

Hands-On Lab

Windows Azure Traffic Manager

Lab version: 1.0.0

Last updated: 4/11/2011



Contents

[Overview 3](#_Toc290308634)

[Getting Started: Preparing and Deploying the Application 7](#_Toc290308635)

[Task 1 – Preparing for Deployment 7](#_Toc290308636)

[Task 2 – Configuring Storage Account Credentials 11](#_Toc290308637)

[Task 3 – Deploying the Application to Multiple Regions 12](#_Toc290308638)

[Task 4 – Configuring the DNS Cache for Testing 20](#_Toc290308639)

[Exercise 1: Routing Traffic for Enhanced Performance 25](#_Toc290308640)

[Task 1 – Creating a Traffic Manager Performance Policy 25](#_Toc290308641)

[Task 2 – Testing the Performance Policy 28](#_Toc290308642)

[Task 3 – Simulating a Service Failure 31](#_Toc290308643)

[Exercise 2: Balancing Traffic across Hosted Services 33](#_Toc290308644)

[Task 1 – Creating a Traffic Manager Round Robin Policy 33](#_Toc290308645)

[Task 2 – Testing the Round Robin Policy 34](#_Toc290308646)

[Exercise 3: Increasing Availability with Failover Policies 37](#_Toc290308647)

[Task 1 – Creating a Traffic Manager Failover Policy 37](#_Toc290308648)

[Task 2 – Testing the Failover Policy 39](#_Toc290308649)

[Exercise 4: Managing Traffic Manager Policies 41](#_Toc290308650)

[Task 1 – Controlling Traffic to Individual Hosted Services 41](#_Toc290308651)

[Task 2 – Disabling a Traffic Manager Policy 44](#_Toc290308652)

[Summary 47](#_Toc290308653)

[Appendix A: Configuring your Windows Azure Management Portal Credentials in Visual Studio 47](#_Toc290308654)

[Appendix B: Configuring the DNS Cache of the Browser 55](#_Toc290308655)

[How to configure the DNS Cache to decrease the lifetime of host entries to 30 seconds 55](#_Toc290308656)

[How to restore the DNS Cache configuration 56](#_Toc290308657)

Overview

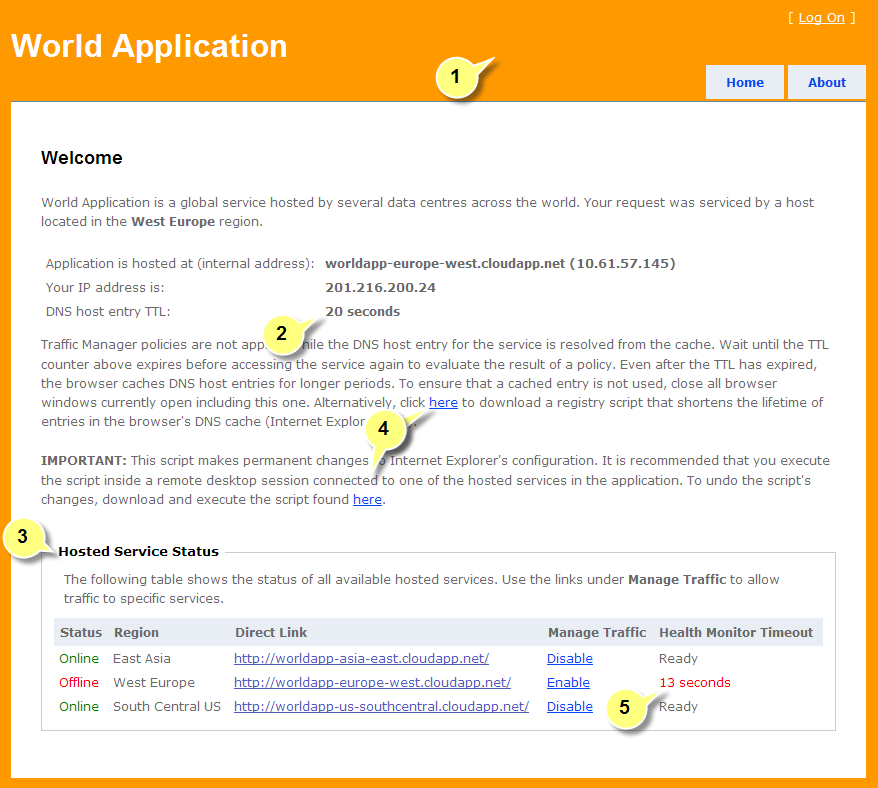
* 1. Windows Azure Traffic Manager is a load balancing solution that enables the distribution of incoming traffic among different hosted services in your Windows Azure subscription, regardless of their physical location. Traffic routing occurs as a the result of policies that you define and that are based on one of the following criteria:
  + **Performance** – traffic is forwarded to the closest hosted service in terms of network latency
  + **Round Robin** – traffic is distributed equally across all hosted services
  + **Failover**  – traffic is sent to a primary service and, if this service goes offline, to the next available service in a list
  1. You assign each policy a DNS name and associate it with multiple hosted services. The load balancer responds to queries for the policy DNS name with the address of one of the associated hosted services that satisfies the criteria for the policy. Traffic Manager constantly monitors hosted services to ensure they are online and will not route traffic to any service that is unavailable.
  2. In this hands-on lab, you will explore Windows Azure Traffic Manager by deploying a very simple application to multiple hosted services, each one in a different geographic region, and then create several Traffic Manager policies to evaluate how the load balancer routes traffic across these services.

# About DNS Caching

The client resolver in Windows caches DNS host entries for the duration of their time-to-live (TTL). Whenever you evaluate Traffic Manager policies, retrieving host entries from the cache bypasses the policy and you could observe unexpected behavior. For example, accessing a service via the endpoint of a round robin policy will continue to yield the same host address for as long as the entry remains in the cache. In general, this is not a problem because the goal for this type of policy is to balance traffic among different clients. However, when you test the policy from a single client, you need to ensure that the DNS name is resolved for each request to observe the round robin behavior.

1. If the TTL of a DNS host entry in the cache expires, new requests for the same host name should result in the client resolver executing a fresh DNS query. However, browsers typically cache these entries for longer periods, even after their TTL has expired. To reflect the behavior of a Traffic Manager policy accurately when accessing the application through a browser, it is necessary to force the browser to clear its DNS cache before each request.
   1. Throughout the lab, you will be required to access policy endpoints repeatedly to evaluate their behavior. To ensure predictable results, it is essential that you do not use a cached entry. Restarting the browser is one way to guarantee this condition, but requires closing every open browser window before each request, even those that display other sites.
   2. To make this process simpler, the lab provides a registry script to shorten the lifetime of entries in the browser’s DNS cache. After executing the script and restarting the browser, evaluating a policy is simply a matter of waiting for the duration of the TTL and then refreshing the browser window. Note, however, that because this script changes Internet Explorer's configuration, you will only execute it inside a remote desktop session connected to one of the hosted services in the application.
   3. For more information, see Appendix B: Configuring the DNS Cache of the Browser.

# About the World Application

* 1. The World Application is a sample ASP.NET application included in this hands-on lab to represent a global application that you could scale across multiple data centers by taking advantage of Windows Azure Traffic Manager. Although it does not implement any specific functionality of its own, the application does include some useful features that can assist you in the evaluation of Traffic Manager.
  2. Some of these features are:
  3. A customizable background that allows you to determine, at a glance, which of the hosted services responds to a request
  4. A timer—restarted with every page refresh—that shows the remaining time on the DNS host entry’s TTL and when it is safe to refresh the page to observe the result of a policy
  5. A control panel showing the status of each hosted service deployed including commands to enable or disable traffic to individual services
  6. Download links for registry scripts to configure the lifetime of DNS host entries in the browser’s cache
  7. A timer—restarted after every change in the status of a service—that shows elapsed time since a change in the state of a service and estimating how long it takes Traffic Manager to become aware of the change
  8. 
  9. Figure
  10. Home page of the World Application
  11. Traffic Manager monitors hosted services by executing periodic HTTP GET requests to an endpoint that you specify when creating a policy. In the simplest case, this endpoint can be the URL to a file served by the application. Traffic Manager considers the service to be available if its monitoring endpoint responds with an HTTP status code of 200 OK within 5 seconds. For the application in this hands-on lab, this monitoring endpoint, located at the URL */AppHealth*, has been implemented so that it can be disabled on demand. During the hands-on lab, you will exploit this feature to simulate a service failure.

# Objectives

* 1. In this hands-on lab, you will learn how to:
  + Define Windows Azure Traffic Manager policies
  + Route traffic to the hosted service that offers the best performance
  + Balance traffic across all hosted services
  + Create a active / standby configuration that routes traffic to a primary service and, in the event of failure, to other secondary services
  + Temporarily disable traffic to specific hosted services

# Prerequisites

* 1. The following is required to complete this hands-on lab:
  + [Microsoft .NET Framework 4.0](http://go.microsoft.com/fwlink/?linkid=186916)
  + [Microsoft Visual Studio 2010](http://msdn.microsoft.com/vstudio/products/)
  + [Windows Azure Tools for Microsoft Visual Studio 1.4](http://go.microsoft.com/fwlink/?LinkID=128752)
  + [Windows Azure subscription](http://www.microsoft.com/windowsazure/offers/default.aspx)

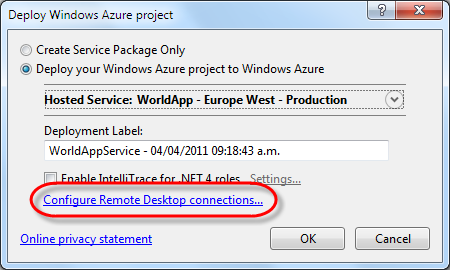
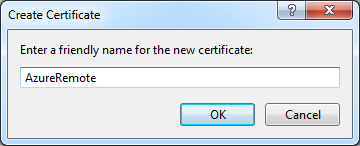
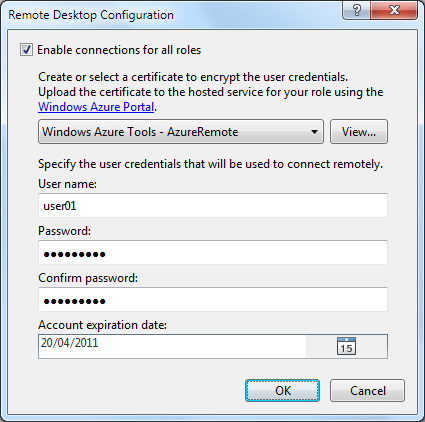
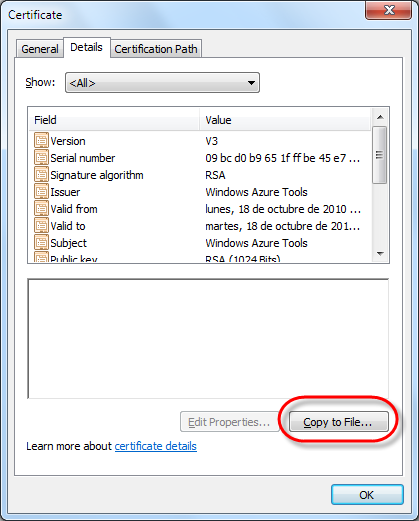
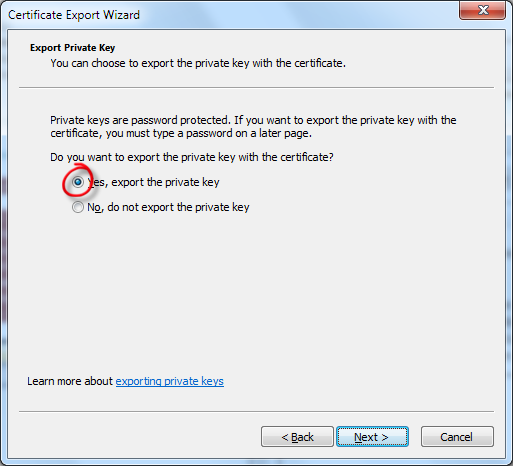
# Exercises

* 1. This hands-on lab includes the following exercises:
  2. Routing Traffic for Enhanced Performance
  3. Balancing Traffic across Hosted Services
  4. Increasing Availability with Failover Policies
  5. Managing Traffic Manager Policies
  6. Estimated time to complete this lab: **60 minutes**.
  7. **Note:** When you first start Visual Studio, you must select one of the predefined settings collections. Every predefined collection is designed to match a particular development style and determines window layouts, editor behavior, IntelliSense code snippets, and dialog box options. The procedures in this lab describe the actions necessary to accomplish a given task in Visual Studio when using the **General Development Settings** collection. If you choose a different settings collection for your development environment, there may be differences in these procedures that you need to take into account.

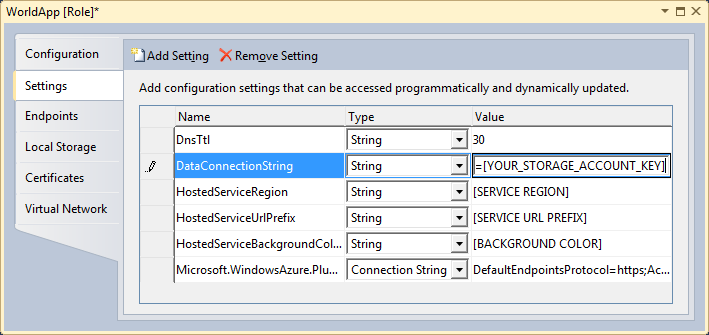
Getting Started: Preparing and Deploying the Application

* 1. During this hands-on lab, you will evaluate Windows Azure Traffic Manager by deploying a sample application to several hosted services. Furthermore, to obtain meaningful results when testing the performance load balancing method, these services should ideally be located in different geographic regions.
  2. Before you begin, you need to configure the application to enable remote desktop. This will allow you to open a remote desktop session to each of your hosted services and test access to the application from other regions.
  3. In this exercise, you deploy the application to multiple hosted services in your Windows Azure subscription. For each hosted service, you configure the application with settings appropriate to its region, create a hosted service in the Windows Azure Management Portal, upload the remote desktop certificate, and then deploy the application.

Task 1 – Preparing for Deployment

* 1. In this task, you prepare the application for deployment by enabling remote desktop access.
  2. Open Visual Studioin elevated administrator mode from **Start | All Programs | Microsoft Visual Studio 2010** by right-clicking the **Microsoft Visual Studio 2010** shortcut and choosing **Run as administrator**.
  3. If the **User Account Control** dialog appears, confirm that you wish to proceed.
  4. In the **File** menu, choose **Open** and then **Project/Solution**. In the **Open Project** dialog, browse to **WorldApp** in the **Source** folder of the lab, select **WorldApp.sln** and then click **Open**.
  5. In **Solution Explorer**, right-click the **WorldAppService** project and select **Publish**.
  6. In the **Deploy Windows Azure project** dialog, click **Configure Remote Desktop connections**.
     1. 
     2. Figure
     3. Opening the Remote Desktop Configuration UI
  7. In the **Remote Desktop Configuration** dialog, check the option labeled **Enable connections for all roles**.
  8. Expand the drop down list labeled **Create or select a certificate to encrypt the user credentials** and select **Create**.
     1. **Note:** If you have previously created a certificate for encrypting remote desktop user credentials, you may choose it instead.
  9. In the **Create Certificate** dialog, enter a name to identify the certificate, for example, *AzureRemote*, and then click **OK**.
     1. 
     2. Figure
     3. Creating a certificate for Remote Desktop connections
  10. Now, back in the **Remote Desktop Configuration** dialog, choose the newly created certificate from the drop down list, enter the name of the user that you will use to connect remotely to your role–this can be any name of your choice–enter a password and confirm it, and leave the account expiration date unchanged.
      1. 
      2. Figure
      3. Configuring Remote Desktop settings
  11. Before you close the dialog, click View next to the certificate drop down list. In the **Certificate** dialog, switch to the **Details** tab and click **Copy to File**.
      1. 
      2. Figure
      3. Exporting the remote desktop certificate
  12. Follow the wizard to export the certificate to a file making sure that you **choose the option to export the private key**. Save the resulting file to a suitable location in your hard disk. You will need to upload this file to the Management Portal shortly, once you create a hosted service for your application.
      1. 
      2. Figure
      3. Exporting the certificate’s private key
  13. Click **OK** to close the **Remote Desktop Configuration** dialog.

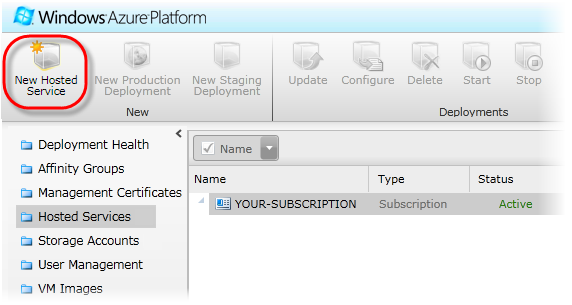
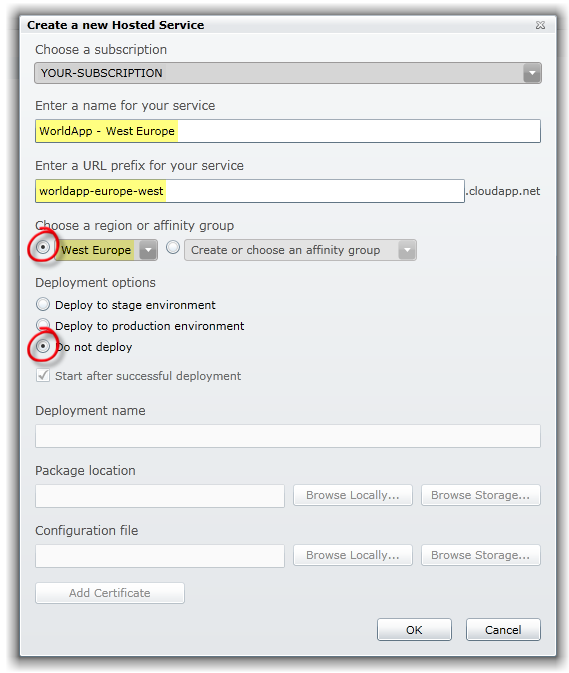
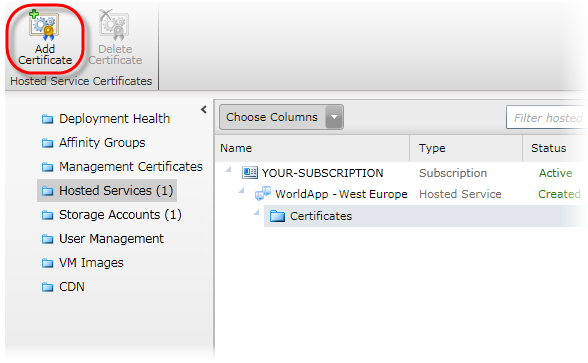
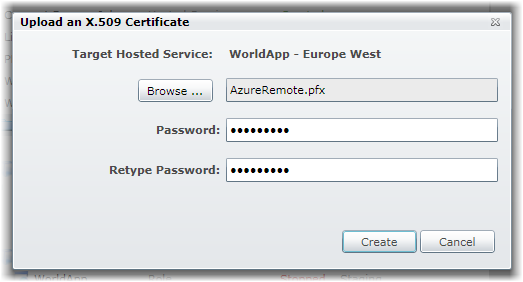
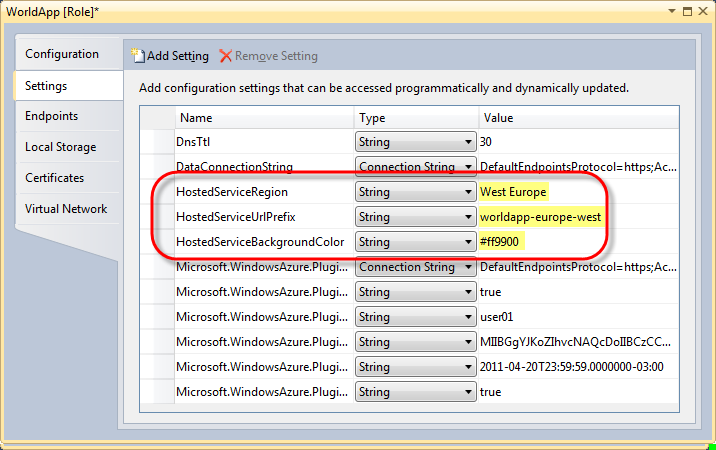
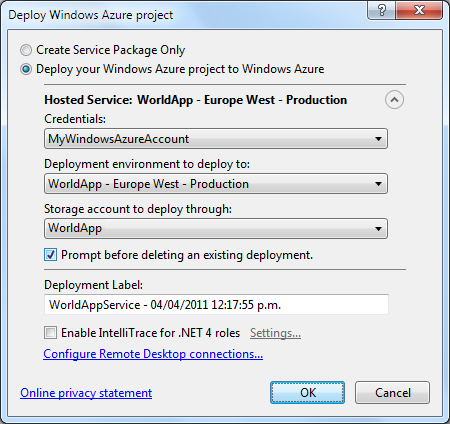
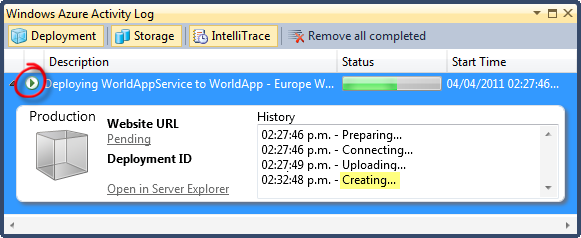
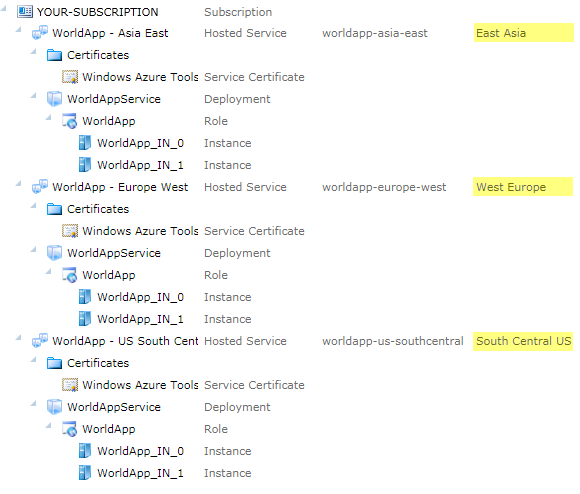
Task 2 – Configuring Storage Account Credentials

* 1. In this task, you configure the credentials used by the application to access your Windows Azure Storage account.
  2. **Note:** All the deployments used in this hands-on lab must share the same *DataConnectionString* setting.
  3. In **Solution Explorer**, expand the **Roles** node in the **WorldAppService** project, and then double-click the **WorldApp** role to open its properties window.
  4. In the **WorldApp [role]** properties window, select the **Settings** tab, and then locate the **DataConnectionString** setting. Replace the placeholder labeled *[YOUR\_STORAGE\_ACCOUNT\_NAME]* with the name of your storage account and *[YOUR\_STORAGE\_ACCOUNT\_KEY]* with its key.
     1. 
     2. Figure
     3. Configuring storage account credentials in Visual Studio
  5. Repeat the previous step to configure the storage credentials for **Microsoft.WindowsAzure.Plugins.Diagnostics.ConnectionString** setting. This is the storage account used for diagnostics.

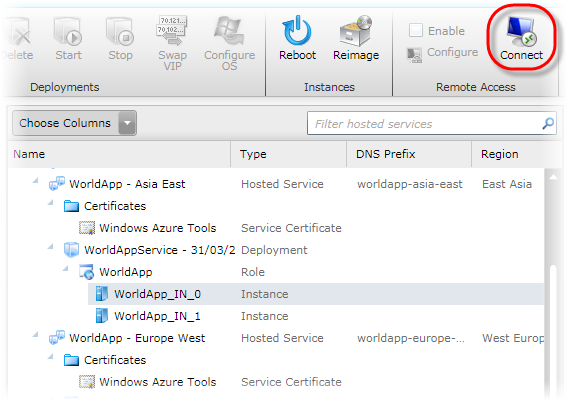
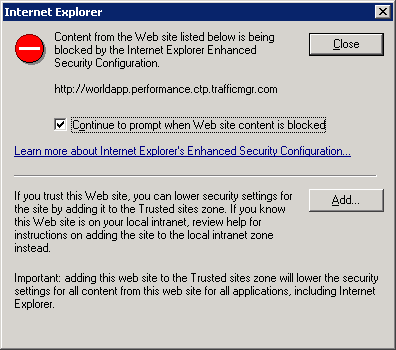
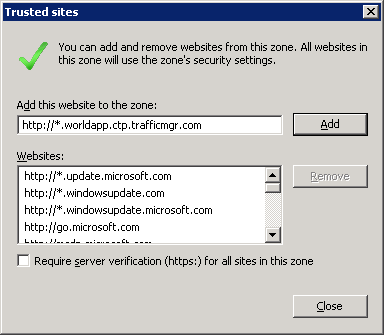
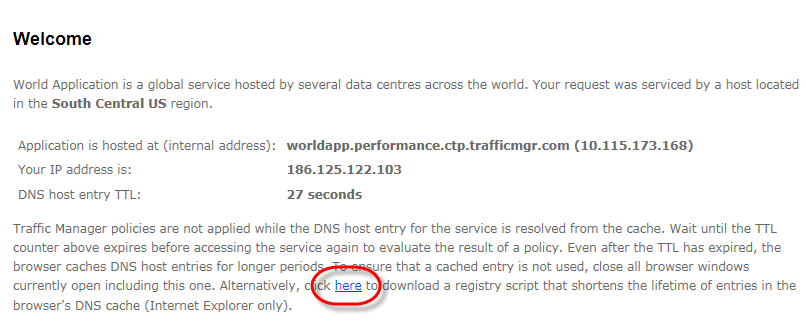
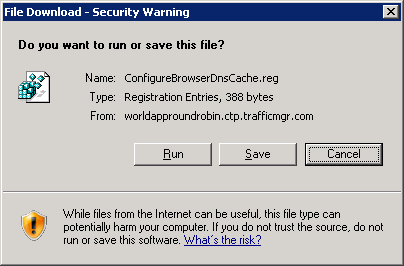
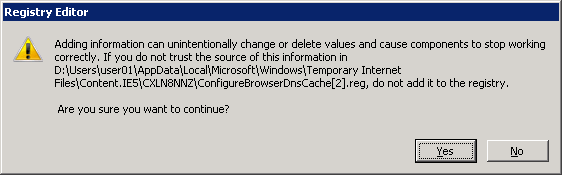
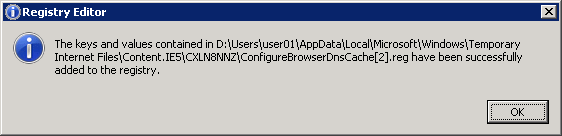
Task 3 – Deploying the Application to Multiple Regions

* 1. In this task, you create the necessary hosted services in your Windows Azure subscription and then deploy the application to different regions. You need to create at least two hosted services to be able to complete the steps in this lab, but three is the recommended number.
  2. Before deploying each hosted service, you update its configuration to specify a number of settings that will allow you to identify which hosted service responds to a given request.
  3. The following procedure describes the steps required to create a hosted service and deploy the application to it. You need to repeat this procedure for each hosted service that you deploy. The table below suggests values for the configuration settings in each deployment, but you may choose different values if desired.

|  |  |  |  |
| --- | --- | --- | --- |
| Service Name | URL prefix | Region | Background |
| [AppName] – West Europe | [appname]-europe-west | West Europe | *#ff9900* |
| [AppName] – South Central US | [appname]-us-southcentral | South Central US | *#5c87b2* |
| [AppName] – East Asia | [appname]-asia-east | East Asia | *#99ff66* |

* 1. The pattern followed by the hosted service URL prefix is ***[appname]-[region]***, where ***[appname]*** is a unique identifier common to all the hosted services that you deploy during this hands-on lab and ***[region]*** is the Windows Azure region where the service is created. The full URL of a hosted service is based on its URL prefix and public so you must choose the ***[appname]*** portion to avoid colliding with hosted services from other users.
  2. To create a hosted service and deploy the application:
  3. Navigate to the [Windows Azure Management Portal](http://windows.azure.com/) and sign in using your Windows Live ID.
  4. In the navigation pane, select the **Hosted Services, Storage Accounts & CDN** tab followed by **Hosted Services**, and then click **New Hosted Service** on the ribbon.
     1. 
     2. Figure
     3. Creating a new hosted service at the Windows Azure Portal
  5. In the **Create a new Hosted Service** dialog, if you have multiple subscriptions, pick the subscription where you will create the hosted services for this hands-on lab.
  6. Next, enter a **Service Name** and the **URL Prefix** for the service. Refer to the table at the start of this task for suggested values. The dialog validates the URL prefix as you type it and warns you if the one you choose is unavailable.
  7. To choose a region for the service, expand the drop down list labeled **Choose a region** and pick the region that corresponds to the service name and URL prefix used.
  8. Finally, in the **Deploy** pane, select the option labeled **Do not deploy**, and then click **OK**.
     1. 
     2. Figure
     3. Creating a new hosted service
  9. Upload the certificate used to encrypt the Remote Desktop password to the newly created service. To do this, expand the node for your hosted service to display and select the **Certificates** node and then click **Add Certificate** on the ribbon.
     1. 
     2. Figure
     3. Configuring hosted service certificates
  10. In the **Upload an X.509 Certificate** dialog, click **Browse** and navigate to the location where you stored the certificate that you created and exported during the previous task when you configured remote desktop, enter the assigned password, confirm it, and then click **Create**.
      1. 
      2. Figure
      3. Uploading the Remote Desktop certificate to the service
  11. Return to Visual Studio and update the configuration settings used to identify the region where the service is located. To do this, in **Solution Explorer**, expand the **Roles** node of the **WorldAppService** project, double-click **WorldApp** to open the properties window for this role, and then select the **Settings** tab.
      1. **Note:** The application includes several configuration settings that you update based on the Windows Azure region where you deploy the service, namely its region label, its URL prefix, and the background color for the UI. The latter setting allows you to identify which hosted service responds to a given request very easily.
  12. In the **Settings** page, set the *HostedServiceRegion*, *HostedServiceUrlPrefix*, and *HostedServiceBackgroundColor* settings to the values in the table at the start of this task that match the region chosen for this service.
      1. 
      2. Figure
      3. Configuring the application based on its Windows Azure deployment region
  13. Once you configure the application, you can proceed to deploy it. To do this, right-click the **WorldAppService** project in **Solution Explorer** and then select **Publish**.
  14. In the **Deploy Windows Azure project** dialog, select the credentials for your Windows Azure subscription from the drop down list, select the **Production** slot for the hosted service created earlier in this task, and choose a storage account to upload the deployment package.
      1. **Note:** Traffic Manager policies only apply to hosted services in the production environment. You cannot route to services running in the staging environment.
  15. Next, enter a new label for the deployment or accept the one suggested in the dialog. Finally, click **OK** to begin the deployment process.
      1. 
      2. Figure
      3. Deploying the application to Windows Azure
      4. **Note:** If you have not previously deployed an application to Windows Azure from Visual Studio, you first need to create and configure the necessary credentials. For more information, see Appendix A: Configuring your Windows Azure Management Portal Credentials in Visual Studio.
  16. After you start a deployment, you can examine the Windows Azure activity log window to determine the status of the operation. If this window is not visible, in the **View** menu, point to **Other Windows**, and then select **Windows Azure Activity Log**. By default, the log shows a descriptive message and a progress bar to indicate the status of the deployment operation. To view detailed information about the operation in progress, click the green arrow on the left side of the activity log entry.
      1. 
      2. Figure
      3. Monitoring the progress of the deployment operation
  17. Repeat the procedure described in this task to deploy the service to other regions referring to the previous steps for a detailed description. These steps have been summarized below:
      1. In the [Windows Azure Management Portal](http://windows.azure.com/), create a hosted service in a different region than the one chosen previously.
      2. Upload the certificate used to encrypt the remote desktop credentials. Use the same certificate that you created previously.
      3. Before proceeding, examine the **Windows Azure Activity Log** to ensure that the previous deployment has completed its “*Uploading…”* phase. This is a precaution to avoid overwriting the service configuration file before it has been completely uploaded.
      4. Next, open the **Settings** page for the **WorldApp** role and update the *HostedServiceRegion*, *HostedServiceUrlPrefix*, and *HostedServiceBackgroundColor* settings to reflect your choices when creating the current hosted service. Refer to the table at the start of this task for suggested configuration settings.
      5. Publish the project to the hosted service.
  18. After you complete this task, the **Hosted Services** view in the Windows Azure Management Portal should list the hosted services for all the regions that you deployed. You should have at least two hosted services in different regions. Make sure that the portal shows the status of all these services as ***Ready***.
      1. 
      2. Figure
      3. Management Portal showing multiple hosted services created in different regions
      4. **Note:** To display the **Region** and **DNS Prefix** in the middle pane of the **Hosted Services** view, choose the corresponding columns in the drop down list labeled **Choose Columns** shown above the hosted service tree view.

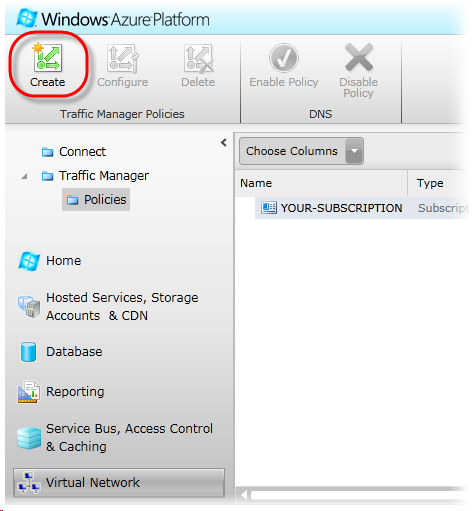
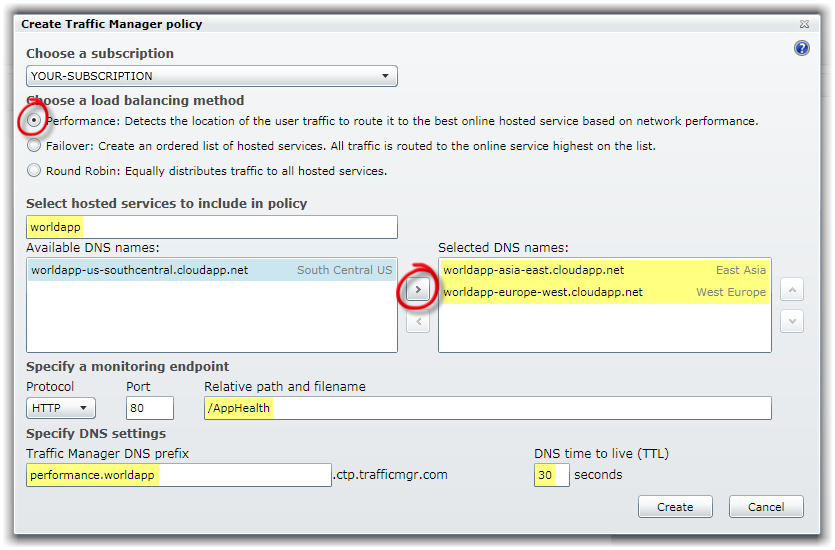
Task 4 – Configuring the DNS Cache for Testing

* 1. In the remainder of this lab, you will open remote desktop sessions to your hosted services and test the application from inside each session. As was described previously, for these tests to be effective the browser in each case needs to be configured by running a registry script that shortens the lifetime of DNS cache entries.
  2. In this task, you open a remote desktop session to a role instance in each of the hosted services and configure the browser suitably.
  3. In the Management Portal, locate the hosted service for the desired region and expand its node to show all its instances. Select the first instance, that is to say *[RoleName]\_IN\_0*, and then click **Connect** on the ribbon to open a remote desktop session to that instance.
     1. 
     2. Figure
     3. Establishing a remote desktop session to a role instance
  4. Inside the remote desktop session, open a browser window and navigate to the URL of the hosted service in the same region, namely *http://[appname].[region].cloudapp.net*, *[appname].[region]* is the DNS prefix name that you configured for the hosted service.
  5. If access to the site is blocked by Internet Explorer’s enhanced security configuration, click **Add**.
     1. 
     2. Figure
     3. Internet Explorer enhanced security configuration warning
  6. In the **Trusted Sites** dialog, click **Add** to insert the URL that you typed in the browser’s address bar to the trusted websites list.
  7. Before you close the **Trusted Sites** dialog, also add the URL for the Traffic Manager’s domain to the list of trusted websites. To do this, type *http://\*.[appname].ctp.trafficmgr.com,* where *[appname]* is the prefix chosen for your application and then click **Add**.
     1. 
     2. Figure
     3. Adding the Traffic Manager to the trusted sites zone
  8. Click **Close** twice to dismiss the enhanced security configuration dialogs and display the home page of the application.
  9. Now, from the home page, download the registry script to configure the browser and shorten the lifetime of host entries in its DNS cache.
     1. 
     2. Figure
     3. Downloading the browser DNS cache configuration script
  10. In the **File Download** dialog, click **Run** to launch the registry configuration script.
      1. 
      2. Figure
      3. Configuring the DNS cache of the browser
  11. When prompted by the Registry Editor, click **Yes** to confirm that you wish to execute the script.
      1. 
      2. Figure
      3. Registry editor security warning
  12. Click **OK** to close the message box confirming the successful update of the registry.
      1. 
      2. Figure
      3. Successful update of the browser’s configuration
  13. Close all open browser windows and restart the browser for the changes to take effect.
  14. Repeat the previous steps until you have configured the browser in each of your hosted services, one for each region.
      1. **Note:** Each hosted service is deployed with 2 instances. Make sure that you always connect to the same instance, for example instance 0, so that you execute your tests in a previously configured session.

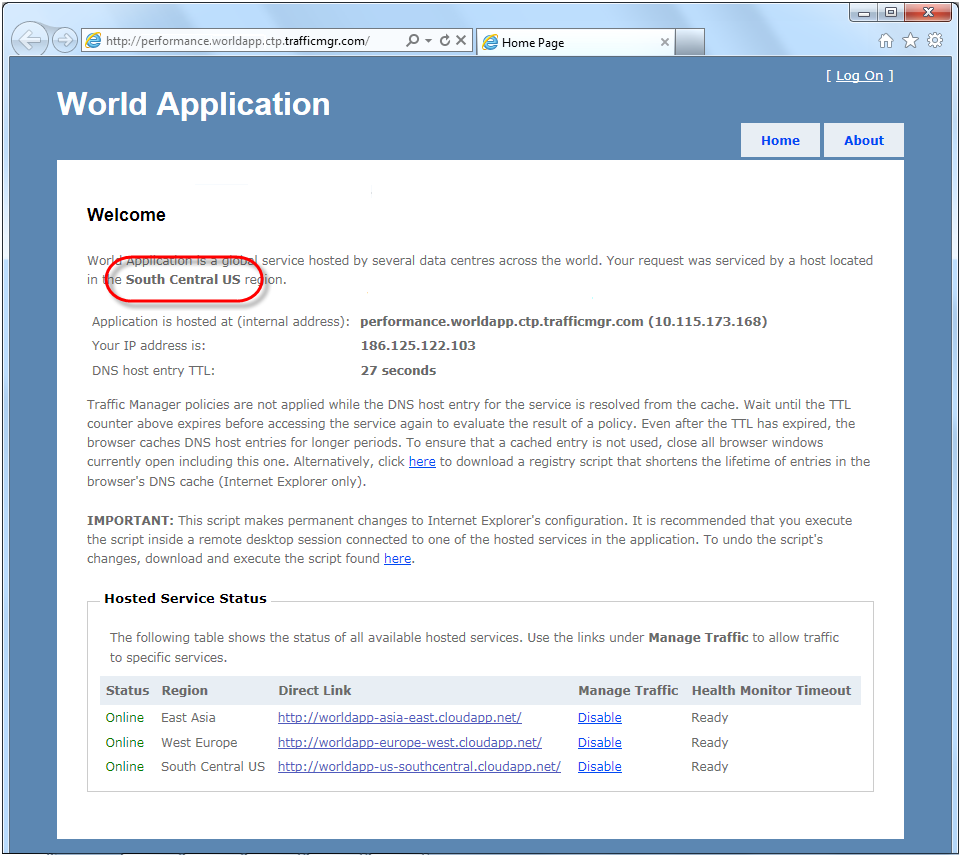
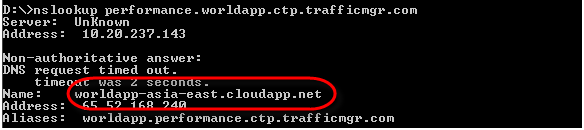
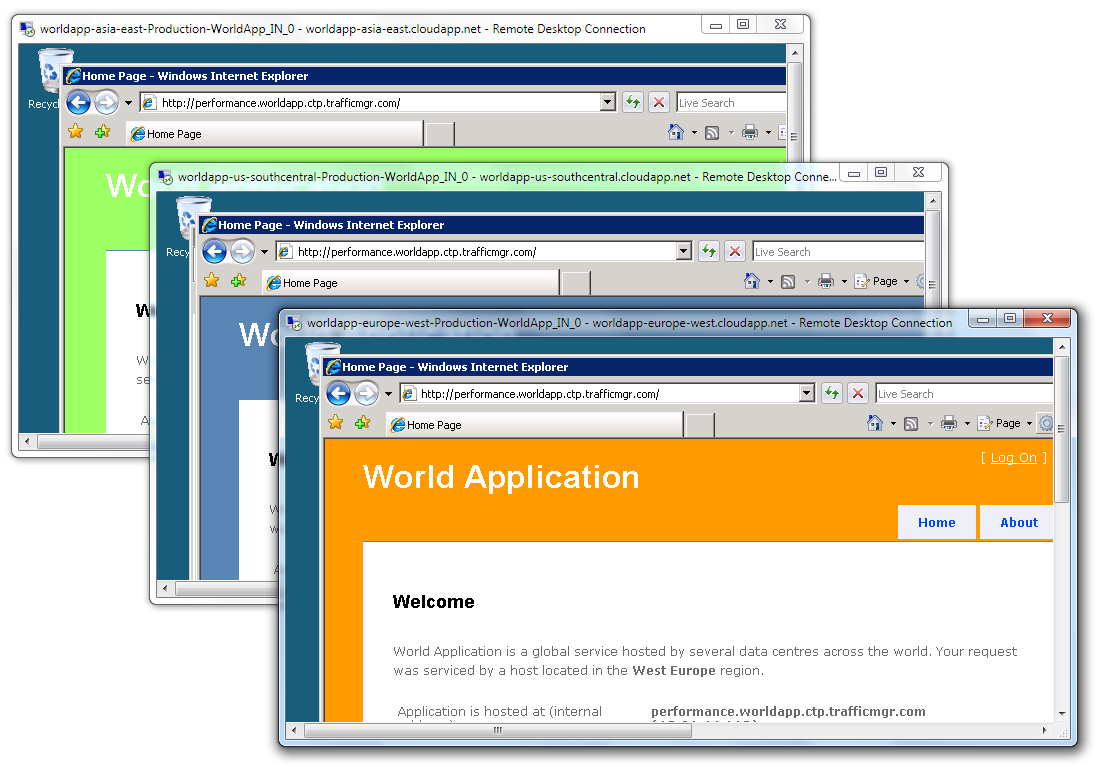
Exercise 1: Routing Traffic for Enhanced Performance

* 1. Traffic Manager maintains a network performance table that it updates periodically and contains the round trip time between various IP addresses around the world and each Windows Azure data center. For the performance load balancing method, Traffic Manager forwards requests to the closest hosted service in terms of its network latency.
  2. Fully evaluating the results of this policy requires accessing the application from different geographic locations. Fortunately, this does not require you to travel around the world to complete this exercise. Instead, because you are deploying the application to multiple hosted services in various regions, you can take advantage of remote desktop to log into a hosted service in each of these regions and then access the application from within the remote session. In this case, the role instance acts as both the client and the server.

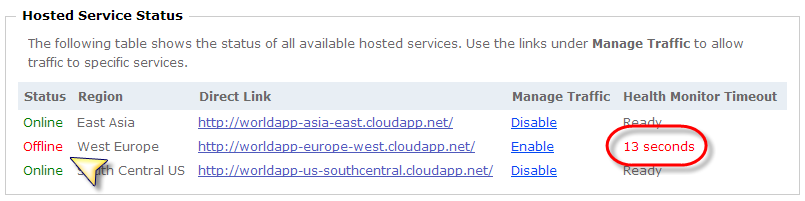
Task 1 – Creating a Traffic Manager Performance Policy

* 1. In this task, you define a Traffic Manager policy that maximizes performance by forwarding traffic to the hosted service that offers the best performance for any given client. The load balancer bases its decision on performance tables that measure the round trip time of different IP addresses around the globe to each Windows Azure data center.
  2. In the Windows Azure Management Portal, select the **Virtual Network** tab. Then, choose the **Policies** option under **Traffic Manager** and click **Create** on the ribbon.
     1. 
     2. Figure
     3. Creating a Windows Azure Traffic Manager policy
  3. In the **Create Traffic Manager Policy** dialog, pick the subscription where you created the hosted services for this hands-on lab.
  4. Specify the **load balancing method** as **Performance**.
  5. Assign the **Hosted services to include in the policy** from the list of **Available DNS names**. Add all the hosted services created for this hands-on lab. To add a service to the policy, select it and then click the arrow button in the middle.
     1. **Note:** For subscriptions that contain a large number of hosted services, you can filter the list by typing the **[appname]** portion of the URL prefix in the text box shown above the list of DNS names.
  6. Set the **Relative path and filename** of the monitoring endpoint to */AppHealth*.
     1. **Note:** Traffic Manager performs an HTTP (GET) request against the monitoring endpoint every 30 seconds to determine the service’s health. The service must respond with a 200 OK HTTP status code within 5 seconds; otherwise, Traffic Manager considers the service unhealthy and removes it from the load balancer rotation.
     2. */AppHealth* is the URL of the monitoring endpoint for the lab’s application that can be disabled on demand to simulate a failure.
  7. Next, choose a name for the **Traffic Manager DNS prefix**, for example, *performance.[appname]*, where *[appname]* is the name chosen for your service.
     1. **Note:** Traffic Manager uses the DNS prefix to build the URL for the service. Client requests to this URL are subject to the policy defined in Traffic Manager.
  8. To complete the policy definition, set the DNS time-to-live (TTL) to *30* seconds.
     1. **Note:** The TTL determines for how long clients and secondary DNS servers cache a DNS host entry. Clients will continue to use a given hosted service until its entry in the cache expires. For this hands-on lab, the TTL is set to the lowest possible value to allow policy results and changes in the status of services to be seen as early as possible. Note, however, that lowering this value increases DNS traffic and that you should consider keeping its default value for your production services.
     2. 
     3. Figure
     4. Creating a Traffic Manager policy for load balancing based on network performance
  9. Click **Create** to save the policy definition.
  10. **Note:** After creating a policy, wait for 2 minutes before you run any tests to allow the policy to propagate to all DNS servers.

Task 2 – Testing the Performance Policy

1. In this task, you test the performance policy defined previously. First, you test the policy from your own computer. Next, to observe the behavior of the policy when accessing the application from different geographic regions, you establish remote desktop sessions to every region where you deployed a hosted service. Inside each remote session, you browse the application to initiate a request originating from the same region as the hosted service.
   1. Open a browser window and navigate to *http://performance.[appname].ctp.trafficmgr.com/*, where *performance.[appname]* is the DNS prefix name that you configured for the performance policy created in the previous task. Notice the distinctive background color and the label in the first paragraph indicating the location of the hosted service that replies to the request. Verify that the response originated from the service closest to your current location.
      1. 
      2. Figure
      3. Testing the Traffic Manager performance policy in your own machine
      4. **Note:** For a performance policy, the load balancer determines which hosted service responds to a client request based on tables that record the round trip time between various IP addresses around the globe and each Windows Azure data center. Note, however, that while there is a strong correlation between distance and network latency and you would normally receive a response from the hosted service closest to your current location, other factors such as network topology and congestion could determine that you receive a response from a service that is further away.
   2. Now, in the navigation pane of the Windows Azure Management Portal, select the **Hosted Services, Storage Accounts & CDN** tab and then the **Hosted Services** option.
   3. Locate one of the hosted services created for this hands-on lab and expand its node to show its instances. Select one of the instances and then click **Connect** on the ribbon to open a remote desktop session to that instance.
   4. Inside the remote desktop session, open a browser window and navigate to *http://performance.[appname].ctp.trafficmgr.com/*, where *performance.[appname]* is the DNS prefix name that you configured for the performance policy. Verify that the response originates from the hosted service in the same region as the remote desktop session.
      1. **Note:** Ensure that you connect to one of the role instances where you previously configured the DNS cache for the browser. Otherwise, click the link to download and run the registry configuration script and then restart the browser. After that, repeat the current step.
   5. Open a command prompt window and type the following command:
      1. Command Prompt
      2. **nslookup http://performance.[appname].ctp.trafficmgr.com/**
      3. where *performance .[appname]* is the DNS prefix name that you configured for the performance policy.
      4. Notice the **Name** and **Address** returned by the DNS query, which should match that of the hosted service nearest to your current location.
      5. 
      6. Figure
      7. Using nslookup to determine the address returned by the performance policy
   6. Next, open a remote desktop session to each of the other regions and repeat the previous steps to access the application from that region. Verify that in all cases, the response originates from the service hosted in the region where the remote desktop session is established.
      1. 
      2. Figure
      3. Testing the performance policy from different geographic regions
   7. Do not close the remote desktop sessions. You will need them for the following exercises.

Task 3 – Simulating a Service Failure

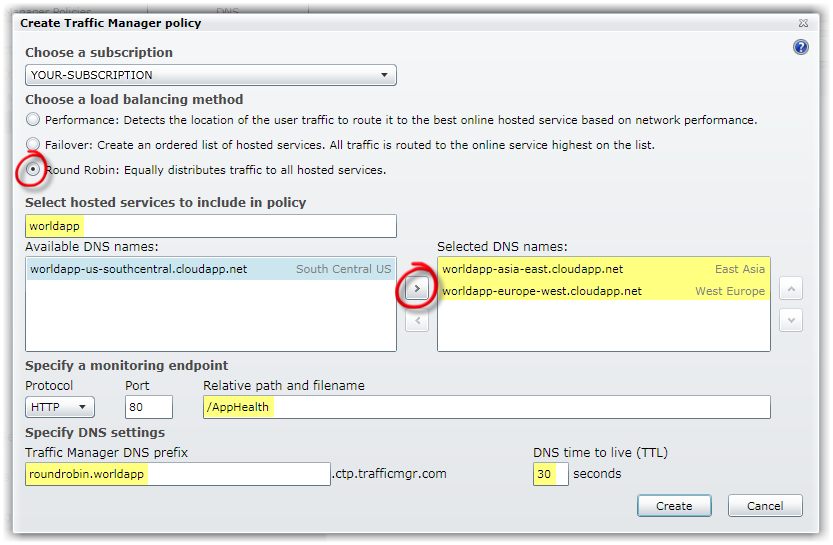
* 1. In this task, you place the hosted service that is currently servicing your requests in an offline state to simulate a failure.
  2. In the remote desktop session, navigate to *http://performance.[appname].ctp.trafficmgr.com/* and determine the region of the hosted service that responds to the request.
  3. Now, locate this region in the **Hosted Service Status** table and click **Disable** to set its state to *Offline*. Notice that the Health Monitor Timeout column starts a timer that shows the elapsed time since the state of the service changed.
     1. 
     2. Figure
     3. Simulating the failure of a hosted service
     4. **Note:** The Health Monitor Timeout provides an estimate of how long it takes Traffic Manager to become aware of the change.
     5. When a hosted service is disabled, its monitoring endpoint stops sending responses to simulate a failure. Traffic Manager performs a check of this endpoint at 30-second intervals and if it fails to receive a response to three consecutive polls, it considers the service as unavailable. Thus, it could take as much as 120 seconds for the service to failover.
     6. After you disable a service, a timer on the page starts showing the elapsed time since the status of the service changed, providing an estimate of how long it takes the Traffic Manager to become aware of the failure.
  4. Wait until the TTL expires and the **Health Monitor Timeout** shows a *Ready* status.
  5. Now, refresh the page and verify that the response now originates from a service located in a different region, presumably, the second closest region in terms of network latency.
  6. Re-enable traffic to the previously disabled region and wait until the **Health Monitor Timeout** indicates that the change should be effective.
     1. **Note:** When a service comes back online, Traffic Manager detects the change in its status within the next polling interval. Thus, the interval shown by the Health Monitor Timeout when switching from offline to online is only 30 seconds.
  7. Refresh the page and confirm that the response is once again from the original hosted service.

Exercise 2: Balancing Traffic across Hosted Services

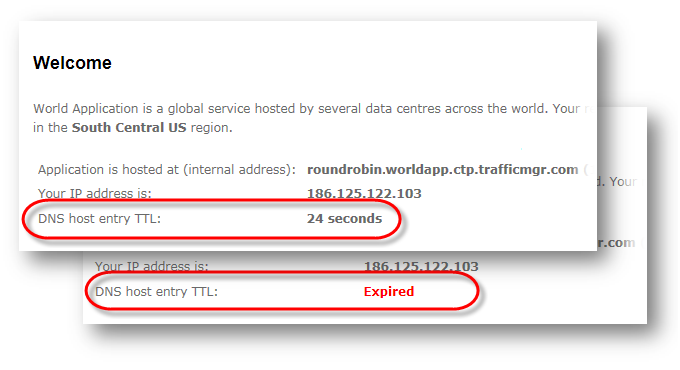
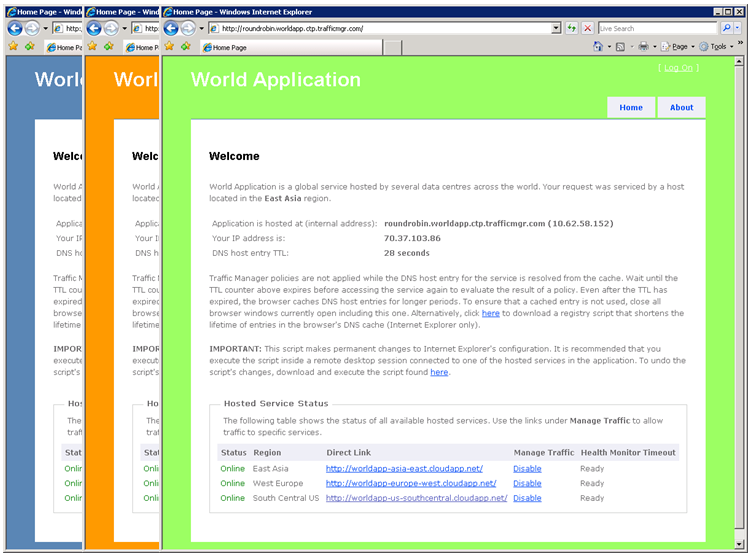
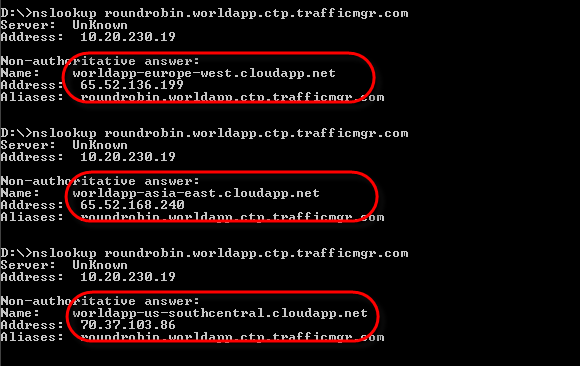
The round robin load balancing method distributes load evenly among each of the hosted services assigned to the policy. It keeps track of the last hosted service that received traffic and sends traffic to the next one in the list of hosted services.

Traffic Manager removes a hosted service from the load balancer’s rotation if it determines that it is offline. Note, however, that if all the services assigned to a policy are unavailable, Traffic Manager will ignore their status and return a response as if they were online.

Task 1 – Creating a Traffic Manager Round Robin Policy

* 1. In this task, you create a Traffic Manager policy that balances traffic evenly across all hosted services assigned to the policy.
  2. In the Windows Azure Management Portal, select the **Virtual Network** tab. Then, choose the **Policies** option under **Traffic Manager** and click **Create** on the ribbon.
  3. In the **Create Traffic Manager Policy** dialog, pick the subscription where you created the hosted services for this hands-on lab.
  4. Specify the **load balancing method** as **Round Robin**.
  5. Assign the **Hosted services to include in the policy** from the list of **Available DNS names**. Add all the hosted services created for this hands-on lab. To add a service to the policy, select it and then click the arrow button in the middle.
     1. **Note:** For subscriptions that contain a large number of hosted services, you can filter the list by typing the **[appname]** portion of the URL prefix in the text box shown above the list of DNS names.
  6. Set the **Relative path and filename** of the monitoring endpoint to */AppHealth*.
  7. Next, choose a name for the **Traffic Manager DNS prefix**, for example, *roundrobin.[appname]*, where *[appname]* is the name chosen for your service.
  8. Now, set the DNS time to live (TTL) to *30* seconds.
     1. 
     2. Figure
     3. Creating a Traffic Manager policy for load balancing traffic equally across hosted services
  9. Click **Create** to save the policy definition.
  10. **Note:** After creating a policy, wait for 2 minutes before you run any tests to allow the policy to propagate to all DNS servers.

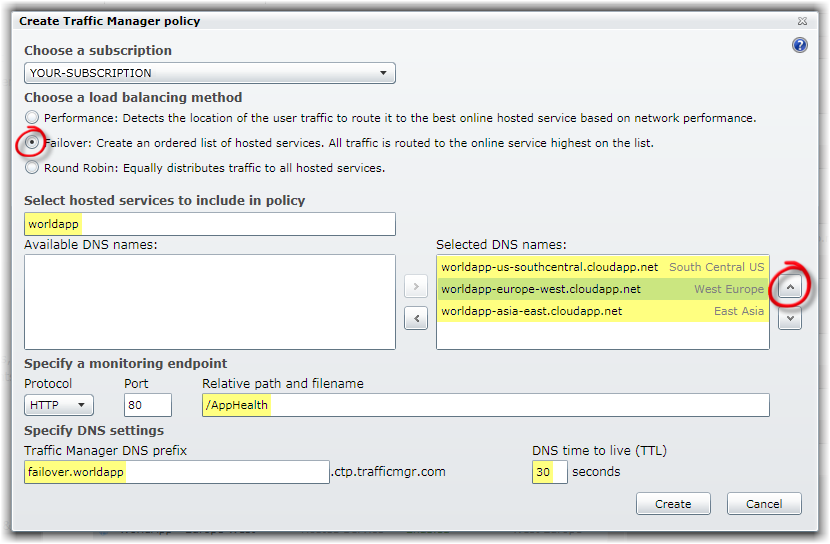
Task 2 – Testing the Round Robin Policy

* 1. In this task, you test the round robin policy created during the previous task.
  2. In a remote desktop session to one of your hosted services, open a browser window and navigate to *http://roundrobin.[appname].ctp.trafficmgr.com/*, where *roundrobin.[appname]* is the DNS prefix name that you configured for the round robin policy.
     1. **Note:** Ensure that you connect to one of the role instances where you previously configured the DNS cache for the browser. Otherwise, click the link to download and run the registry configuration script and then restart the browser. After that, repeat the current step.
  3. Take note of the service that responds to the current request.
  4. Wait for at least 30 seconds, which is the TTL configured for the current policy. For your convenience, a timer on the page provides a visual estimate of the remaining time.
     1. 
     2. Figure
     3. DNS host entry TTL
  5. Once the TTL expires, refresh the page and notice that the background color of the page changes indicating that the response now originates from a hosted service located in a different region.
  6. Wait until the TTL expires again and refresh the page one more time to verify that origin of the response has changed yet again.
     1. 
     2. Figure
     3. Testing the round robin performance policy
  7. Open a command prompt window and execute the following command. Take note of the name and address returned.
     1. Command Prompt
     2. **nslookup roundrobin.[appname].ctp.trafficmgr.com**
  8. Repeat the command several times while waiting at least 30 seconds between attempts to allow the TTL to expire. Verify that the returned host cycles between each of the hosted services assigned to the round robin policy.
     1. 
     2. Figure
     3. Using nslookup to evaluate the round robin policy

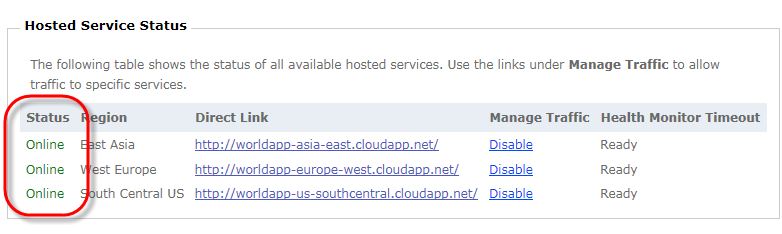
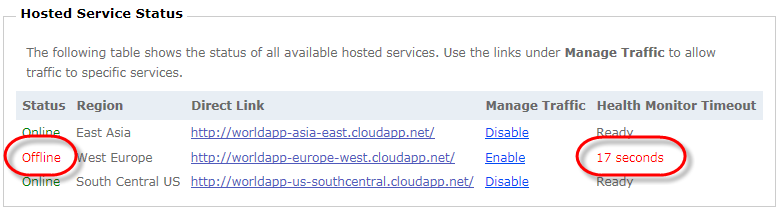
Exercise 3: Increasing Availability with Failover Policies

* 1. When using a failover policy, if the primary hosted service is offline, traffic is sent to the next one in a sequence defined by the policy. To test this policy, the application provided with the lab includes a monitoring endpoint that Traffic Manager polls to determine whether the hosted service is available.

Task 1 – Creating a Traffic Manager Failover Policy

* 1. In this task, you create a Traffic Manager policy that contains an ordered list of hosted services. Traffic is normally forwarded to the first service in this list—the primary service. If a service becomes unavailable, the load balancer switches to the next service in the list, and if that fails too, it will continue with the next in order.
  2. In the Windows Azure Management Portal, select the **Virtual Network** tab. Then, choose the **Policies** option under **Traffic Manager** and click **Create** on the ribbon.
  3. In the **Create Traffic Manager Policy** dialog, pick the subscription where you created the hosted services for this hands-on lab.
  4. Specify the **load balancing method** as **Failover**.
  5. Assign the **Hosted services to include in the policy** from the list of **Available DNS names**. Add all the hosted services created for this hands-on lab. To add a service to the policy, select it and then click the arrow button in the middle.
     1. **Note:** For subscriptions that contain a large number of hosted services, you can filter the list by typing the **[appname]** portion of the URL prefix in the text box shown above the list of DNS names.
  6. Set the **Relative path and filename** of the monitoring endpoint to */AppHealth*.
  7. Next, choose a name for the **Traffic Manager DNS prefix**, for example, *failover.[appname]*, where *[appname]* is the name chosen for your service.
  8. Now, set the DNS time to live (TTL) to *30* seconds.
  9. Unlike the performance and round robin policies, where order does not matter, the load balancer chooses an active service based on its position in the list of selected DNS names. Choose the hosted service that will act as the primary and then move it to the top of the list. Arrange the remaining (standby) services in the order in which they should be activated in case the preceding services in the list fail. To change the order of a hosted service, select its DNS name in the list and then click the up arrow button until the service is in the desired position.
     1. 
     2. Figure
     3. Creating a failover Traffic Manager policy
  10. Click **Create** to save the policy definition.
  11. **Note:** After creating a policy, wait for 2 minutes before you run any tests to allow the policy to propagate to all DNS servers.

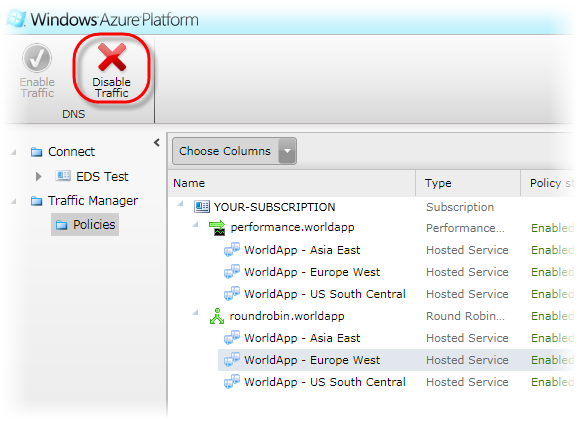
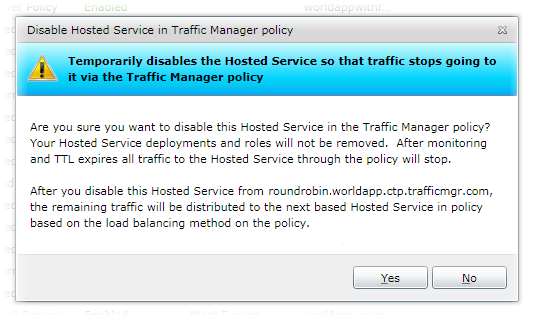
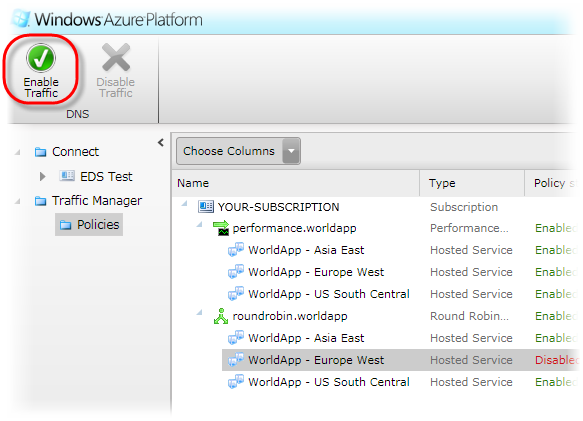
Task 2 – Testing the Failover Policy

* 1. In this task, you test the failover policy created previously.
  2. In a remote desktop session to one of your hosted services, open a browser window and navigate to *http://failover.[appname].ctp.trafficmgr.com/*, where *failover.[appname]* is the DNS prefix name that you configured for the failover policy.
     1. **Note:** Ensure that you connect to one of the role instances where you previously configured the DNS cache for the browser. Otherwise, click the link to download and run the registry configuration script and then restart the browser. After that, repeat the current step.
  3. Verify that the hosted service that responds to the request is the one that you defined as primary by placing it at the top of the list of DNS names when you created the failover policy.
  4. In the home page, the bottom half shows the status of each of your hosted services. Make sure that they are all *Online*.
     1. 
     2. Figure
     3. Examining the status of the hosted services
  5. In the **Hosted Service Status** list, locate the primary service and then click **Disable** under **Manage Traffic**. Notice that the status of the service immediately changes to *Offline* and the **Health Monitor Timeout** column begins displaying a timer.
     1. 
     2. Figure
     3. Disabling a hosted service to simulate a failure
     4. **Note:** When a hosted service is disabled, its monitoring endpoint stops sending responses to simulate a failure. Traffic Manager performs a check of this endpoint at 30-second intervals and if it fails to receive a response to three consecutive polls, it considers the service as unavailable. Thus, it could take as much as 120 seconds for the service to failover.
     5. After you disable a service, a timer on the page starts showing the elapsed time since the status of the service changed, providing an estimate of how long it takes the Traffic Manager to become aware of the failure and switch to the next service in the list.
  6. Wait until the **Health Monitor Timeout** expires and its status is shown again as *Ready.* Then, refresh the page and verify that the response now originates from the second hosted service in the policy list.
  7. Continue disabling hosted services following the same order as the policy list until all your hosted services are offline except one. After you disable each service, wait until the **Health Monitor Timeout** expires and then refresh the page. Confirm that the response is always from the next active (online) service in the policy list.
  8. Disable the last remaining service, wait until the **Health Monitor Timeout** expires, and then refresh the page one more time. Notice that when there are no more services available, the load balancer acts as if all the services are online and routes traffic to the first hosted service in the policy list again.
  9. Re-enable each service again, one at a time, testing the policy after each change.
     1. **Note:** When a service comes back online, Traffic Manager detects the change in its status within the next polling interval. Thus, the interval shown by the Health Monitor Timeout when switching from offline to online is only 30 seconds.

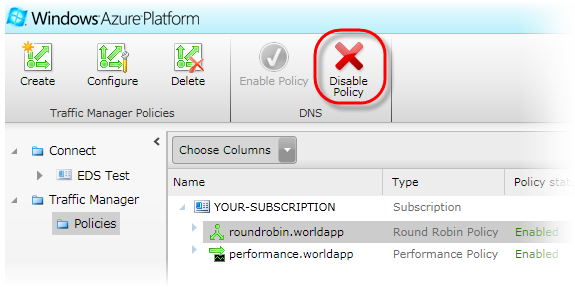
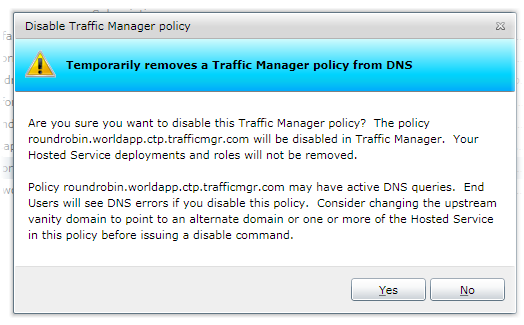
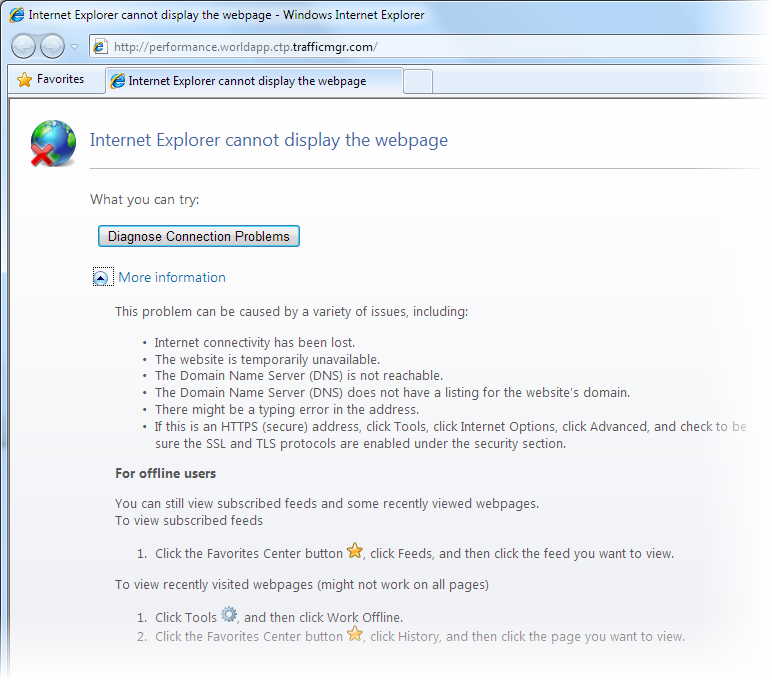
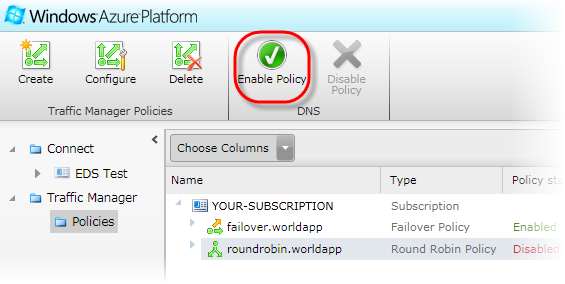
Exercise 4: Managing Traffic Manager Policies

* 1. In this exercise, you learn how to disable one of the hosted services in a policy to remove it from the load balancer rotation. Next, you disable a Traffic Manager policy to prevent routing through its URL.

Task 1 – Controlling Traffic to Individual Hosted Services

* 1. You can disable individual hosted services assigned to a Traffic Manager policy.
  2. In this task, you disable one of the hosted services in the round robin policy and then verify that traffic stops being routed to this service when accessed through this policy.
  3. Open the Traffic Manager interface in the Management Portal by selecting **Virtual Network** followed by the **Policies** option inside **Traffic Manager**.
  4. Locate and expand the round robin policy. Then, select one of the hosted services assigned to this policy and click **Disable Traffic** on the ribbon.
     1. 
     2. Figure
     3. Managing traffic to individual host services in a policy
  5. In the **Disable Hosted Service in Traffic Manager policy** message box, click **Yes** to confirm that you wish to disable traffic to the selected service.
     1. 
     2. Figure
     3. Disabling traffic to a hosted service in a Traffic Manager policy
  6. Now, repeat the steps in you used to test the round robin policy to confirm that the disabled hosted service is no longer included in the load balancer sequence.
     1. **Note:** Disabling a hosted service in a Traffic Manager policy can be useful for temporarily removing a malfunctioning service or during maintenance tasks.
  7. Next, open a remote desktop session to an instance of the hosted service that you disabled in the round robin policy.
  8. In the remote desktop session, open a browser window and then access the service using the endpoint of the performance policy at *http://performance.[appname]. ctp.trafficmgr.com/*. Notice that despite being disabled in the round robin policy, the service is still available when accessed through a different policy.
  9. Re-enable the hosted service in the round robin policy. To do this, in the Management Portal, expand the round robin policy, select the disabled hosted service, and then click **Enable Traffic** on the ribbon.
     1. 
     2. Figure
     3. Re-enabling traffic to a hosted service
  10. Repeat the verification steps to confirm that the previously disabled hosted service is once again included in the load balancer sequence.

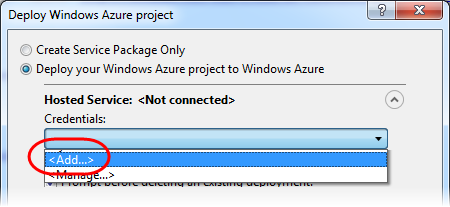
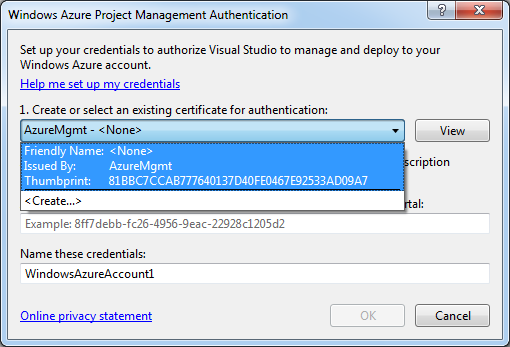
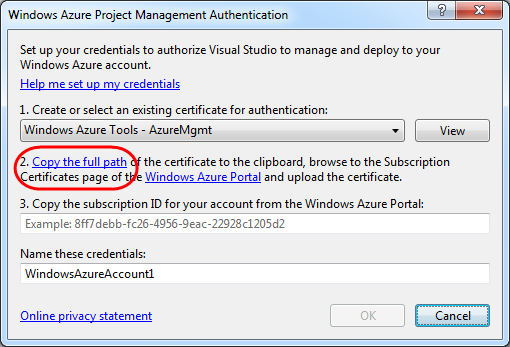
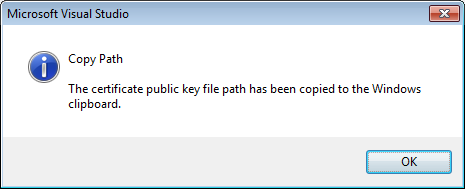
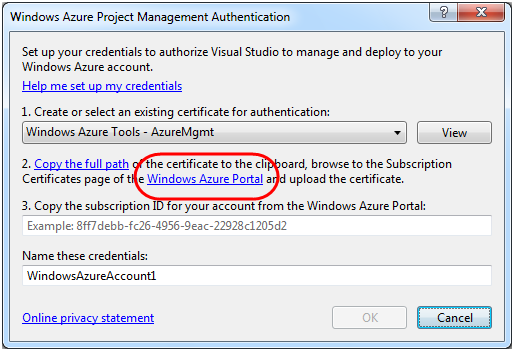
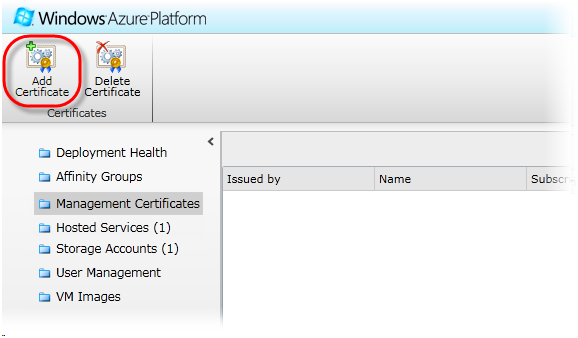
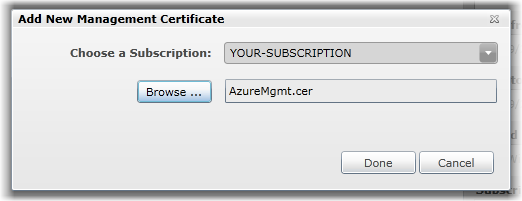
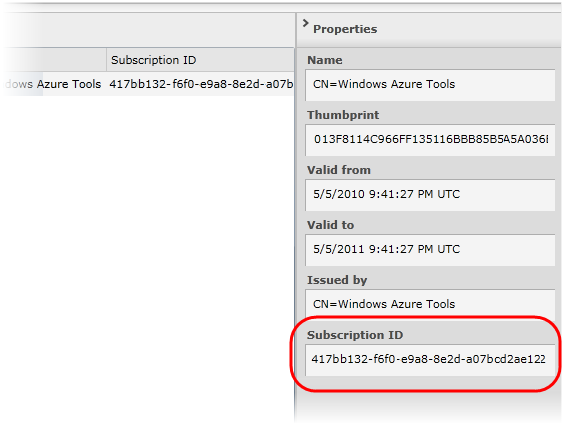
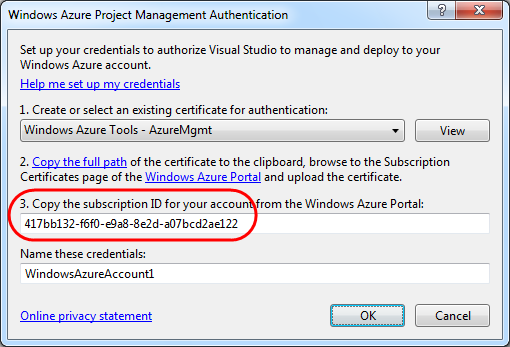
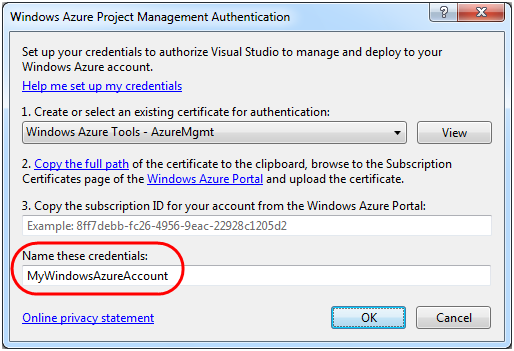
Task 2 – Disabling a Traffic Manager Policy

* 1. You can temporarily disable a Traffic Manager policy to prevent it from routing traffic and then re-enable it again later.
  2. In a remote desktop session to one of your hosted services, open a browser window and navigate to *http://roundrobin.[appname].ctp.trafficmgr.com/*, where *failover.[appname]* is the DNS prefix name that you configured for the failover policy. Confirm that you are able to access the application.
  3. Now, in the Management Portal, open the Traffic Manager interface by selecting **Virtual Network** followed by the **Policies** option inside **Traffic Manager**.
  4. Select the round robin policy and then click **Disable Policy** on the ribbon.
     1. 
     2. Figure
     3. Disabling a Traffic Manager policy
  5. In the **Disable Traffic Manager policy** message box, click **Yes** to confirm that you wish to disable traffic to the specified policy.
     1. 
     2. Figure
     3. Disabling traffic via the URL of a Traffic Manager policy
  6. In the remote desktop session, refresh the page to restart the TTL timer and then wait until it expires to ensure that you are not using a cached DNS entry.
  7. Refresh the page again and notice that the browser cannot display the application because the disabled policy no longer routes traffic with the specified URL.
     1. 
     2. Figure
     3. Browser window showing that the application is no longer accessible through the disabled policy
  8. Re-enable the round robin policy. To do this, in the Management Portal, select the policy and then click **Enable Policy** on the ribbon.
     1. 
     2. Figure
     3. Enabling a Traffic Manager policy
  9. Refresh the page and confirm that the application is once again available at the policy’s URL.
     1. **Note:** It may take up to 2 minutes for the policy to propagate to all DNS servers.

Summary

* 1. Windows Azure Traffic Manager enables you to manage and distribute incoming traffic to your Windows Azure hosted services whether they are deployed in the same data center or in different regions across the world. In this hands-on lab, you explored available load balancing policies and learned how to use them to enhance performance, increase availability, and balance traffic to your hosted services.

Appendix A: Configuring your Windows Azure Management Portal Credentials in Visual Studio

* 1. Follow these steps to save the credentials that allow you to deploy an application to your Windows Azure subscription from Visual Studio. The credentials are stored in your Visual Studio configuration so will only need to do this once for each subscription.
  2. In **Solution Explorer**, right-click the Windows Azure project and select **Publish**.
     1. **Note:** Ensure that you click the Windows Azure cloud project and not one of its associated roles.
  3. In the **Deploy Windows Azure project** dialog, make sure that the option labeled **Deploy your Windows Azure project to Windows Azure** is selected, expand the drop down list labeled **Credentials**, and then select **Add**.
     1. 
     2. Figure
     3. Configuring Windows Azure account credentials
  4. In the **Windows Azure Project Management Authentication** dialog, expand the drop down list below the label **Create or select an existing certificate for authentication** and choose **Create**.
  5. In the **Create Certificate** dialog, set the name to **AzureMgmt** and click **OK**.
  6. Choose the newly created certificate from the drop down list.
     1. 
     2. Figure
     3. Choosing a certificate for authentication
  7. Now, click the link labeled **Copy the full path** to store the path to the certificate file in the clipboard.
     1. 
     2. Figure
     3. Copying the path of the generated certificate public key file to the clipboard
     4. **Note:** Visual Studio stores the public key file for the certificate it generates in a temporary folder inside your local data directory.
  8. Click **OK** to dismiss the confirmation message box and then save the path in the clipboard to a safe location. You will need this value shortly, when you upload the certificate to the portal.
     1. 
     2. Figure
     3. Confirmation that the file path was copied to the clipboard successfully
  9. Now, in the **Windows Azure Project Management Authentication** dialog, click the link labeled **Windows Azure Portal** to open a browser window and navigate to the Management Portal.
     1. 
     2. Figure
     3. Navigating to the Windows Azure Management Portal
  10. At the Management Portal, if you have not already done so, sign in using your Windows Live ID.
  11. Select the **Hosted Services, Storage Accounts & CDN** option, click **Management Certificates**, and then click **Add Certificate** on the ribbon.
      1. 
      2. Figure
      3. Adding a new management certificate to your subscription
  12. In the **Add New Management Certificate** dialog, click **Browse**, change the file name to the path for the public key (.cer) file generated by Visual Studio that you copied earlier to your clipboard, and then click **Open**. Click **Done** to upload the certificate to the Management Portal.
      1. 
      2. Figure
      3. Uploading a management certificate
  13. In the **Properties** pane, make a record of your subscription ID. You will need it for the next step.
      1. 
      2. Figure
      3. Retrieving your subscription ID from the Management Portal
  14. Go back to the **Windows Azure Project Management Authentication** dialog in Visual Studio and paste the value retrieved from the Management Portal into the subscription ID textbox.
      1. 
      2. Figure
      3. Entering your Windows Azure subscription ID
  15. Provide a name to identify your credentials, for example, *MyWindowsAzureAccount*, and then click **OK**.
      1. 
      2. Figure
      3. Saving the Windows Azure credentials

Appendix B: Configuring the DNS Cache of the Browser

The following steps describe how to configure the lifetime of DNS host entries in the browser’s cache using a registry script that you download from the home page of the lab’s application.

This procedure is optional. If you choose not to execute it, you can force the browser to clear its DNS cache by closing all open windows and restarting the browser before each request.

* 1. **Important:** Running this script makes changes to your browser that can have an effect on your browsing experience. **You should only execute this script inside a remote desktop session to one of your hosted services.**
  2. Nevertheless, if you choose to apply the script in your local machine, you can undo its changes by applying a complementary script that is also provided. Note that both scripts assume that you have not previously customized Internet Explorer’s DNS cache configuration. Otherwise, running these scripts could overwrite your original configuration.
  3. For more information, see [How Internet Explorer uses the cache for DNS host entries](http://support.microsoft.com/kb/263558).
  4. You can find both scripts in the **Assets** folder of the lab or download them from the home page of the application.
  5. The scripts are specific to Internet Explorer, so if you use a different browser, you will need to restart the browser before each request to clear its DNS cache.

How to configure the DNS Cache to decrease the lifetime of host entries to 30 seconds

* 1. In the home page of the application, download the registry script to shorten the lifetime of host entries in the browser’s DNS cache.
  2. In the **File Download** dialog, click **Run** to launch the registry configuration script.
  3. When prompted by the Registry Editor, click **Yes** to confirm that you wish to execute the script.
  4. Click **OK** to close the message box confirming the successful update of the registry.
  5. Close all open browser windows and restart the browser for the changes to take effect.

How to restore the DNS Cache configuration

* 1. In the home page of the application, download the registry script to restore the browser’s DNS cache configuration.
  2. In the **File Download** dialog, click **Run** to launch the registry configuration script.
  3. When prompted by the Registry Editor, click **Yes** to confirm that you wish to execute the script.
  4. Click **OK** to close the message box confirming the successful update of the registry.
  5. Close all open browser windows and restart the browser for the changes to take effect.