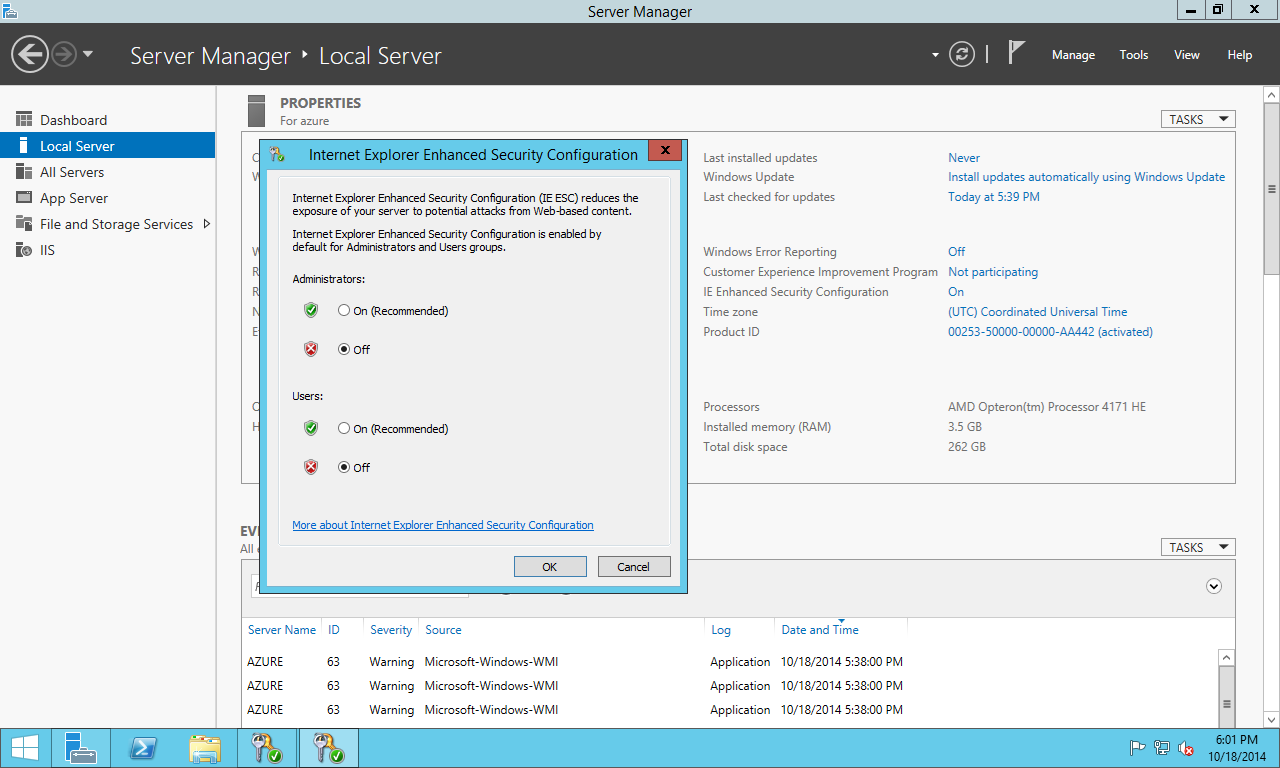
In this exercise, you will learn how to deploy a simple ASP.NET MVC4 application in the IIS of the Azure Virtual Machine you have previously configured.

**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.

#### Task 1 - Deploying a Simple MVC4 Application

In this task, you will deploy the MVC4 application to the IIS Virtual Machines.

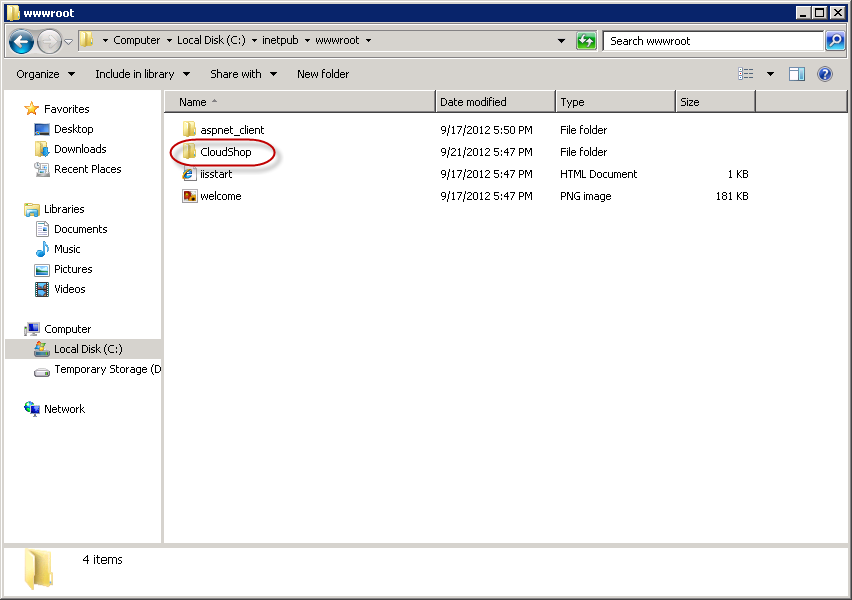
1. In the Azure Portal, Click **Virtual Machines** on the left menu.
2. You will see a list with your existing Virtual Machines. Select the first one you created in Exercise 1 and click **Connect**. If you used the proposed name, this Virtual Machine's should be named **iisvm1**.
3. You will be prompted to download the remote desktop client. Click **Open** and log on using the Admin credentials you defined when creating the Virtual Machine.
4. In the Server Manager, Local Server, **Internet explorer Enhanced Security** dialog, turn **off** enhanced security for **Administrators** and click **OK**.



Internet Explorer Enhanced Security

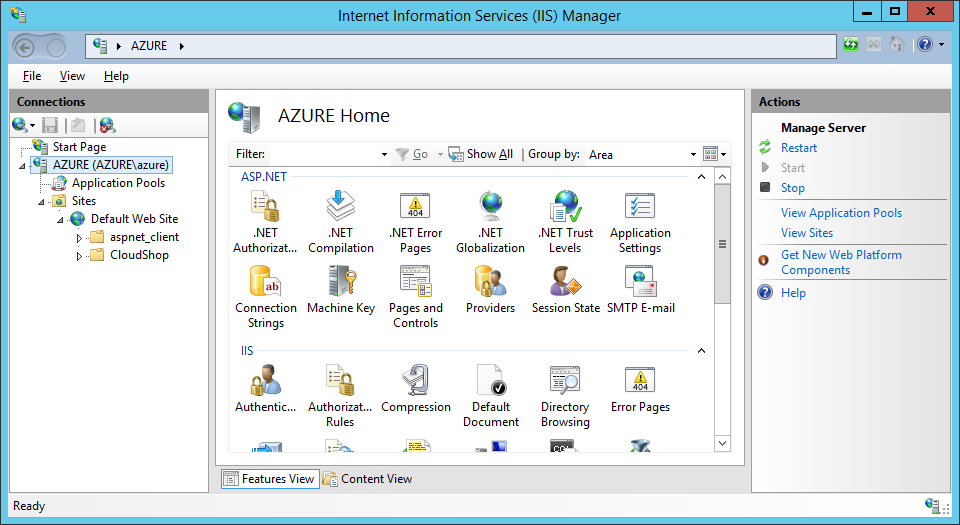
**Note:** Modifying Internet Explorer Enhanced Security configurations is not a good practice and it only for the purpose of this particular lab. The correct approach would be to download the files locally and then copy them to a shared folder or directly to the Virtual Machine.

1. Once **.Net Framework 4.5** installation finishes, open **wwwroot** folder located at **C:\inetpub\** and copy the file **7 - Hand On Lab - Dev Test StarterKit.zip** located in the **current** **folder** of this lab. To do this, copy **7 - Hand On Lab - Dev Test StarterKit.zip** (**Ctrl + C**) and paste it (**Ctrl + V**) in the Virtual Machine's **wwwroot** folder. Extract all files to **C:\inetpub\wwwroot\CloudShop** folder.

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/wwwroot-folder.png?raw=true)

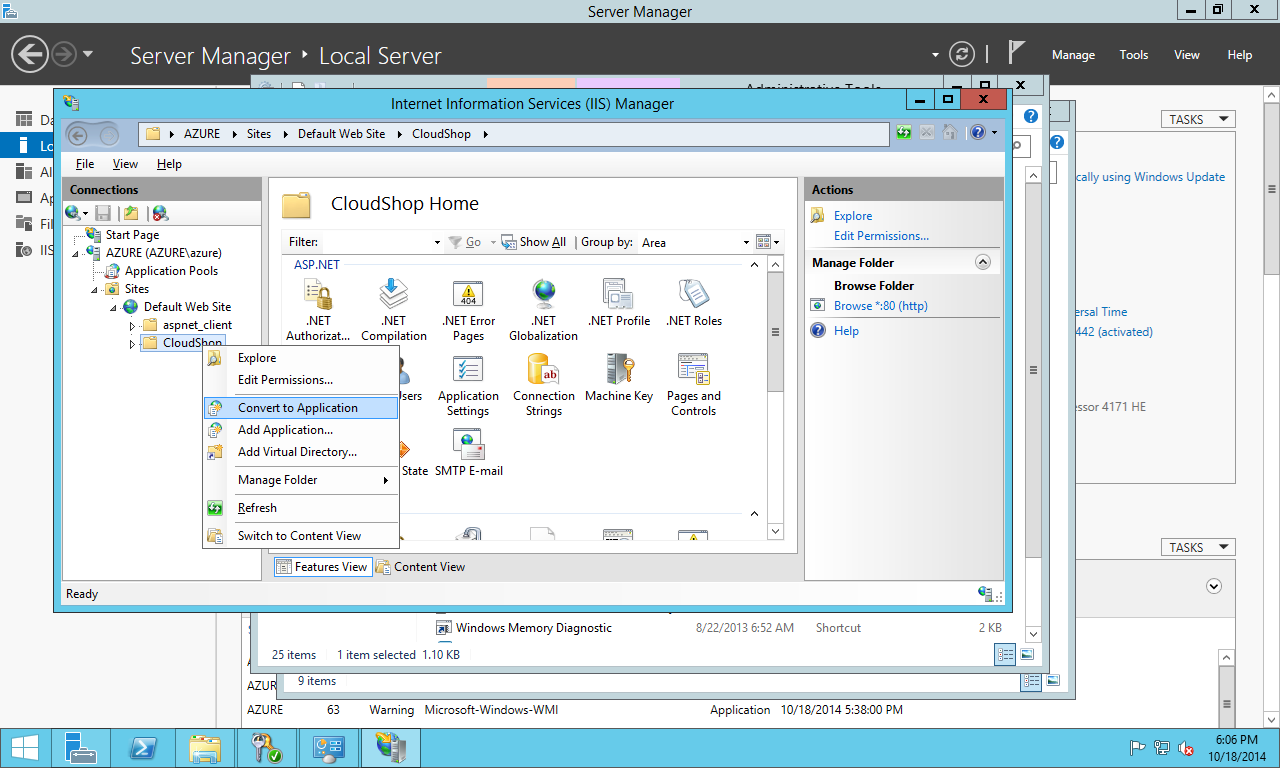
wwwroot folder

1. Open with **Notepad** the **Web.config** file located in **C:\inetpub\wwwroot\CloudShop**. Replace the connection strings placeholder with the name of your SQL Server (by default, is the Virtual Machine's name).
2. <connectionStrings>
3. <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=["<ListenerName>,<EndpointPort];initial catalog=AdventureWorks2012;Uid=CloudShop;Password=Azure$123;multipleactiveresultsets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />
4. <add name="DefaultConnection" connectionString="Data Source=["<ListenerName>,<EndpointPort];initial catalog=AdventureWorks2012;Uid=CloudShop;Password=Azure$123;MultipleActiveResultSets=True" providerName="System.Data.SqlClient" />
5. </connectionStrings>
6. Open the **Internet Information Services (IIS) Manager** from **Start | Administrative Tools**.
7. 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.



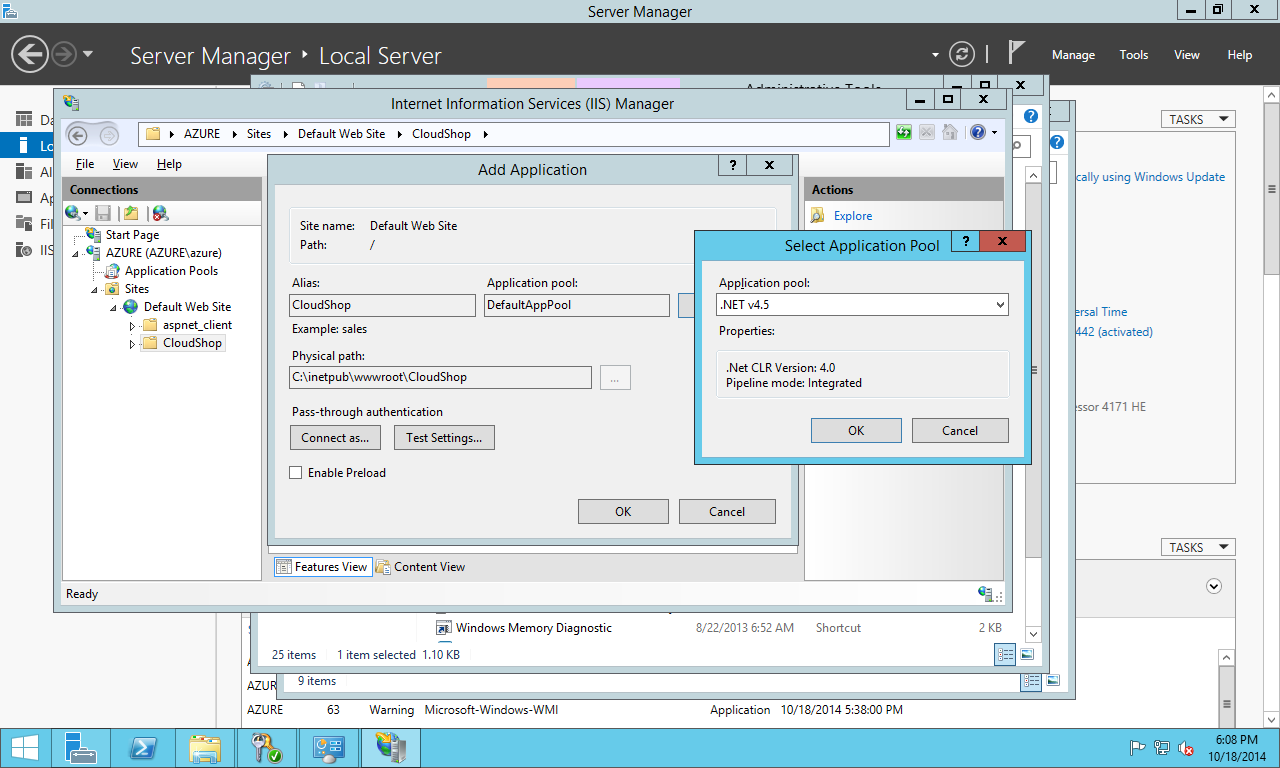
IIS Manager

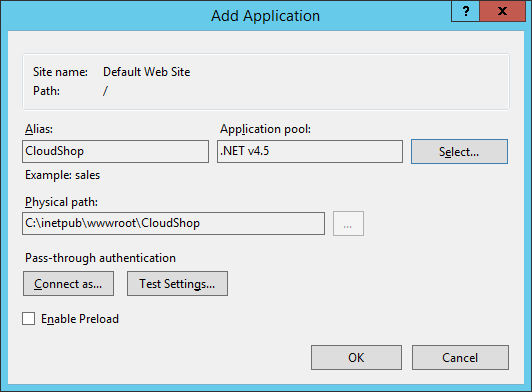
1. Right-click **CloudShop** folder and select **Convert to Application**.



IIS Manager - Convert to Application

1. In the **Add Application** dialog, Change the Application Pool to **ASP.Net v4.0** **OK**.





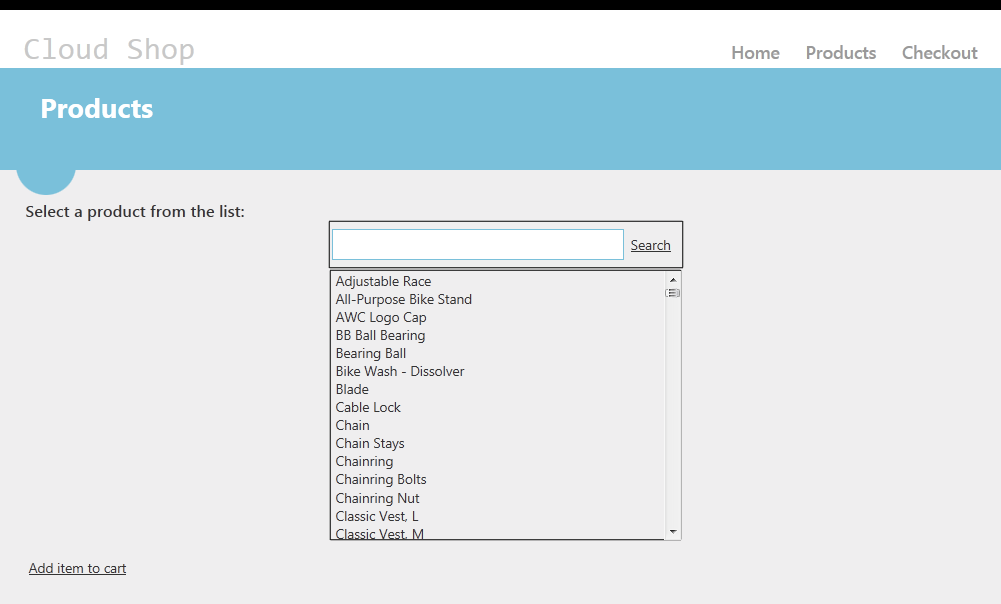
Add Application dialog

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

#### Verification

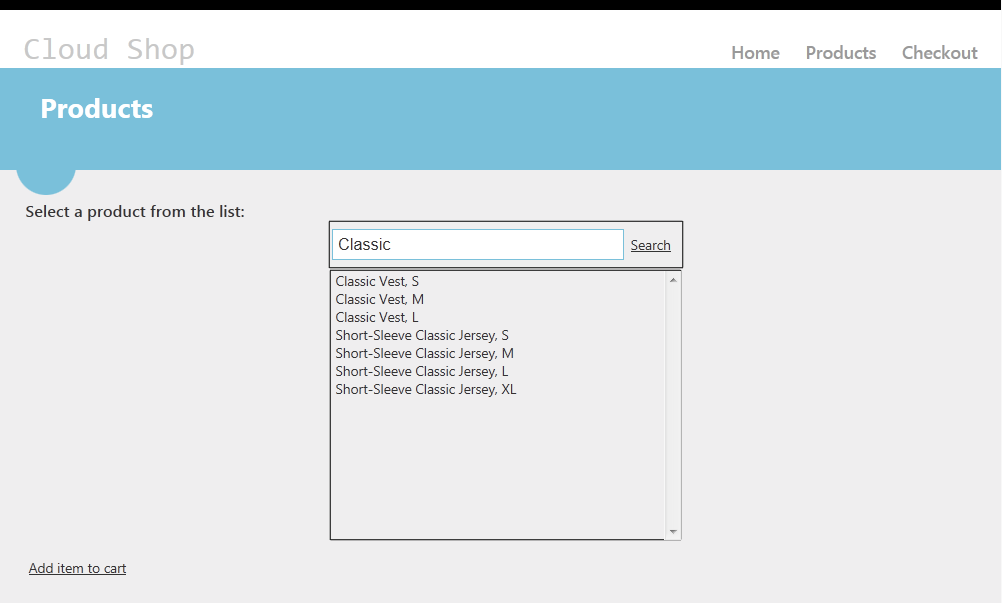
In this task, you will test the Cloud Shop MVC4 application you deployed in the previous task.

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).

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/MVC4-Application-running-in-the-Web-Farm.png?raw=true)

MVC4 Application running in the Web Farm

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

[](https://github.com/WindowsAzure-TrainingKit/HOL-IntroToWindowsAzureVirtualMachines/blob/master/Images/searching-products2.png?raw=true)

Searching Products

## Summary

In this hands-on Lab, you have learnt how to deploy a simple ASP.NET MVC 4 Web application to a Web server hosted in Microsoft Azure with Azure Resource Manager Model, using SQL Server and configuring load balancing. Finally, you automated the deployment with ARM Template for IIS.