Capstone-Project 1
By: Ankit Ranjan

Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Problem Statement:

You work in XYZ Corporation. Your company is facing some issues and wish for You to use Azure cloud expertise to solve them.

Issues:

1. They wish to have a centralized store to store all their developer tools in. This store should be such that developers can mount it on their file system.

2. They wish to store large volumes of image data. They wish to have low latency access to frequently accessed images i.e., images that have been accessed in the last 14 days. If an image is not accessed within the last 14 days they wish to archive them.

3. They wish to lower the latency of their website. They have noticed that users who are far away from their web server have complained that images take a lot of time to load.

4. They wish to serve another website on Azure's VMs.

5. You also want to have two VMs in different networks. They wish for you to deploy those VMs and enable communication between them.

6. They wish to use Azure to resolve their site with domain 'simple-site.tk' to its IP address.

7. They wish for both the VMs serving their website to be more reliable so that if one VM fails the traffic is automatically routed to the other one.

8. They wish for you to find a way to assign and manage credentials for Azure for all 10 employees in the company.

9. Finally they have two applications that need to pass messages between one another on an on-demand basis i.e., an application will send the message and other applications will receive and process it when it can. You need to set up a service in such a way that these applications can do so (you are provided with the code). All you need to do is make changes to the config file.

You need to use services from Azure cloud to help your company resolve all these issues.

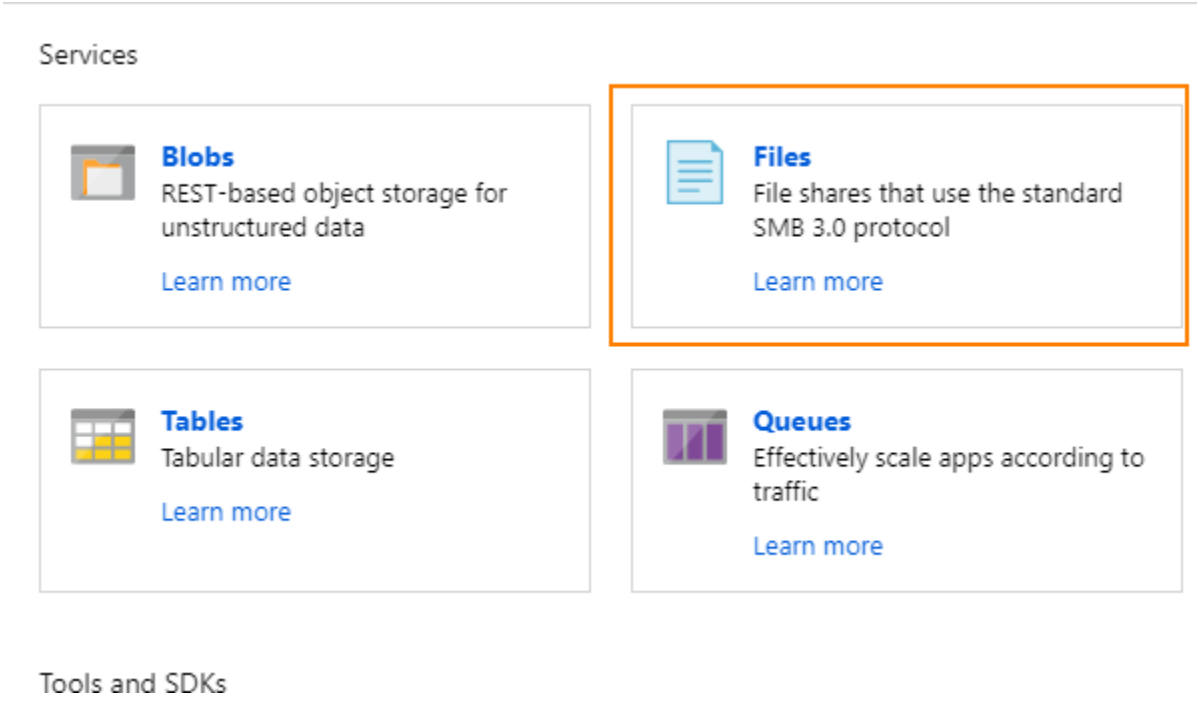
Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Issue 1: They wish to have a centralized store to store all their developer tools in. This store should be such that developers can mount it on their file system.

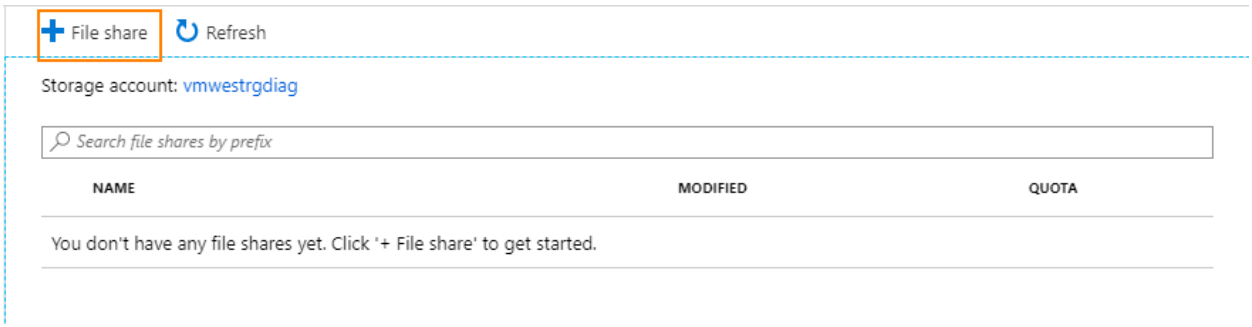
Solution:

Step 1: Create and open your storage account

Step 2: Click on Files

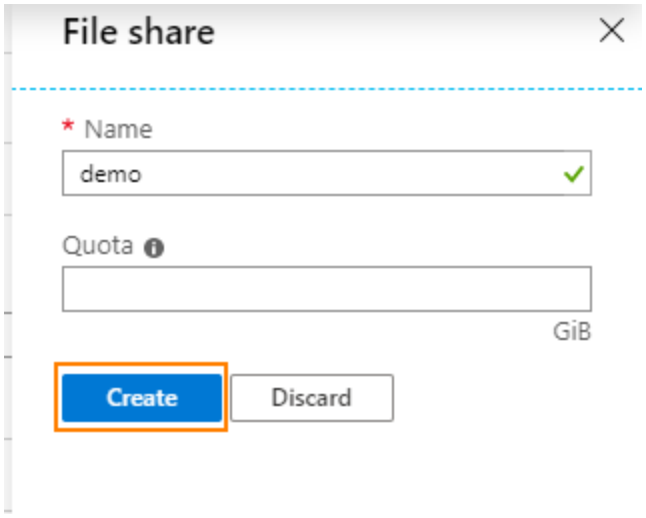


Step 3: Click on +FileShare



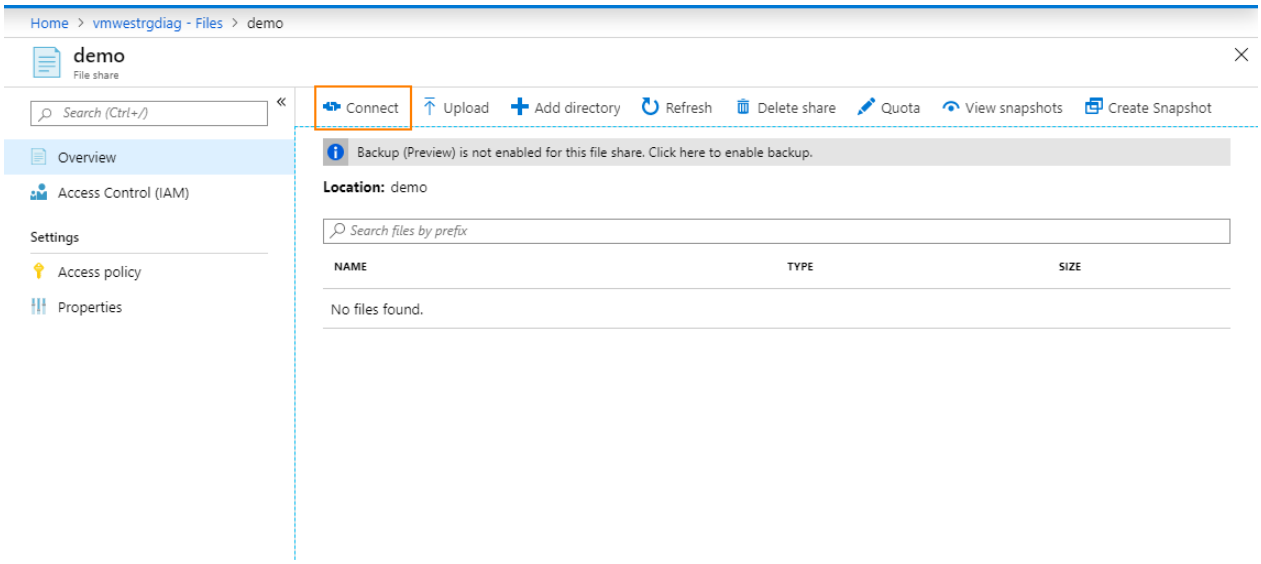
Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Step 4: Enter the details and click on Create



Step 5: Mount the File Share on your PC

Step 5.1: Open the File Share in Azure Portal
Step 5.2: Click on Connect



Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Step 5.3: Copy the command and check you have URL, username and password

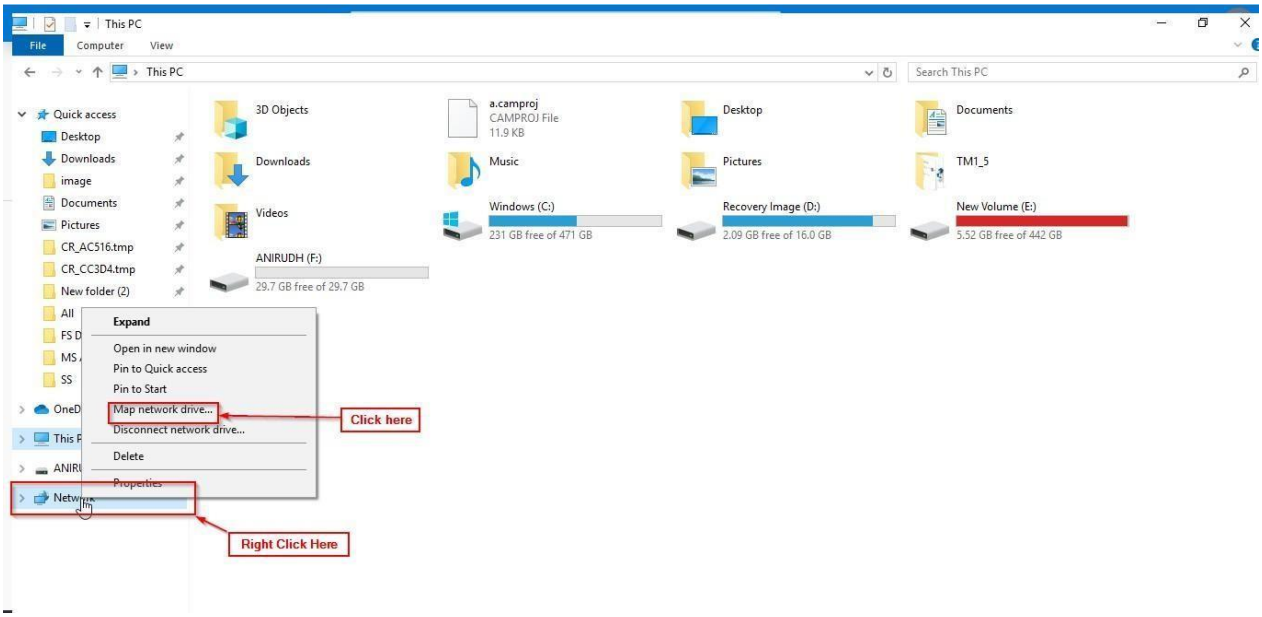
```
Test-NetConnection -ComputerName fsdemo1234.file.core.windows.net -Port 445
# Save the password so the drive will persist on reboot
Invoke-Expression -Command "cmdkey /add:fsdemo1234.file.core.windows.net /user:Azure\fsdemo1234
/pass:sb+fZUElz5o6CDRC/ViFYptpKshhKnq4yJoMCPw8mHJIz2wPCDwofeKdhCGsjQ/g7b7KWLRnLtdCC1+kfX6hpw=="
# Mount the drive
New-PSDrive -Name Z -PSProvider FileSystem -Root "\\fsdemo1234.file.core.windows.net\demo"
```

Username

Password

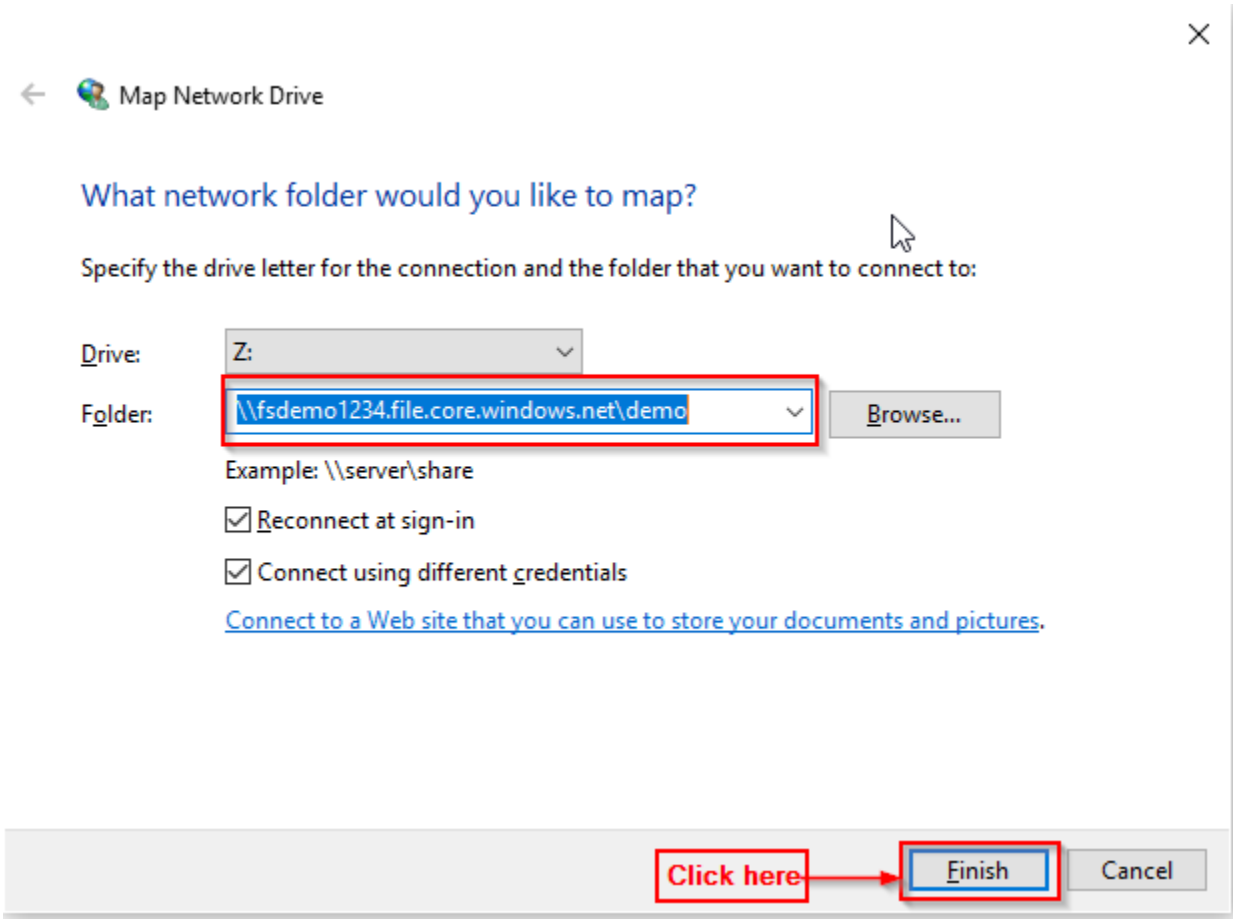
Folder URL

Step 5.4: Open Windows Explorer

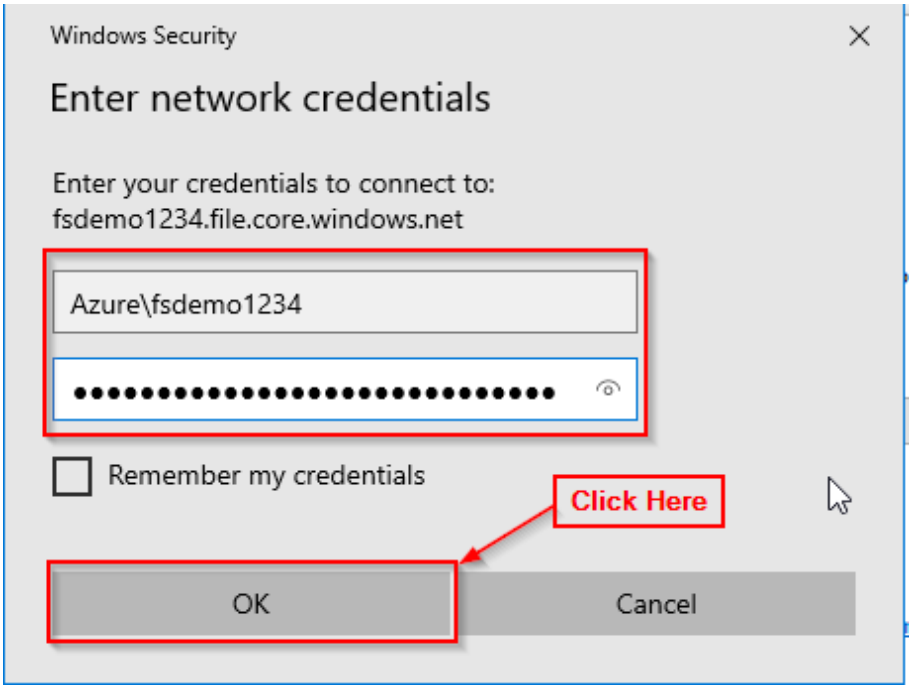


Step 5.5: Right Click on Network and Select Map network drive

Step 5.6: In the folder section, paste the URL copied from the command



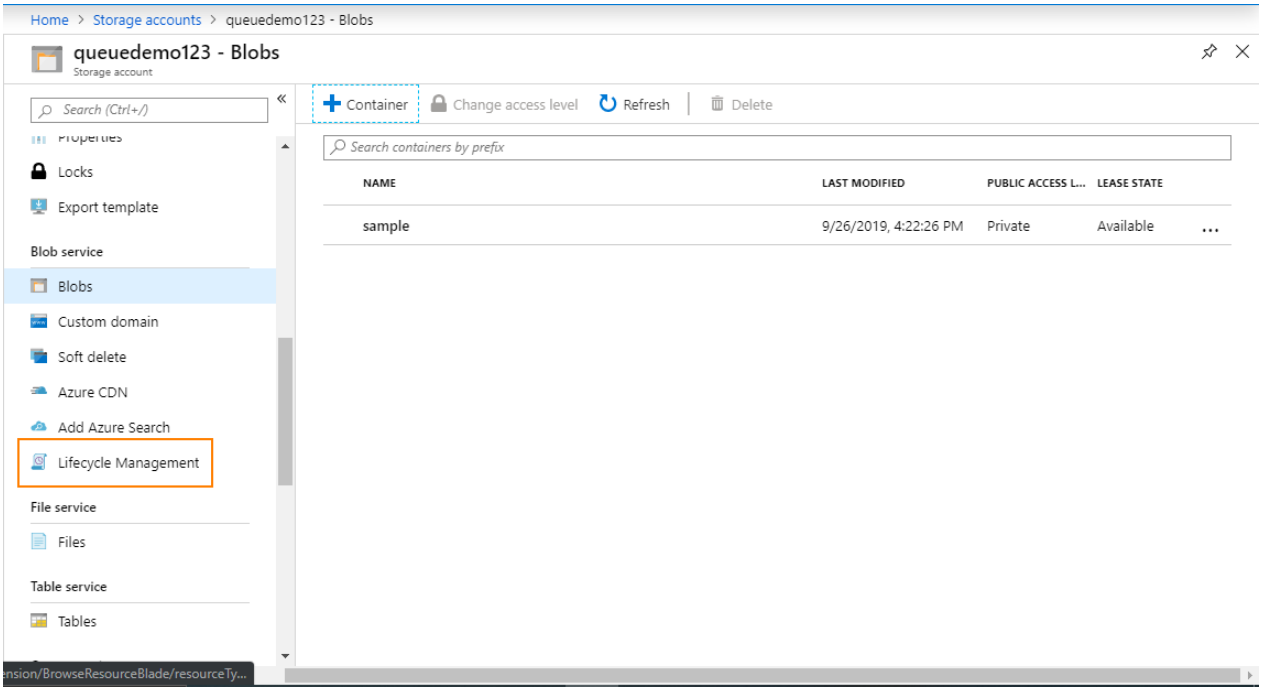
Step 5.7: Enter username and password from the command and click OK



Issue 2: They wish to store large volumes of image data. They wish to have low latency access to frequently accessed images i.e. images that have been accessed in the last 14 days. If an image is not accessed within the last 14 days they wish to archive them.

Solution:

Step 1: Create a Blob container
Step 2: Add images to that Blob
Step 3: Under Blob Service in the sidebar, select Lifecycle Management



Step 4: Select Add rule

Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

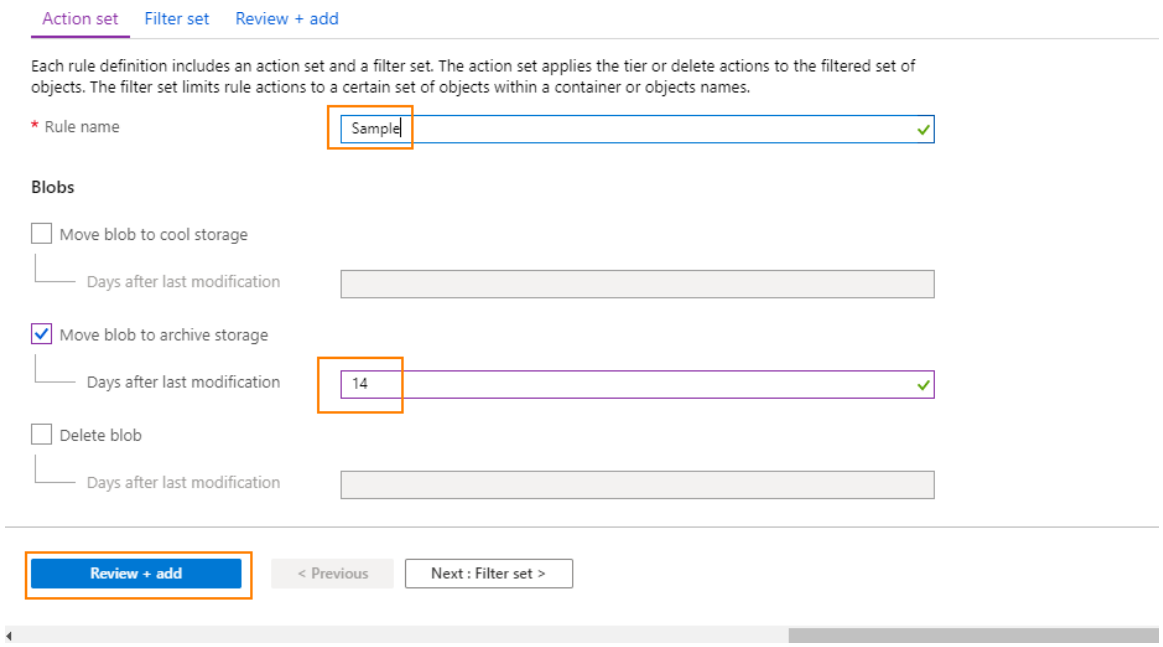
Step 5: Under the action set tab fill in the following details:

Step 5.1: Add rule name

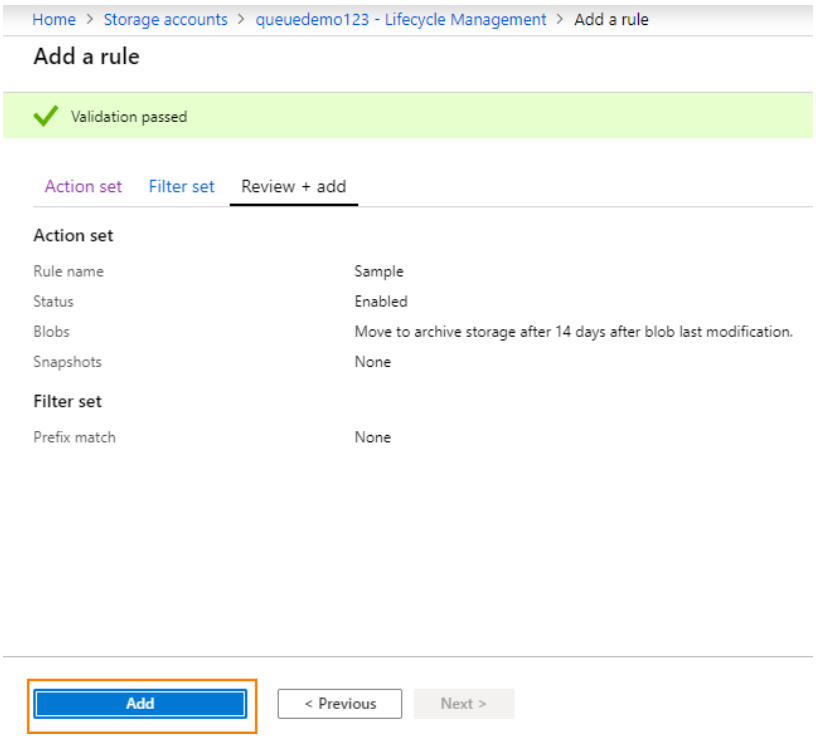
Step 5.2: Select Move blob to archive storage

Step 5.3: In Days after last modification fill 14

Step 5.4: Click on Review + add



Step 5.5: Click on Add



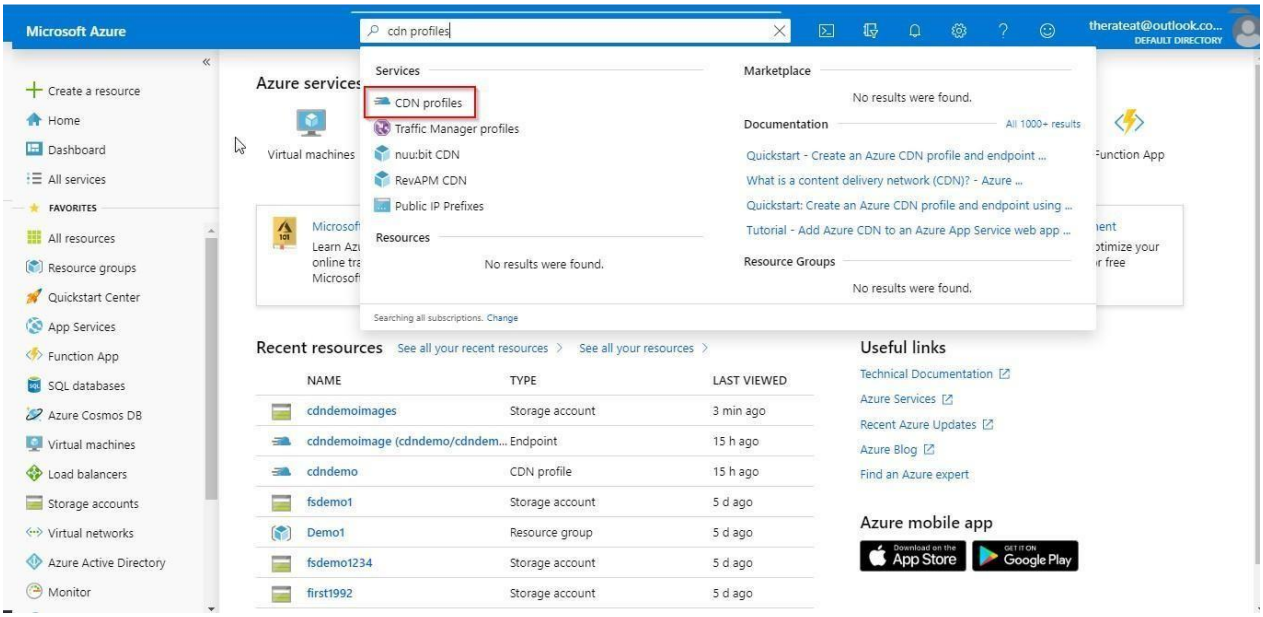
Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Issue 3: They wish to lower the latency of their website. They have noticed that users who are far away from their web server have complained that images take a lot of time to load.

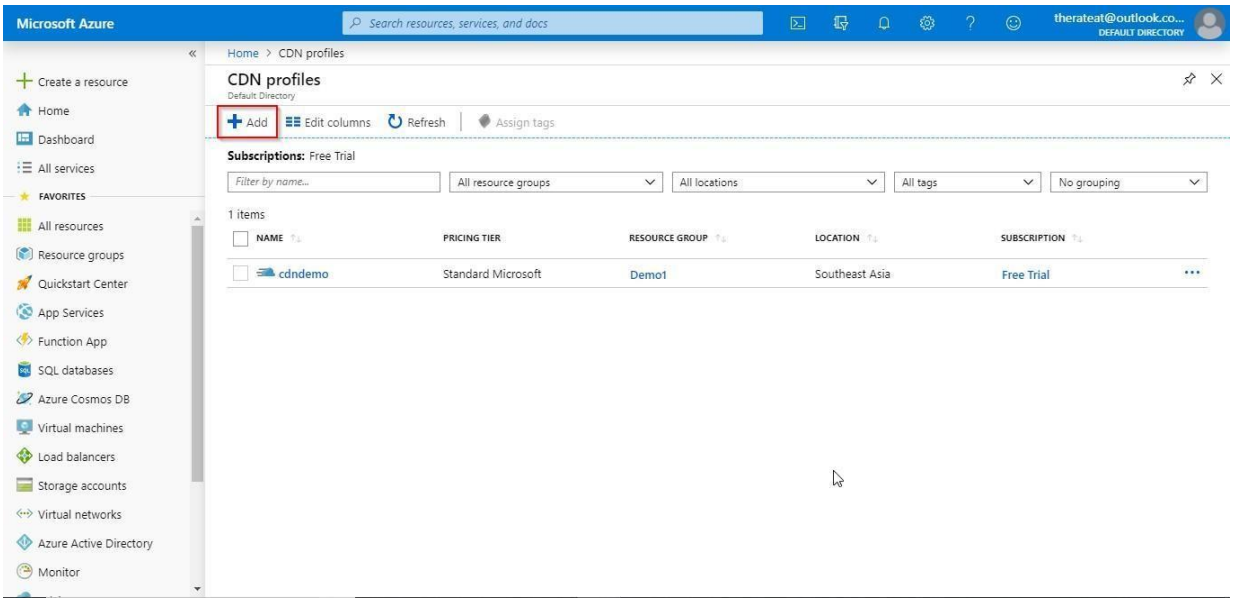
Solution:

Step 1: Create a storage account for Blob Storage and upload a few images

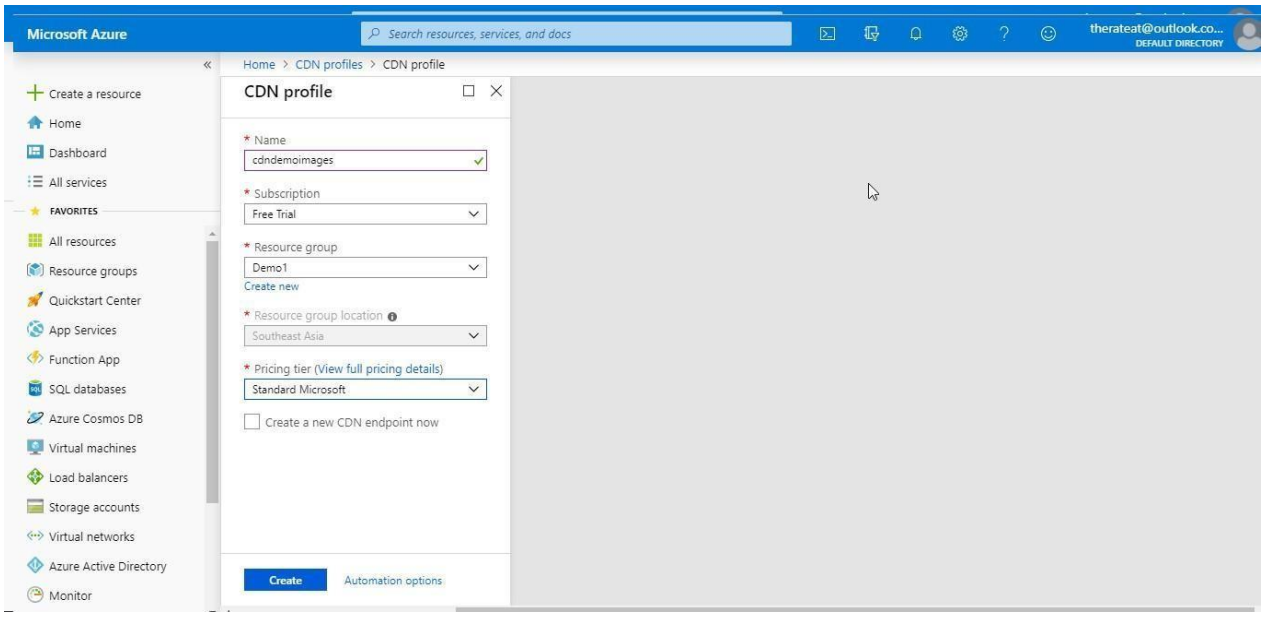
Step 2: In the Azure Portal's search for CDN Profiles and open it



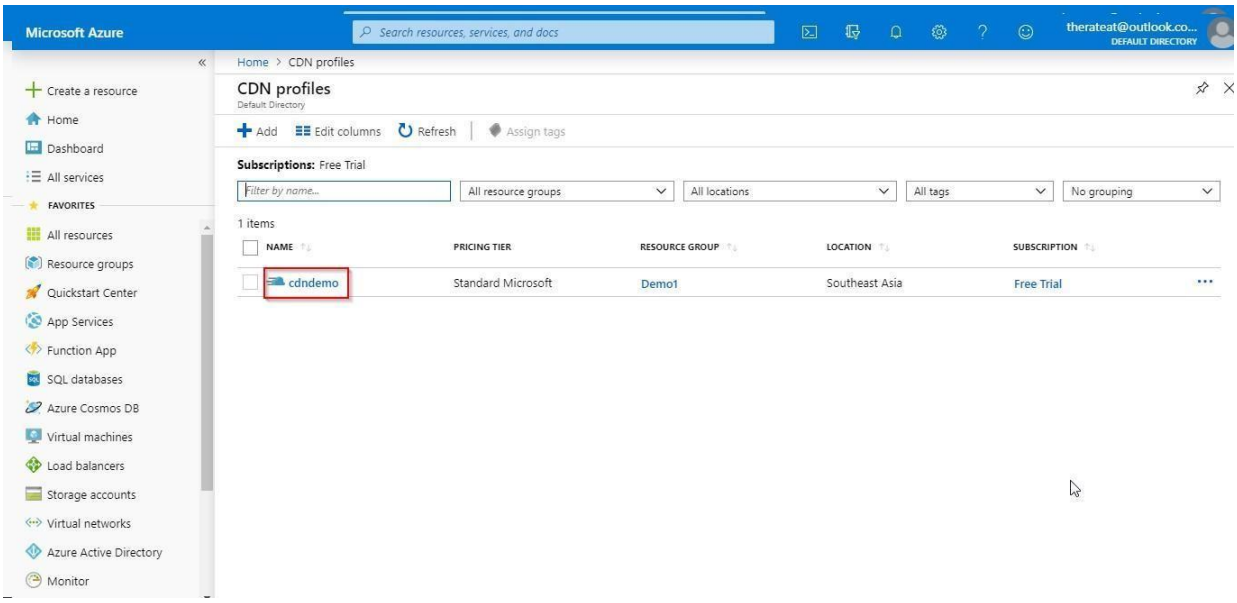
Step 3: Click on + Add



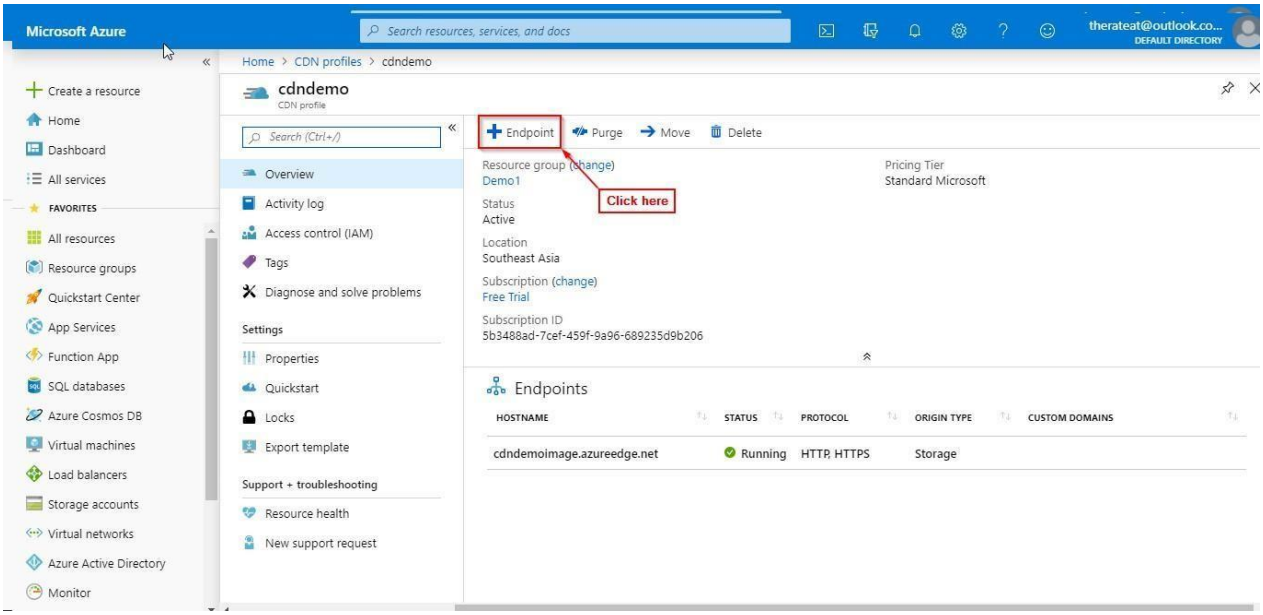
Step 4: Fill the details and click on Create



Step 5: Click on the CDN Profile just created



Step 6: Click on +Endpoint

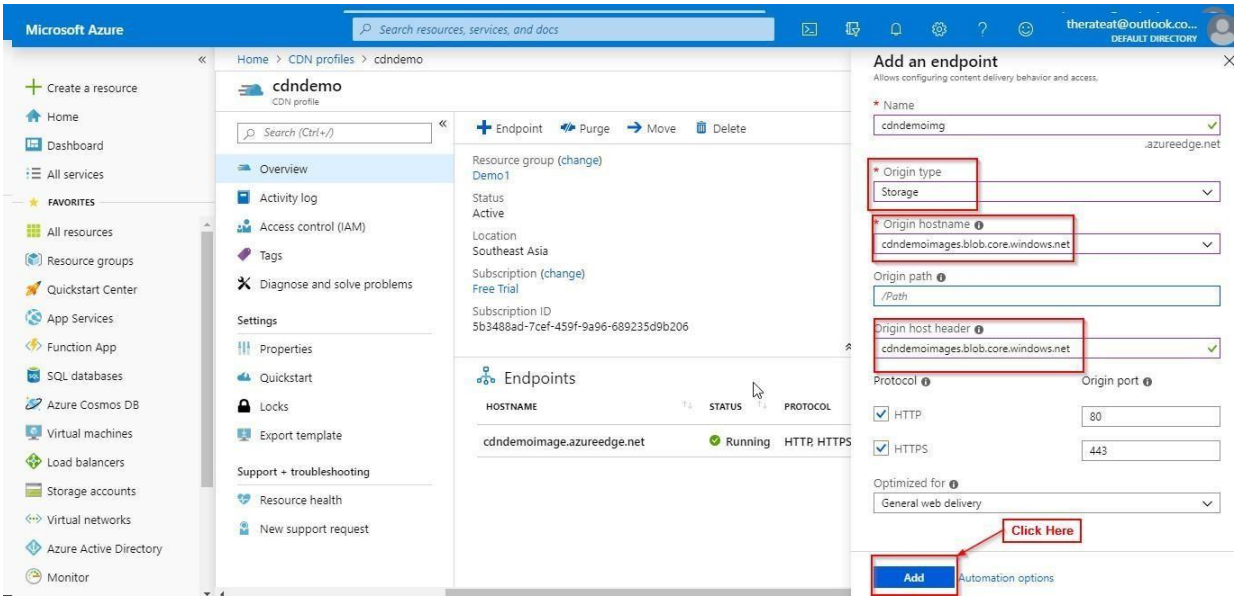


Step 7: Enter the details:

Origin type : Storage
Origin hostname : URI of Blob storage created in previous step

Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Step 8: Click on add



Issue 4: They wish to serve another website on

Azure's VMs.

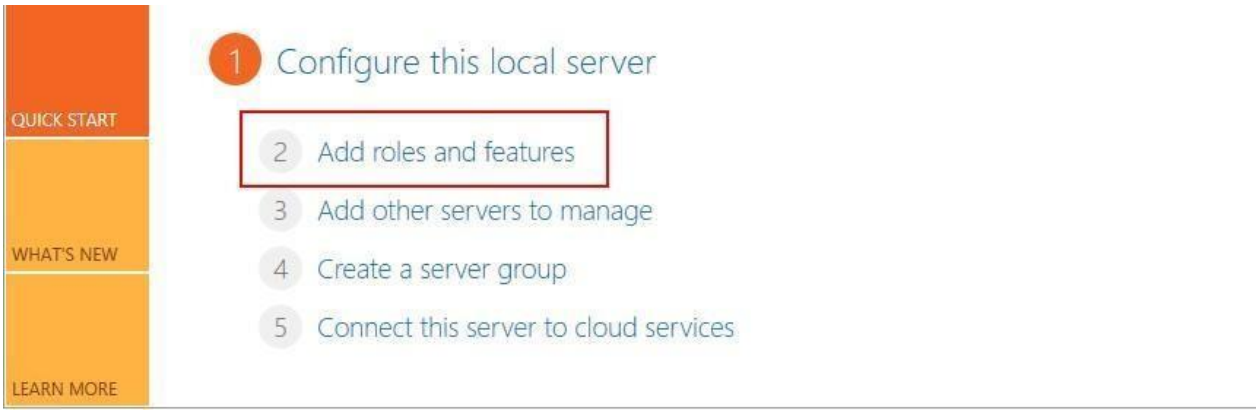
Solution:

Step 1: Create a VM using Azure Portal (Make sure to enable HTTP and RDP protocol for connecting)

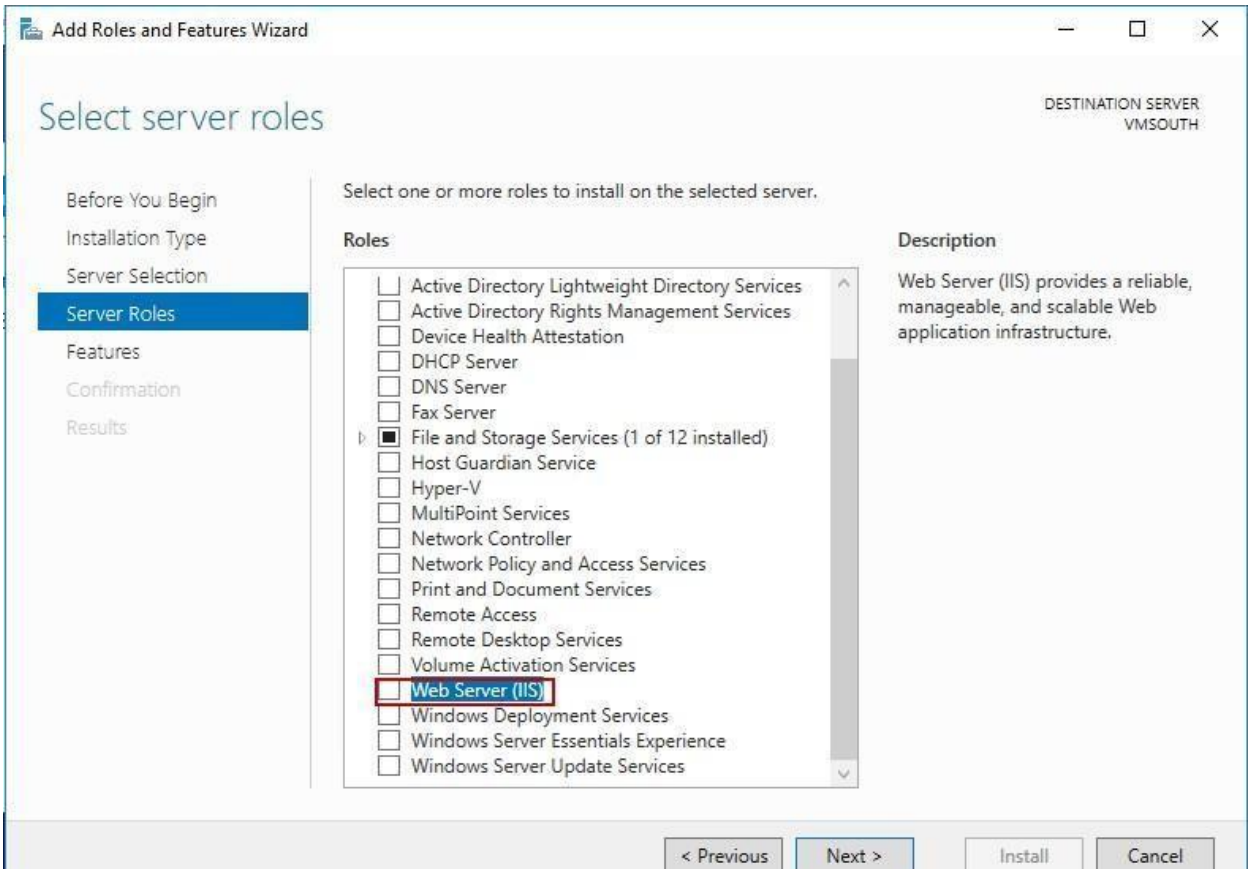
Step 2: Open the VM using RDP

Step 3: Open Server Manager

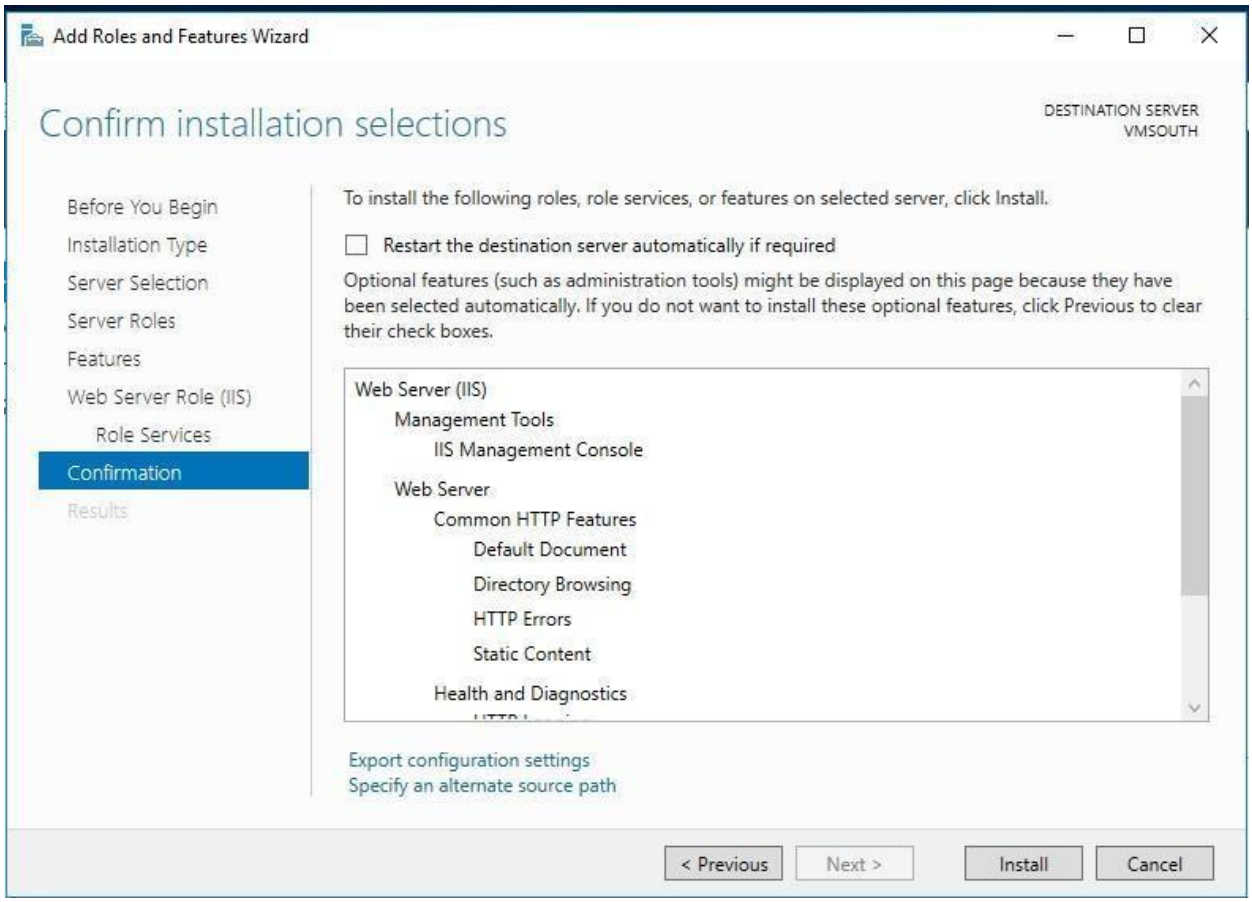
Step 4: Click on 'Add Roles and Features'



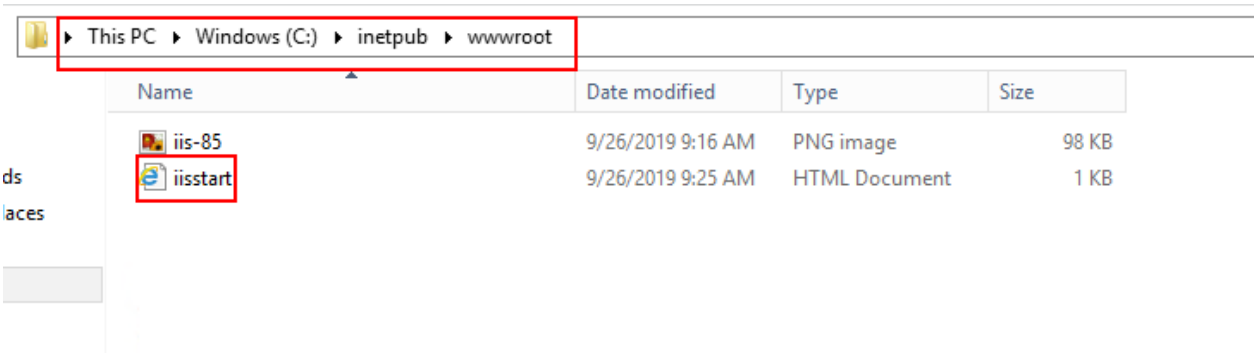
Step 5: In the wizard click on 'Next' 3 times and then in 'roles' select Web Server (IIS)



Step 6: Click on Next and Install



Step 7: After installation finishes Open ‘C:\inetpub\wwwroot\’, this is the root folder of IIS

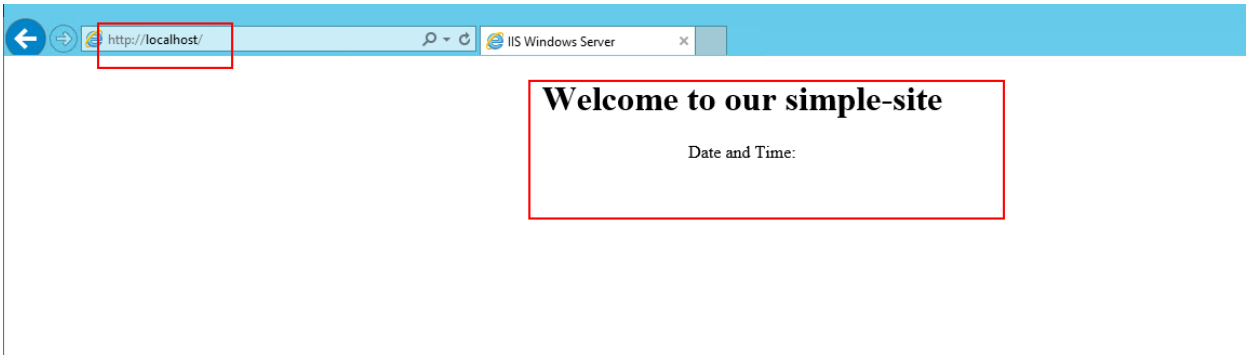


Step 8: Open iisstart.html in Notepad

Step 9: Make changes to the code by replacing the body tag with following code and save it

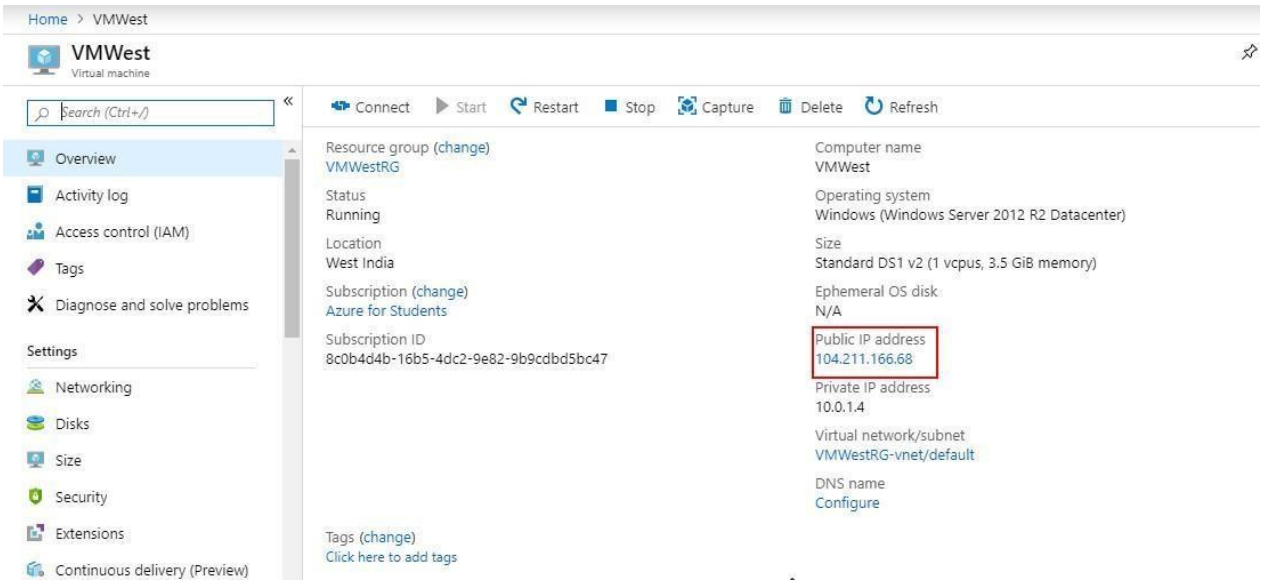
```
<body>
<div id="container">
<h1>Welcome to our simple-site</h1>
<span>Date and Time: </span>
<p id='time'></p>
</div>
<script>
        let el = document.getElementById('time');
        setInterval(() => el.innerHTML = (new Date()).toLocaleString(), 1000);
</script>
</body>
```

Step 10: Open localhost in Internet Explorer in VM to see if it works.

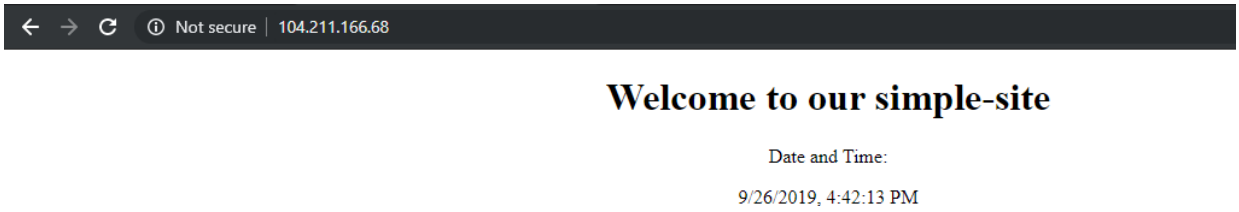


Step 11: Open the VM page in Azure Portal

Step 12: Copy the public IP Address of the VM and open it in the browser



Step 13: See that your simple-site is being served by the web server



Issue 5: They also want to have two VMs in different networks. They wish for you to deploy those VMs and enable communication between them.

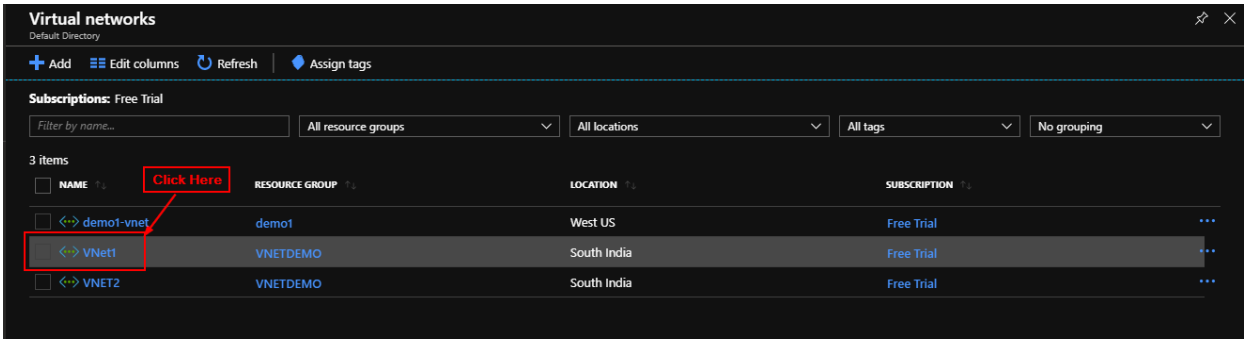
Solution:

Step 1: Create two VNets in the same region

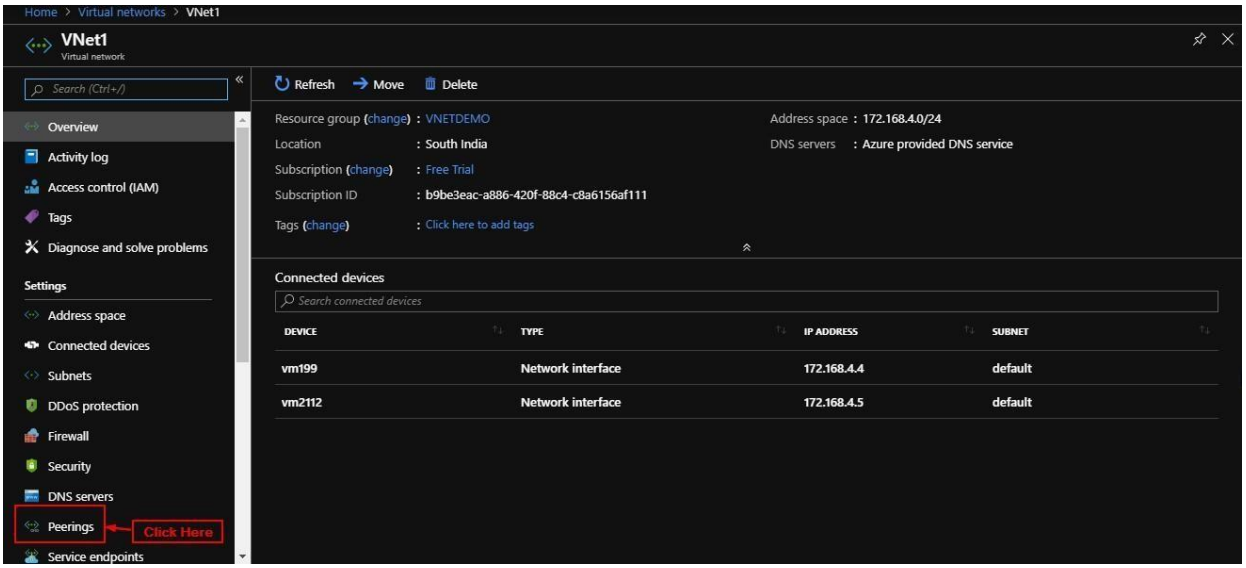
Step 2: Create two VMs, one in each separate VNet

Step 3: Open Virtual Network page in Azure Portal

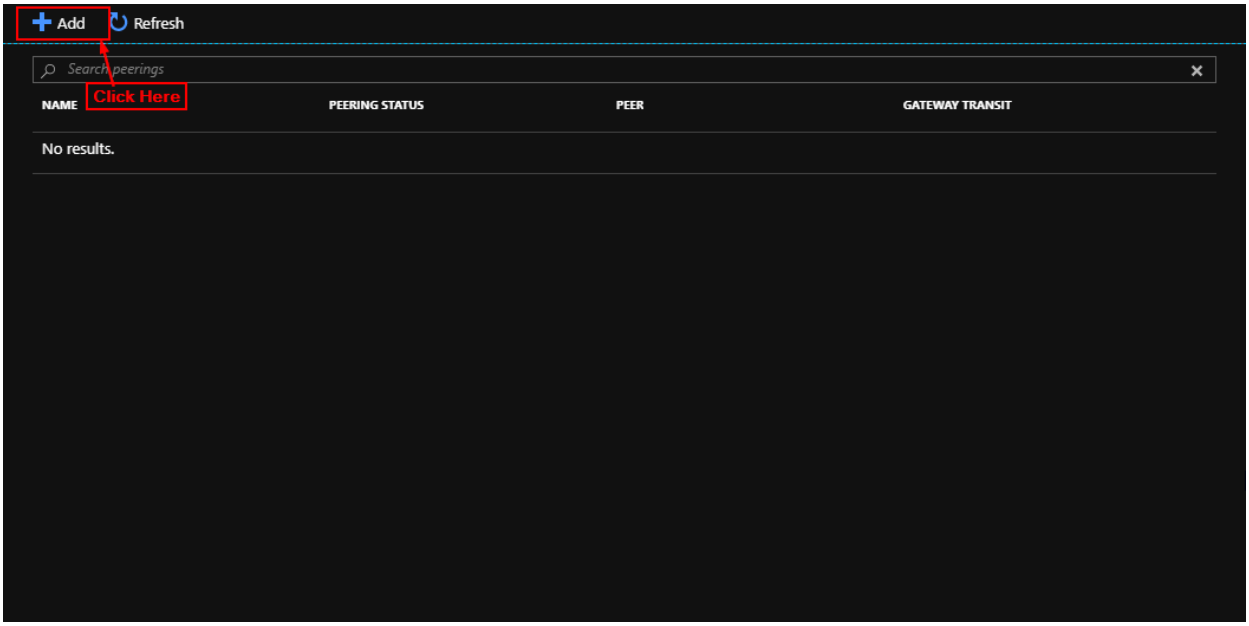
Step 4: Click and open one of the VNets created in Step 1



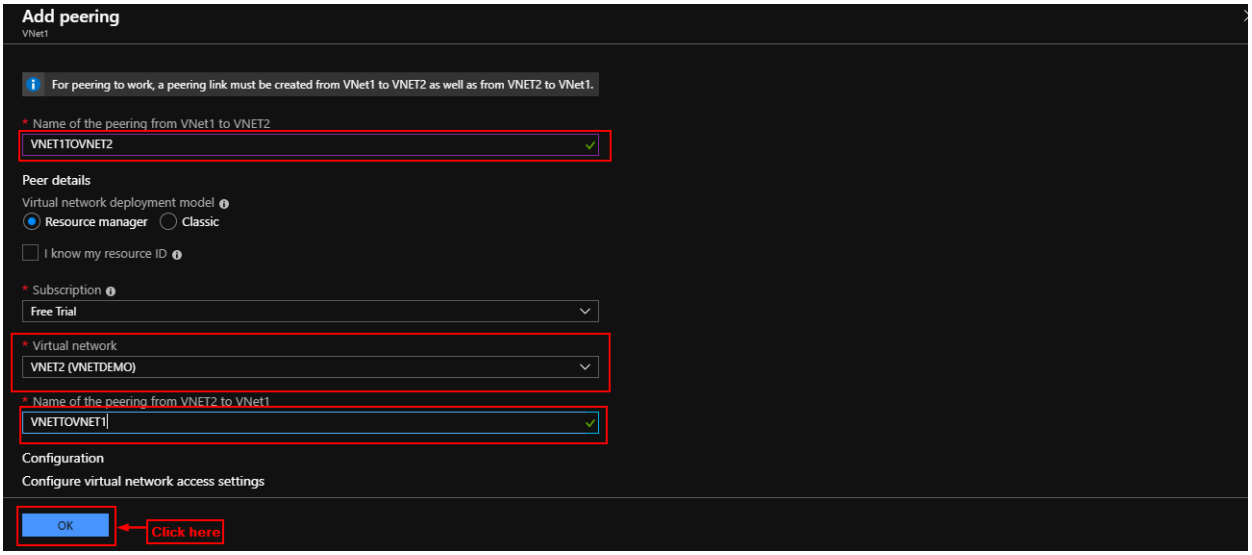
Step 5: In the sidebar click on Peerings



Step 6: Click on + Add



Step 7: Enter the details, make sure you add details for both peerings to enable bi-directional communication



Step 8: Click on OK

Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Issue 6: They wish to use Azure to resolve their site with a domain of your choice to its IP address

Solution:

-- GET A FREE DOMAIN --

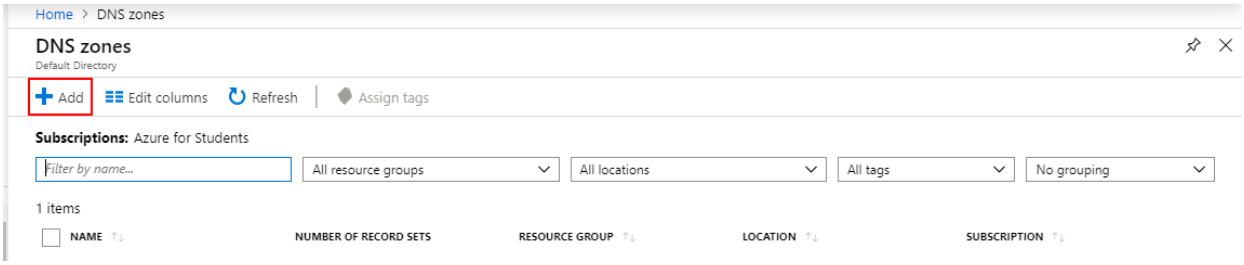
Step 1: Open and sign up at freenom.com

Step 2: Buy a free domain (for our purposes it can be anything)

-- Configure Azure DNS –

Step 1: In the Azure Portal search for DNS Zones and open it

Step 2: Click on + Add

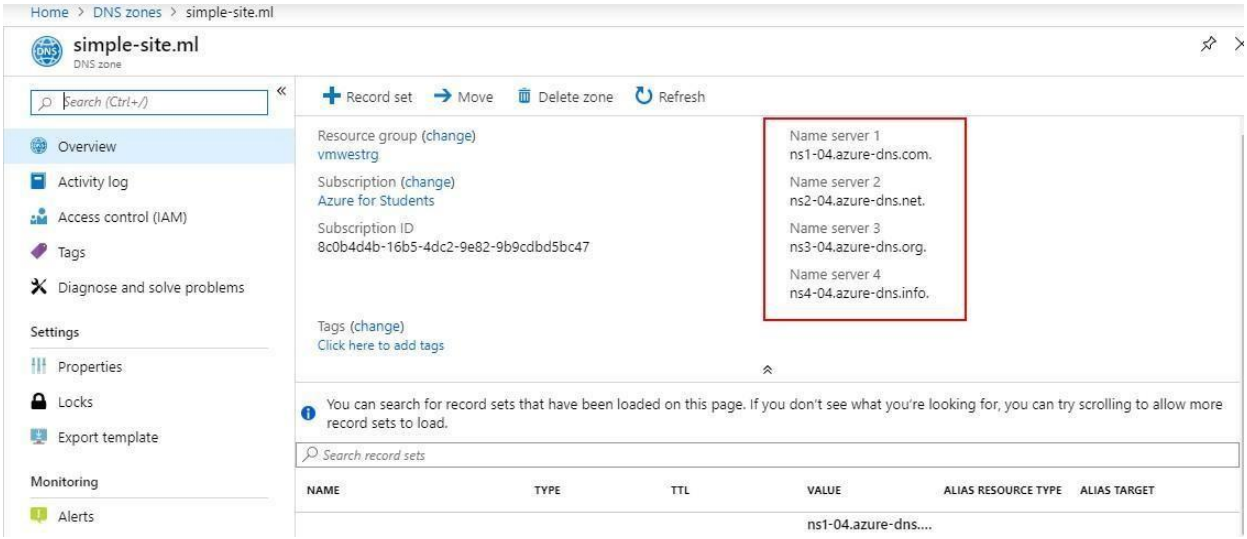


Step 3: Enter the following details and click on ‘Review + create’

- Resource Group: Your Resource Group.
- Name: you domain name e.g. ‘simple-site.ml’

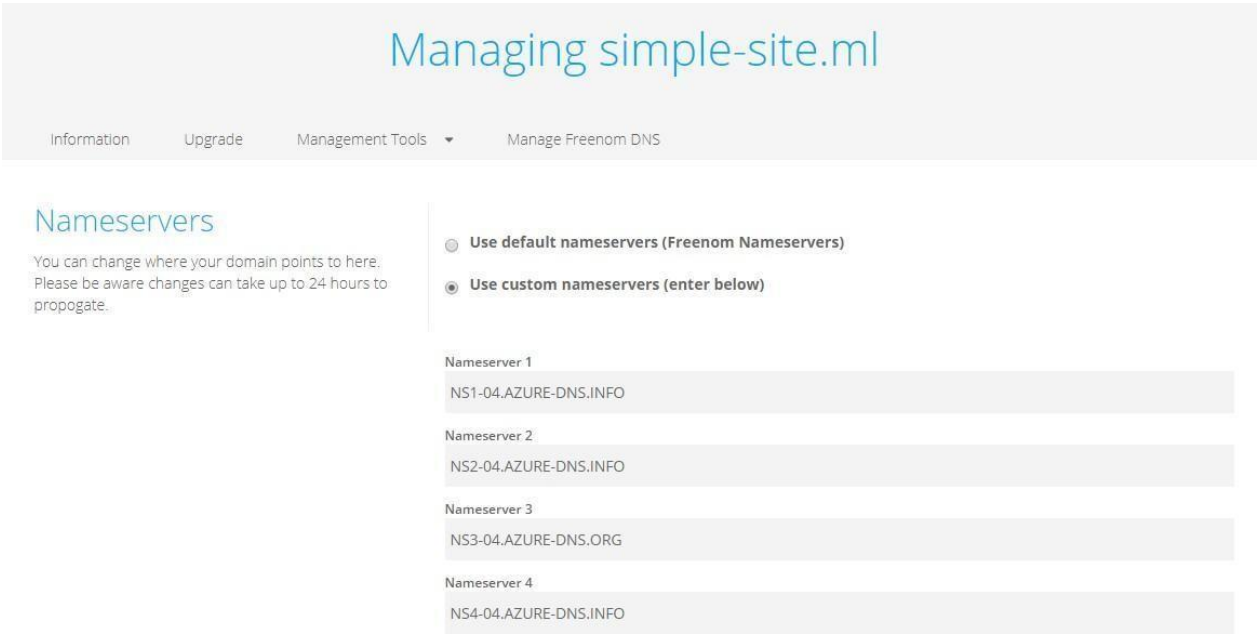
Step 4: Click on Create

Step 5: Open the DNS Zone and copy Name Server Addresses

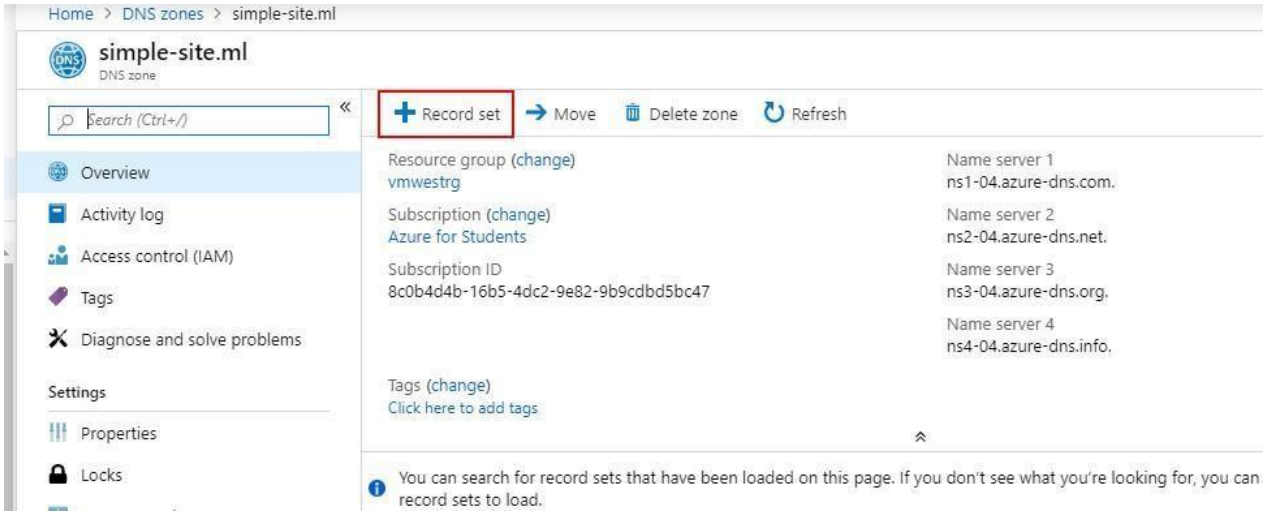


Step 6: Open your domain name provider's admin panel

Step 7: Change the name server addresses to the Azure Name Servers



Step 8: Open the DNS Zone and click on '+ Record Set'



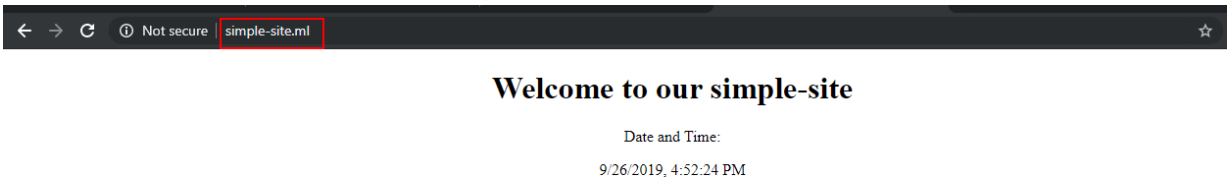
Step 9: Enter details:

- Name: www.
- Type: A

-IP Address as the Public IP of the VM on which app is deployed (Public IP of VM Created in solution of issue #4).

Step 10: Click on OK

Step 11: Now open the domain



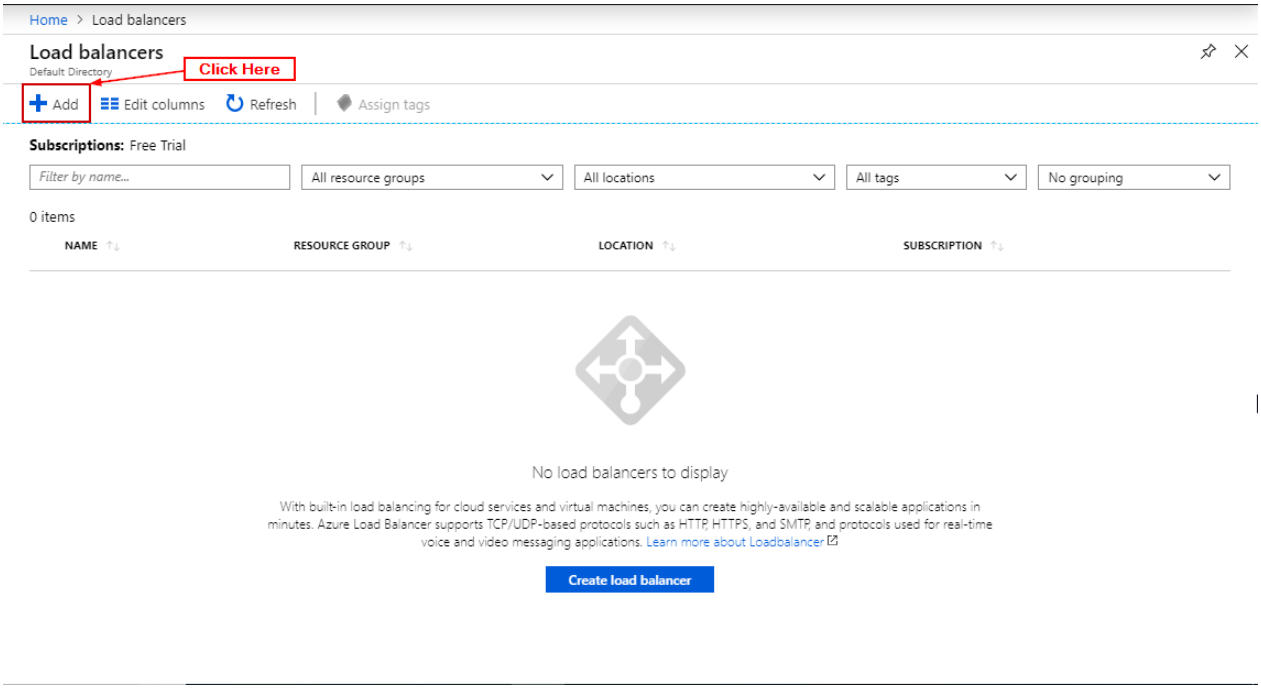
Issue 7: They wish for both the VMs serving their website to be more reliable so that if one VM fails the traffic is automatically routed to the other one.

Solution:

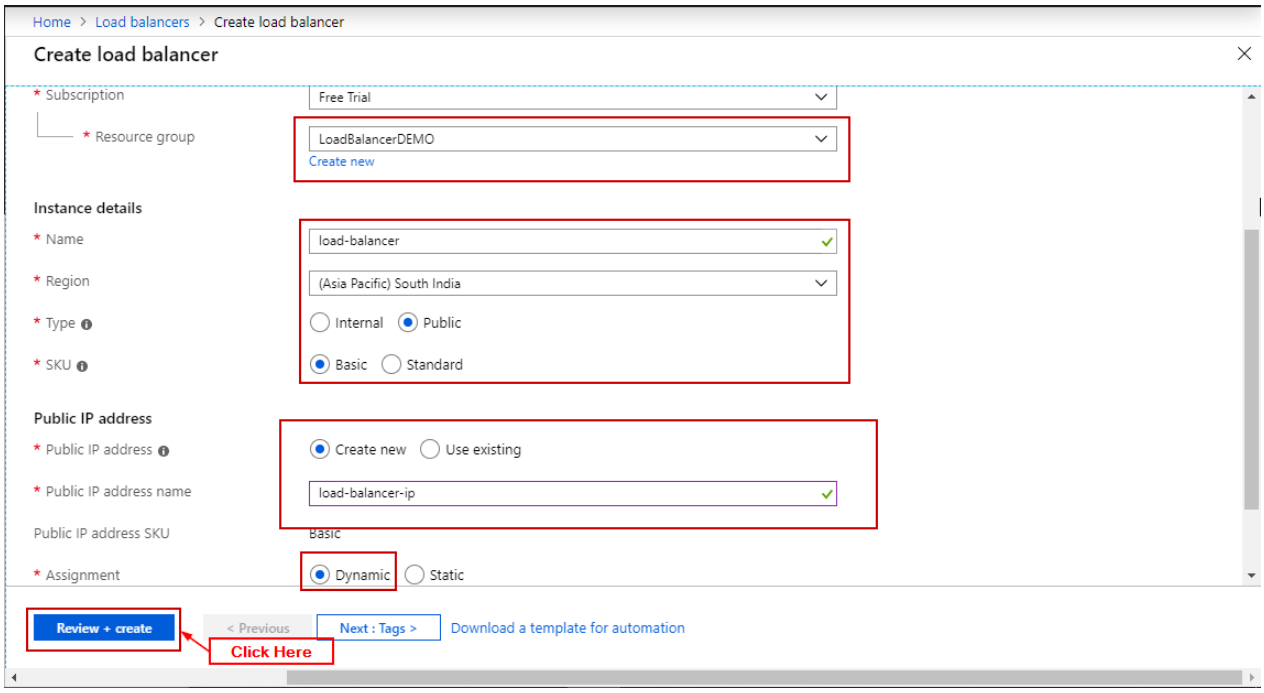
Step 1: Create two virtual machines within a single availability set and same virtual network

Step 2: Search Load Balancer in Azure Portal and click on it

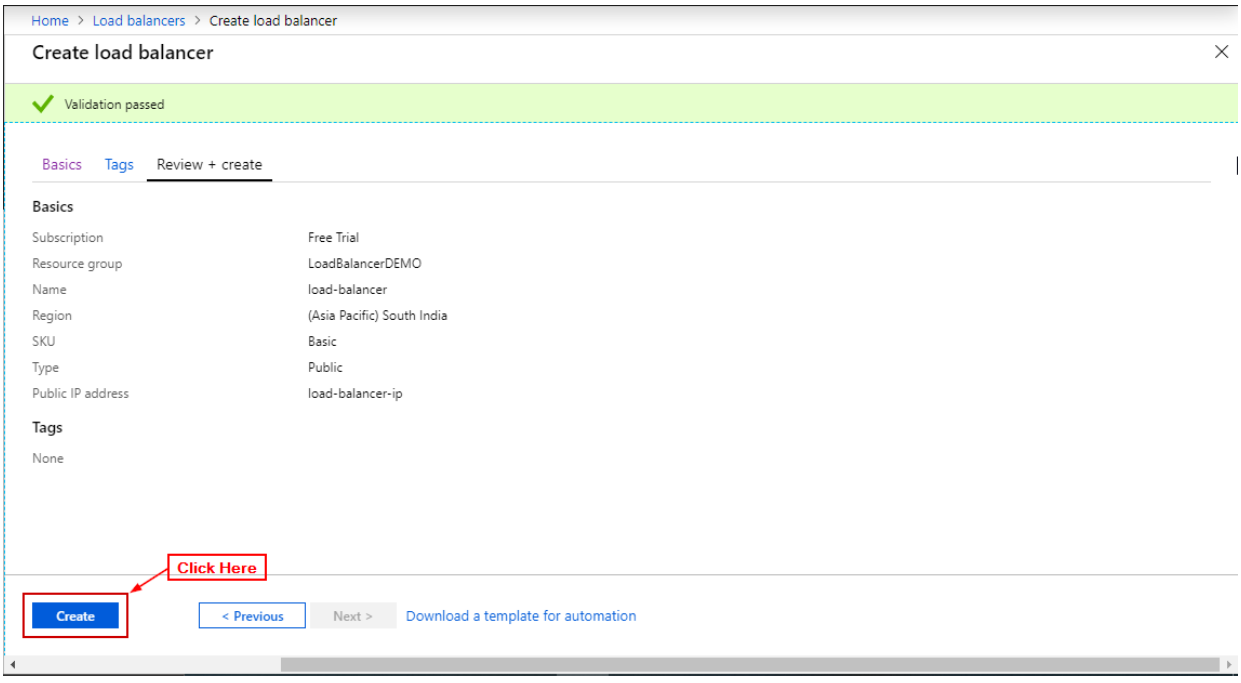
Step 3: Click on + Add



Step 4: Add details and click Review + create



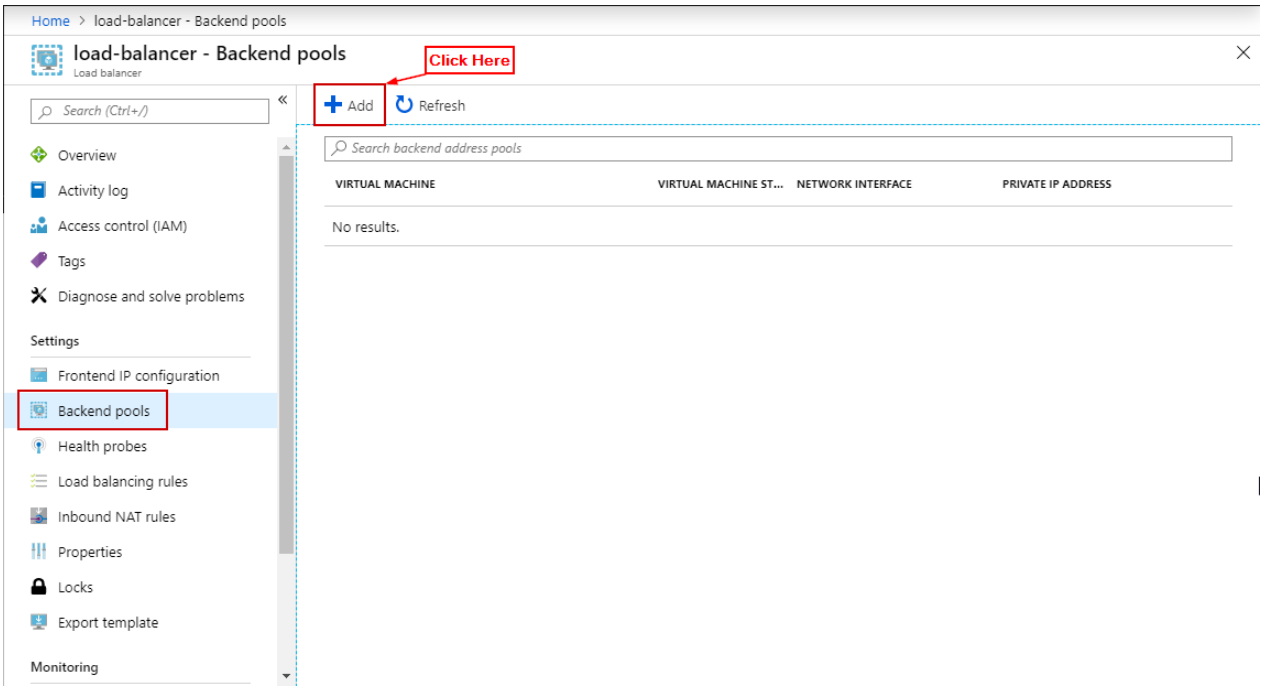
Step 5: Click on Create



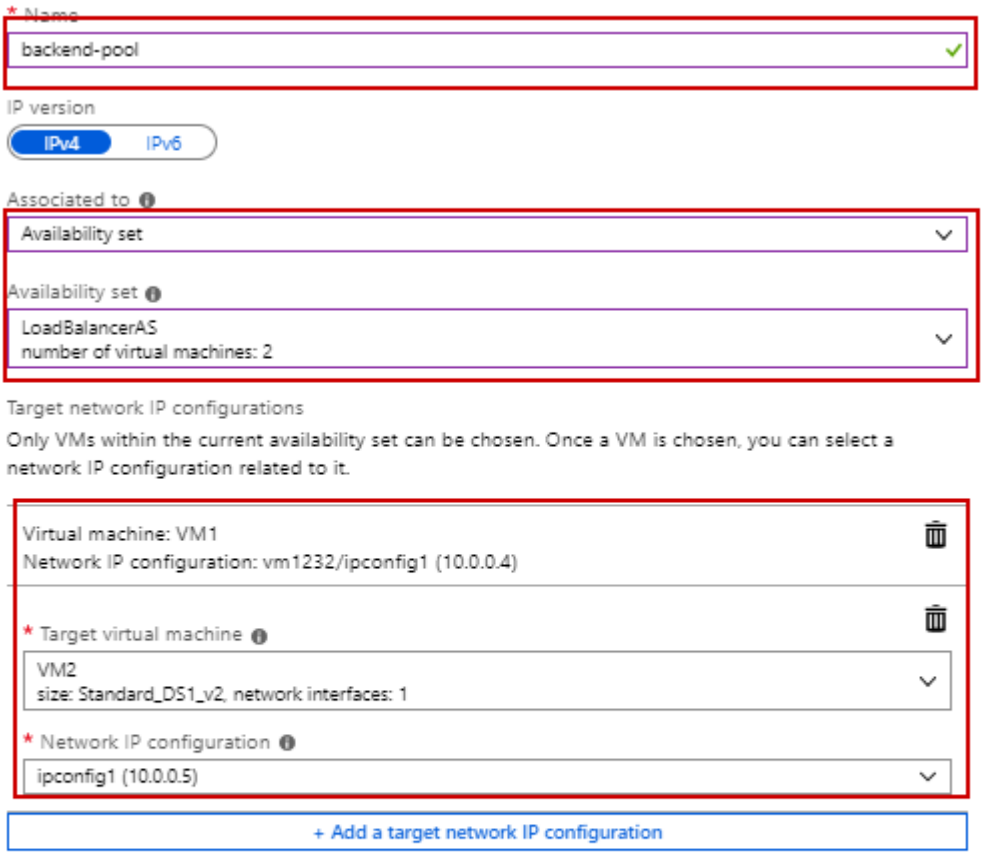
Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Step 6: Open the load balancer

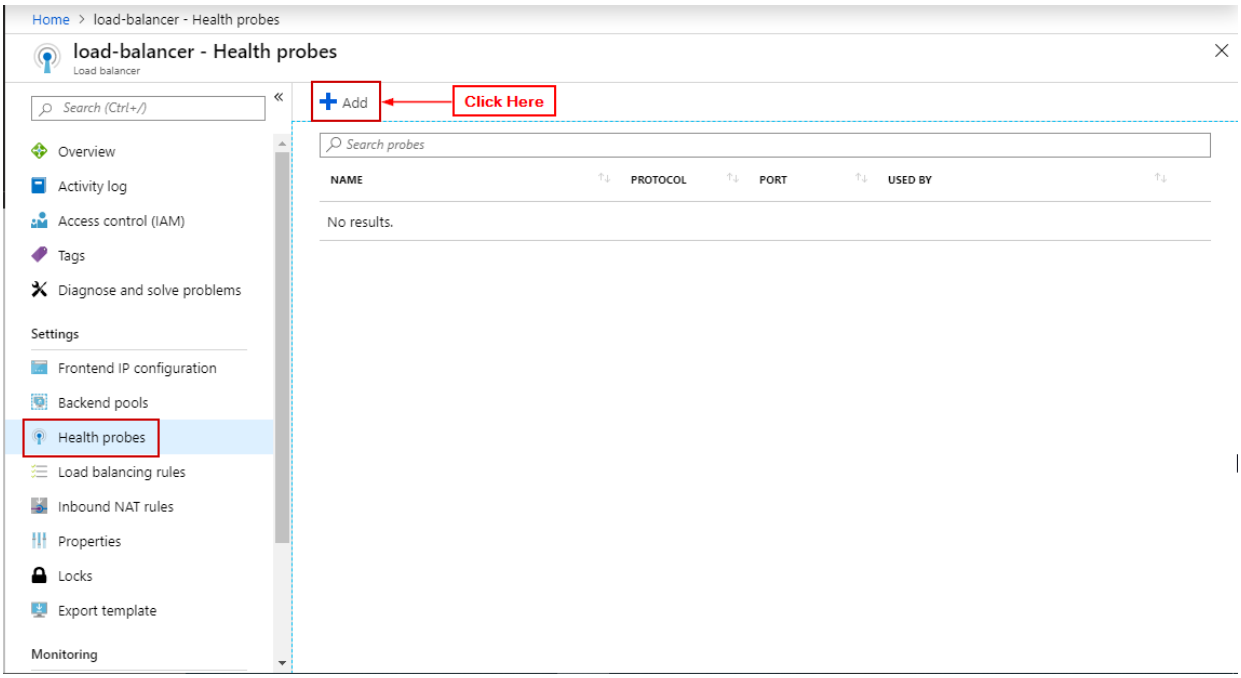
Step 7: Click on backend pools and click on + Add



Step 8: Enter the details (availability Set and in target IP Configuration, add both V

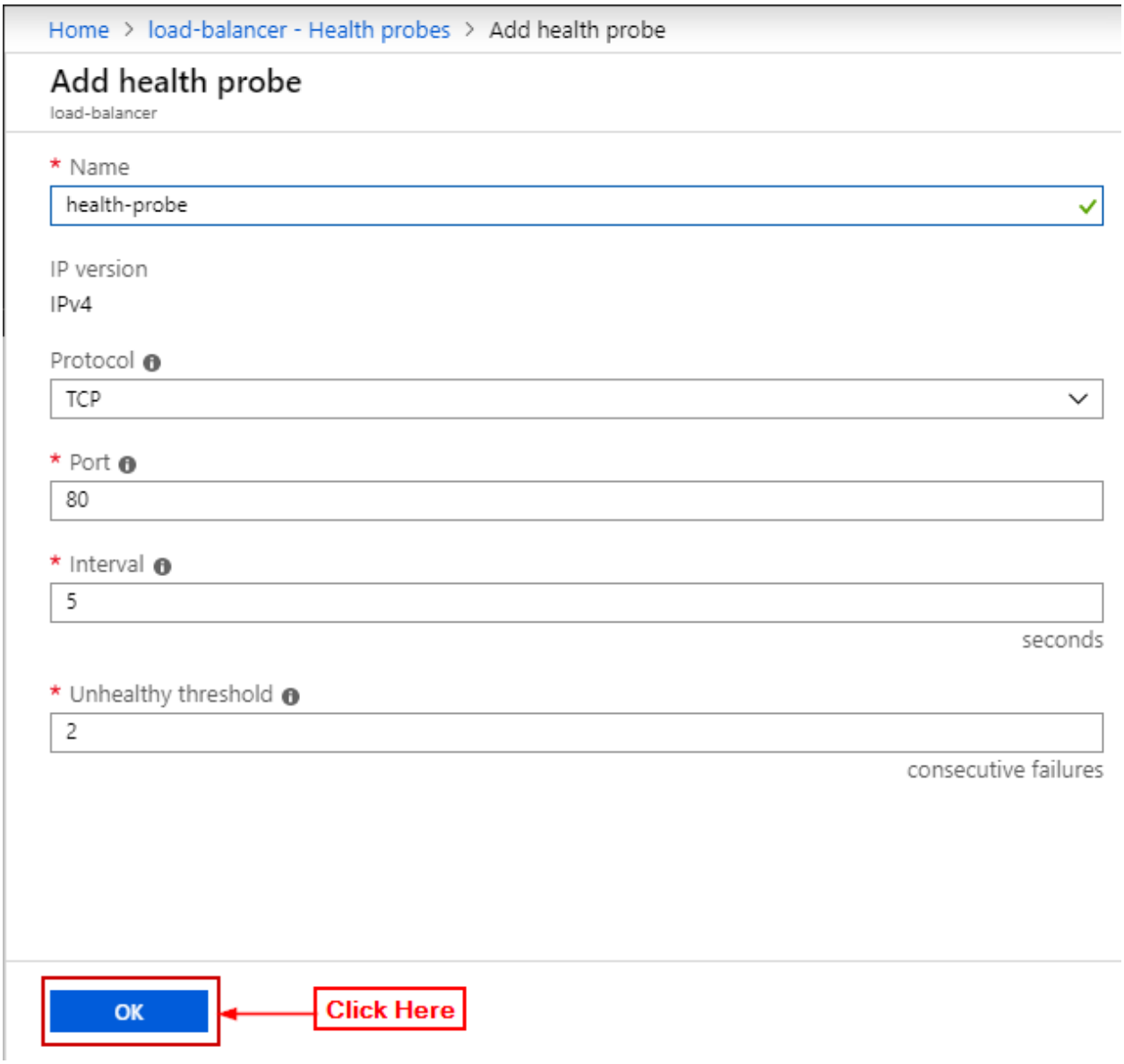


Step 9: In load balancer click on health probes and click on + Add

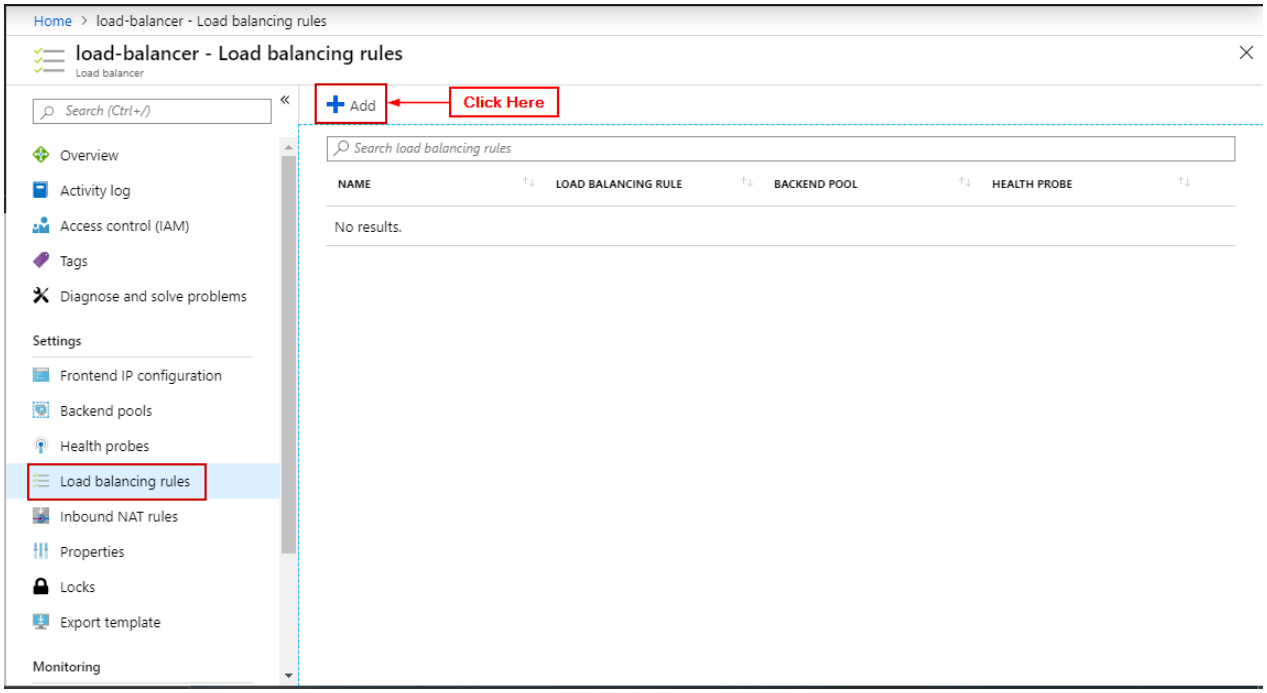


Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

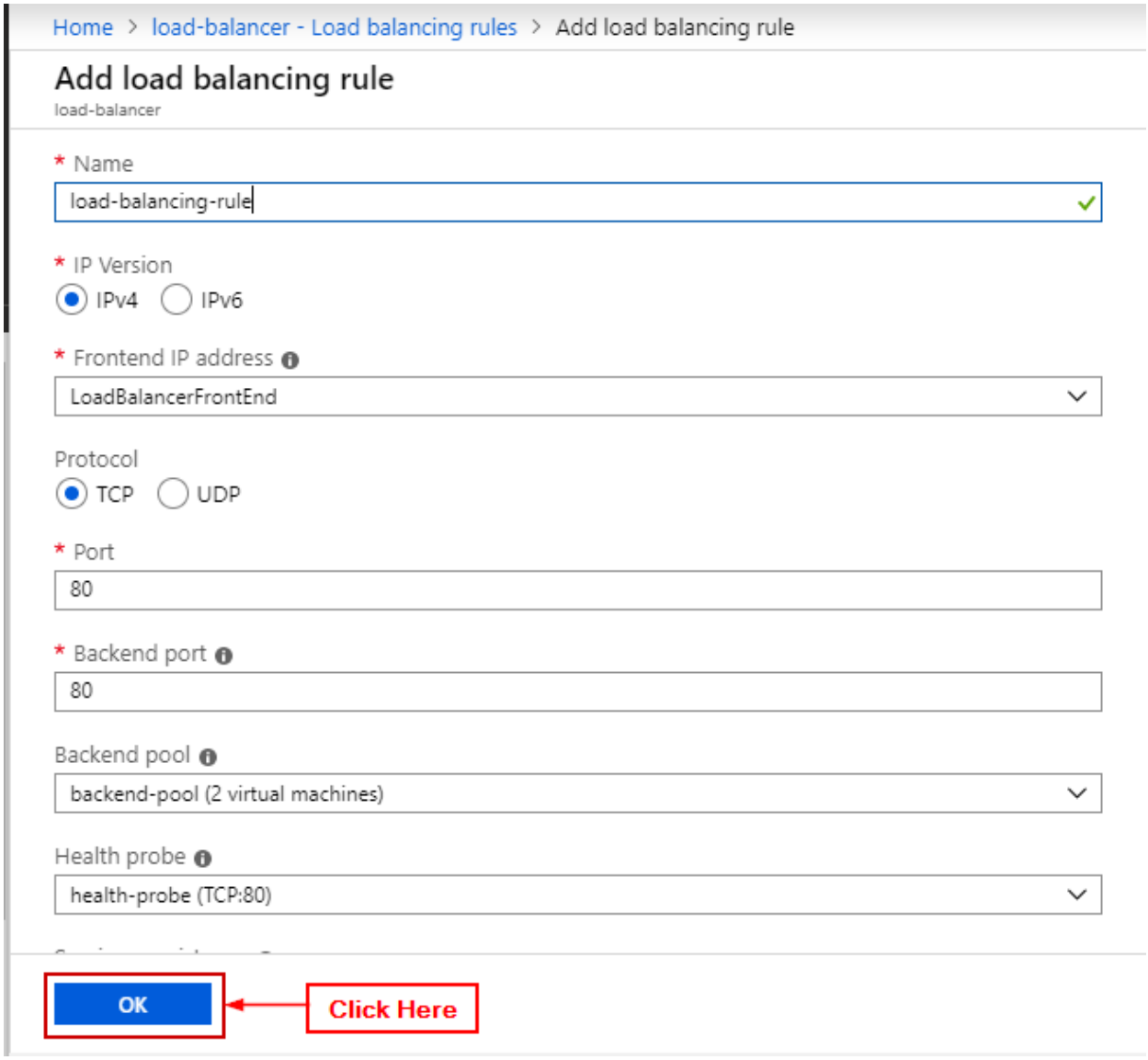
Step 10: Enter details click on OK



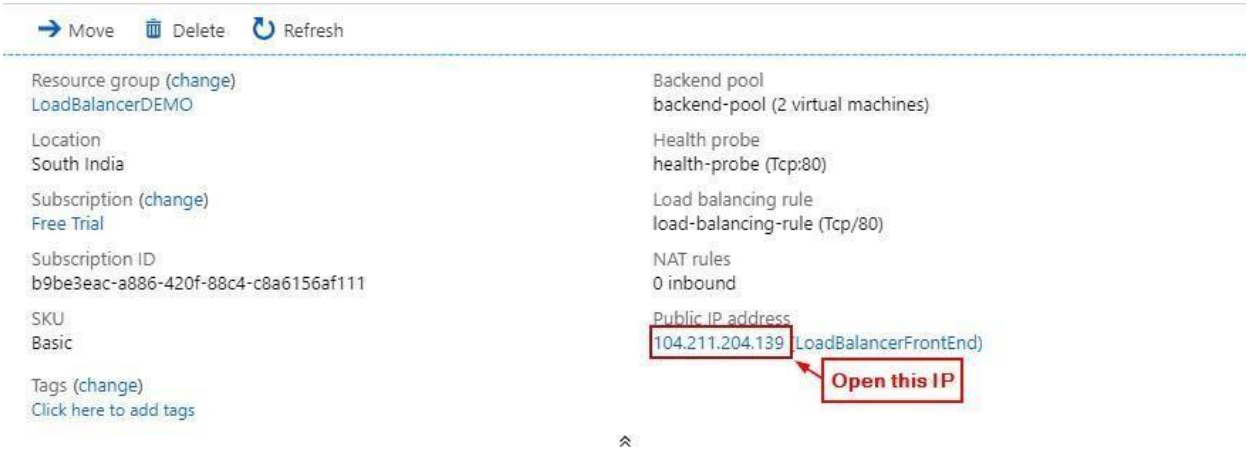
Step 11: In load balancer click on Load Balancing Rule and click on + Add



Step 12: Enter details click on OK



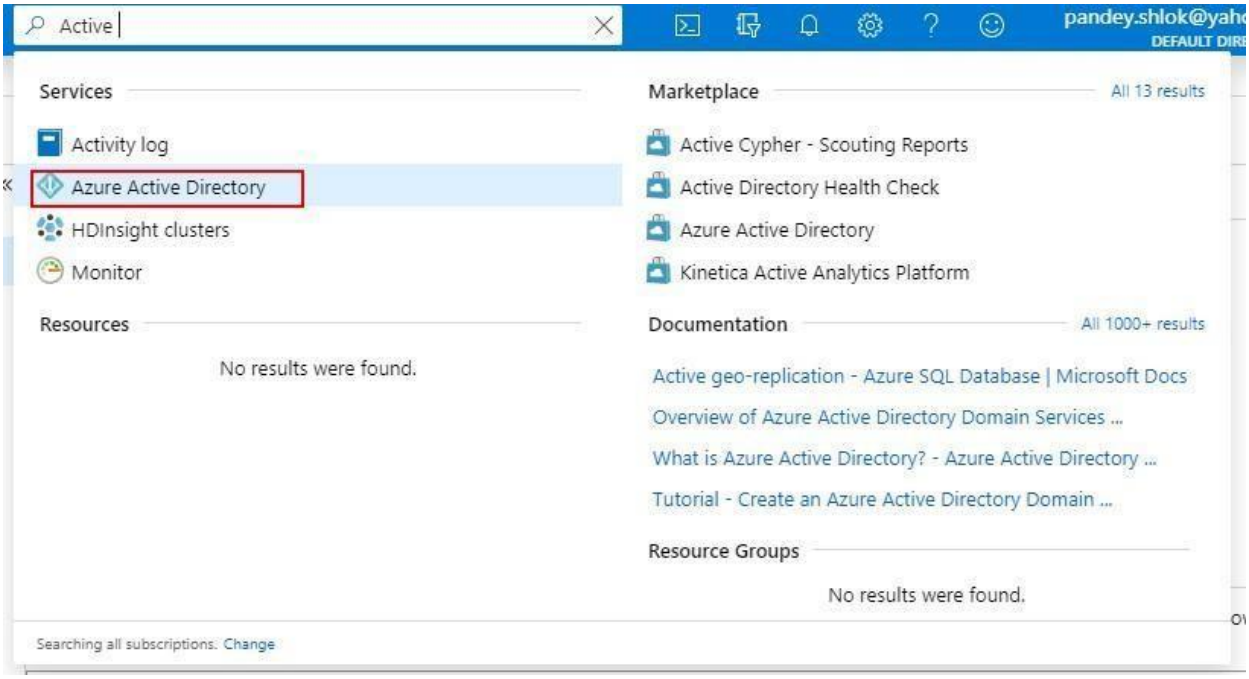
Step 13: Open load balancer and open the public IP Address



Issue 8: They wish for you to find a way to assign and manage credentials for Azure for all 10 employees in the company

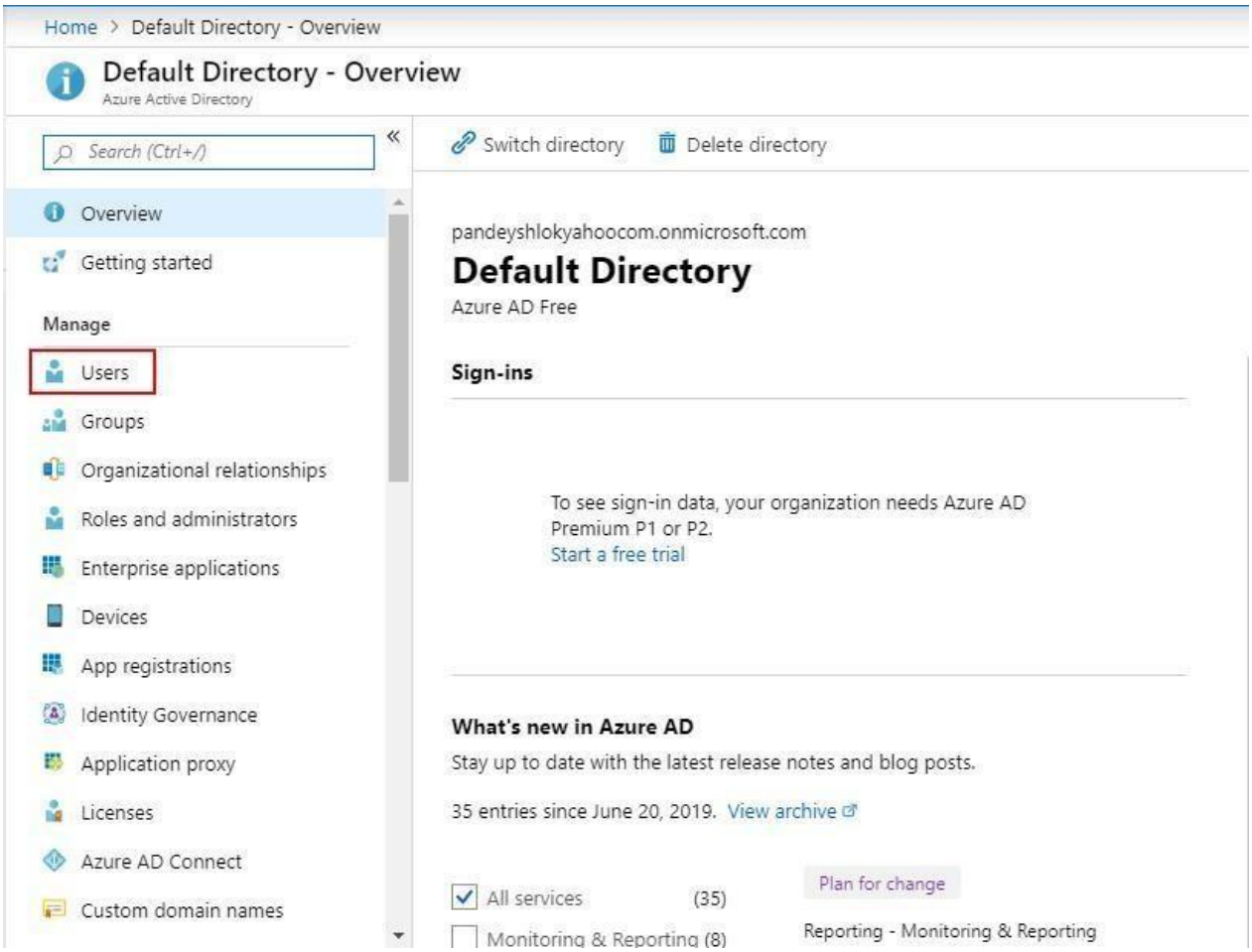
Solution:

Step 1: In Azure Portal search for Azure Active Directory

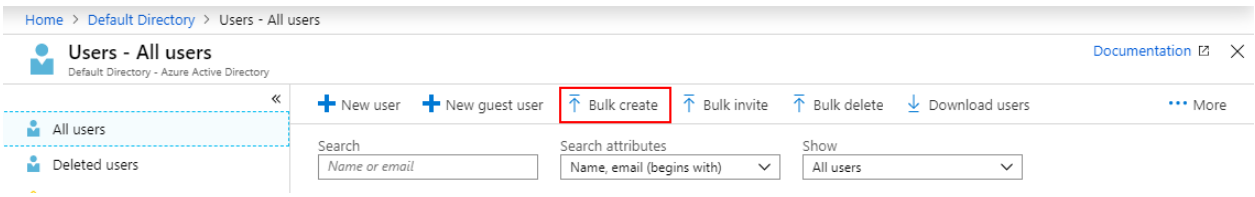


Step 2: Click on it and open it

Step 3: In the sidebar, click on Users

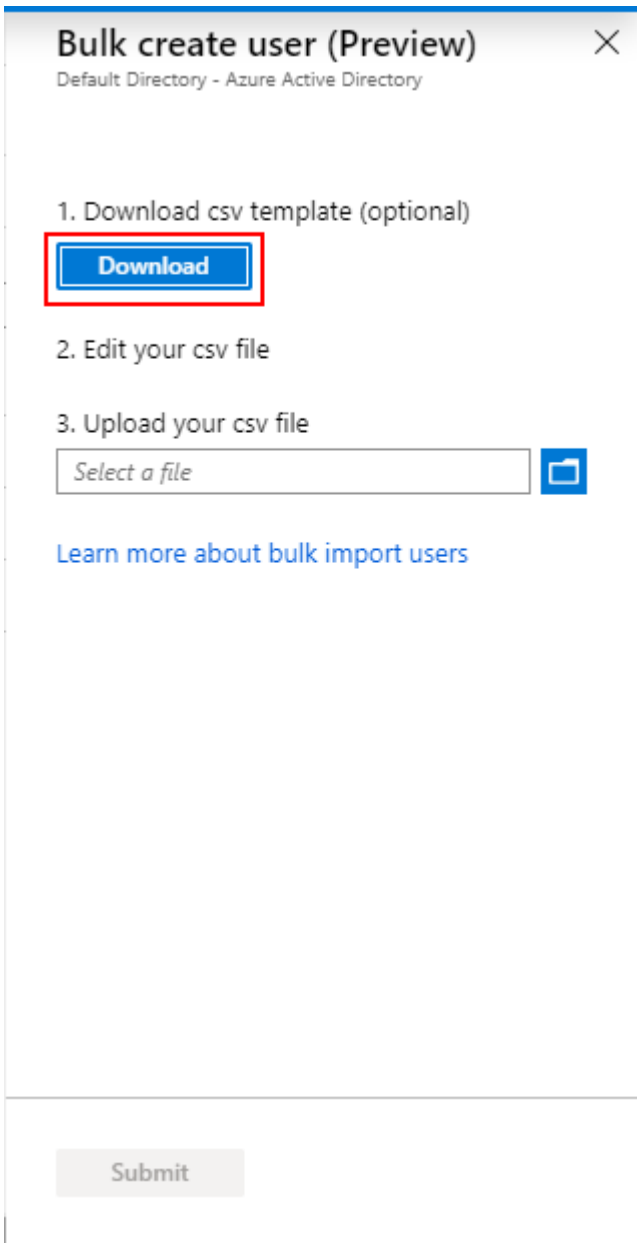


Step 4: Click on Bulk Create

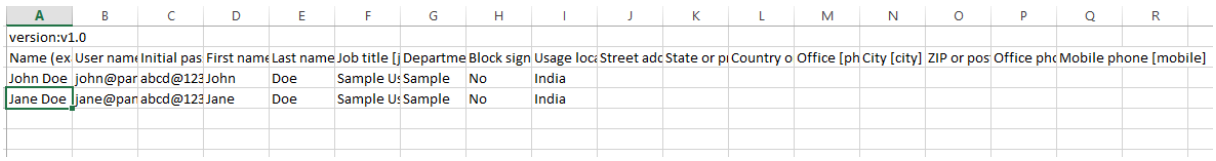


Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Step 5: Click on Download CSV Template



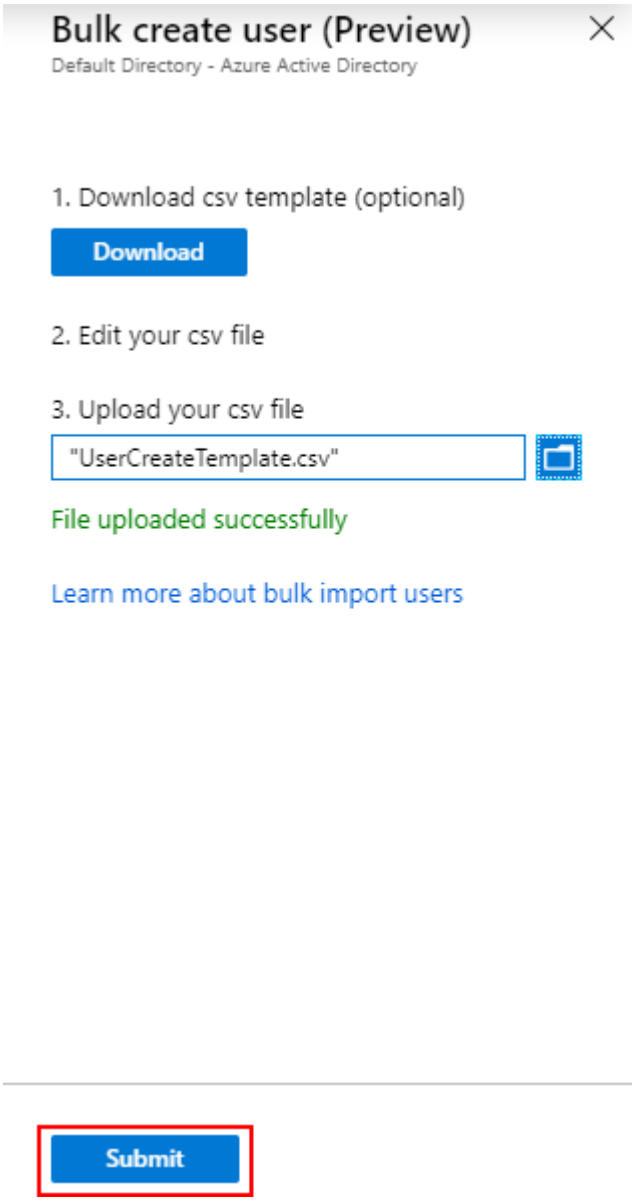
Step 6: Edit the CSV Template and add details of users



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	version:v1.0																	
2	Name (ex	User nam	Initial pas	First name	Last name	Job title [j	Departme	Block sign	Usage loc	Street adc	State or pi	Country o	Office [ph	City [city]	ZIP or pos	Office phc	Mobile phone [mobile]	
3	John Doe	john@par	abcd@123	John	Doe	Sample Us	Sample	No	India									
4	Jane Doe	jane@par	abcd@123	Jane	Doe	Sample Us	Sample	No	India									

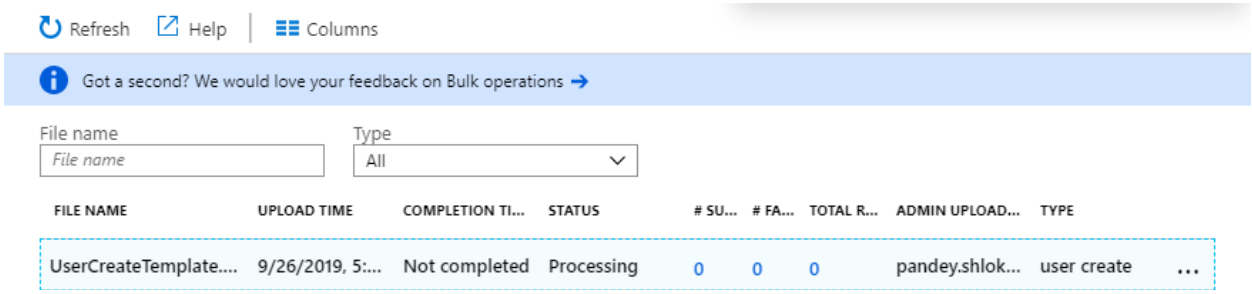
Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

Step 7: Upload the CSV File

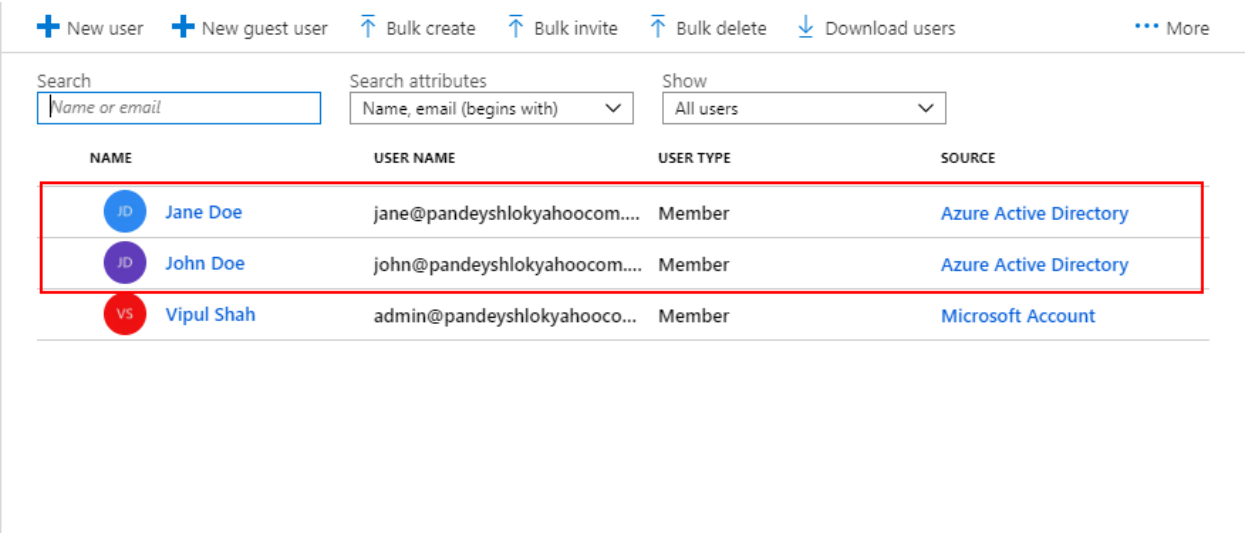


Step 8: Click on Submit

Step 9: In the sidebar click on Bulk Operation Details to view the status



Step 10: After the operation is successful, open the User page to view the created users



Contact here: ankitranjanjobs@gmail.com LinkedIn: https://www.linkedin.com/in/ankit-ranjan05/

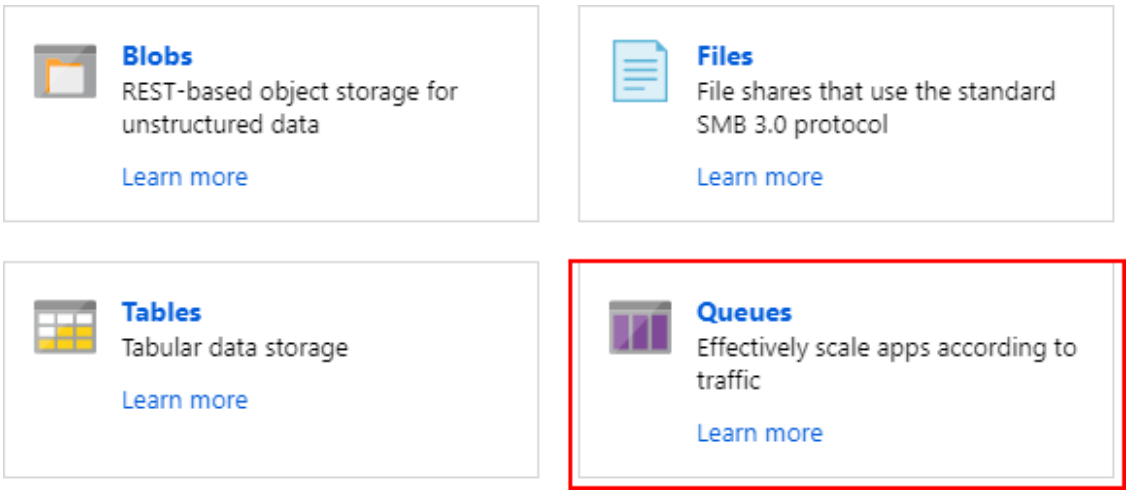
Issue 9: Finally they have two applications that need to pass messages between one another on an on-demand basis i.e. an application will send the message and another application will receive and process it when it can. You need to set up a service in such a way that these applications can do so (you are provided with the code). All you need to do is make changes to the config.js file.

Solution:

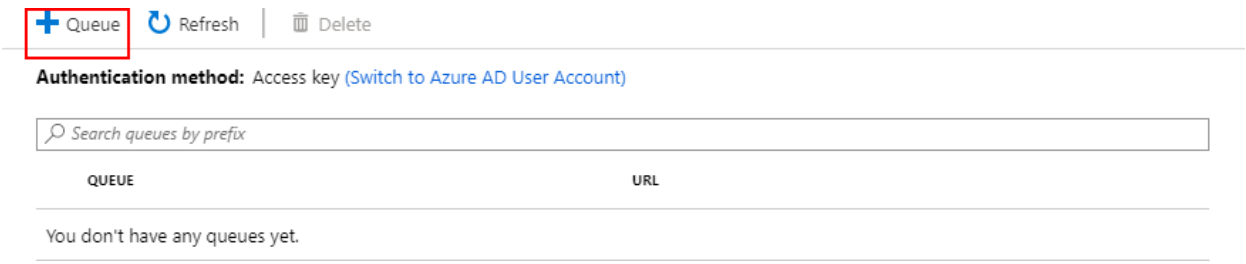
-- Make sure you have node js installed and run npm install in the project directory –

Step 1: Open Storage Account.

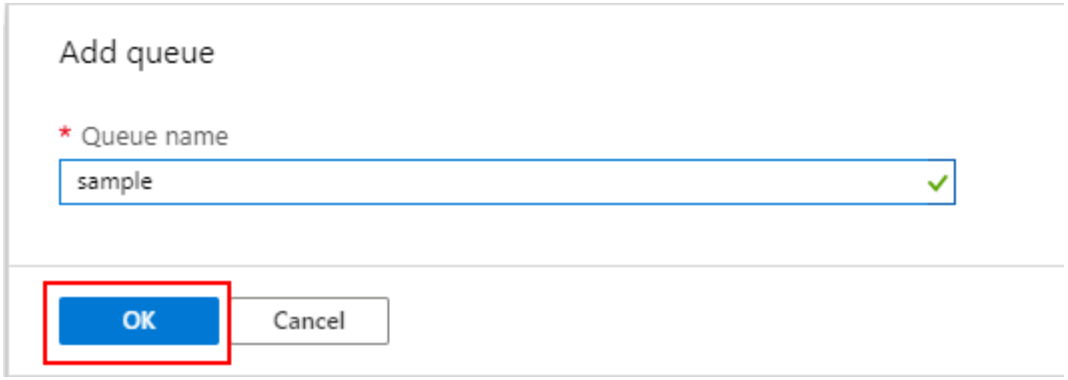
Step 2: Click on Queues.



Step 3: Click on + Queue

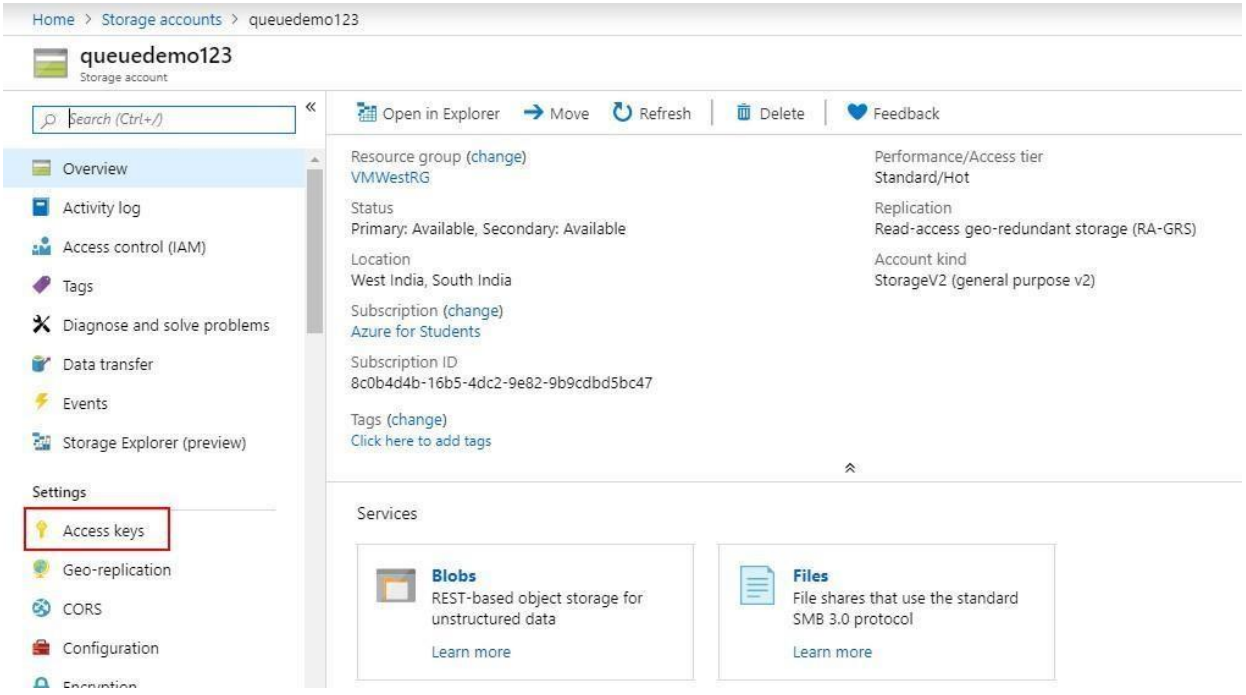


Step 4: Enter details and click OK



Step 5: Open the Storage Account

Step 6: In the Settings section click on Access Keys



Step 7: Copy one of the keys

Step 8: Open config.js file in the sample app.

Step 9: Replace information:

- queueName: Name of your Queue you just created.

- azureStorageAccount: Name of your Storage Account.

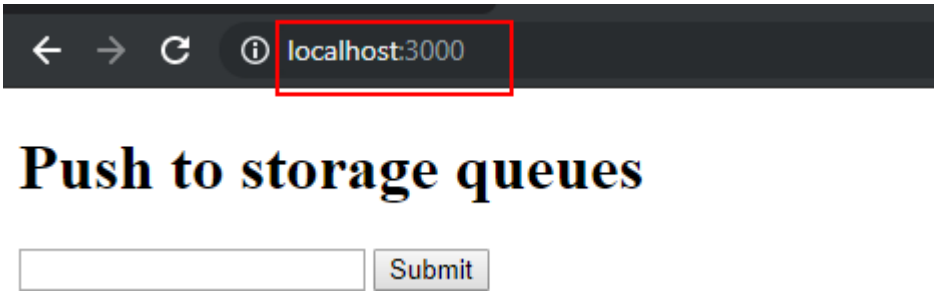
- azureStorageAccessKey: Access Key for your storage account.

```
module.exports = {
    queueName: 'sample',
    azureStorageAccount: 'queuedemo123',
    azureStorageAccessKey: 'Nyb1zpwQbMNyb/
    eGrv4qwkjVxExPu0CNM1x3nnZC0vLCMgD7q0CiLPDF9YscEBHLGm3D+o17WOBY7VCIF4KlfQ==',
};
```

Step 10: Run the app to check if it works (To run the app run the command: node server.js)

```
Intellipaat-Team@DESKTOP-FPNPKTE MINGW64 ~/Desktop/Storage Queue Demo
$ node server.js
Listening at port 3000
```

- Open http://localhost:3000



- Open http://localhost:3000/admin