**UWB – CSE 490A - Intro to Cloud Computing**

**Azure VMs – Auto-Scaling project**

**Objective**

In this project, you will demonstrate basic understanding of setting up Azure VMs and horizontal auto-scaling as the load increases and decreases in a queue based producer-consumer scenario.

**Cloud concepts covered**

1. Creating VMs (Azure VMs)
2. Building a Cloud service using several VMs (Azure Service)
3. Configuring Cloud service for auto-scaling (Azure Availability Set)
4. Cloud Apps (Producer and Consumer using VS, C# and Azure SDK)
5. Cloud Queue (Azure Service Bus) – for communication between producer and consumer apps
6. Cloud Storage (Azure Blob) – for consumer apps to log in cloud
7. Scripting common Cloud tasks (Azure PowerShell)
8. Managing cloud resources (Azure Portal)
9. Testing cloud scenario (Auto-scaling in phases and end-to-end)

**Overview of Final Delivery**

In this “final” delivery of the “Azure VM – Auto scaling project”, you will use the Producer and Consumer apps you had already built in the CP1 delivery to demonstrate the auto-scaling in action. You will need to update your Consumer app to log its activities in a cloud blob storage. Each Consumer app will have its own blob for its logs. You will still run multiple copies of your Producer app from your local laptop feeding the messages into the cloud queue. You will create several (say 3 with bare minimum being 2) VMs in your Azure subscription under a single cloud service (meaning all those VMs will be behind a single VIP and load balancer). You will first manually connect to each of these VMs and get them ready to run your Consumer app. You will also develop a few PowerShell scripts to automate the launching of this app on these VMs when the auto-scaling logic will start them. You will then set up the auto-scaling rules in your Azure portal to define when additional VMs will be started and stopped based on the length of the message queue. Assuming there was nothing in the message queue in the beginning, other than one VM all other VMs will quickly shut down based on the policy you had setup for auto-scaling. After verifying this, you will then launch several Producers from your laptop to build up the load in the message queue. You will then successful demonstrate that Azure will automatically launch more VMs to consume the messages from the queue as per the auto-scaling rules. You will then kill the Producers and demonstrate that slowly other than one, all other VM will again stop.

**Details of Final delivery**

1. You already have a folder with Producer.exe and Consumer.exe (along with all the other dlls and configuration files they need) from CP1 and you should have already tested the reading and writing into the cloud queue with multiple Producers and Consumers.
2. From the left pane of the Azure portal choose “Storage” and create a storage name. Once it is created click on it and create a container (by choosing “Container” menu at the top) for the logs coming out of the Consumer apps running on various VMs (you can name it something like “Consumer-App-Logs.txt”). Remember in auto-scaling you won’t be starting your Consumer app manually from a console window. Instead the system will run the app at startup as a background process without a console. Whatever you were printing earlier to your console window won’t be visible. To compensate for this, your consumer app needs to be updated to

* Create a blob with the above storage name that you just created OR use if one already existed (it will be good to have your VM name in the blob name you create)
* Update app.config to add another entry in the appsettings section of configuration where you had earlier added an entry for the queues. It will look something like this and you will have to add your account name and key:

<configuration>

<appSettings>

<add key="StorageConnectionString"

value="DefaultEndpointsProtocol=https;AccountName=account-

name;AccountKey=account-key" />

</appSettings>

</configuration>

* All the consumer apps will write the log entries to their individual blobs by appending at the tail end. The blob will continue collecting the logs across VM start, shutting and the restart.
* Log an entry to mark the start of a Consumer app with the time it started.
* Log an entry for each message written and when
* Here is a fake example for one of the log file from one of the VM. There will similarly be several logs one for each VM:

TestVM1Blob.log:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* “TestVM1” started at 14:40pm \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TestVM1 Received: Producer#1 Msg#2 at 14:41pm

TestVM1 Received: Producer#2 Msg#3 at 14.41pm

..

..

..

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* “TestVM1” started at 15:20pm \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

TestVM1 Received: Producer#2 Msg#27 at 15:21pm

TestVM1 Received: Producer#3 Msg#32 at 15.22pm

..

..

..

1. With the same storage name you created in previous step, click on it and create another container for the automation scripts (name it something like “myscripts”). You will need this in the later steps.
2. Create N VMs from Azure Portal under a single Azure service (From Gallery -> Windows Server 2012 R2 Data Center -> choose a VM name, an appropriate size for VM, give an admin name and password for the VM -> For first VM choose a Cloud Service Name, Create an Availability Set and Create The VM. Then create the second and subsequent VMs with different VM names but sharing the Cloud Service and the Availability Set).
3. Connect to each VM and
   1. disable IE’s enhanced Sec (using Server Manager – Local Server – IE Enhanced Security to Off)
   2. Install .NET 4.5.2 on each VM (search the web or try <http://www.microsoft.com/en-us/download/details.aspx?id=42643> ). This may need a reboot.
   3. Create a folder on each VM (e.g. c:\temp) and copy updated Consumer app (everything from the VS’s bin/debug folder) in this folder
4. You had already installed PowerShell with CP1 setups on your laptop. Run it and connect it with your Azure subscription (add-azureaccount).
5. Develop a PowerShell script (.ps1) that will help launch the Connumer.exe whenever a VM starts
6. Develop a PowerShell Script to copy the above .ps1 file to the storage container (e.g. “myscripts”) you created above.
7. Develop and run another PowerShell Script from your laptop that will help setup the startup action for each VM to launch the Consumer app in the background using the .ps1 file you have just copied to the storage container. This will also launch Consumer app on your running VMs.
8. At this point you should be able to run task manager on each VM and see that the Consumer.exe is running in the list of background tasks.
9. Setup the Auto-scaling rules from Azure portal for your cloud service under which three VMs are running. From Azure Manage Portal, go to Cloud Services, select your service, click scale tab, choose “Queue” from “Scale by Metric”, fill-up relevant info there and setup a number for “Target per Machine” (e.g. 50). You should also set Scale Up and Scale Down time to say 5 mins. You will notice in your Virtual Machine page in the portal that after 5 mins, one of the running VM will be stopped by Azure. If you had created more than 2 VMs, after subsequent 5 mins, more VMs will be stopped leaving only one running.
10. When only one VM is left running, start several Producer apps from your laptop and see the number in your queue to start going up as only one instance of Consumer.exe is running. After a while as the queue start to get filled and goes above 50, second VM (and second instance of Consumer.exe will start up) and then the third one etc.
11. When all your VMs have started, kill the Producer apps and ensure that after a while queue length goes down to 0 and VMs start to shut down again.
12. Use <https://poprtal.azure.com> to go to the Storage Account (classic) on the left pane and drill down from there to go to your blob. Download the blob from there and match up your log entries against the activities you had done.

**Submission of Final Delivery**

1. Assigned on Oct 21st 2015
2. Due by end of the day Oct 28th 2015
3. 50% grade weightage of the whole project
4. Make sure your Cloud service and VMs are working properly and availability set is defined properly for the VMs
5. Make sure scripts are doing the right thing for setting up the startup actions for each VM
6. Make sure your consumer app is running properly on each VM
7. Make sure auto-scaling rules are setup properly
8. Make sure end-to-end scenario of VMs starting and stopping as a result of queue based auto-scaling events have been tested with multiple Producers.
9. Make sure the log blobs have right logs.
10. Make sure you have made me the co-admin to your azure subscription
11. Submit a compressed zip folder in canvas containing
    1. Your Producer and Consumer projects solution folder
    2. scripts folder containing all the automation scripts you have used

**Extra Credit (20%)**

1. Develop and submit an app that will read the log entries from a Blob log of the above project. This could be in the form of downloading the file (e.g. .txt) and view with an existing app or printing each log line on the console.(10%)
2. Develop and use automation scripts to create above VMs and copy the consumer apps on those (10%)