Project Report

On

Secure Hosting of Web App via Azure Application Gateway using Hub and Spoke Topology

Submitted by

Piyush Kumar Dey

User ID: CT_CSI_CI_4819

Domain: Cloud infra & Security

Email: piyushmgm2000@gmail.com

Internship Period: June 2025 - August 2025

Date of Submission: 23rd July 2025

INTRODUCTION

In today's time, secure hosting of websites is very important. Companies want to make sure their web applications run safely and only allow trusted access. To handle this, cloud platforms like Microsoft Azure provide strong tools that help manage and protect the entire setup. This project focuses on hosting a secure web app using Azure's network features.

This setup follows a **hub-and-spoke** model. The hub works like the central controller, and the **spokes** carry the actual content and storage. In the hub, services like **DNS, firewall, and routing** are placed. In the spokes, the **web app** and **storage** are kept safe but still connected. This helps in better security and traffic control. The web traffic first goes to a simulated firewall, which checks everything before sending it to the web app.

The goal of the project is to design and build a full **secure network structure** in Azure. It includes steps like creating virtual networks (**VNets**), connecting them (peering), setting up firewalls (simulated with NSG), routing traffic, and finally hosting a website behind a secure gateway. Each part of the project is tested and shown with screenshots to make sure the setup works properly.

OBJECTIVE

The main aim of this project is to create a secure network setup in Microsoft Azure for hosting a web application. This setup should allow only trusted traffic to reach the website, while blocking anything unsafe.

The project also focuses on using Azure's basic tools like virtual networks, security groups, and routing to control how data moves inside the system. It uses a simulated firewall to check all incoming traffic before it reaches the web server.

Lastly, the goal is to complete the full system within the time and resource limits of Microsoft Learn Sandbox, while making sure the solution is safe, working properly and easy to manage.

PREREQUISITIES

Azure Sandbox Environment

- Used for deploying and testing the application within the Azure cloud platform.

Resource Groups and Azure Regions

- All project resources were deployed in the West US region under structured resource groups for easy management.

Basic Networking Concepts

- Understanding of VNets, NSGs, subnets, and routes was needed to build secure and connected cloud infrastructure.

Remote Desktop Protocol (RDP) Access

- Allowed testing of VM connectivity through port 3389, confirming routing and firewall rules were working properly.

METHODOLOGY

This project was executed step-by-step using Microsoft Azure's Learn Sandbox, following the Hub-and-Spoke topology model. Each stage was carefully planned and tested to ensure secure hosting of a web application through an Application Gateway. Below is the summarized methodology:

1. **Topology Planning and Resource Setup**

Planned the network structure using Hub-and-Spoke model. Created 3 VNets: Hub-VNet, Spoke-Web-VNet, and Spoke-Storage-VNet.

2. **Creation Subnets of the Virtual Networks (VNets)**

Created required subnets for each VNet including HubSubnet, WebSubnet, and StorageSubnet. Special care was taken to avoid NSGs on system-reserved subnets.

3. **Peering Connections**

Established peering between VNets to allow communication between spokes and the centralized hub network.

4. **Network Security Groups (NSGs)**

Created two NSGs:

- **WebVM-NSG** (with RDP port 3389 open)
- **SimulatedFirewall-NSG** (with ports 3389 and 8080 open)

5. **Simulated Firewall Setup**

Due to limitations in the Learn Sandbox, Azure Firewall was replaced by a VM and NSG-based simulation.

6. **Route Table Configuration**

Added a custom route to direct all traffic (0.0.0.0/0) through the SimulatedFirewallVM using a user-defined route table.

7. **Virtual Machines Deployment**

Deployed two VMs:

- **SimulatedFirewallVM** in Hub-VNet
- **WebVM** in Spoke-Web-VNet

8. **Testing and Validation**

Tested end-to-end communication using ping, RDP, and browser access to ensure routing and security configurations were correct.

IMPLEMENTATION

This project was implemented using a step-by-step approach inside the Azure Learn Sandbox environment. All configurations followed the Hub-and-Spoke topology to securely host a web application.

After logging in **Microsoft Azure** and **Activating the Learn Sandbox** with required resource group selected, we will proceed with the main workflow for this project.

Step 1: Create Resource Groups and Virtual Networks (VNets)

Three VNets were created: one Hub-VNet and two Spoke-VNets (Web and Storage). These represent the core of the network design .
At the Azure portal home page, we will see different services and resource groups, so we will navigate to the Virtual networks icon as shown below Fig1:

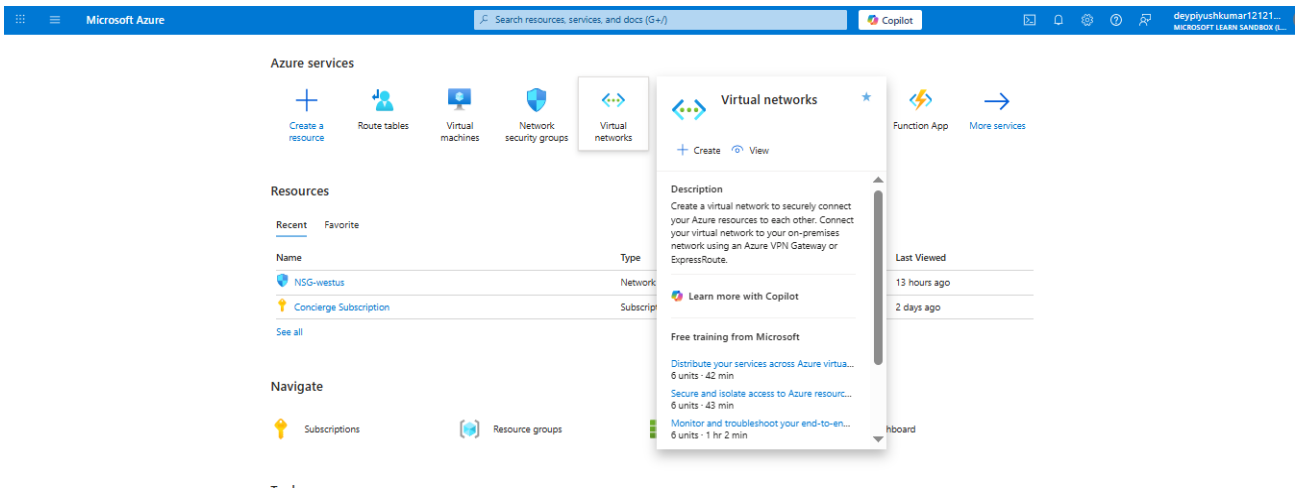


Figure:1

Next**, Step 2: Create Subnets inside each VNet**

After creation of Vnets, each VNet was divided into subnets , shown in the below snapshot as:-:

- **HubSubnet** (for firewall simulation)
- **WebSubnet** (for the web app)
- **StorageSubnet** (for future storage or app needs)

After clicking on Virtual Networks, a new window opens see Figure.2. Here, We would click on the Create(+) button to create three VNets.

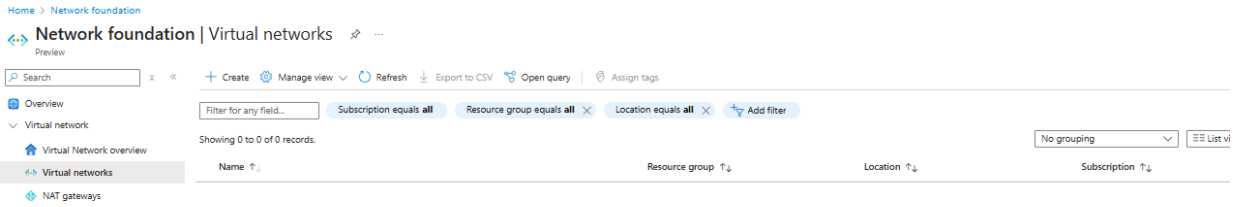
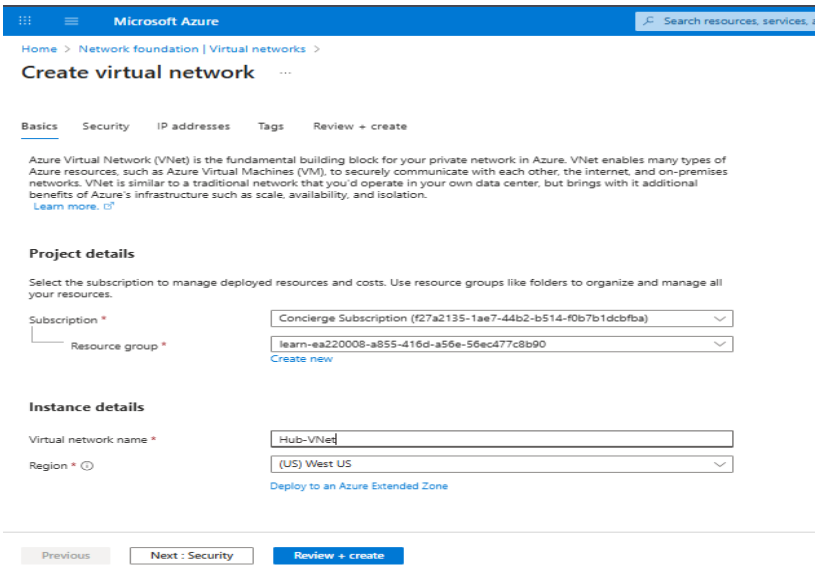


Figure: 2

Then, we would proceed with creation of Hub-VNet with all its details along with the subnet HubSubnet for firewall simulation, with its ip addesses.



Creation of Subnets with all its necessary details, shown in Figure:3 as

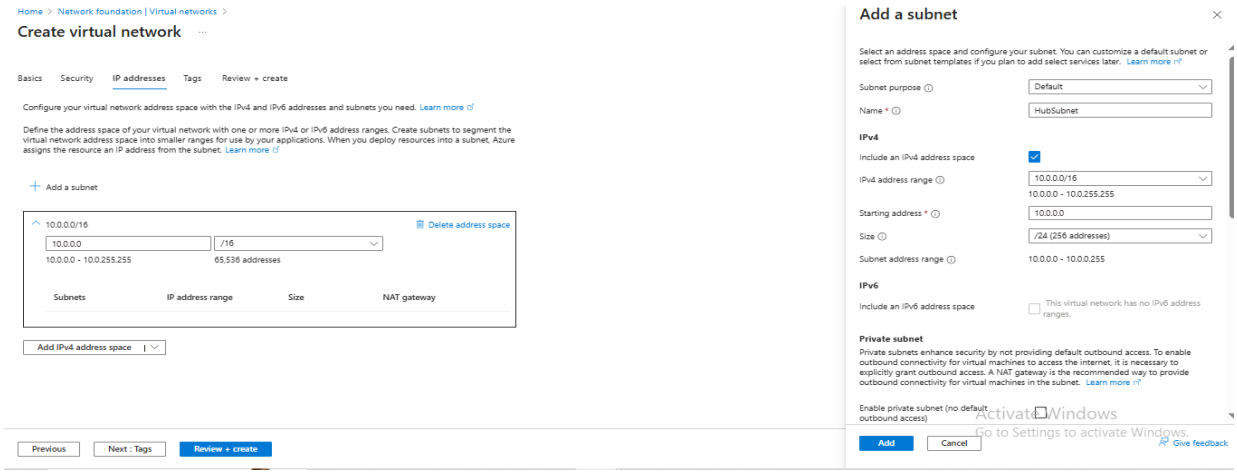


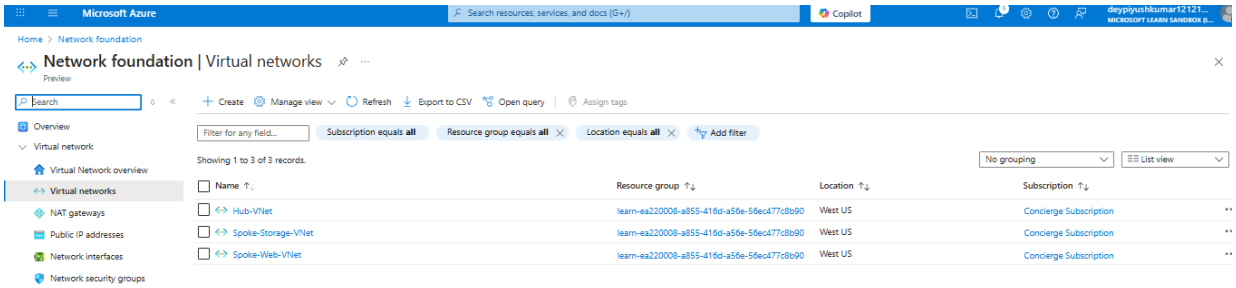
Figure: 3

After creation of Hub-VNet and its subnets, similarly we would create the other two Vnets and its subnets named i) Spoke-Web-Vnet (WebSubnet) and,

ii) Spoke-Storage-VNet(StorageSubnet)

Step 3: Configure Peering Between VNets

Peering is established between Hub-VNet and Spoke-Web-VNet, and between Hub-VNet and Spoke-Storage-VNet to allow secure, private communication across networks.



After successful creation of Vnets and its subnets, now peering between them will be done.

Here, inside Hub-VNet, in the left pane, we would select 'Peering' button, then Add(+) shown in Figure 4 as,

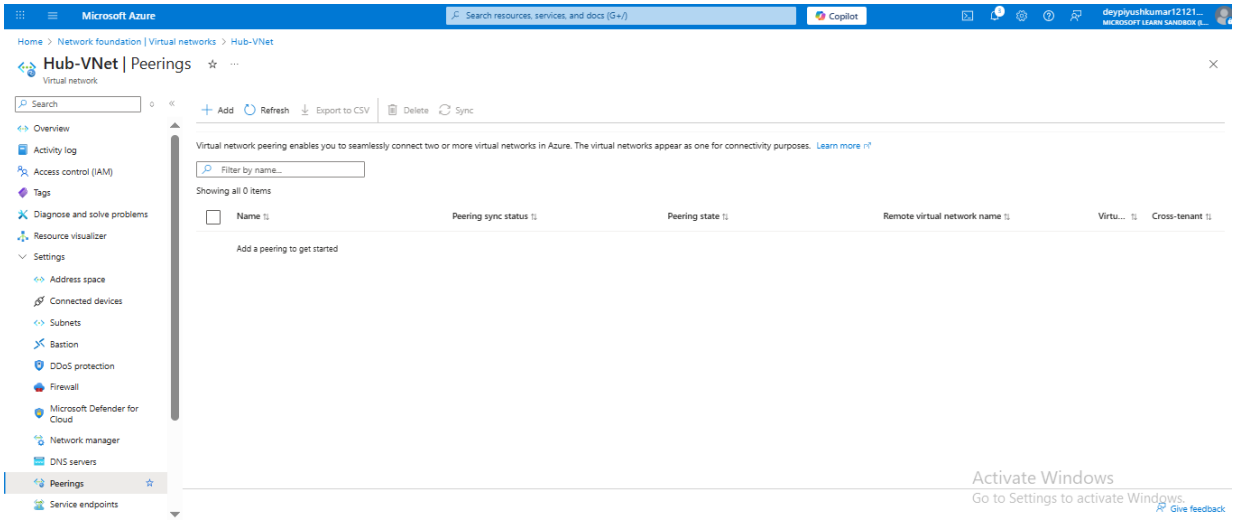
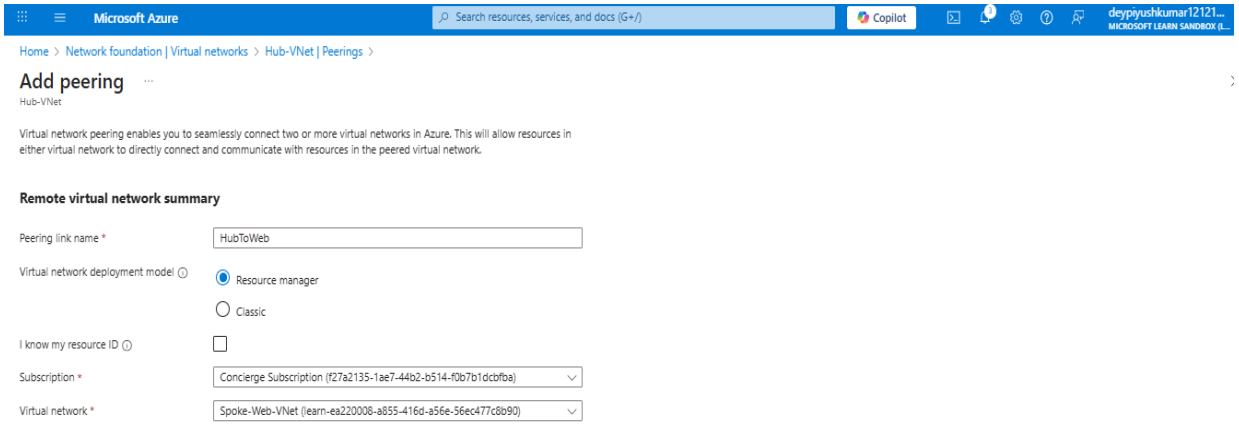


Figure:4

We will create two peerings, Hub-VNet ⟷ Spoke-Web-Vnet and another peering of Hub-VNet ⟷ Spoke-Storage-VNet.

This shows the creation of Hub-VNet peering, Similarly we had created with Spoke-Storage-VNet also.

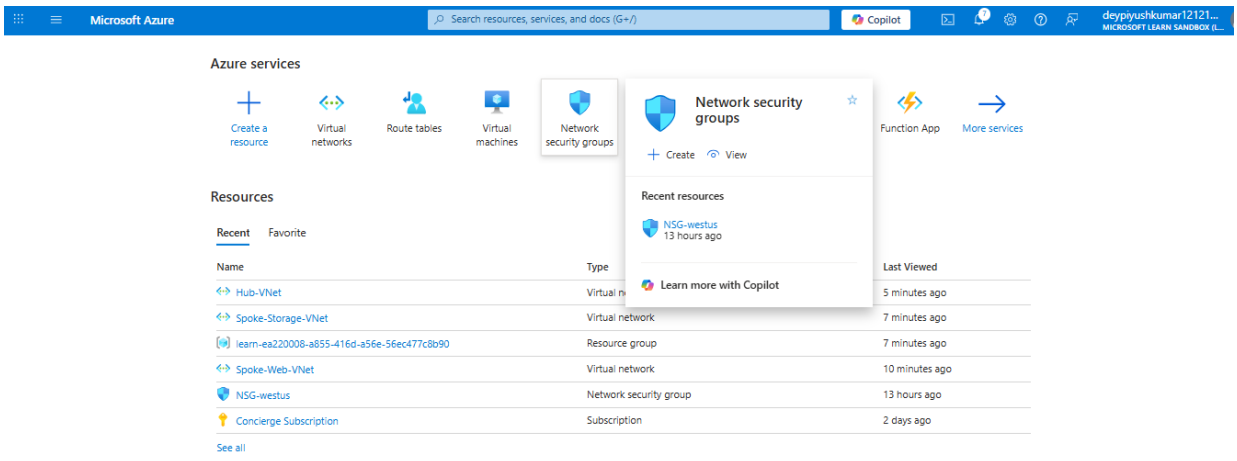


Step 4: Create Network Security Groups (NSGs)

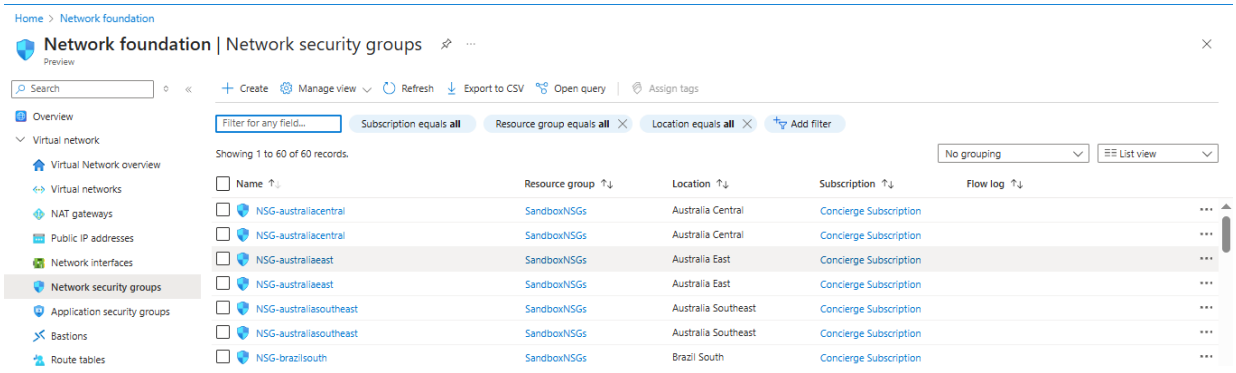
Two NSGs should be created as:

- **WebVM-NSG** →which allows RDP on port 3389.
- **SimulatedFirewall-NSG** → allows ports 8080 (HTTP) & 3389 (RDP).

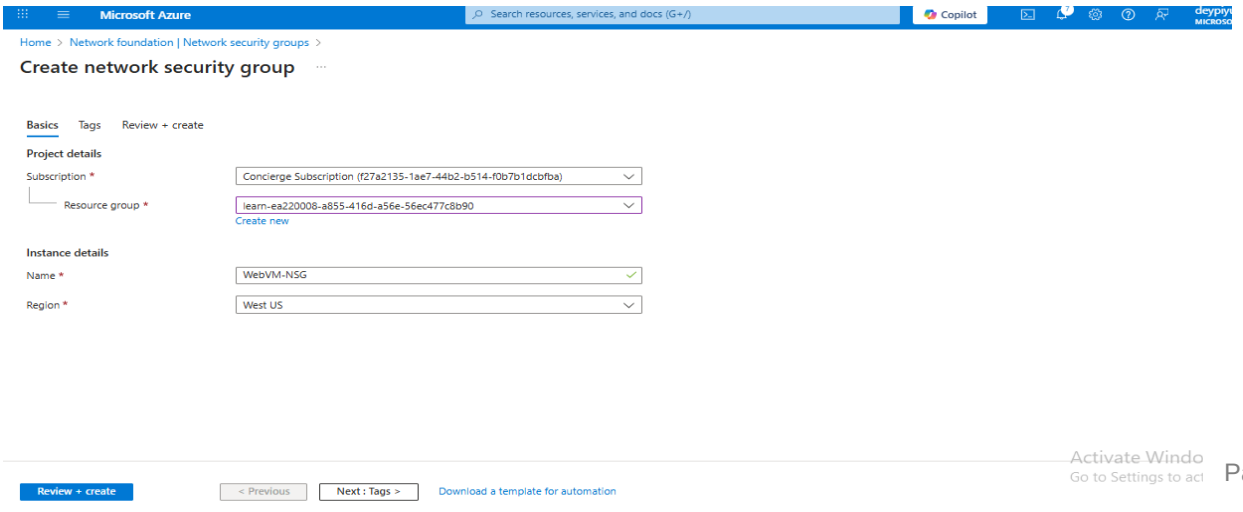
For this, at home page, we would select Network Security Groups(NSGs) tab as shown below.



Then new window opens, where we would click on Create(+) button to create new NSG.



Upon clicking on create button, window appears where we would create NSG named as **WebVM-NSG**, then click on review + create button.



Also we have to attach Inbound Security rules to the **WebVM-NSG** for specific port 3389 , then click on add(+) button shown in Figure: 5 as

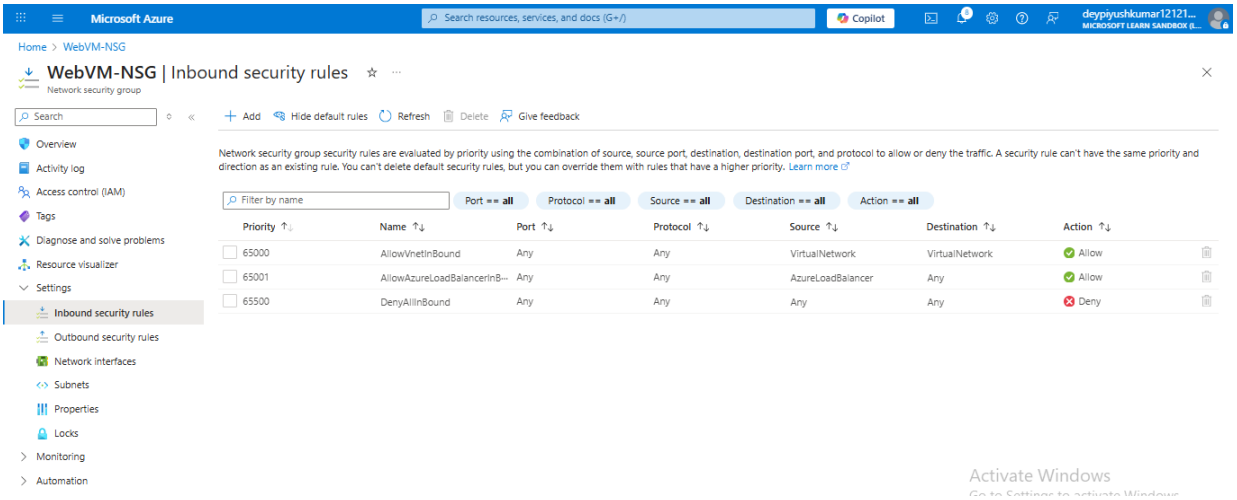
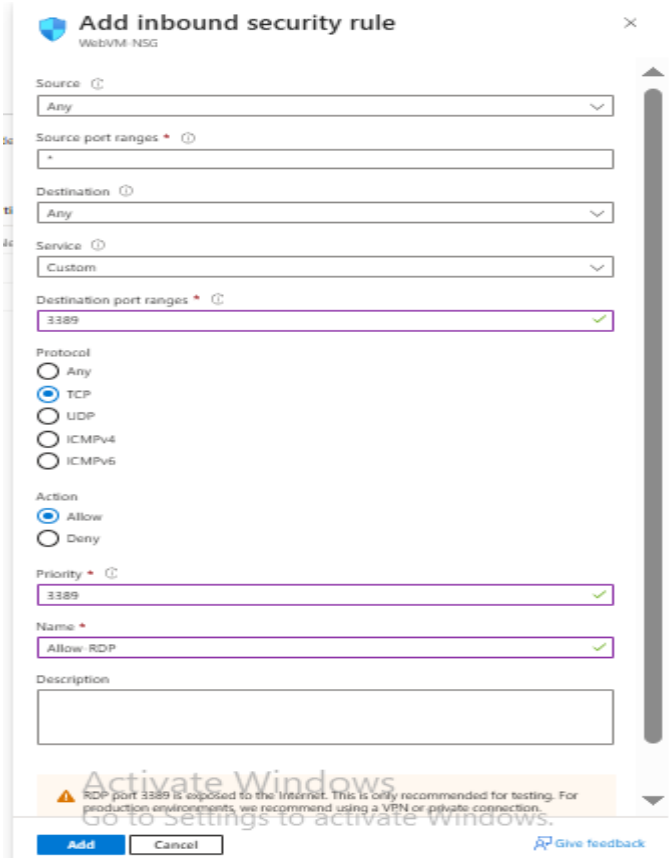


Figure: 5

We would fill the required values in the rule with port number and name.



Then Click on 'Add' button. The **WebVm-NSG** would be successfully created with its inbound security rules. Similarly we would create another NSG name '**SimulatedFirewall-NSG'** with ports 3389 and 8080 for firewall simulation.

Next, we would proceed with Route Table creation.

Step 5: Create Route Table for Simulated Firewall

To simulate firewall routing, and addition of a custom route table is required to route all outbound traffic from WebVM through the SimulatedFirewallVM, with association of the subnets.

For this, we would select the Route Table tab at azure home page as shown in Figure: 6

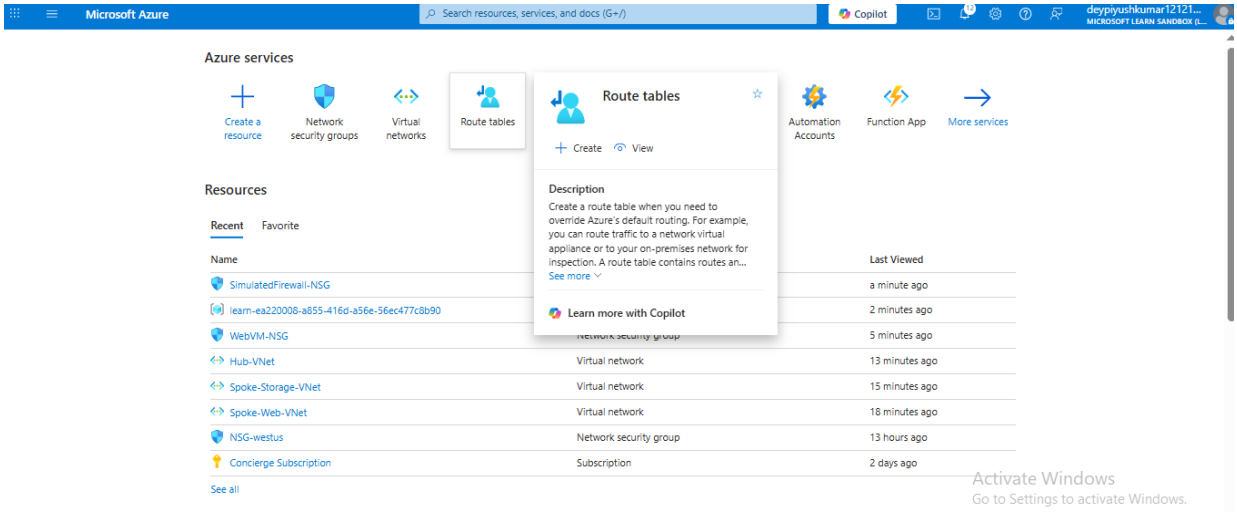
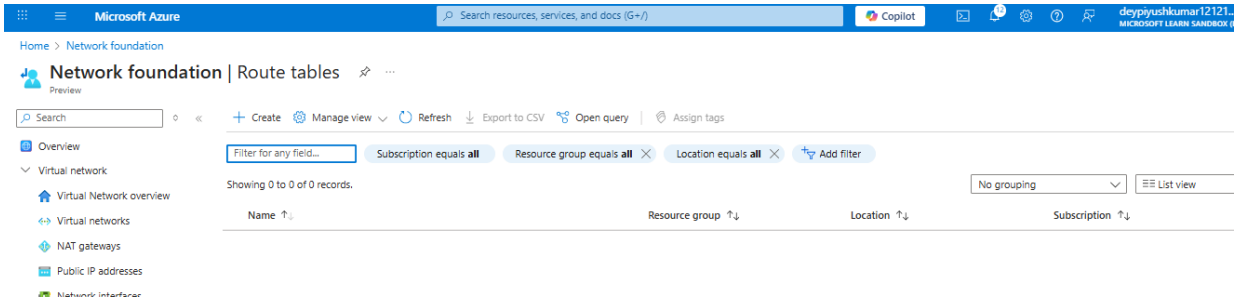
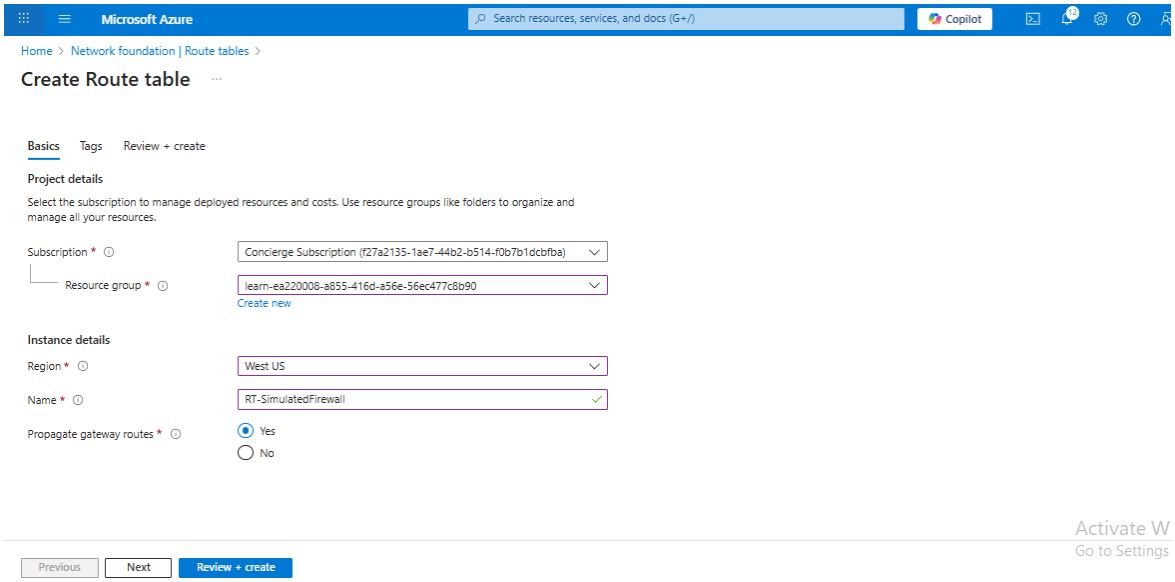


Figure: 6

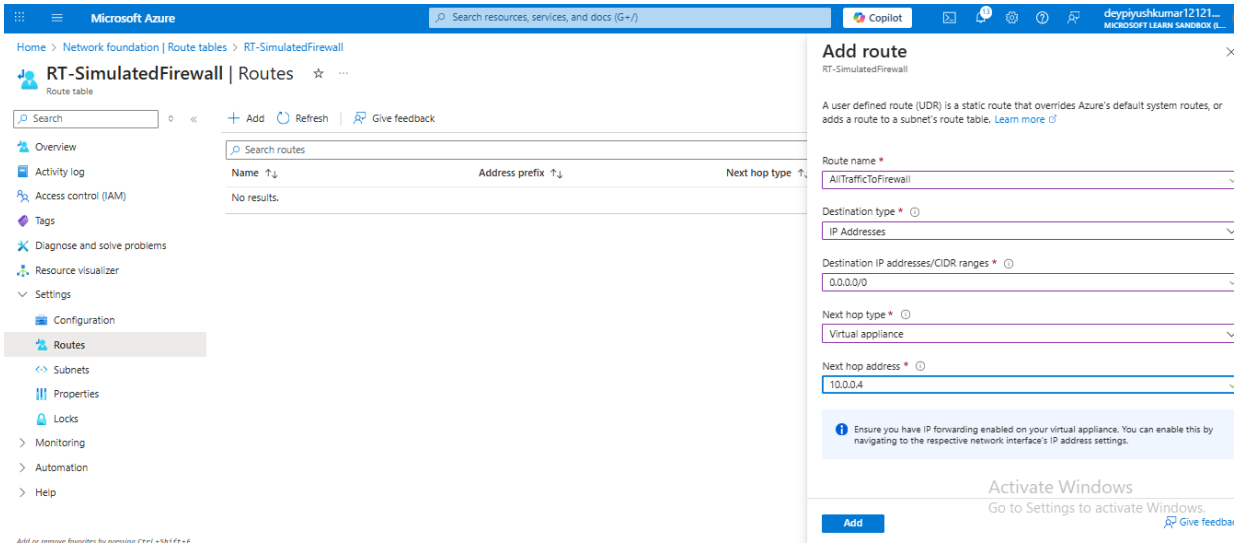
On Clicking on the Route table button, a window appears. Now, we would create(+) a new route for the firewall simulation.



Now, we would create a route table named **RT-SimulatedFirewall,** and then review + create.



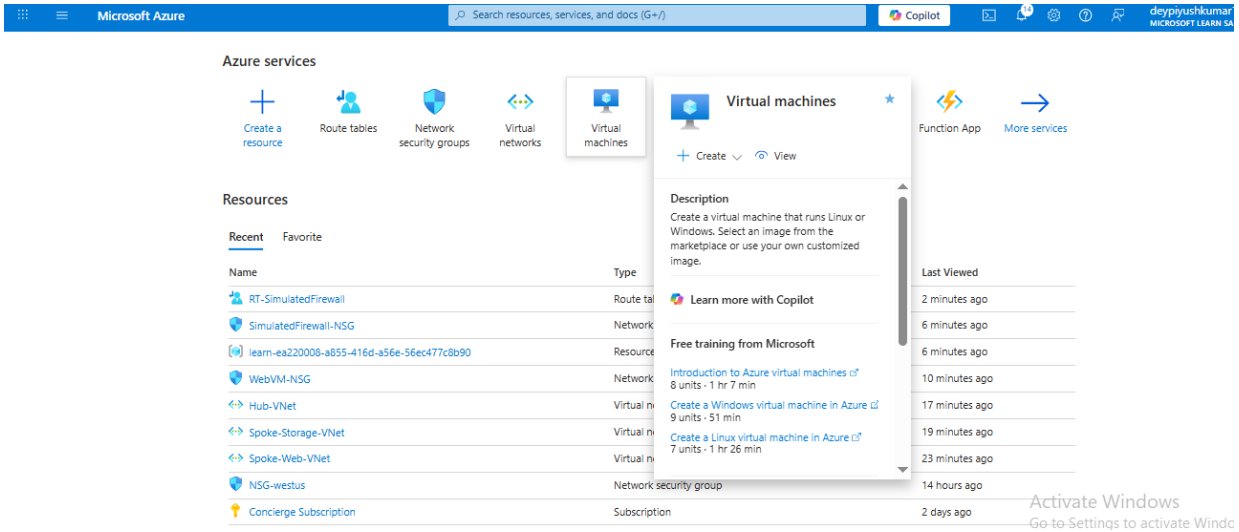
After successful route table creation, on the left panel “Routes” menu, we have to add Routes to our table with name **AllTrafficToFirewall,** clicking on Add(+) button alongwith its destination address 0.0.0.0/0 and ports as 10.0.0.4.



Step 6: Create Virtual Machines

Two virtual machines are needed – SimulatedFirewallVM with Hub-Vnet to simulate a firewall, and WebVM with Spoke-VNet to host the web application.

At the azure home page, we would select Virtual machines.



Now, we have to create two virtual machines for this project. After clicking on '+' button, a new window appears as see Figure: 7 and we would create vm named '**WebVM'** for the web application**,** with operating system, storage capacity details and username with password**.**

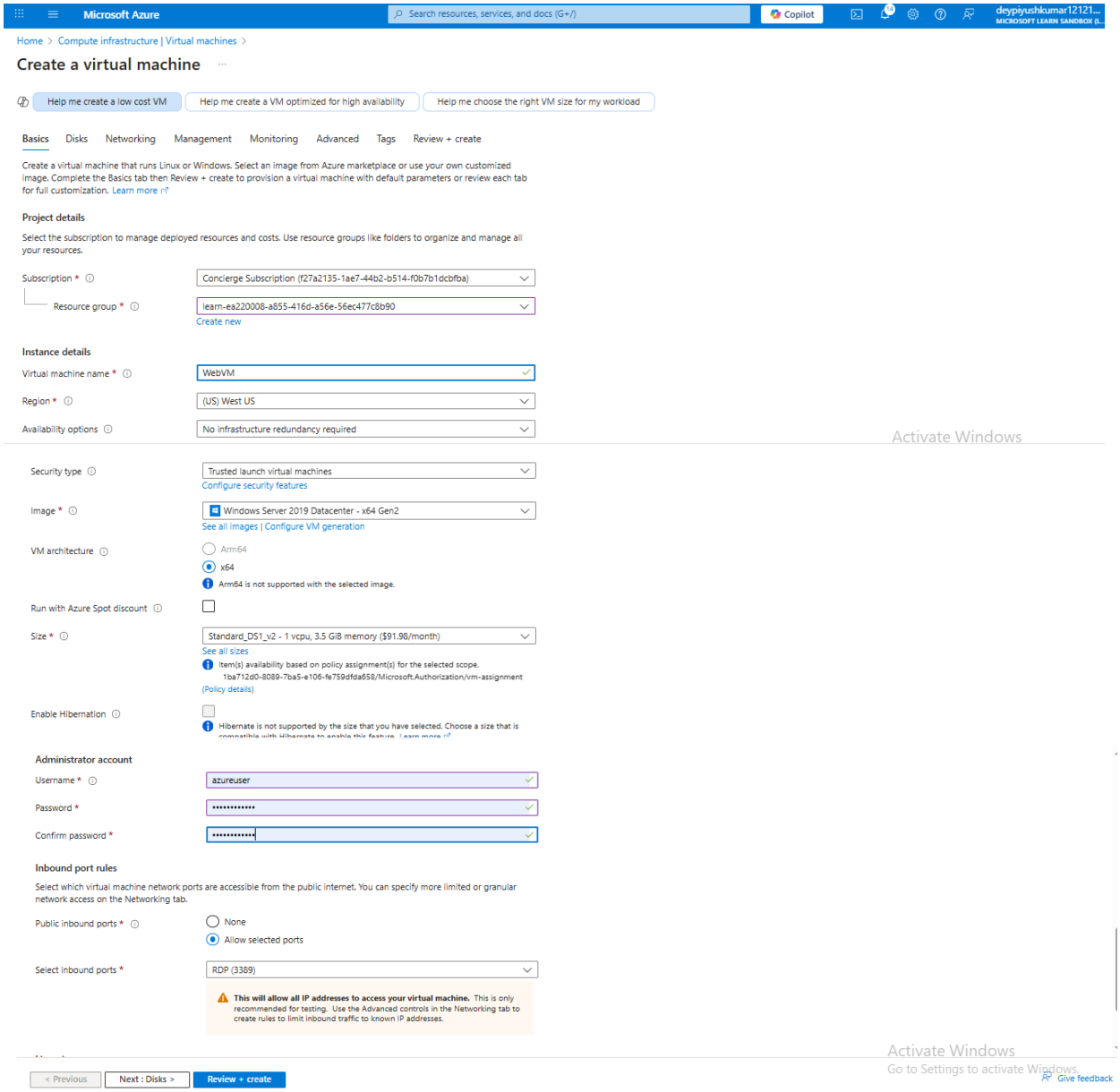
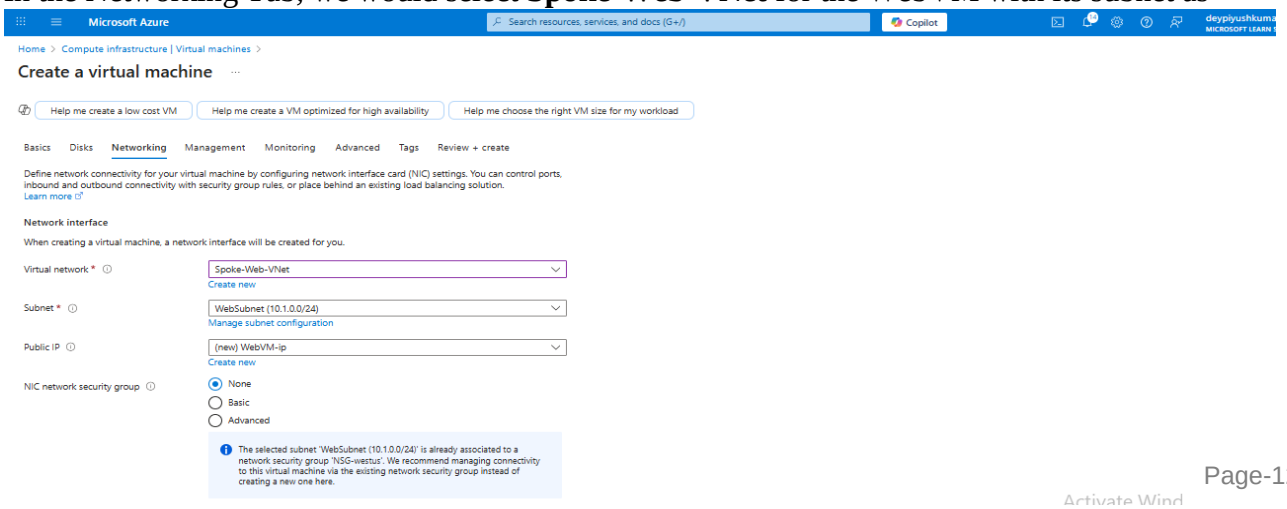


Figure: 7

In the Networking Tab, we would select **Spoke-Web-VNet** for the WebVM with its subnet as



Now, we would click on Review+ create, then it will take around 1-3 minutes to set up the virtual machine. Similarly, we would create another VM named SimulatedFirewallVM with Hub-VNet for firewall simulation. After creation of both virtual machines it would look like see in Figure: 8.

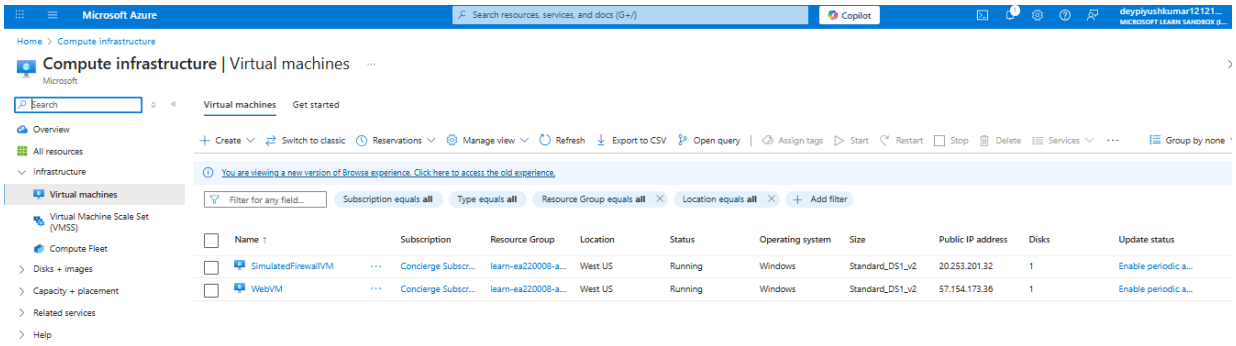
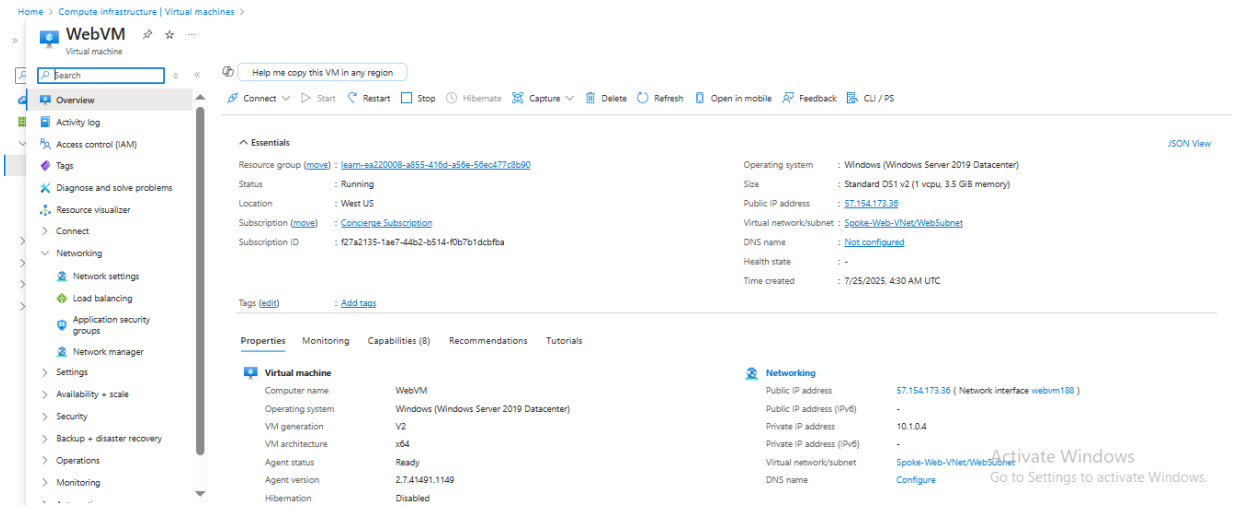
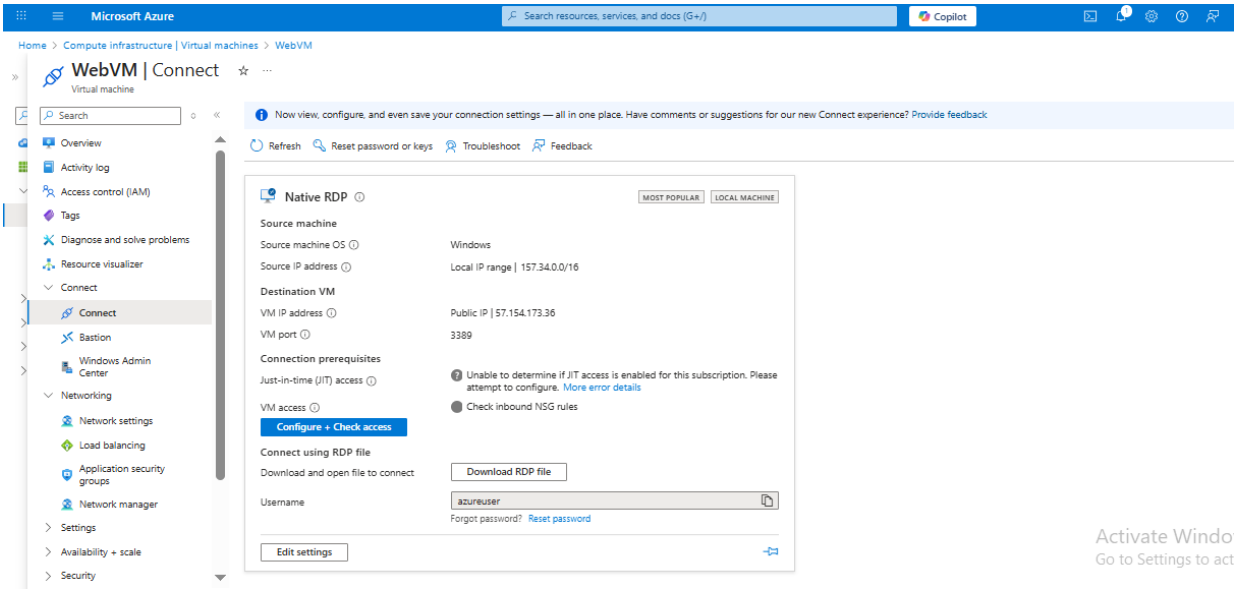


Figure: 8

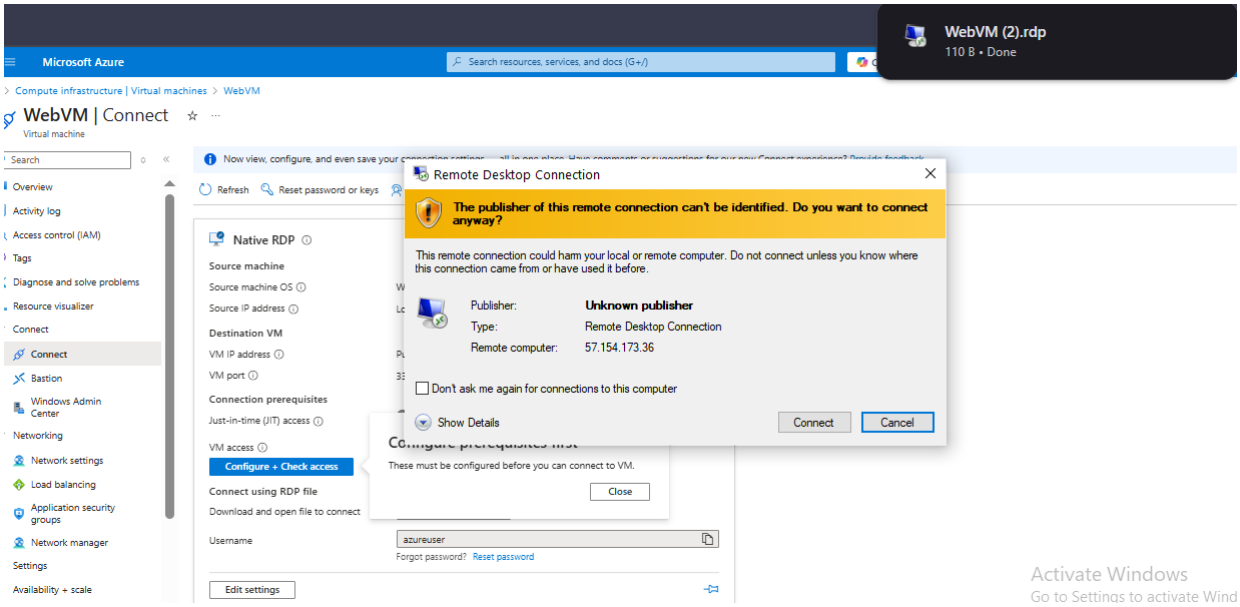
Now, at last for RDP testing, we have to go to WebVM and click on connect button.



We would download **.rdp file,** to test the virtual machine along with firewall setup.



On opening the **rdp** file, we would see the Remote Desktop as

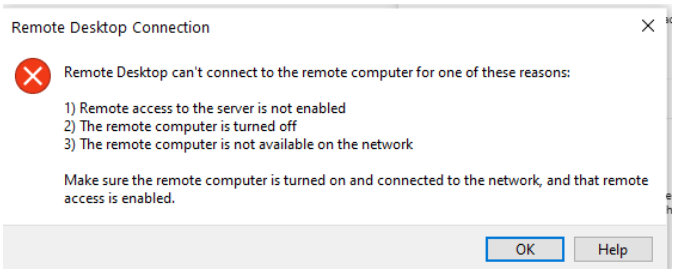


Remote Desktop Limitation (Microsoft Learn Sandbox)

Due to limitations in Microsoft Learn Sandbox, Remote Desktop Protocol (RDP) connections could not be established.

- NSGs were correctly configured with port **3389 open**.
- Public IP(57.154.173.36) was assigned to WebVM to get RDP start.
- However, attempts to RDP into the VM resulted in a **timeout**.
- This is a known sandbox policy restriction and does not reflect a misconfiguration.

This step was tested and verified as far as the sandbox permits.



Remote Desktop (RDP) was not functional due to sandbox restrictions, though all configurations were correctly completed.

CONCLUSION

This project showed how secure hosting of a web application can be done using Microsoft Azure's cloud services. The complete setup followed the hub-and-spoke network design. Each component was created step-by-step, from VNets and subnets to peering and routing. The application hosting was simulated with clear network control using NSGs, even without using the real Azure Firewall due to sandbox limits.

All configurations were tested within the free Microsoft Learn Sandbox. Even though remote access (RDP) could not be completed because of sandbox restrictions, the entire system was logically set up with correct NSG rules and routing. Custom route tables were also added to direct traffic through a simulated firewall VM, allowing centralized control of data movement across the environment.

This hands-on project helped understand Azure services like virtual networks, peering, NSGs, route tables, and VM deployment in a secure way. It followed best practices for traffic filtering and safe network design, and it used available tools smartly despite the sandbox restrictions. The overall experience improved knowledge of cloud security and hosting models in real-world cloud infrastructure projects.

REFERENCE

- Microsoft Learn – Secure your Azure network resources with NSGs and route tables

- Microsoft Learn – Design a hub and spoke network topology in Azure