Lab 06 - Implement Traffic Management Report

Yunis Mohamed

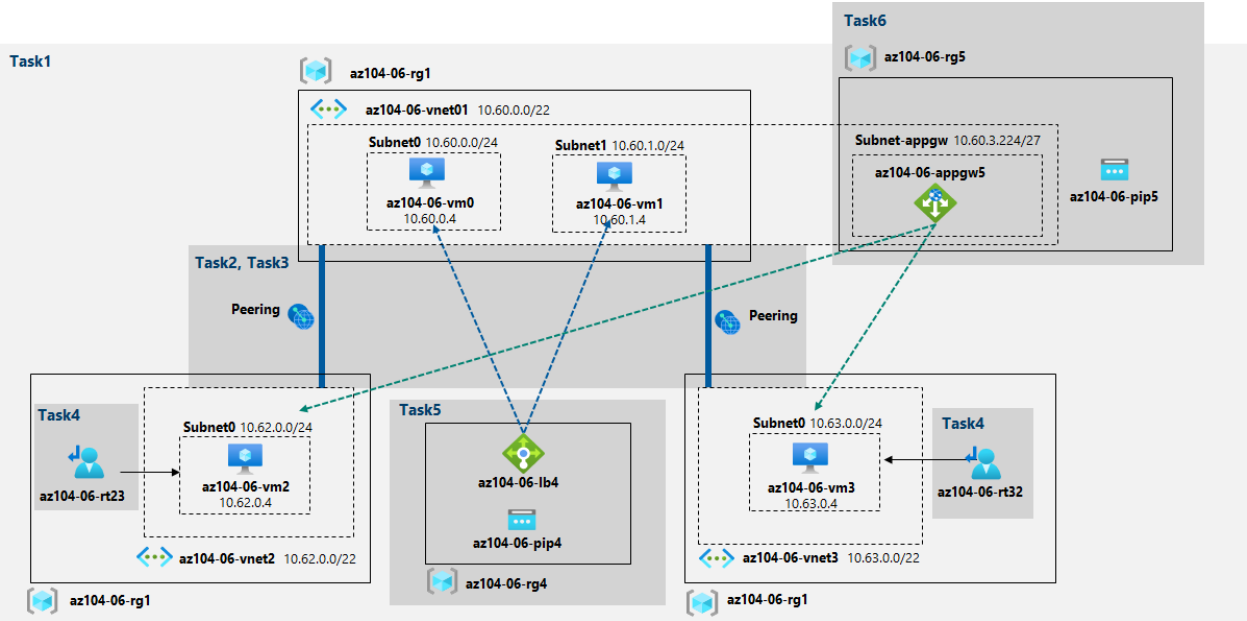
MICROSOFT AZURE LAB 06

Table of Contents

Introduction

The Implement Traffic Management Lab 06 in Azure comprehensive lab provides an environment to practices and gain knowledge on cloud Traffic Management. Through a series of tasks designed in the lab I will be able to complete the Lab's objective. Task 1 focuses on provisioning the lab environment, ensuring I have everything necessary to proceed. Task 2 involves configuring the hub and spoke network topology to establish a robust and scalable network foundation. Task 3 tests the connectivity between different virtual networks using virtual network peering. Task 4 involves optimizing the flow of network traffic by configuring routing in the hub and spoke topology. Task 5 provides an opportunity to implement Azure Load Balancer, a high-performance load balancing solution. Finally, in Task 6, I will explore Azure Application Gateway, a powerful application delivery controller. By completing these tasks, I hope to gain valuable knowledge and hands-on experience in implementing traffic management techniques in Azure.

Architecture diagram



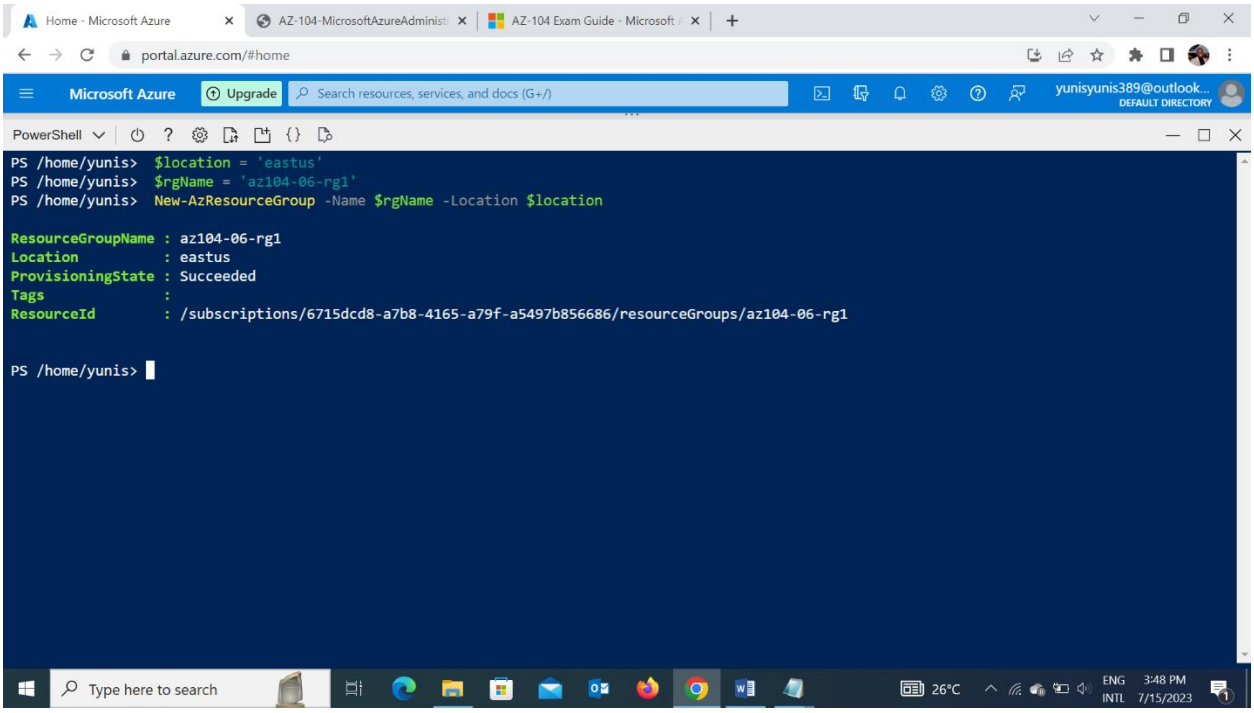
Task 1: Provision the lab environment

In this task, I will deploy four virtual machines into the same Azure region. The first two will reside in a hub virtual network, while each of the remaining two will reside in a separate spoke virtual network.

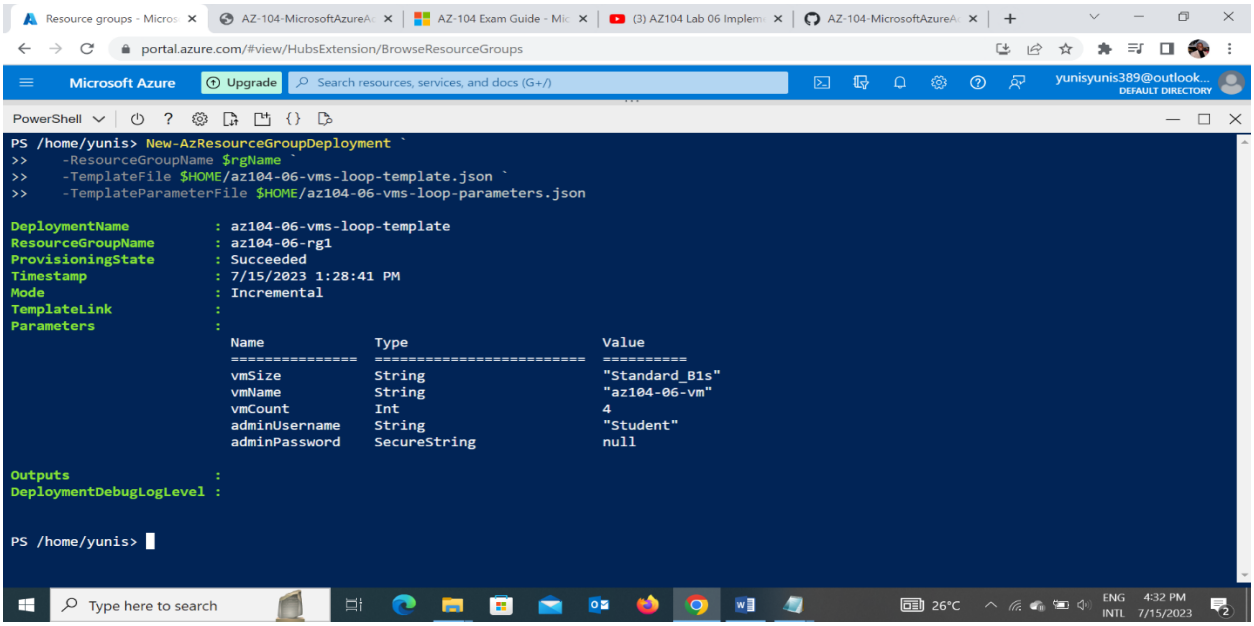
1. Sign in to the Azure portal.
2. In the Azure portal, open the **Azure Cloud Shell** by clicking on the icon in the top right of the Azure Portal.
3. If prompted to select either **Bash** or **PowerShell**, select **PowerShell**.
4. In the toolbar of the Cloud Shell pane, click the Upload/Download files icon, in the drop-down menu, click Upload and upload the files \Allfiles\Labs\06\az104-06-vms-loop-

template.json and \Allfiles\Labs\06\az104-06-vms-loop-parameters.json into the Cloud Shell home directory.

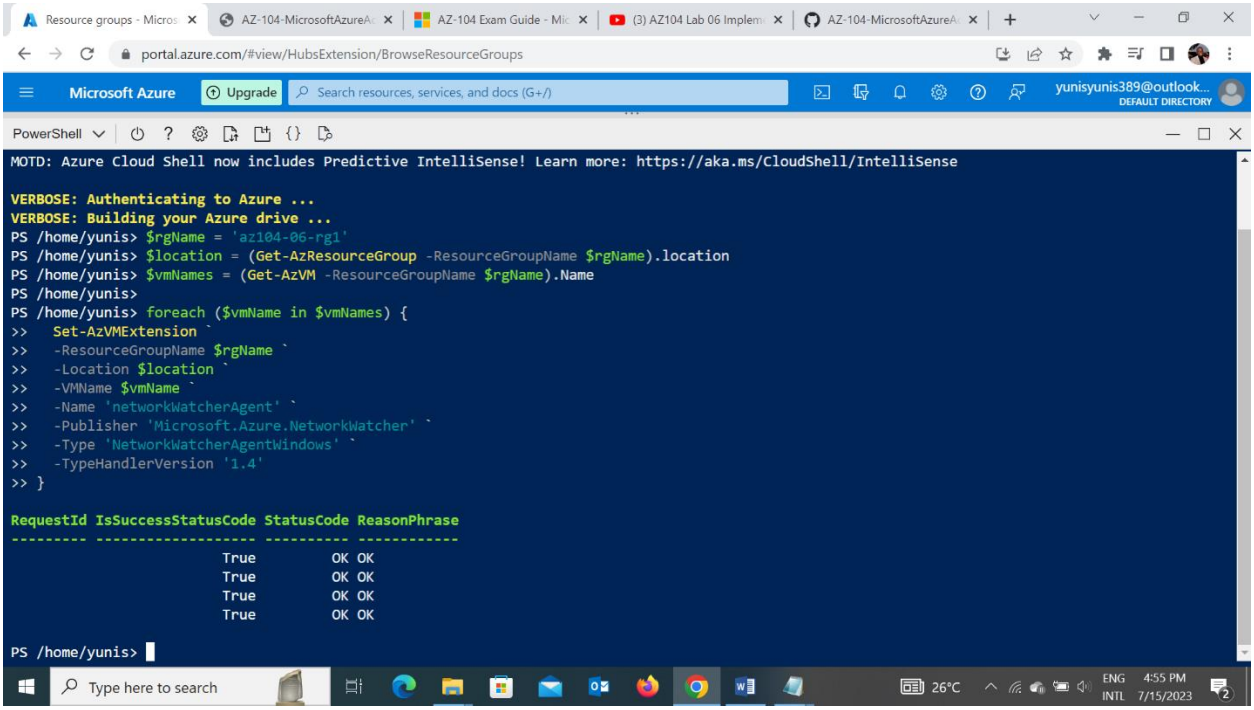
5. From the Cloud Shell pane, run the following to create the first resource group that will be hosting the lab environment (replace the '[Azure_region]' placeholder with the name of an Azure region where you intend to deploy Azure virtual machines)(you can use the "(Get-AzLocation).Location" cmdlet to get the region list):



6. From the Cloud Shell pane, run the following to create the three virtual networks and four Azure VMs into them by using the template and parameter files you uploaded:



7. From the Cloud Shell pane, run the following to install the Network Watcher extension on the Azure VMs deployed in the previous step:

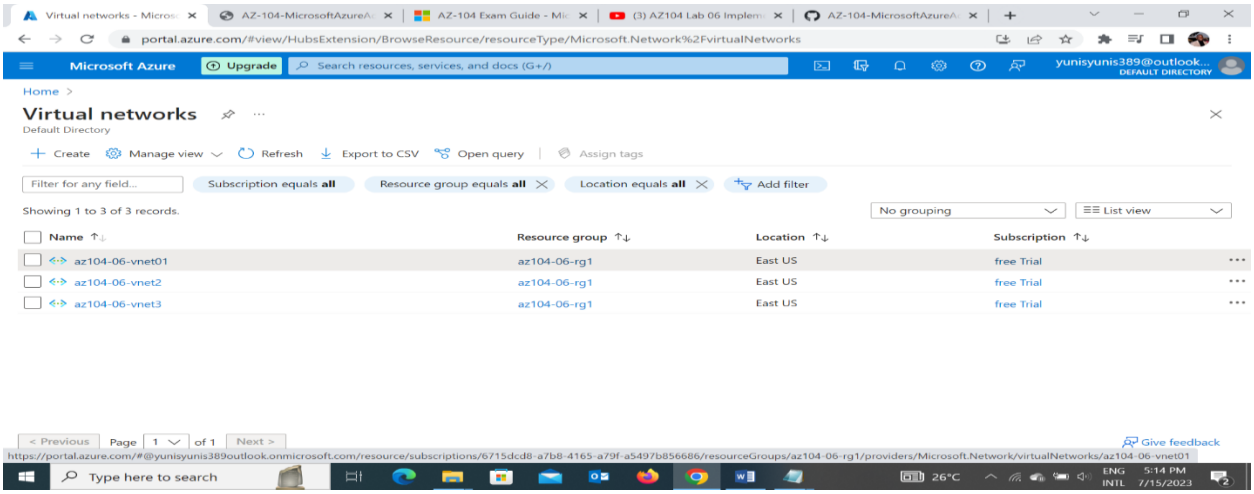


8. Close the Cloud Shell pane.

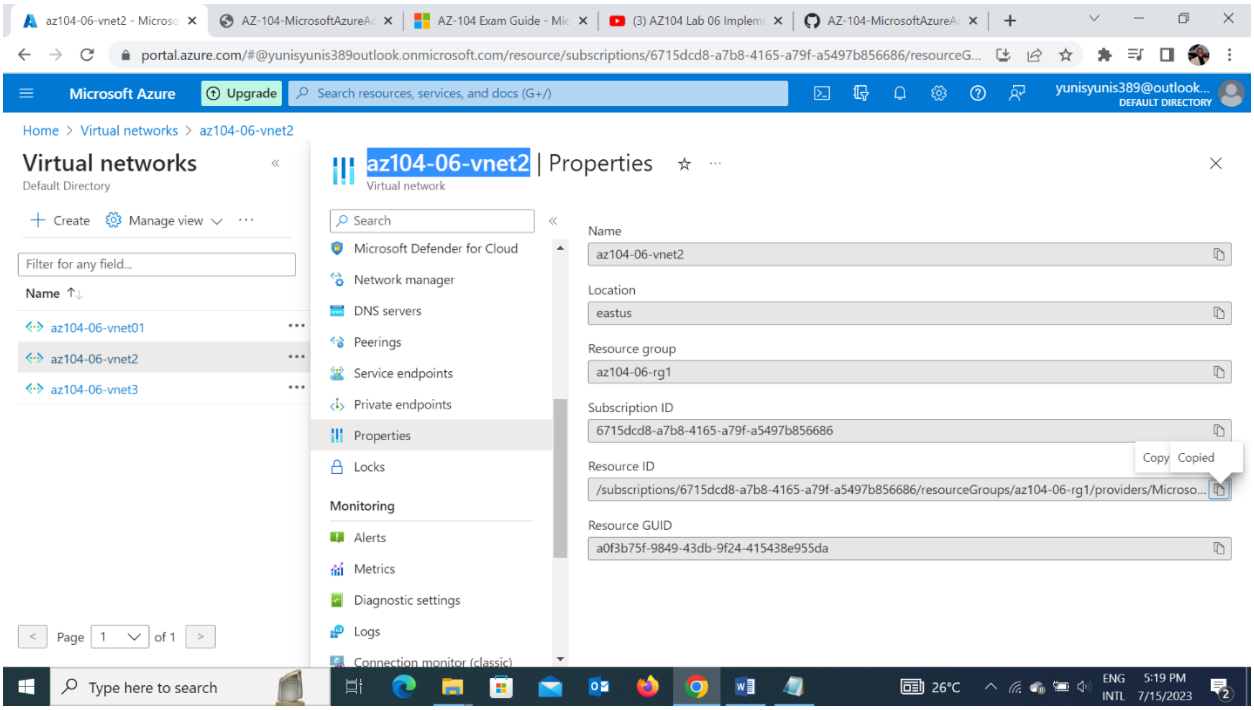
Task 2: Configure the hub and spoke network topology

In this task, I will configure local peering between the virtual networks you deployed in the previous tasks in order to create a hub and spoke network topology.

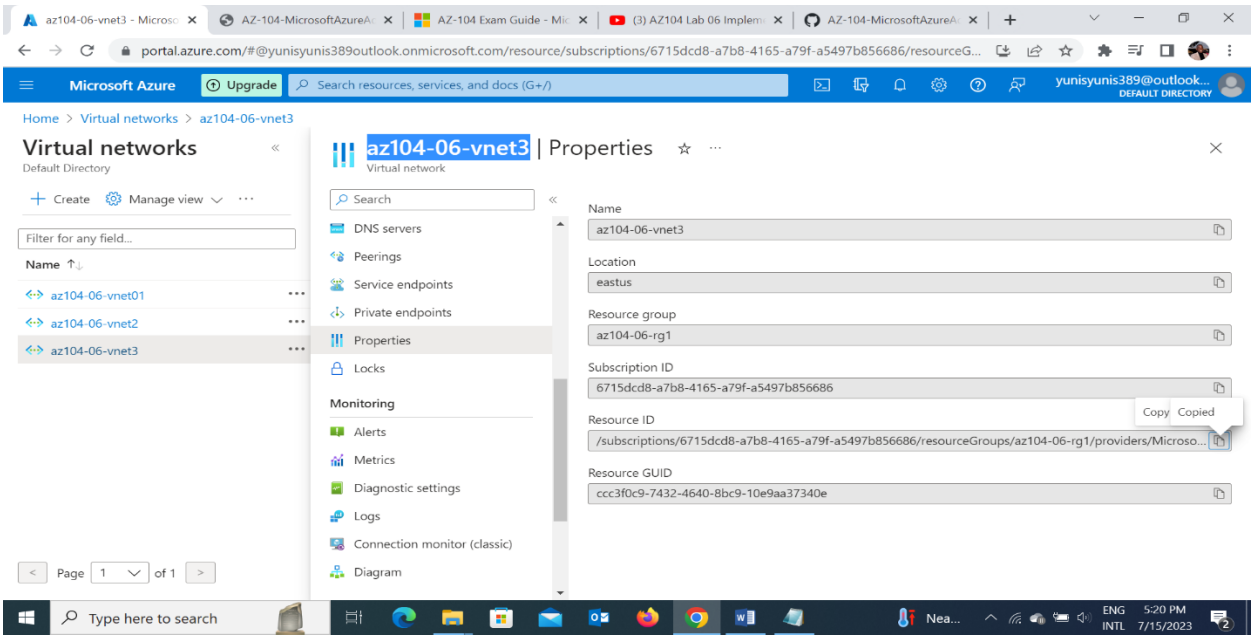
1. In the Azure portal, search for and select **Virtual networks**.
2. Review the virtual networks you created in the previous task.
3. In the list of virtual networks, select **az104-06-vnet2.**



4. On the **az104-06-vnet2** blade, select Properties.
5. On the **az104-06-vnet2** | Properties blade, record the value of the Resource ID property.



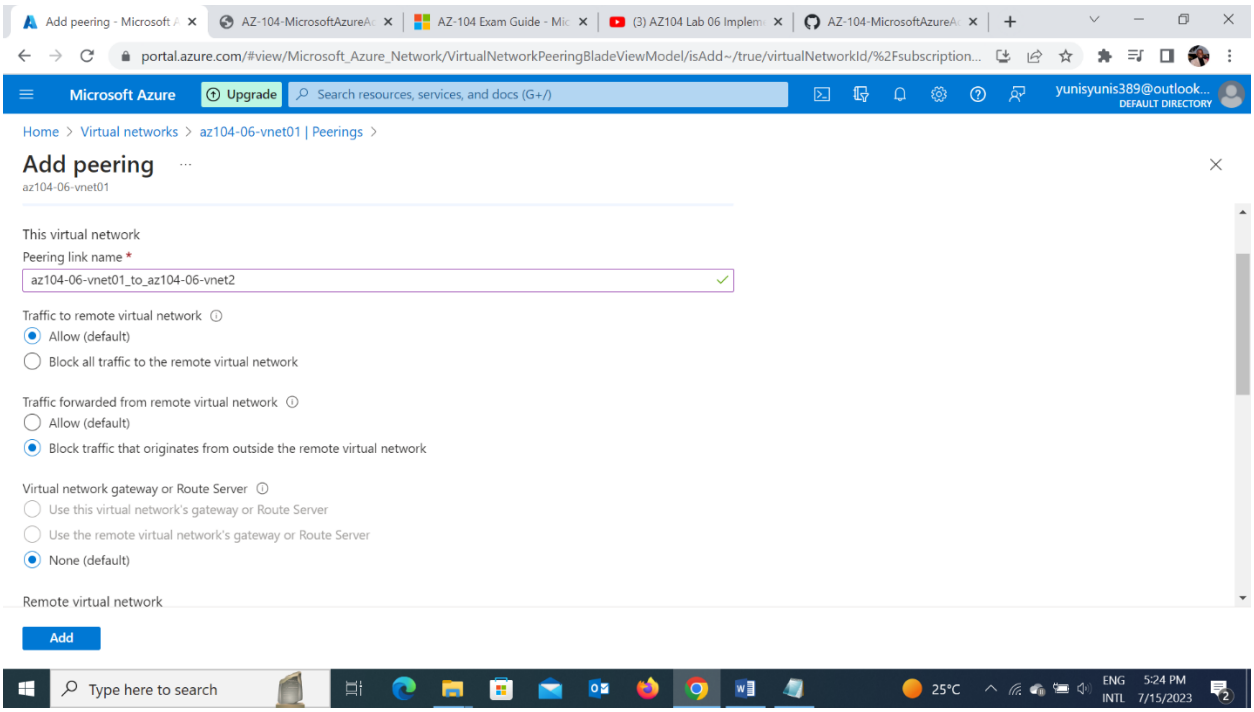
6. Navigate back to the list of virtual networks and select az104-06-vnet3.
7. On the **az104-06-vnet3** blade, select Properties.
8. On the **az104-06-vnet3** | Properties blade, record the value of the Resource ID property.

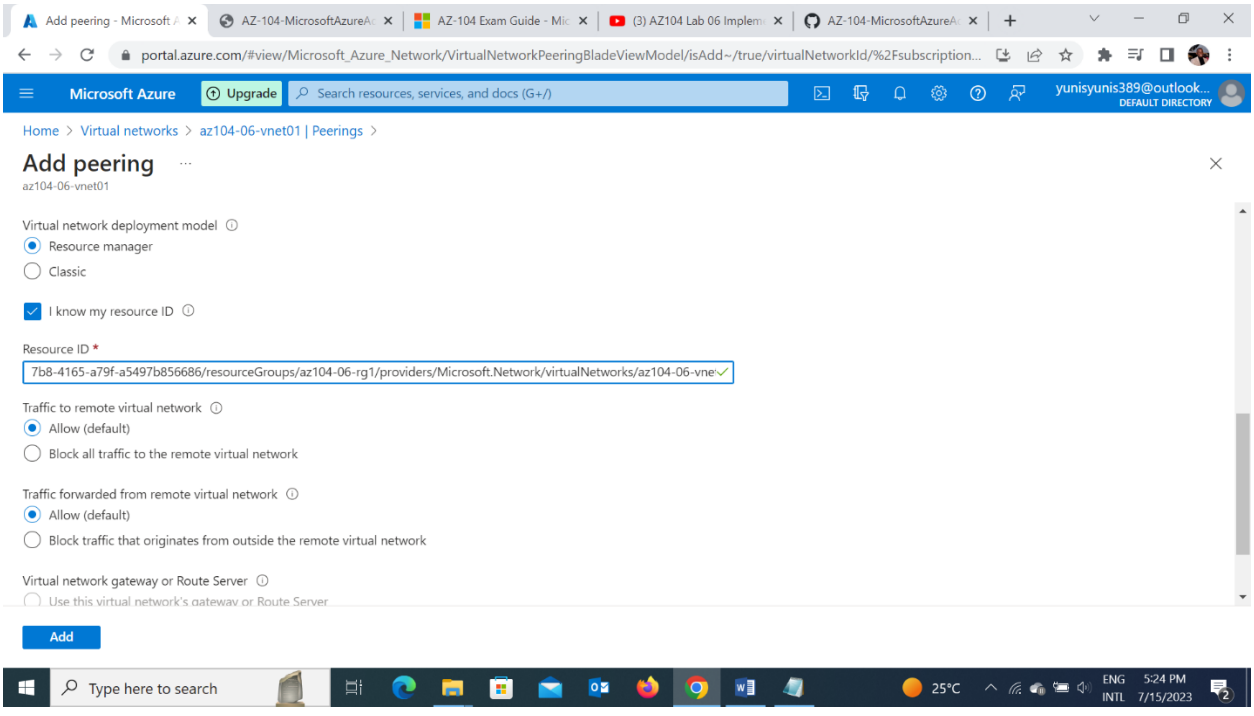


9. In the list of virtual networks, click **az104-06-vnet01**.

10. On the **az104-06-vnet01** virtual network blade, in the Settings section, click Peerings and then click + Add.

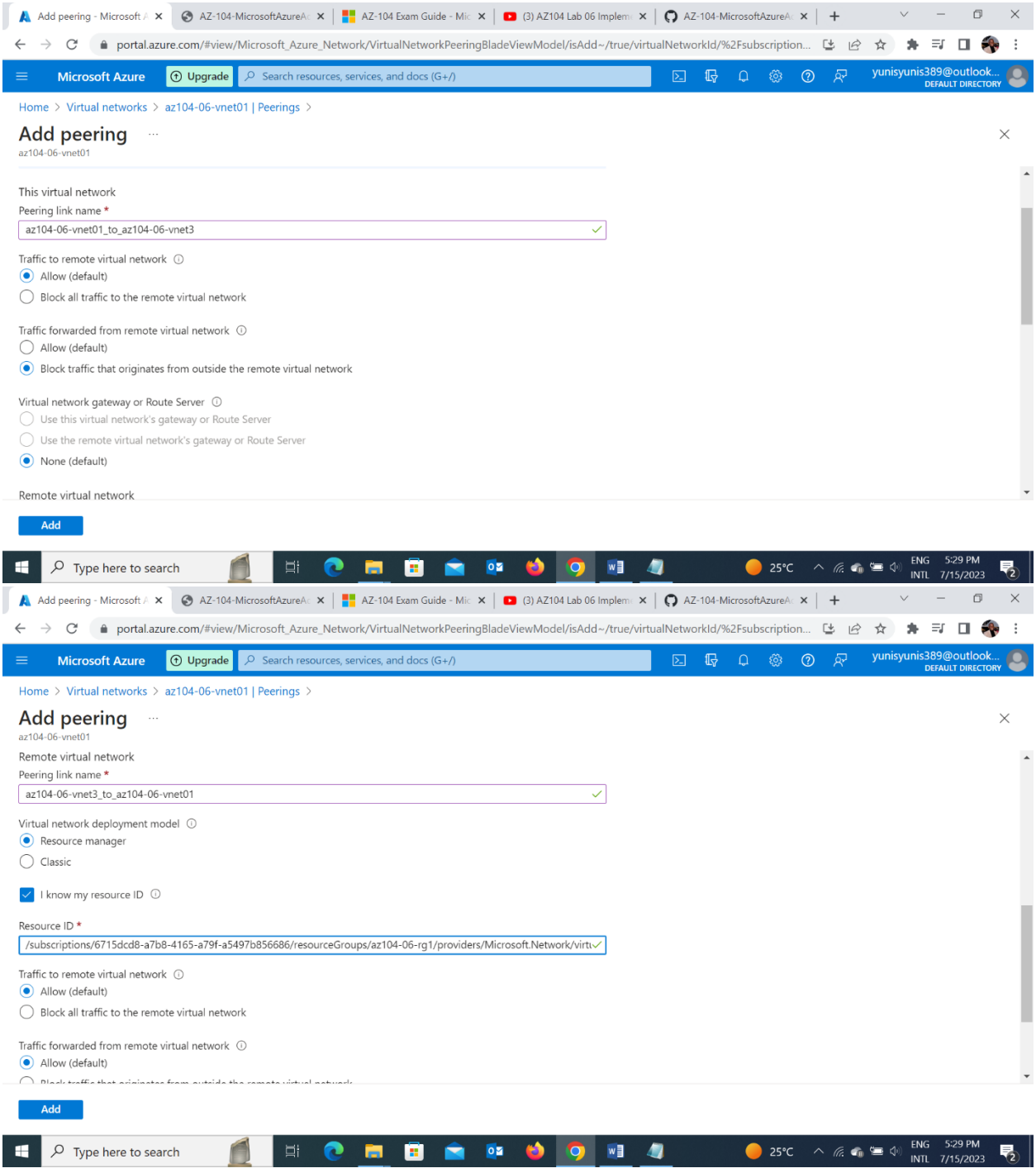
11. Add a peering with the following settings (leave others with their default values) and click Add:





12. On the **az104-06-vnet01** virtual network blade, in the Settings section, click Peerings and then click + Add.

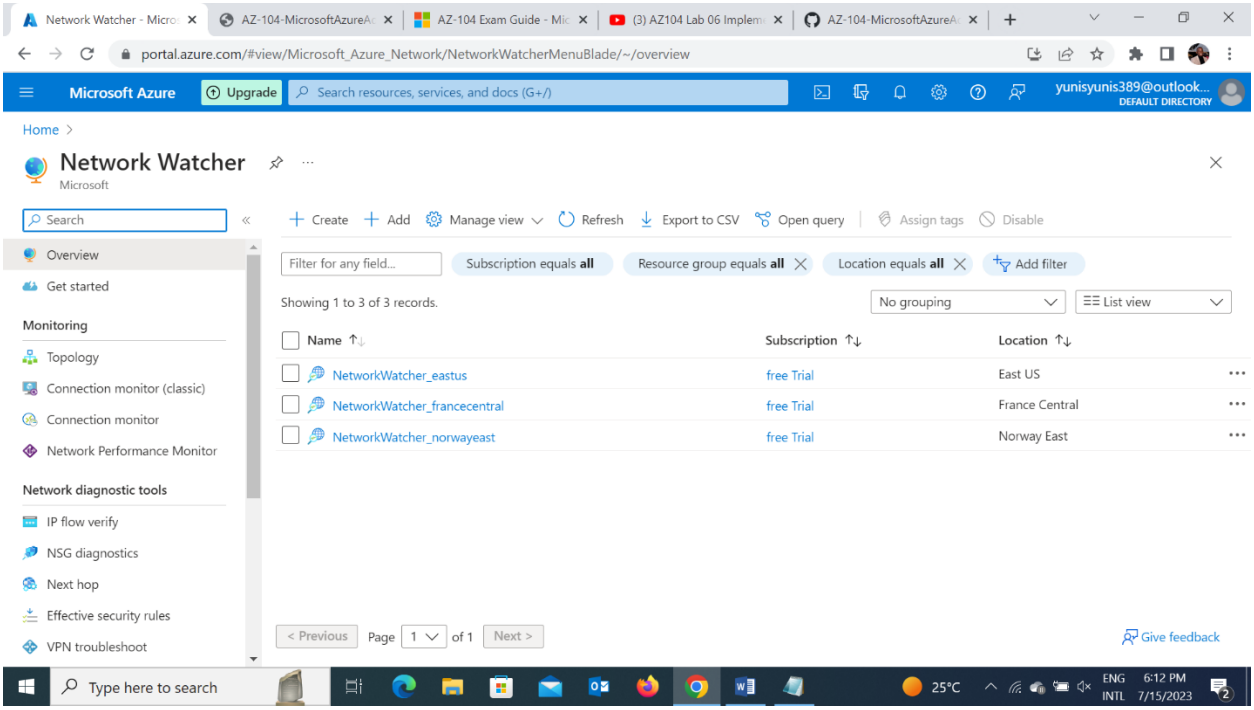
13. Add a peering with the following settings (leave others with their default values) and click Add:



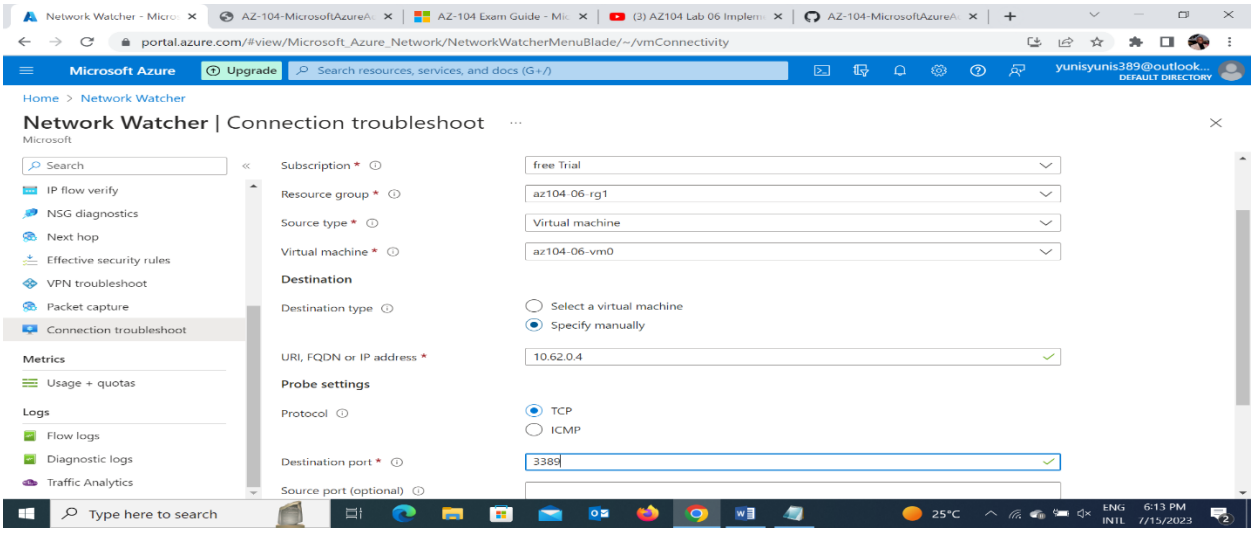
Task 3: Test transitivity of virtual network peering

In this task, I will test transitivity of virtual network peering by using Network Watcher.

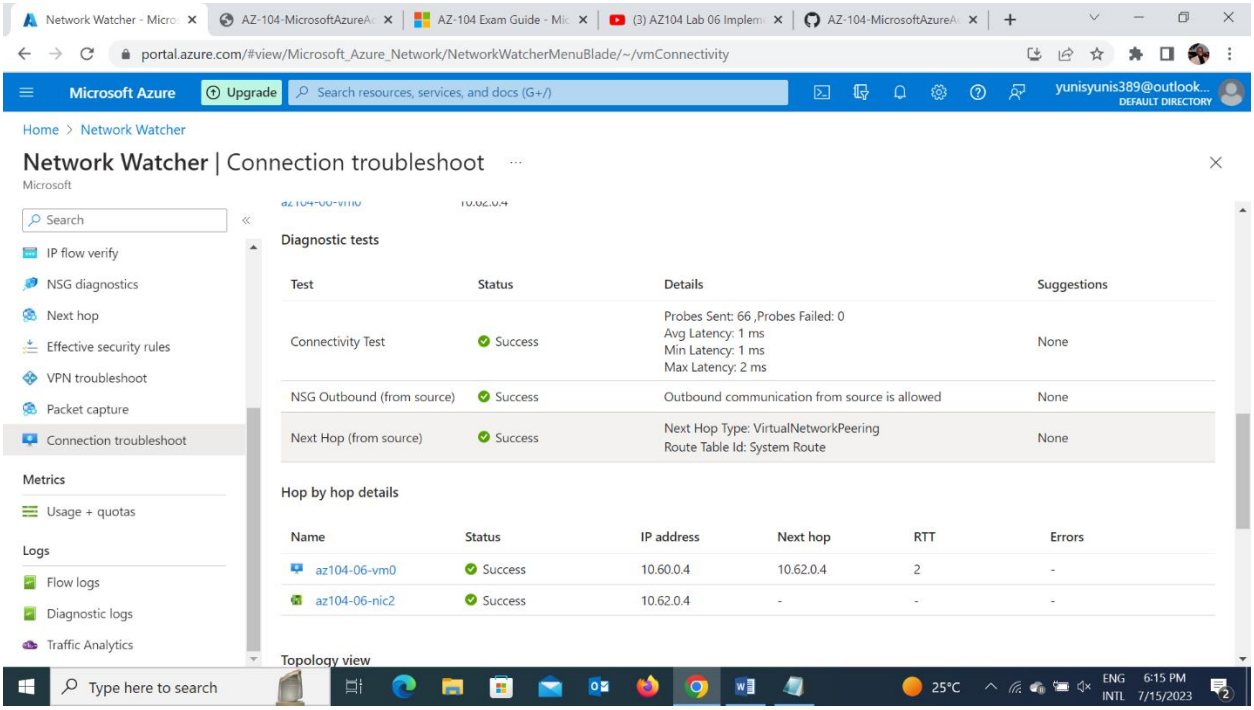
1. In the Azure portal, search for and select **Network Watcher**.
2. On the **Network Watcher** blade, expand the listing of Azure regions and verify the service is enabled in region you are using.



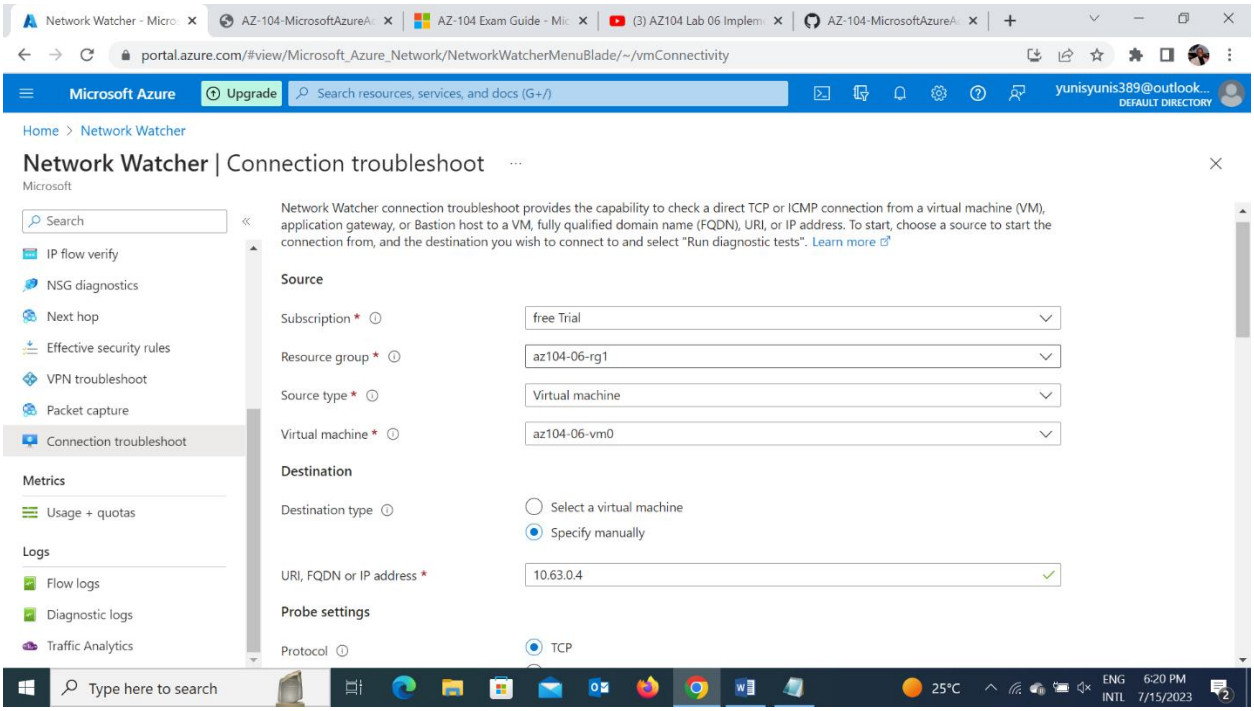
3. On the **Network Watcher** blade, navigate to the **Connection troubleshoot**.
4. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings (leave others with their default values): **10.62.0.4** represents the private IP address of **az104-06-vm2**



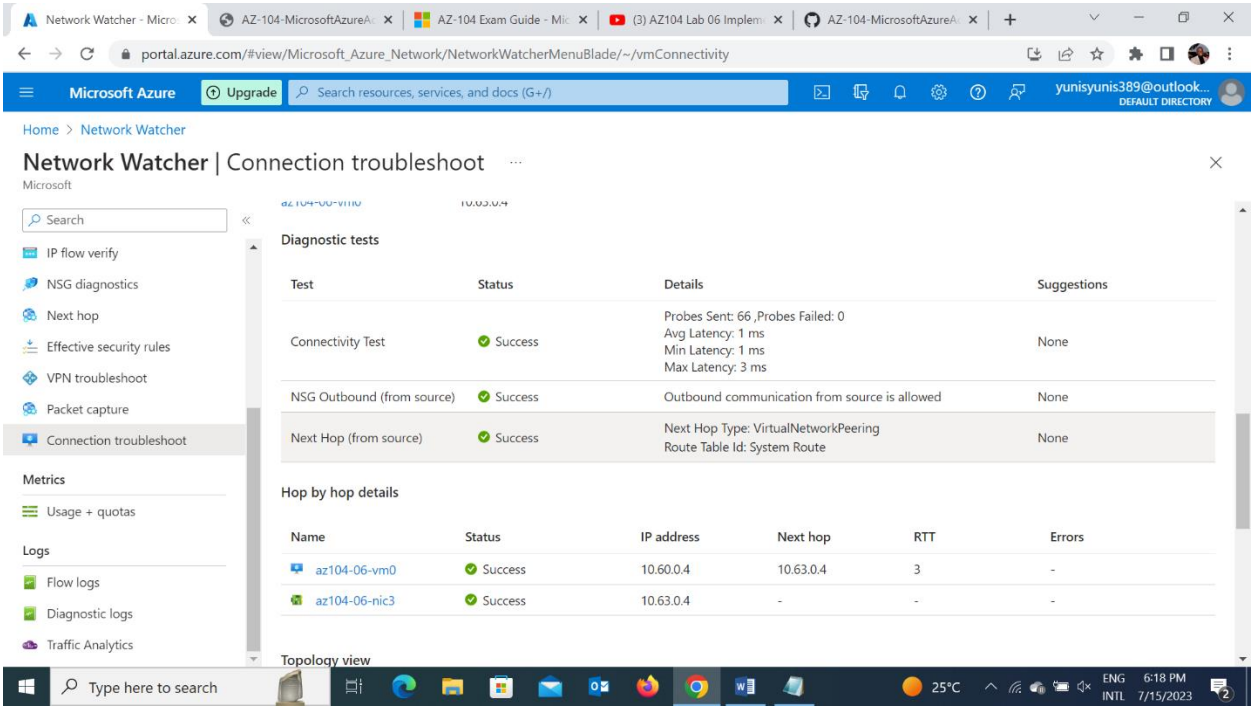
5. Click **Run diagnostic tests** and wait until results of the connectivity check are returned. Verify that the status is **Success**. Review the network path and note that the connection was direct, with no intermediate hops in between the VMs.



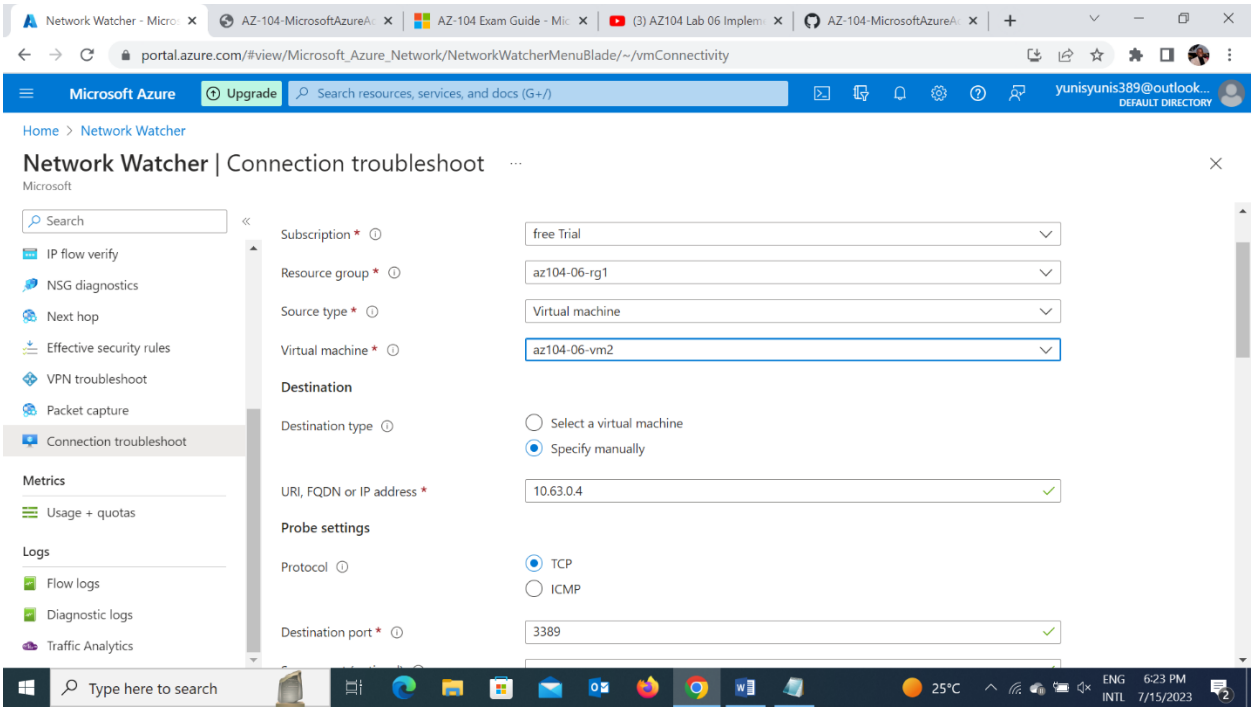
6. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings (leave others with their default values): **10.63.0.4** represents the private IP address of **az104-06-vm3**



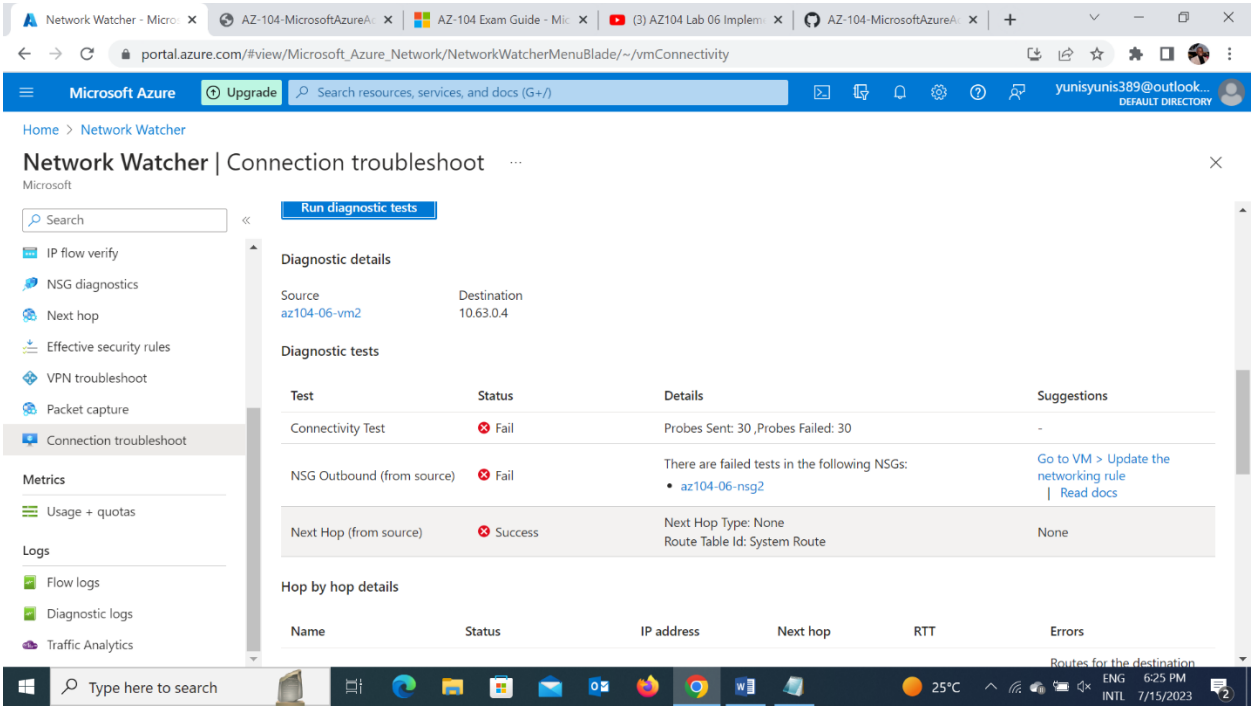
7. Click **Run diagnostic tests** and wait until results of the connectivity check are returned. Verify that the status is **Success**. Review the network path and note that the connection was direct, with no intermediate hops in between the VMs.



8. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings (leave others with their default values):



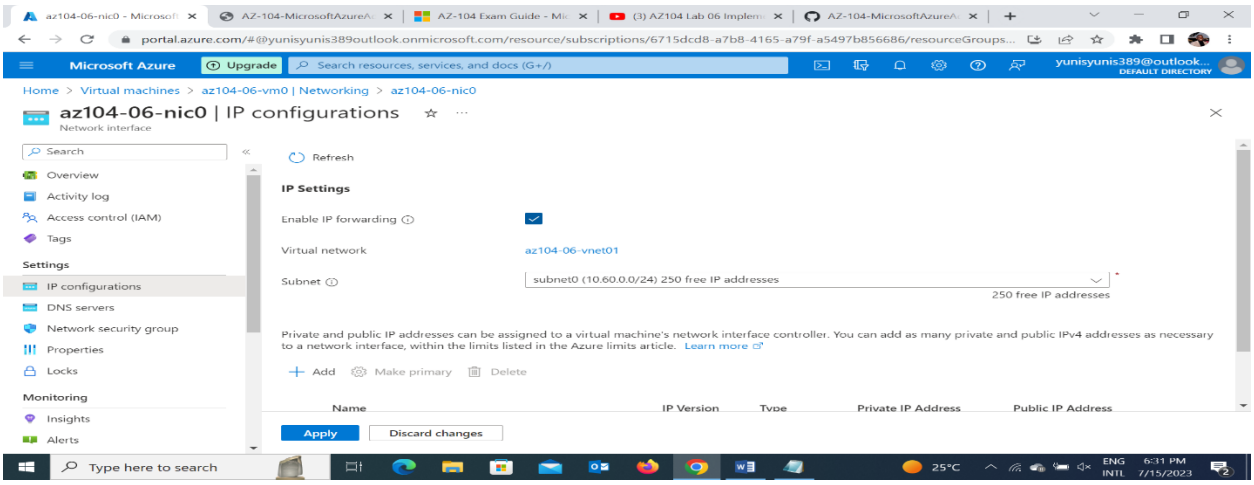
9. Click **Run diagnostic tests** and wait until results of the connectivity check are returned. Note that the status is **Fail**. ***This is expected, since the two spoke virtual networks are not peered with each other (virtual network peering is not transitive).***



Task 4: Configure routing in the hub and spoke topology

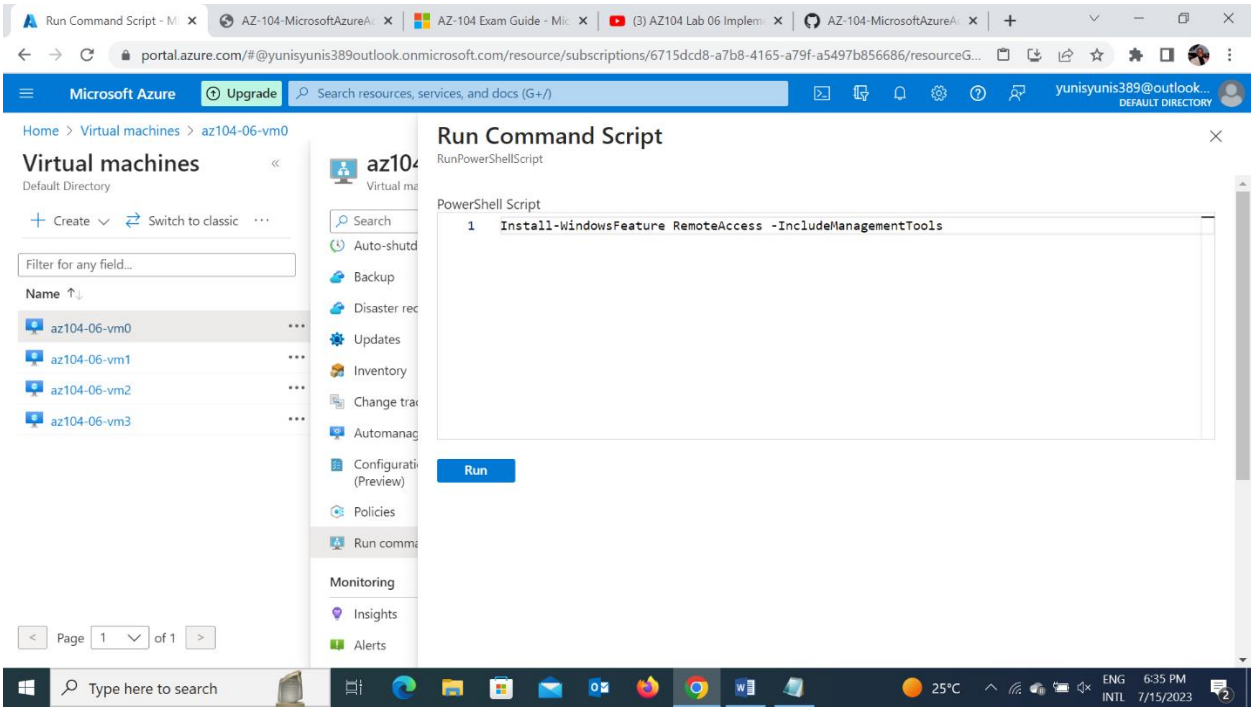
In this task, I will configure and test routing between the two spoke virtual networks by enabling IP forwarding on the network interface of the **az104-06-vm0** virtual machine, enabling routing within its operating system, and configuring user-defined routes on the spoke virtual network.

1. In the Azure portal, search and select **Virtual machines**.
2. On the **Virtual machines** blade, in the list of virtual machines, click **az104-06-vm0**.
3. On the **az104-06-vm0** virtual machine blade, in the **Settings** section, click **Networking**.
4. Click the **az104-06-nic0** link next to the **Network interface** label, and then, on the **az104-06-nic0** network interface blade, in the **Settings** section, click **IP configurations**.
5. Set **IP forwarding** to **Enabled** and save the change. ***This setting is required in order for az104-06-vm0 to function as a router, which will route traffic between two spoke virtual networks.***

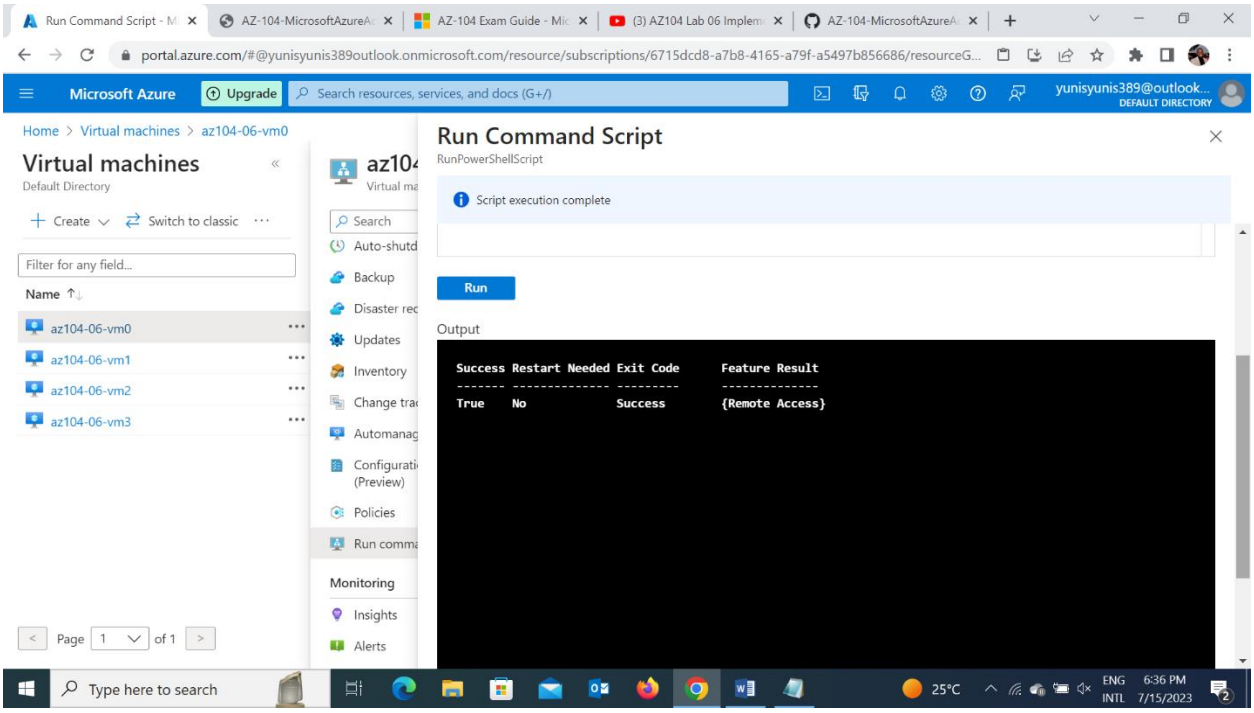


6. In the Azure portal, navigate back to the az104-06-vm0 Azure virtual machine blade and click Overview.

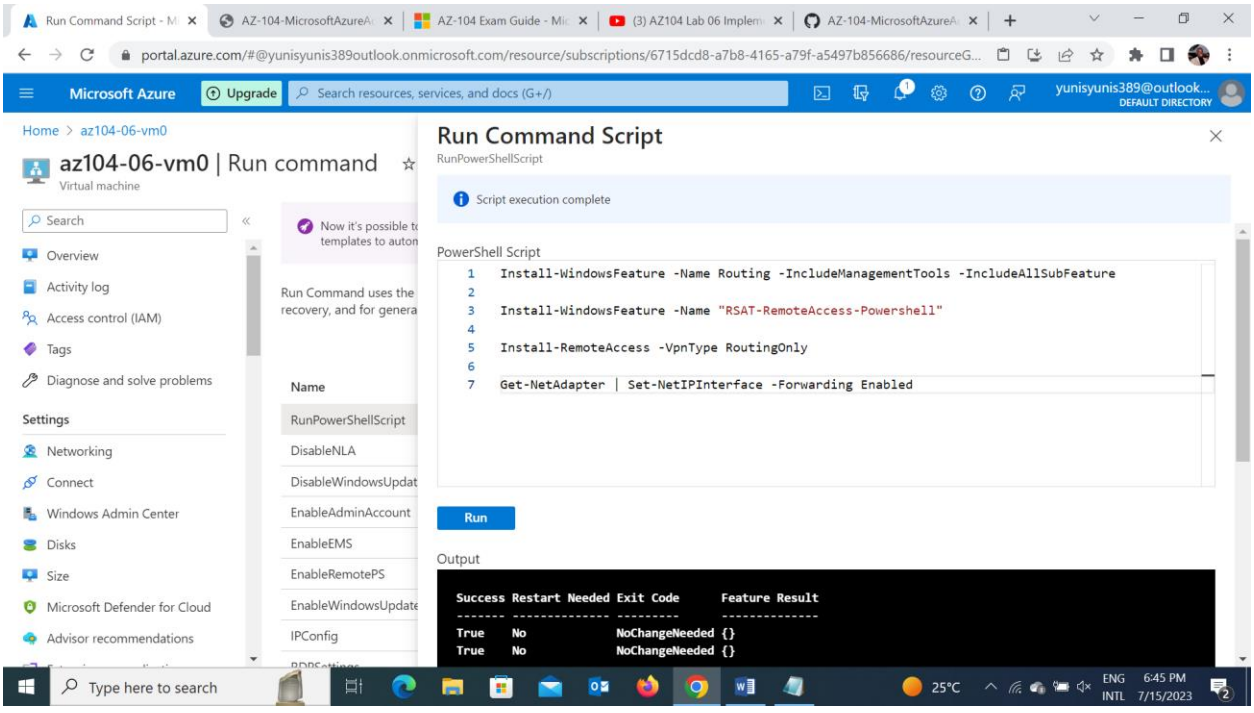
7. On the az104-06-vm0 blade, in the Operations section, click Run command, and, in the list of commands, click RunPowerShellScript.



8. On the Run Command Script blade, type the following and click Run to install the Remote Access Windows Server role.

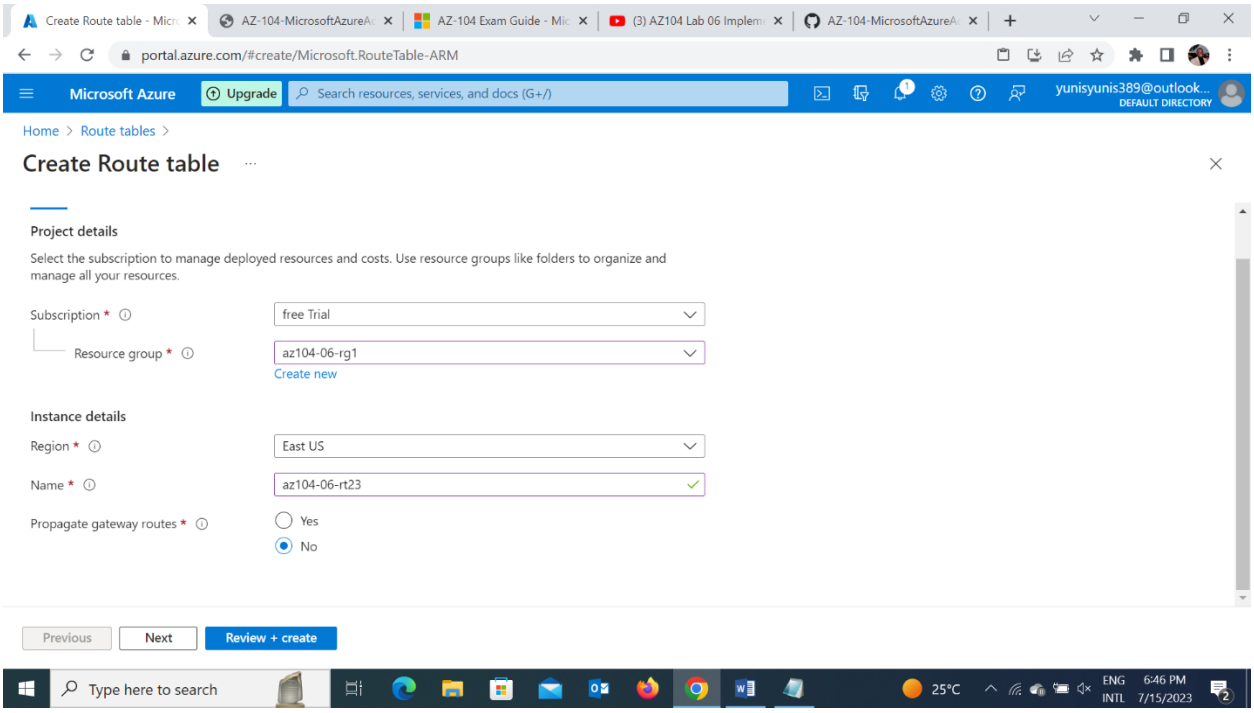


9. On the **Run Command Script** blade, type the following and click **Run** to install the Routing role service.

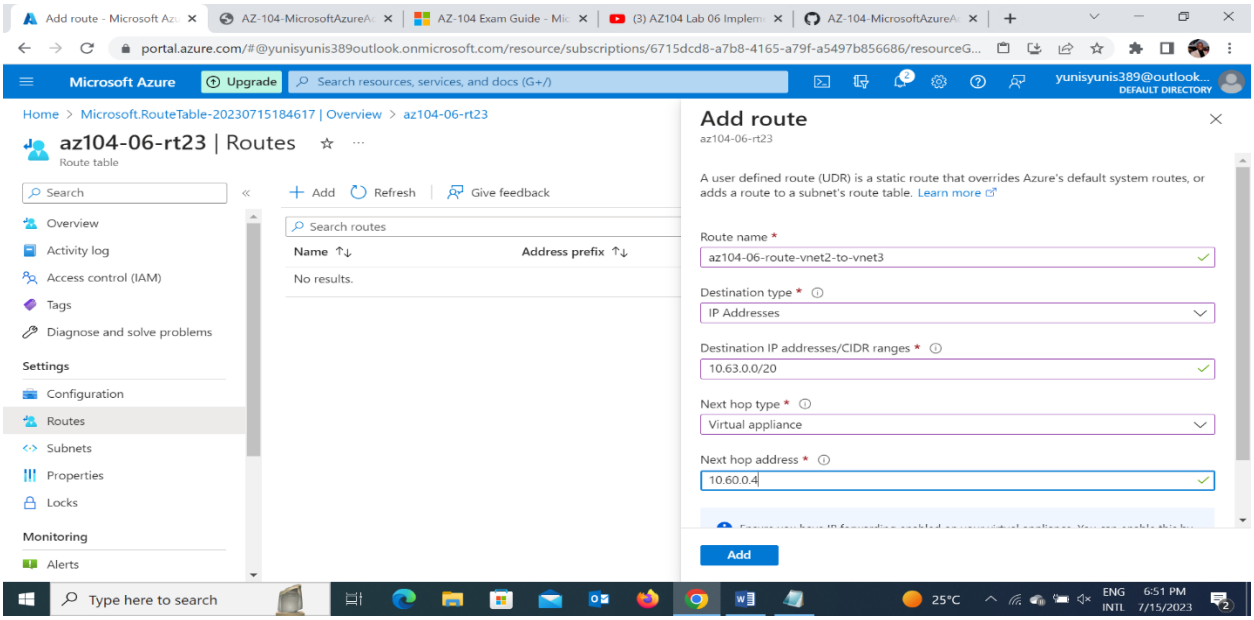


10. In the Azure portal, search and select Route tables and, on the Route tables blade, click + Create.

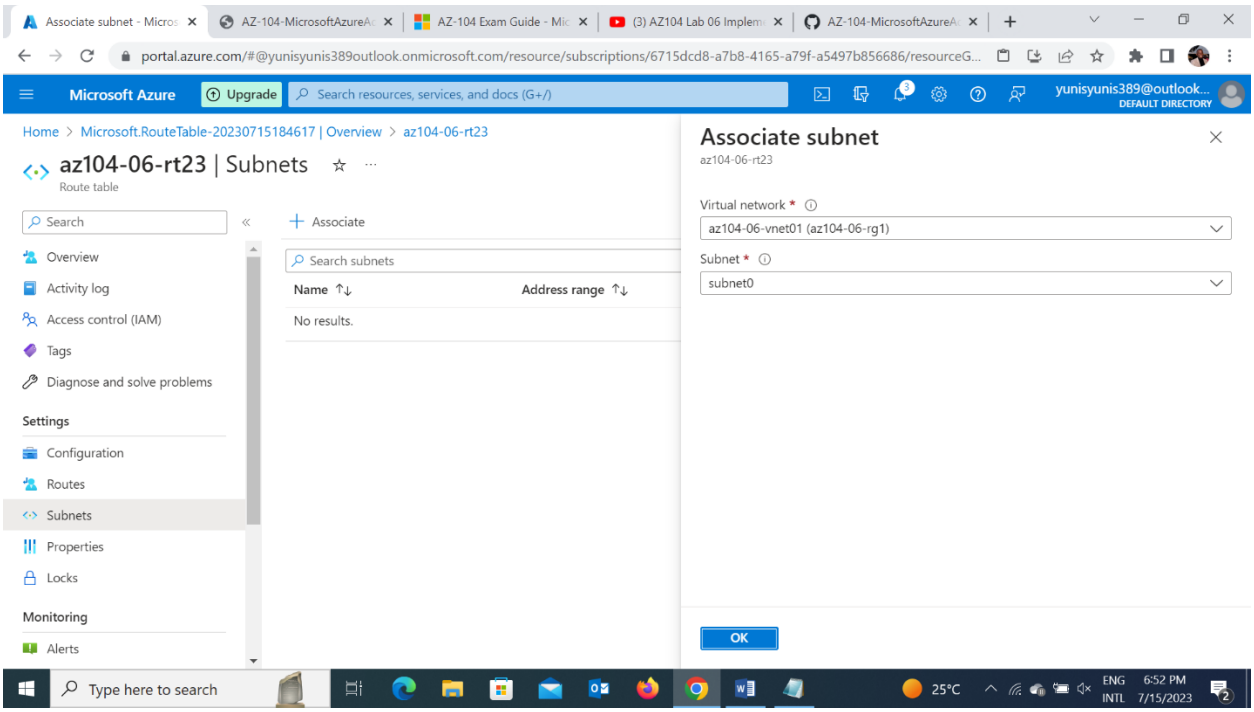
11. Create a route table with the following settings (leave others with their default values):



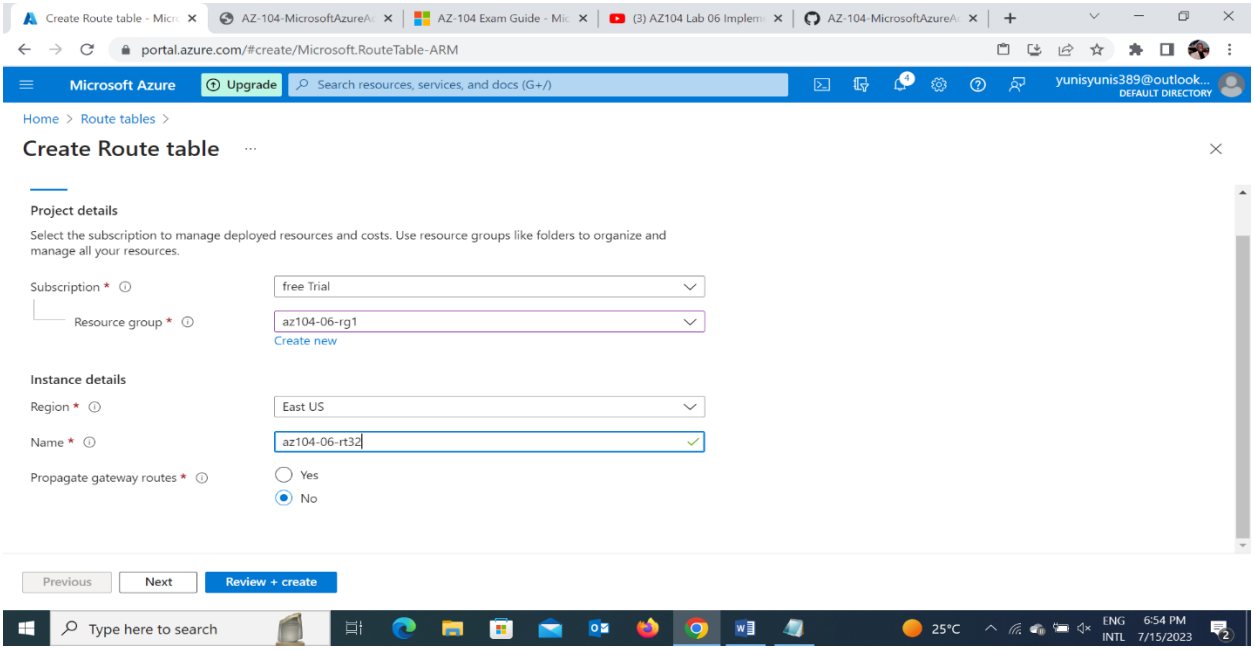
12. Click **Review and Create**. Let validation occur, and click **Create** to submit your deployment.
13. Click Go to resource.
14. On the az104-06-rt23 route table blade, in the Settings section, click Routes, and then click + Add.
15. Add a new route with the following settings:
16. Click Add



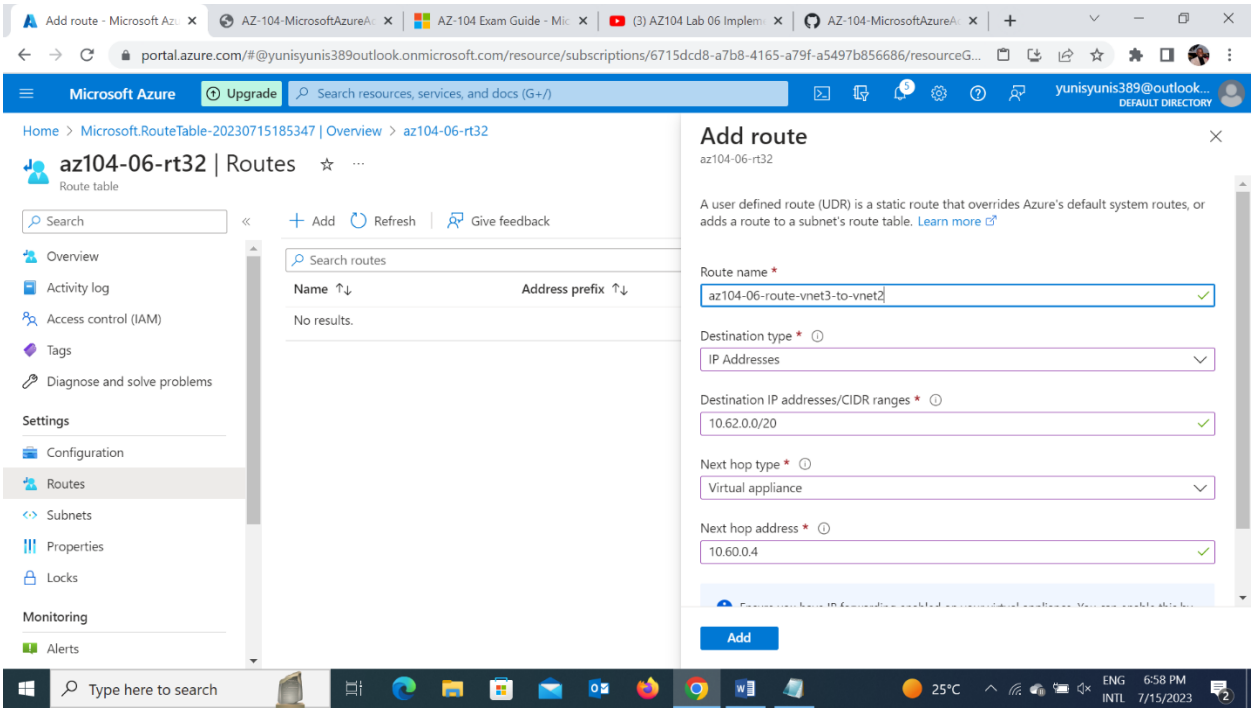
17. Back on the az104-06-rt23 route table blade, in the Settings section, click Subnets, and then click + Associate.
18. Associate the route table az104-06-rt23 with the following subnet:
19. Click Add



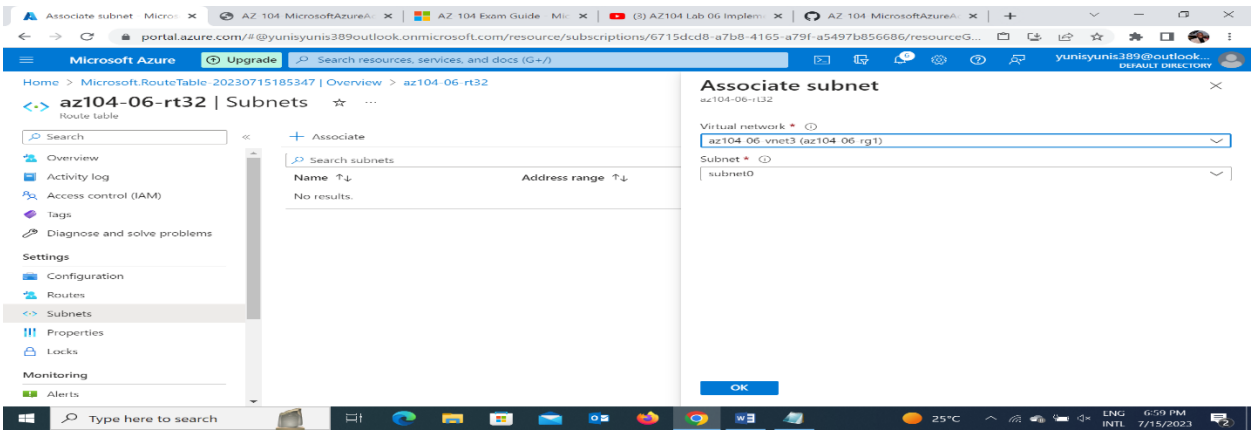
20. Navigate back to Route tables blade and click + Create.
21. Create a route table with the following settings (leave others with their default values):



22. Click Review and Create. Let validation occur, and hit Create to submit your deployment.
23. Click Go to resource.
24. On the az104-06-rt32 route table blade, in the Settings section, click Routes, and then click + Add.
25. Add a new route with the following settings:
26. Click OK

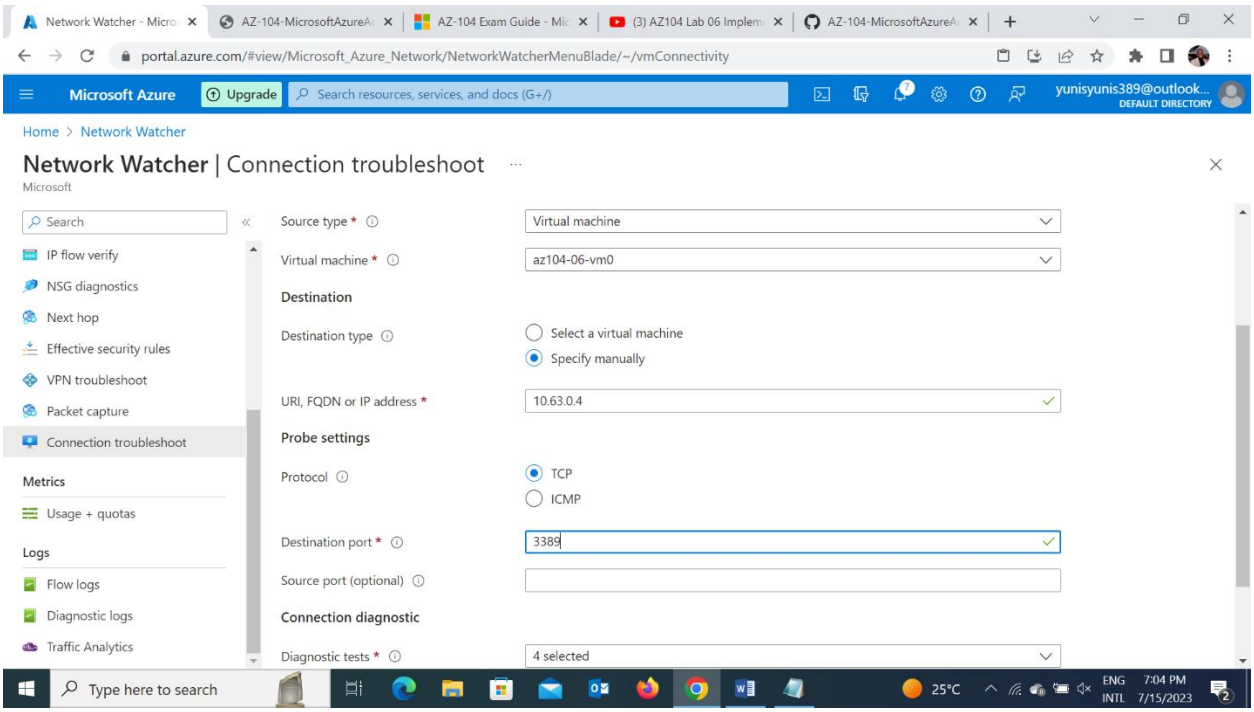


27. Back on the az104-06-rt32 route table blade, in the Settings section, click Subnets, and then click + Associate.
28. Associate the route table az104-06-rt32 with the following subnet:
29. Click OK

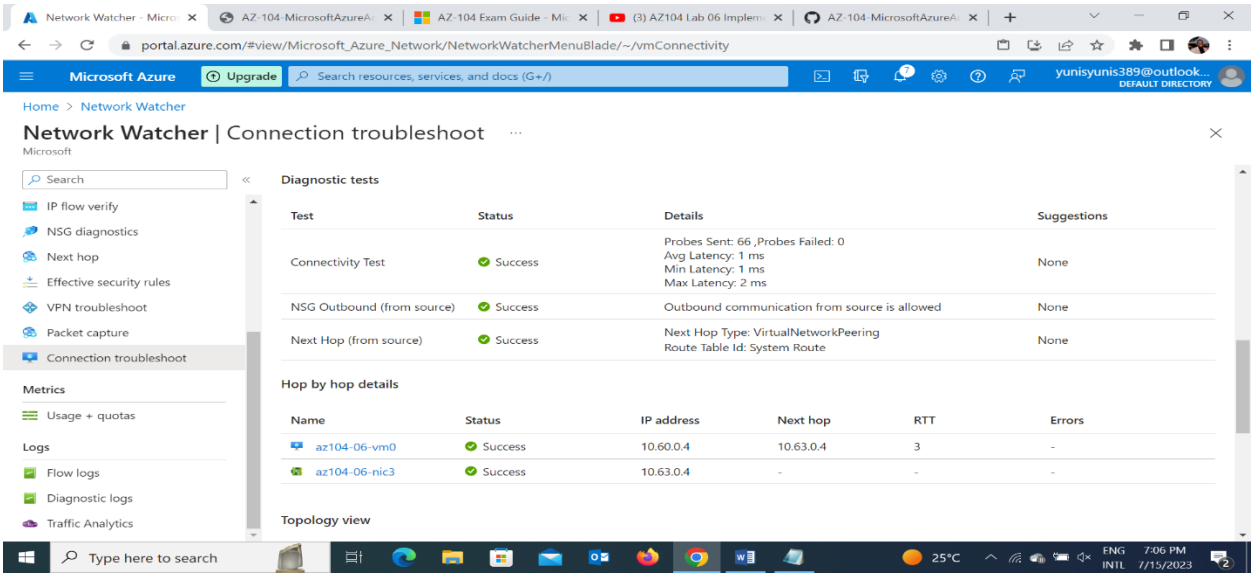


30. In the Azure portal, navigate back to the Network Watcher - Connection troubleshoot blade.

31. On the Network Watcher - Connection troubleshoot blade, use the following settings (leave others with their default values):



32. Click **Run diagnostic tests** and wait until results of the connectivity check are returned. Verify that the status is **Success**. Review the network path and note that the traffic was routed via **10.60.0.4**, assigned to the **az104-06-nic0** network adapter. If status is **Fail**, you should stop and then start az104-06-vm0.

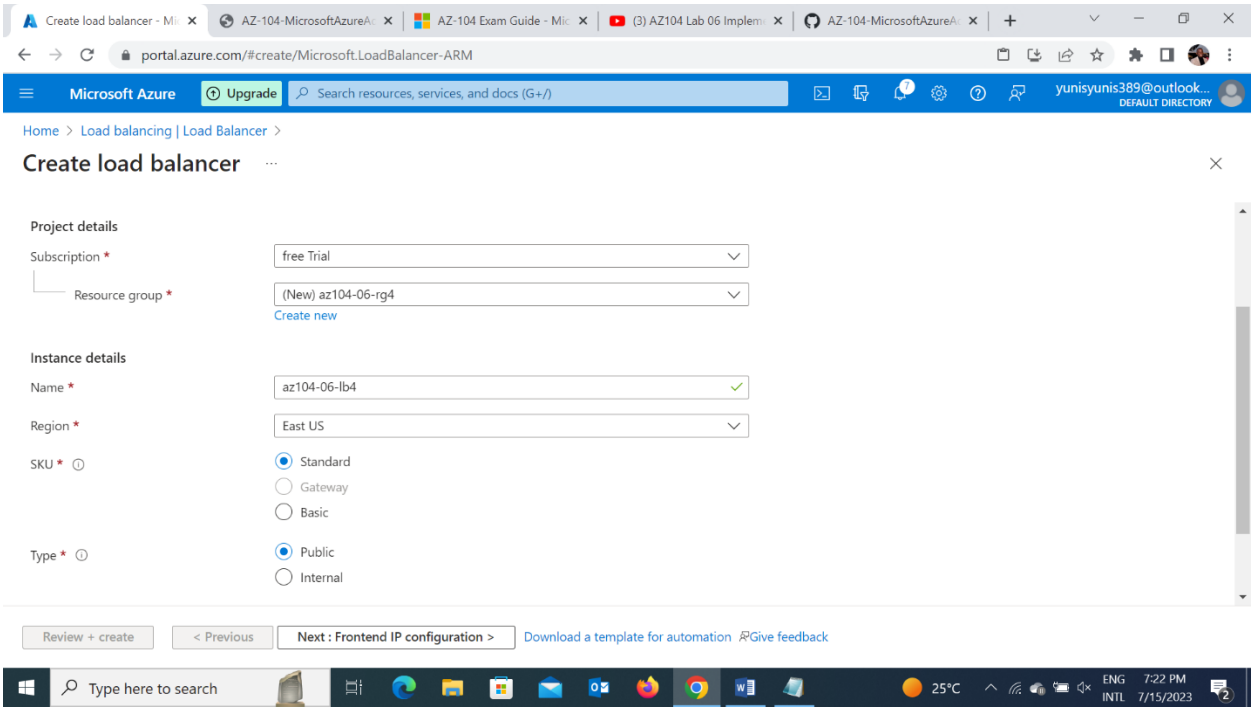


Task 5: Implement Azure Load Balancer

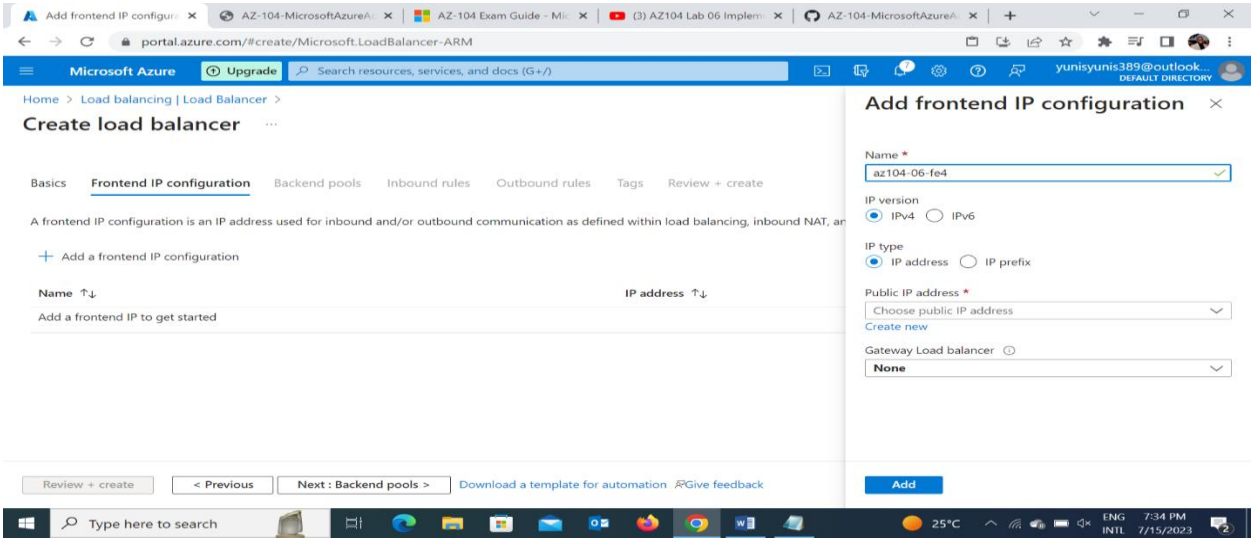
In this task, I will implement an Azure Load Balancer in front of the two Azure virtual machines in the hub virtual network.

1. In the Azure portal, search for and select **Load balancers** and, on the **Load balancers** blade, click **+ Create**.

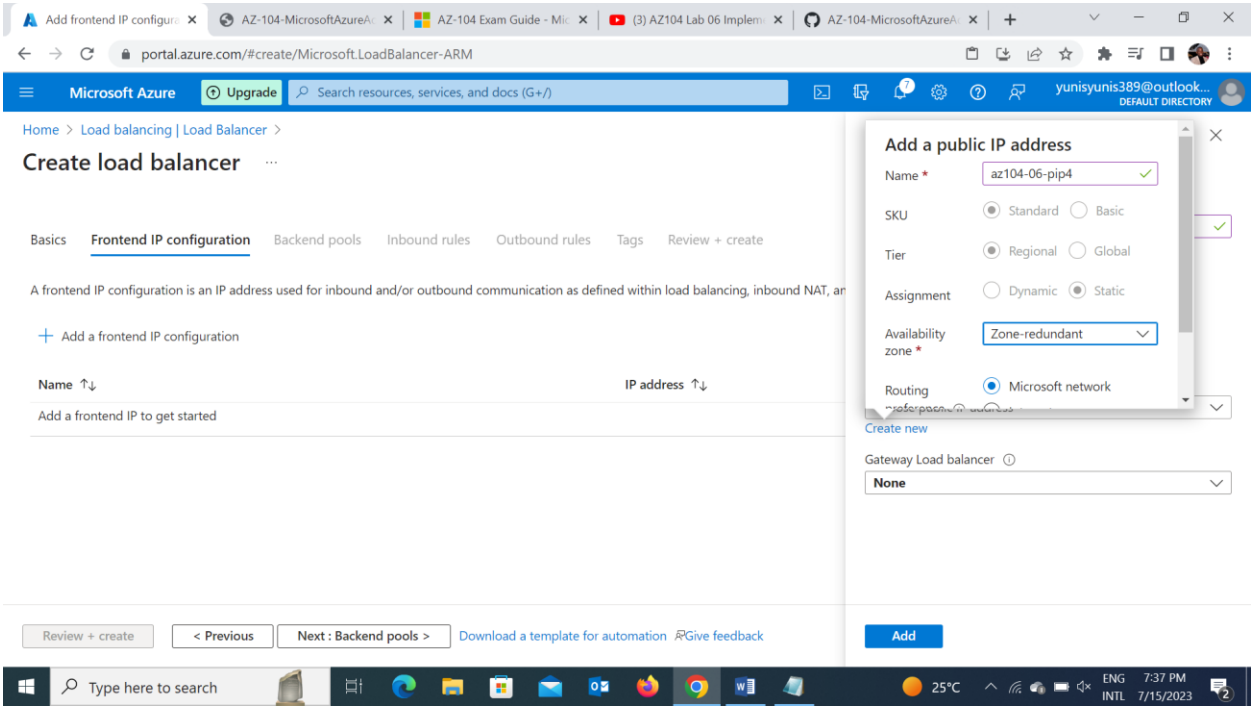
2. Create a load balancer with the following settings (leave others with their default values) then click **Next : Frontend IP configuration**:



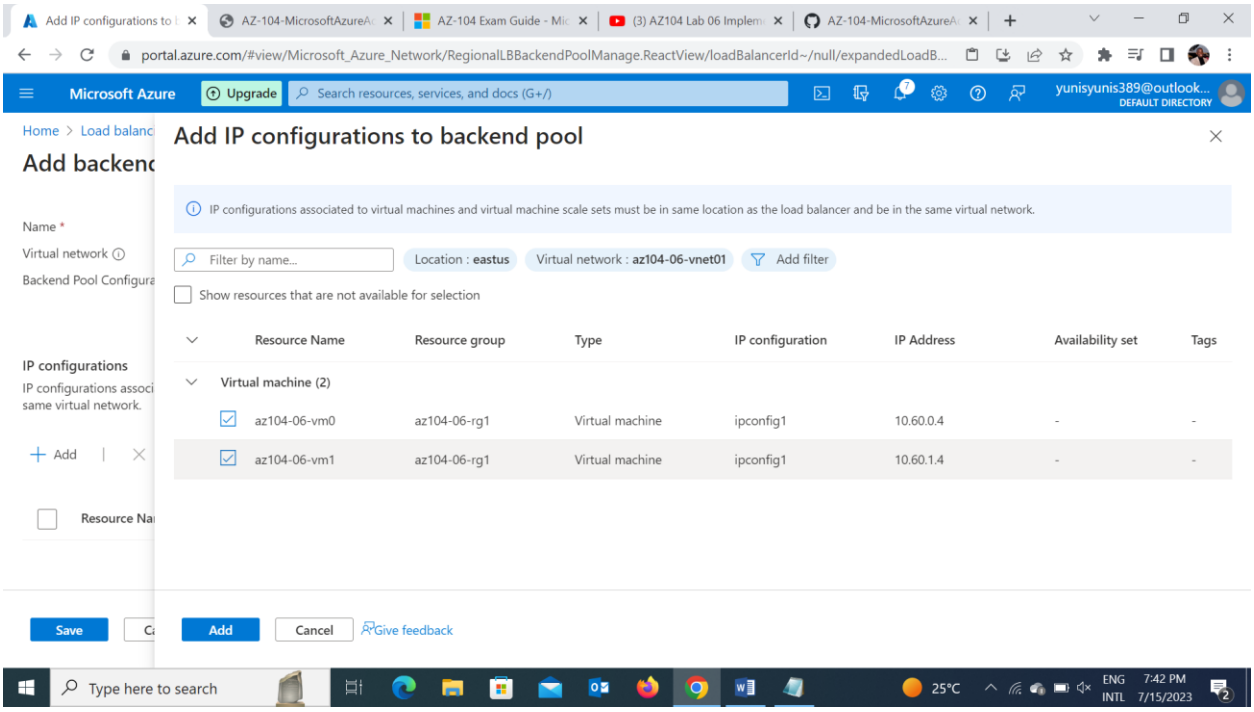
3. On the **Frontend IP configuration** tab, click **Add a frontend IP configuration** and use the following settings:



4. On the **Add a public IP address** popup, use the following settings before clicking **OK** and then **Add**. When completed click **Next: Backend pools**.



5. On the **Backend pools** tab, click **Add a backend pool** with the following settings (leave others with their default values). Click **+ Add** (twice) and then click **Next:Inbound rules**.

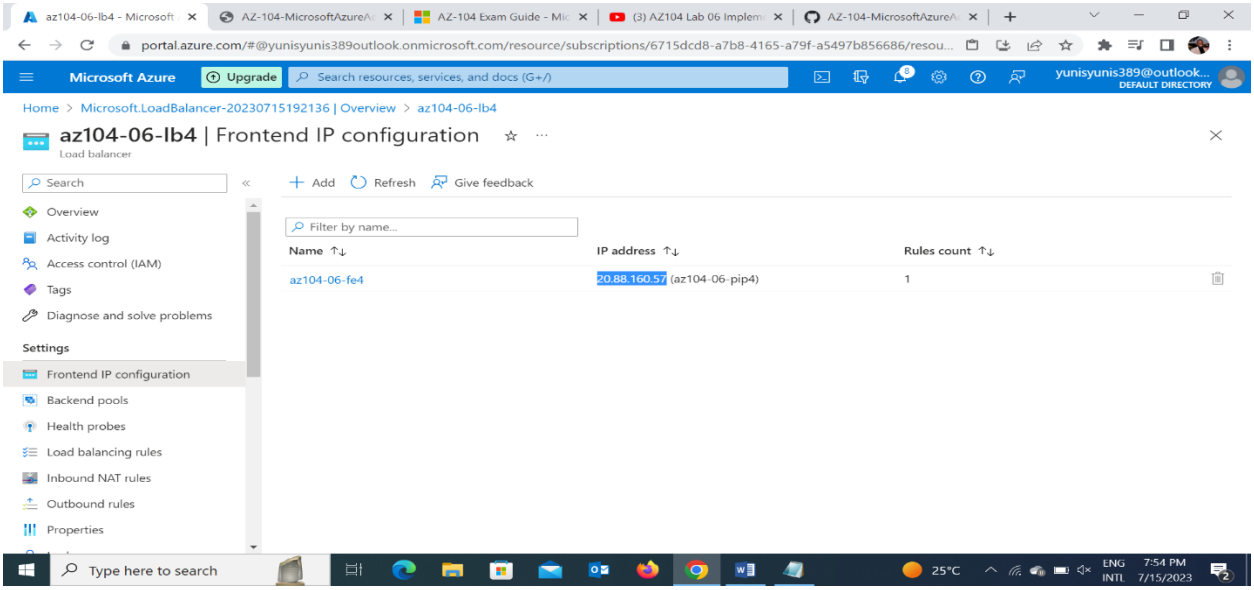


6. On the **Inbound rules** tab, click **Add a load balancing rule**. Add a load balancing rule with the following settings (leave others with their default values). When completed click **Add**.

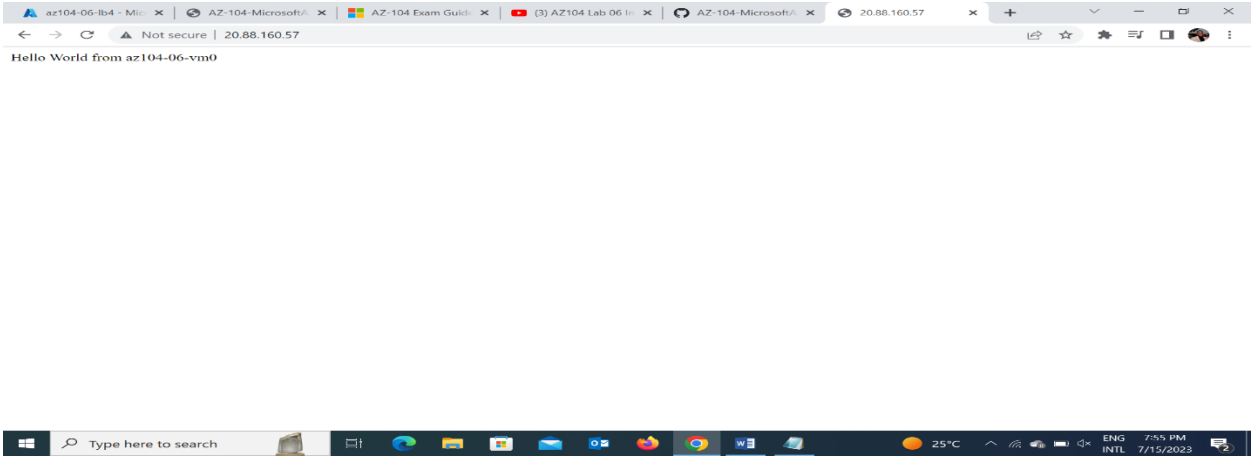
7. As you have time, review the other tabs, then click Review and create. Ensure there are no validation errors, then click Create.

8. Wait for the load balancer to deploy then click Go to resource.

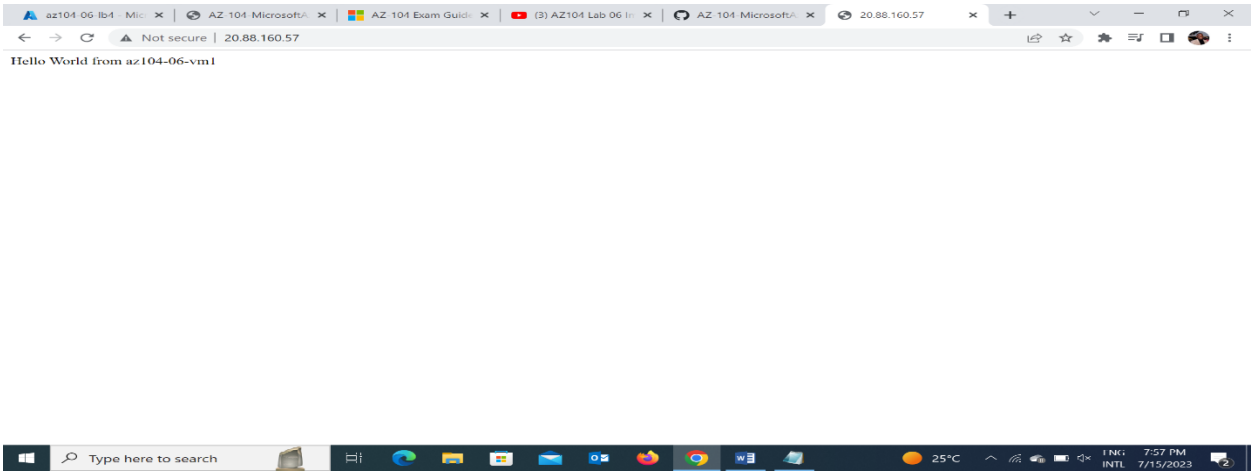
9. Select Frontend IP configuration from the Load Balancer resource page. Copy the IP address.



10. Open another browser tab and navigate to the IP address. Verify that the browser window displays the message Hello World from az104-06-vm0 or Hello World from az104-06-vm1.



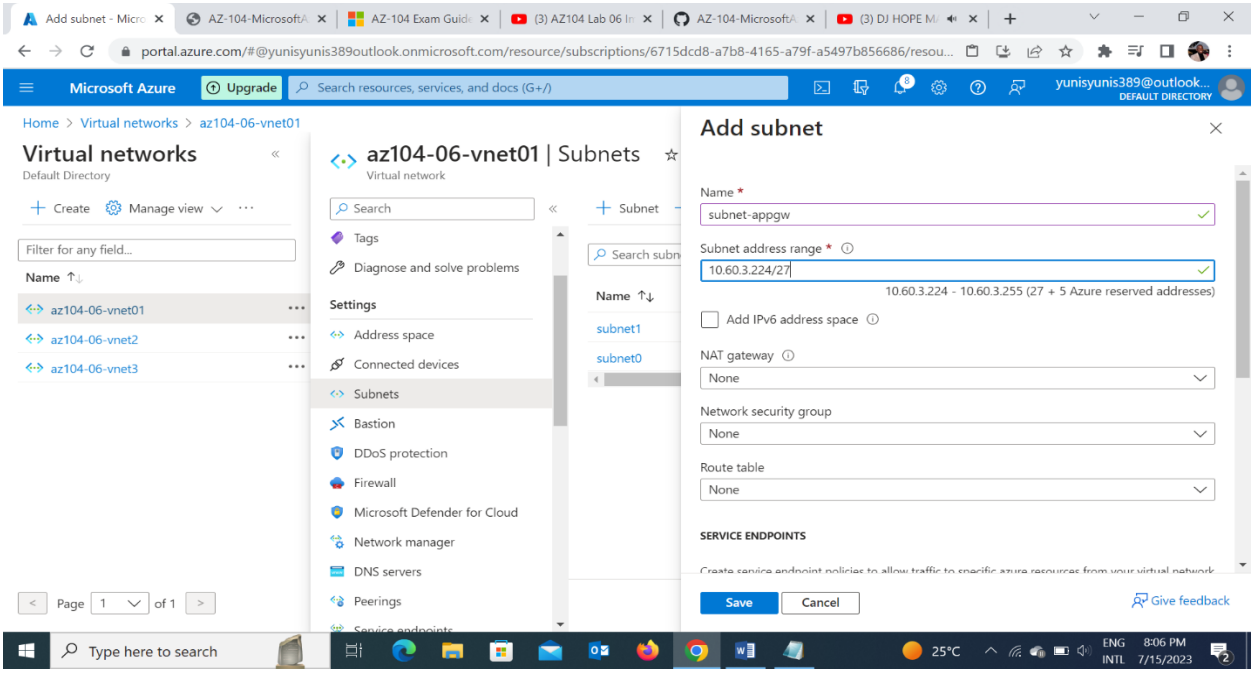
11. Refresh the window to verify the message changes to the other virtual machine. This demonstrates the load balancer rotating through the virtual machines.



Task 6: Implement Azure Application Gateway

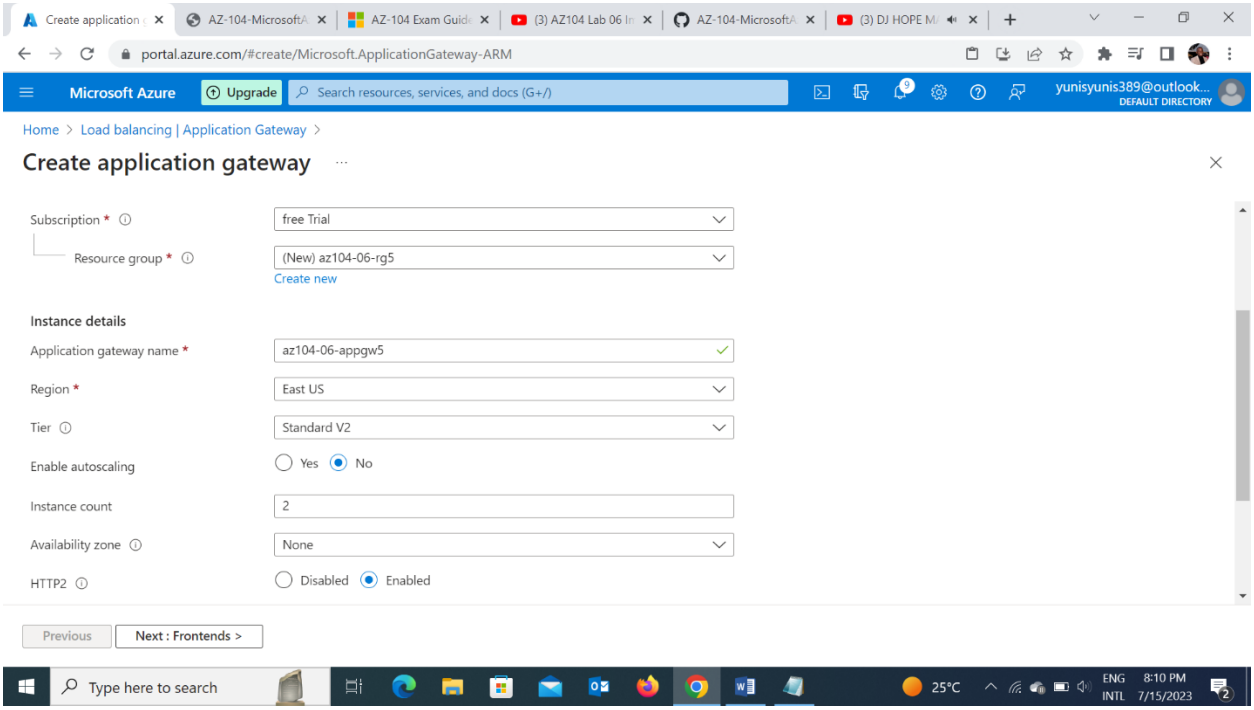
In this task, I will implement an Azure Application Gateway in front of the two Azure virtual machines in the spoke virtual networks.

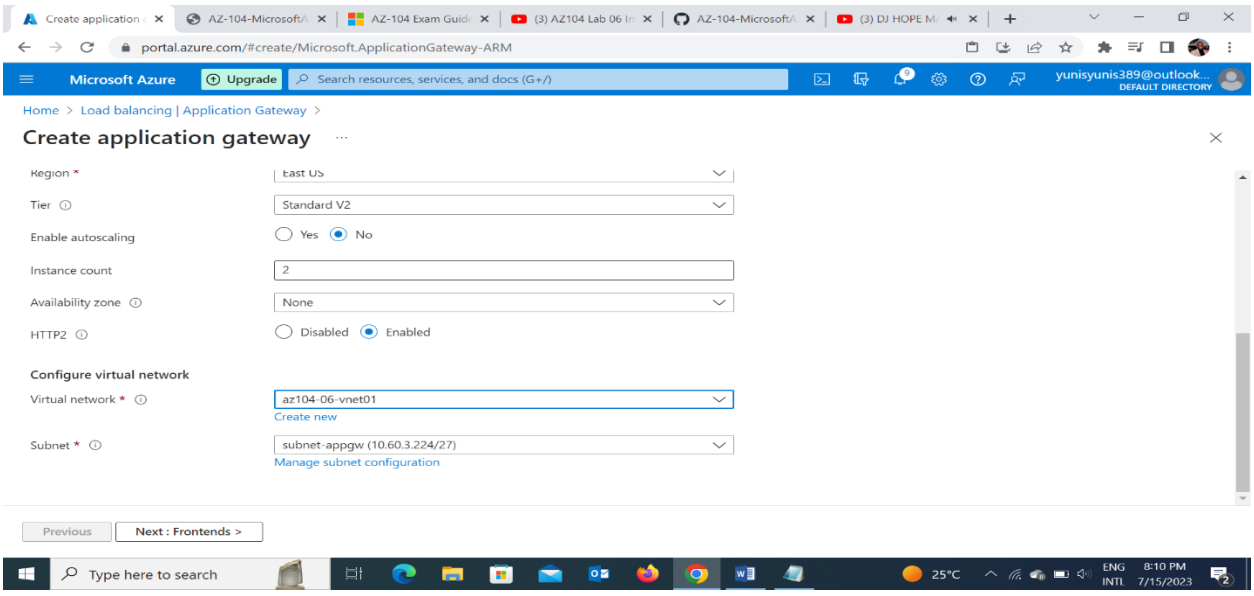
1. n the Azure portal, search and select **Virtual networks**.
2. On the **Virtual networks** blade, in the list of virtual networks, click **az104-06-vnet01**.
3. On the **az104-06-vnet01** virtual network blade, in the **Settings** section, click **Subnets**, and then click **+ Subnet**.
4. Add a subnet with the following settings (leave others with their default values):
5. Click **Save**



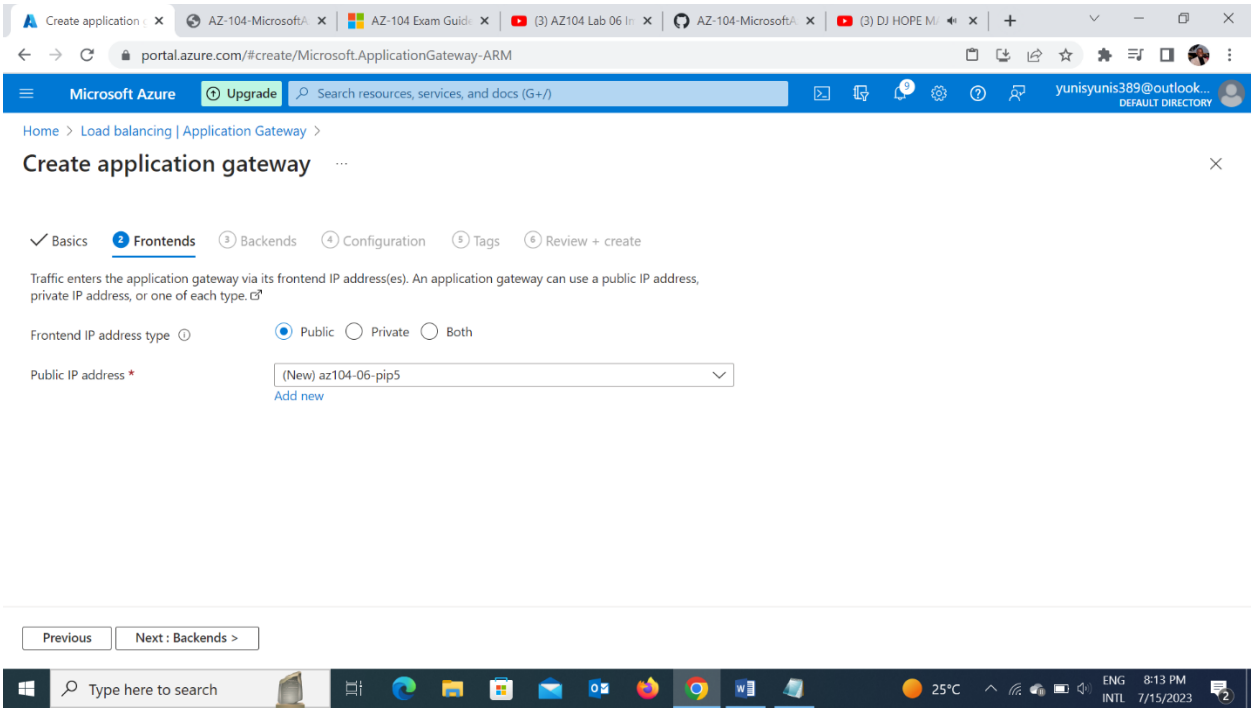
6. In the Azure portal, search and select Application Gateways and, on the Application Gateways blade, click + Create.

7. On the Basics tab, specify the following settings (leave others with their default values):

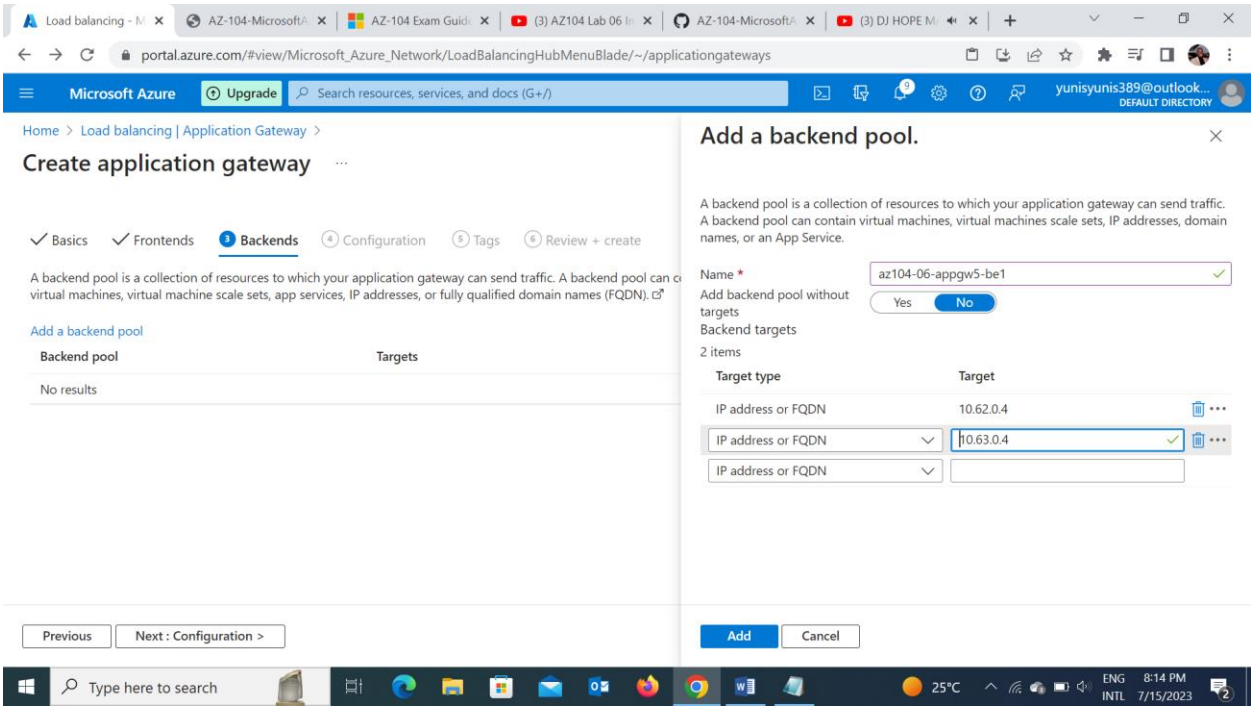




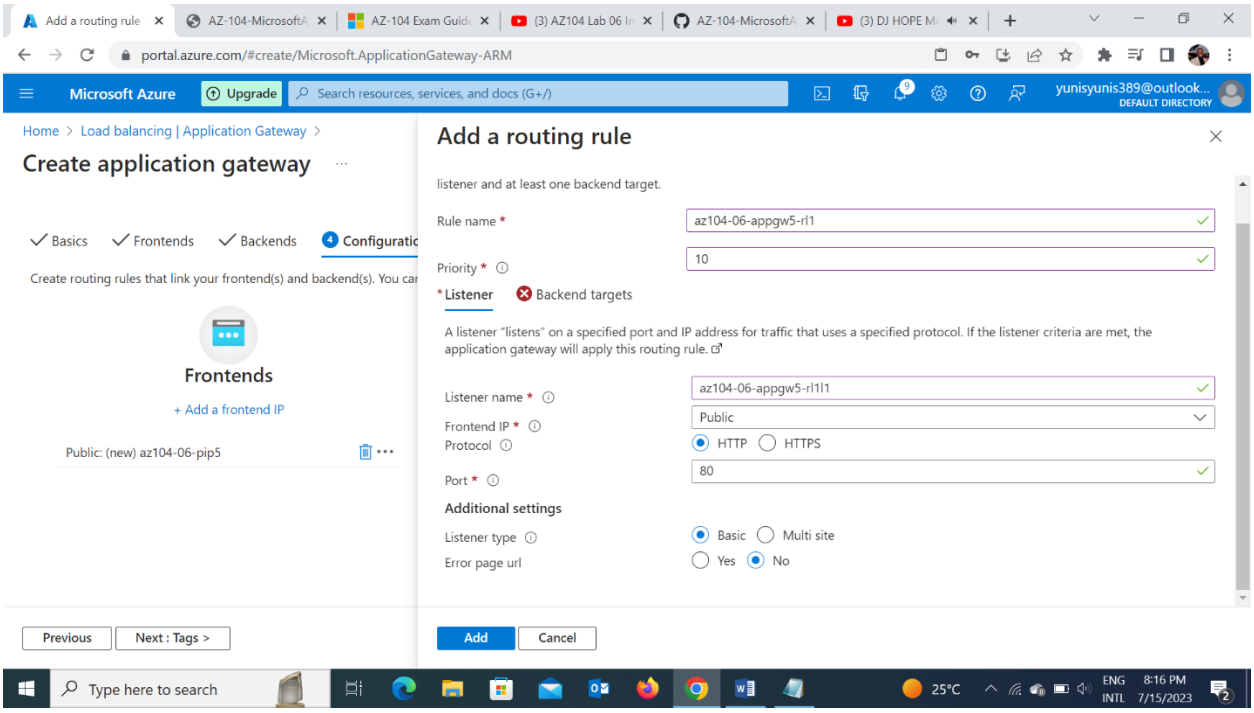
8. Click **Next: Frontends >** and specify the following settings (leave others with their default values). When complete, click **OK**.



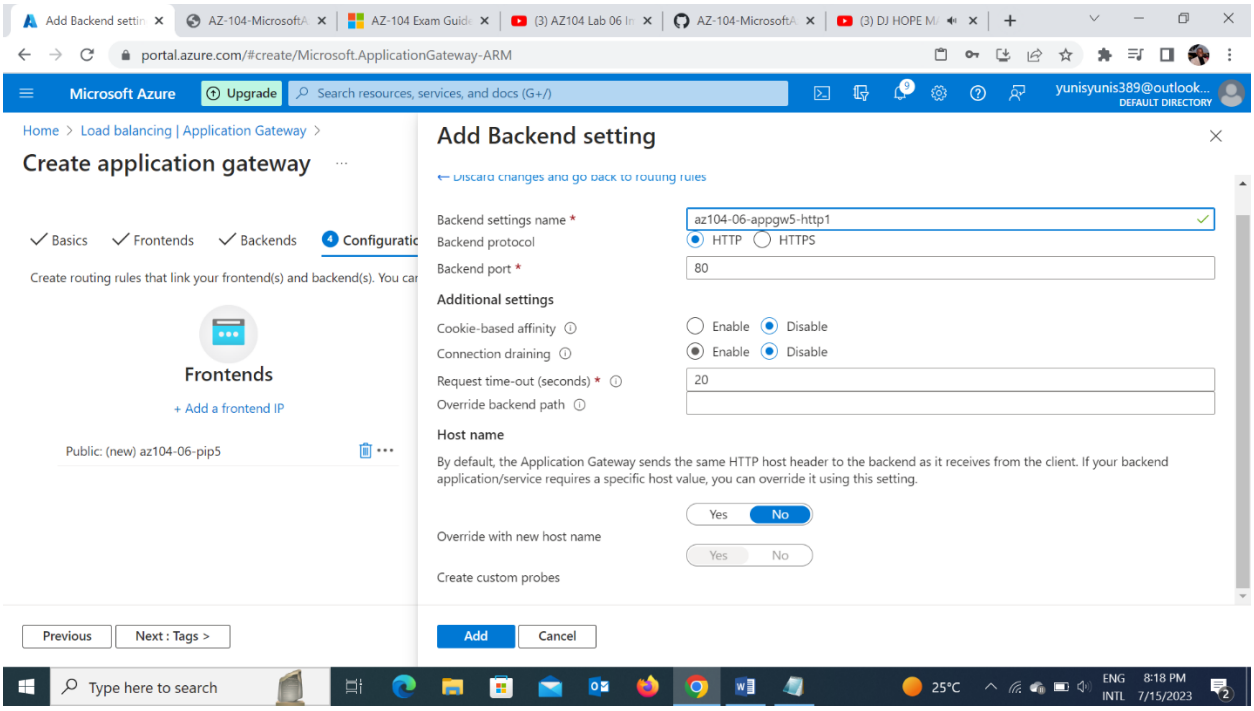
9. Click **Next: Backends >** and then **Add a backend pool**. Specify the following settings (leave others with their default values). When completed click **Add**.



10. Click **Next: Configuration >** and then **+ Add a routing rule**. Specify the following settings:

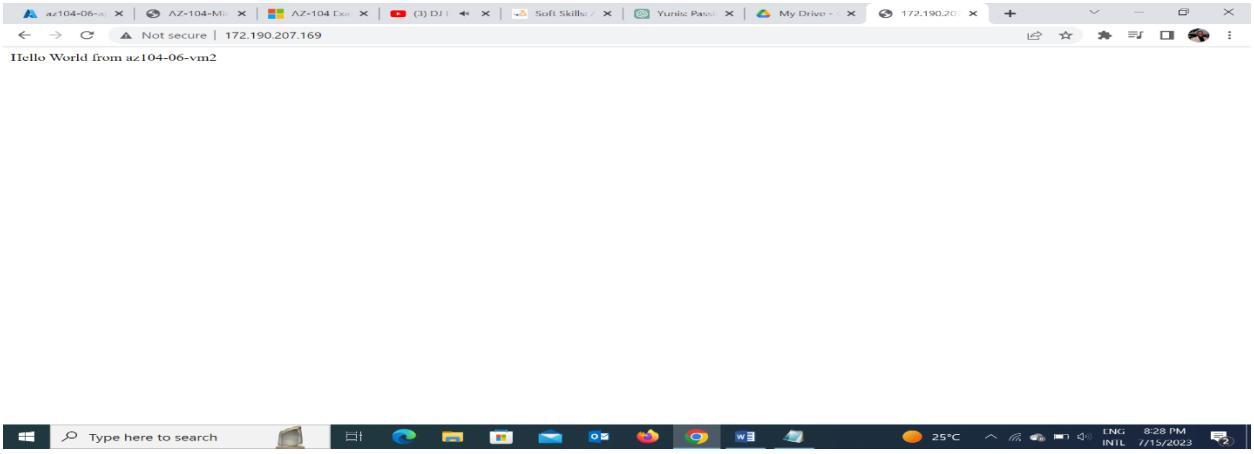


11. Switch to the **Backend targets** tab and specify the following settings (leave others with their default values). When completed click **Add** (twice).

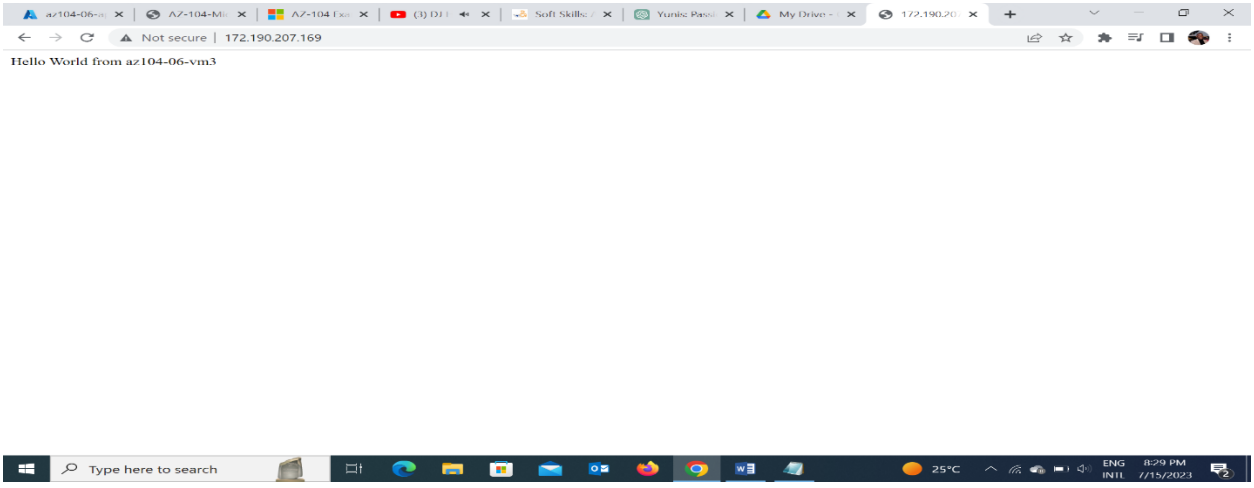


12. Click **Next: Tags >**, followed by **Next: Review + create >** and then click **Create**.

13. In the Azure portal, search and select Application Gateways and, on the Application Gateways blade, click az104-06-appgw5.
14. On the az104-06-appgw5 Application Gateway blade, copy the value of the Frontend public IP address.
15. Start another browser window and navigate to the IP address you identified in the previous step.
16. Verify that the browser window displays the message Hello World from az104-06-vm2 or Hello World from az104-06-vm3.



17. Refresh the window to verify the message changes to the other virtual machine.



Conclusion

In conclusion, Lab 06 - Implement Traffic Management in Azure has been an enlightening and enriching experience, providing me with valuable insights and practical skills in managing traffic within our Azure environment. Through a series of tasks including provisioning the lab environment, configuring network topologies, testing connectivity, optimizing routing, and implementing load balancing and application delivery controllers, I have gained a comprehensive understanding of traffic management techniques. I have learned the importance of distributing traffic effectively, enhancing application availability and performance, and ensuring fault tolerance. The Azure Application Gateway has the ability to intelligently route traffic to backend resources based on various criteria, such as URL path, host header, or HTTP header. This enables administrators to implement sophisticated routing rules and efficiently direct traffic to different applications or services based on specific requirements. The Azure Load Balancer ensures that each resource receives an optimal share of traffic, minimizing the chances of overloading any single resource and optimizing overall performance. This lab has emphasized the significance of proactive monitoring and continuous optimization to deliver seamless user experiences. Overall, this experience has equipped me with the knowledge and tools to tackle the challenges of traffic management in Azure.