

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:

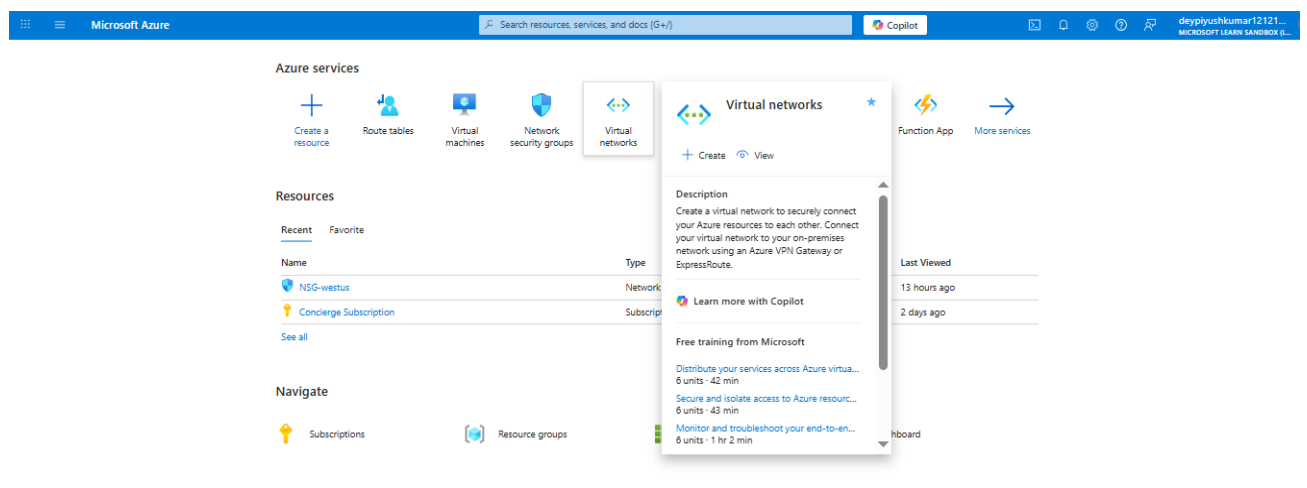


Figure:1

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.

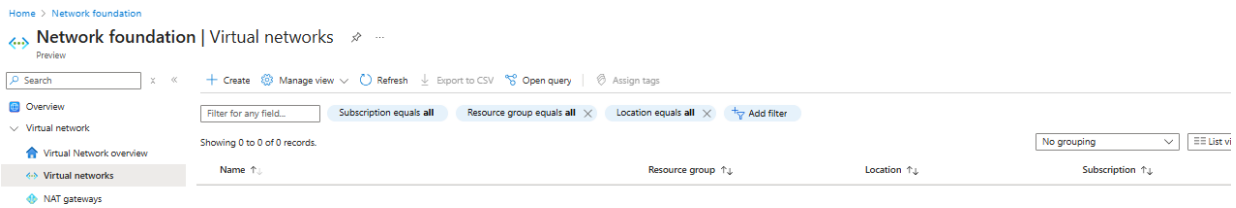
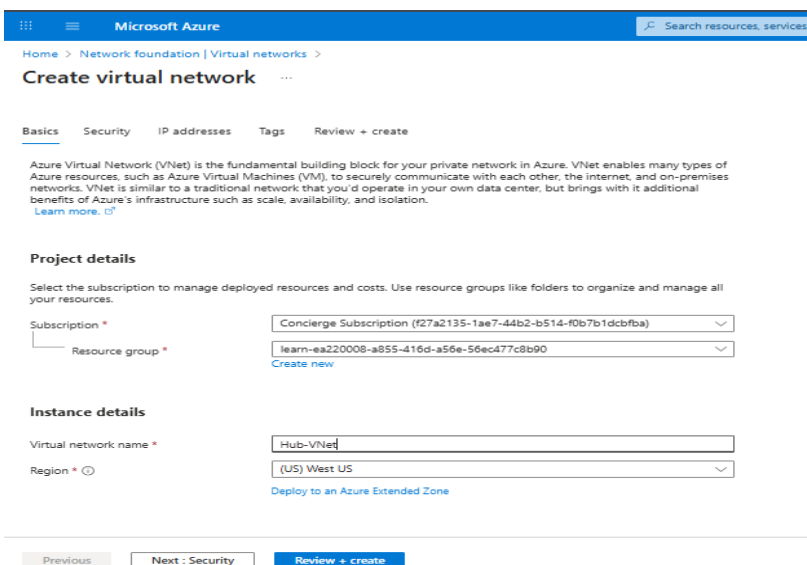


Figure: 2

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



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

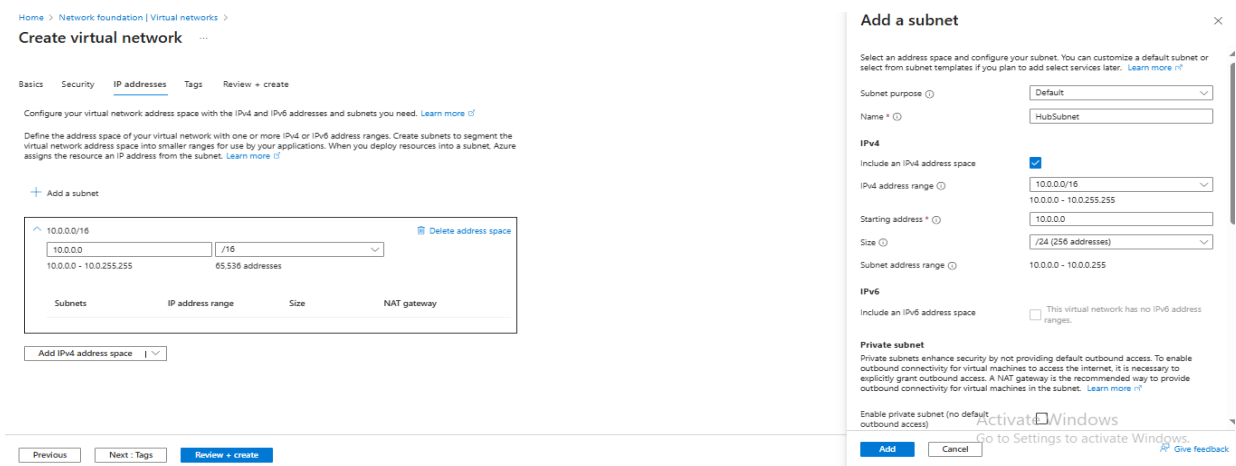


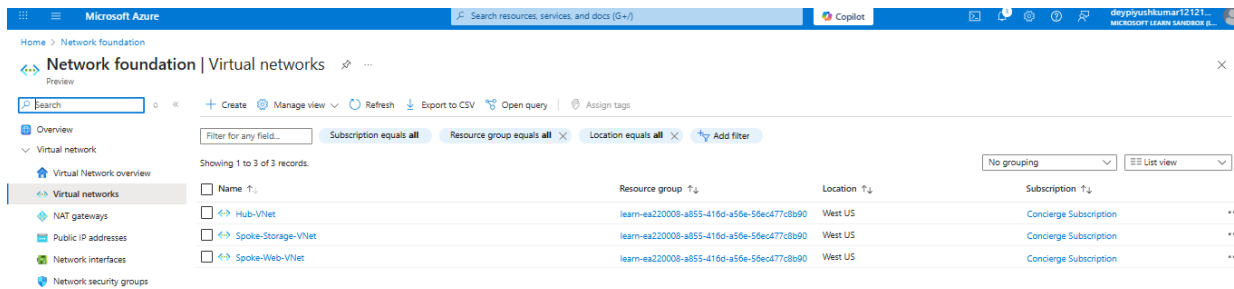
Figure: 3

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,

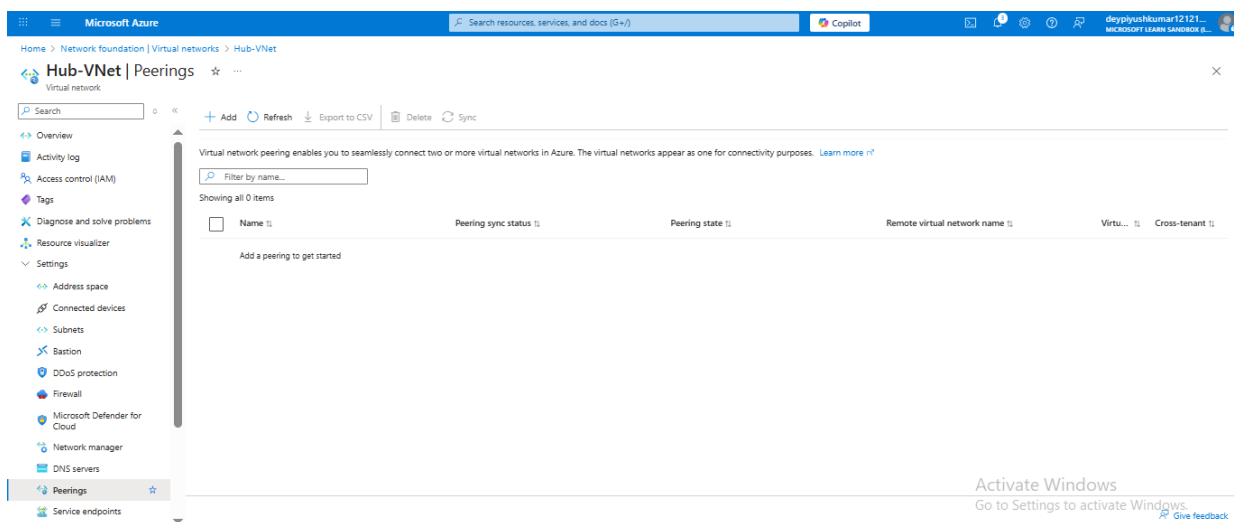
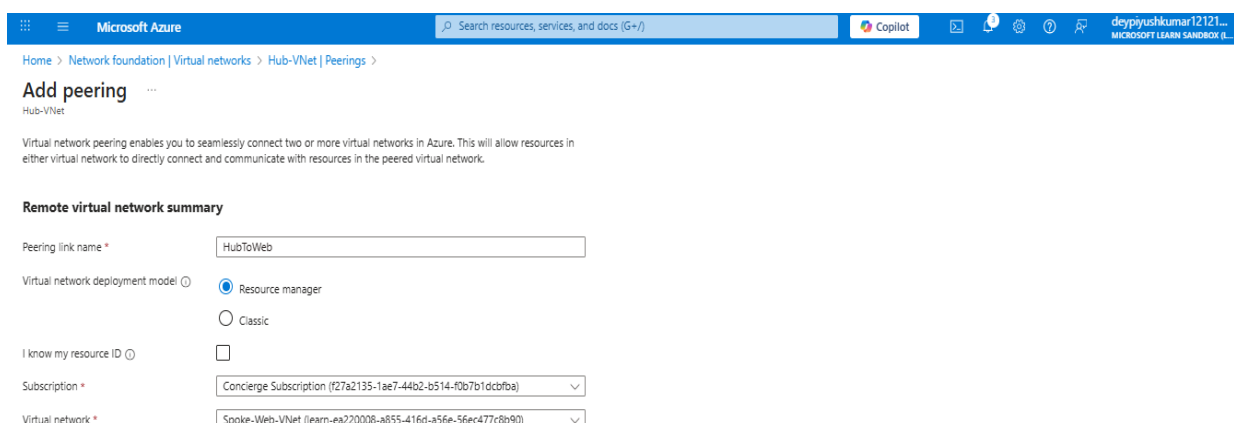


Figure:4

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.

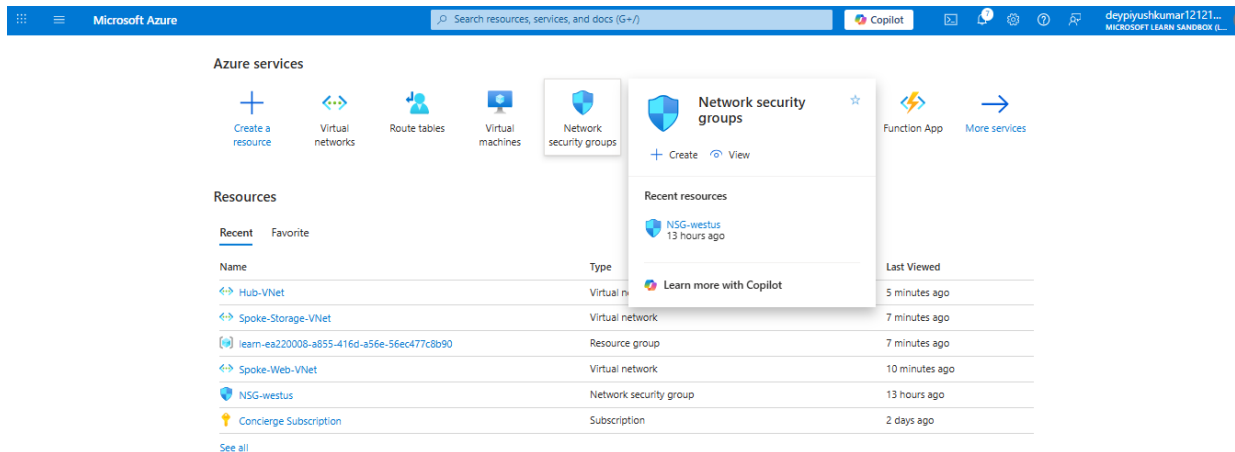


Step 4: Create Network Security Groups (NSGs)

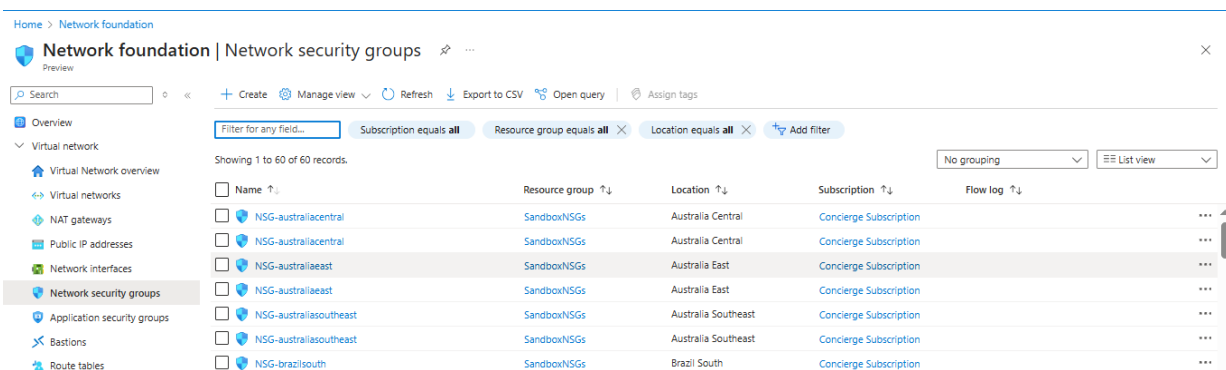
Two NSGs should be created as:

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

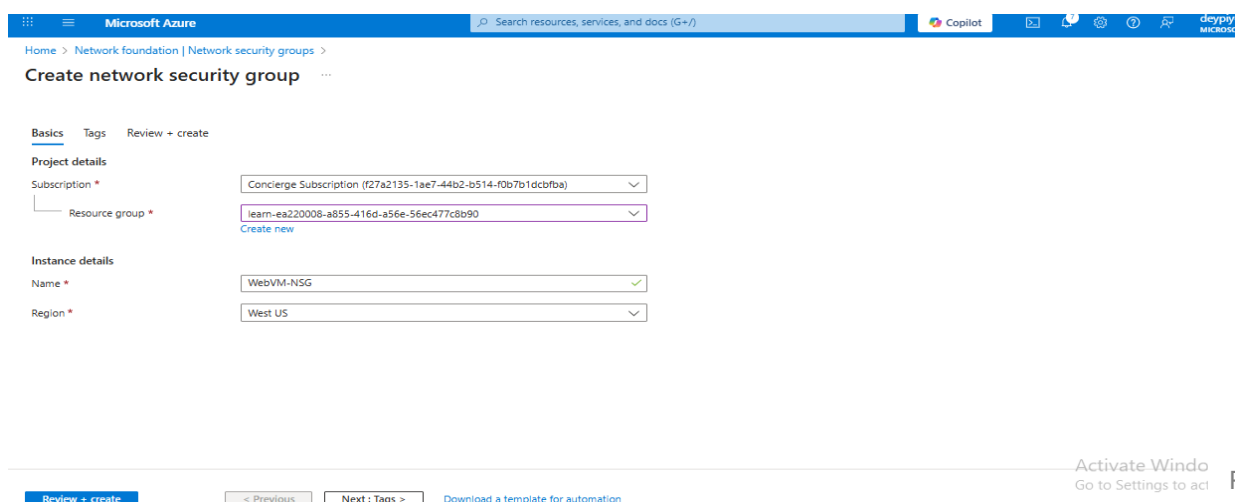
For this, at home page, we would select Network Security Groups(NSGs) tab as shown below.



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

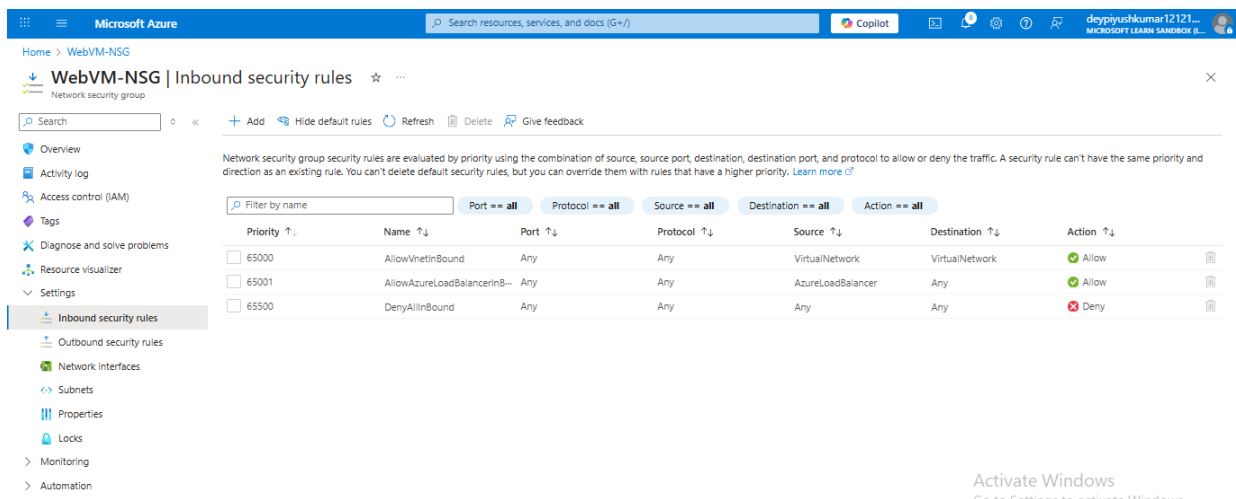


Figure: 5

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

Add inbound security rule

Source: Any

Source port ranges: *

Destination: Any

Service: Custom

Destination port ranges: 3389

Protocol: TCP

Action: Allow

Priority: 3389

Name: Allow-RDP

Description:

Add **Cancel**

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

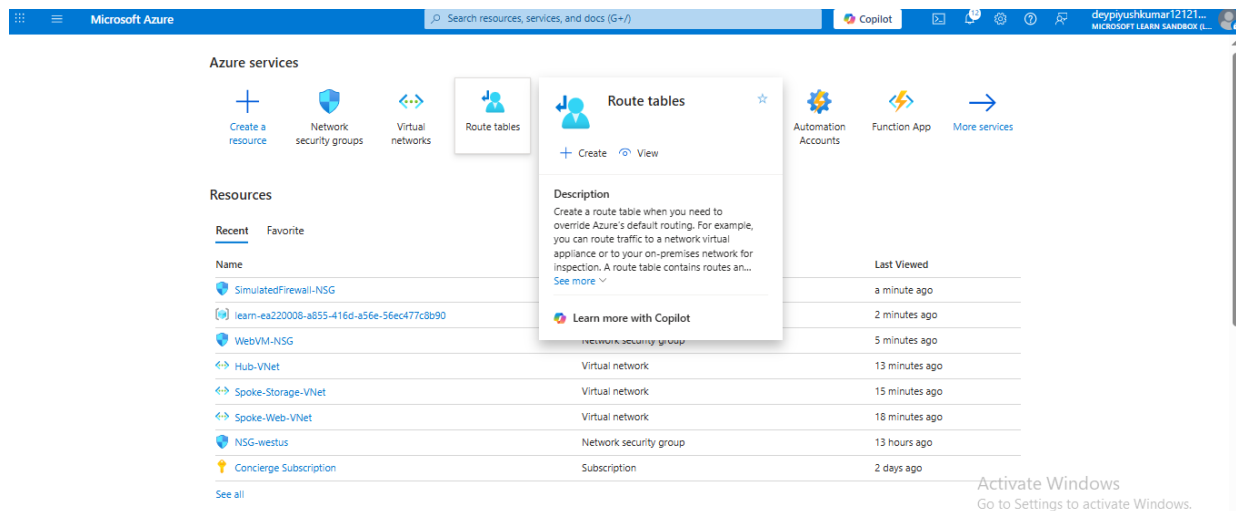
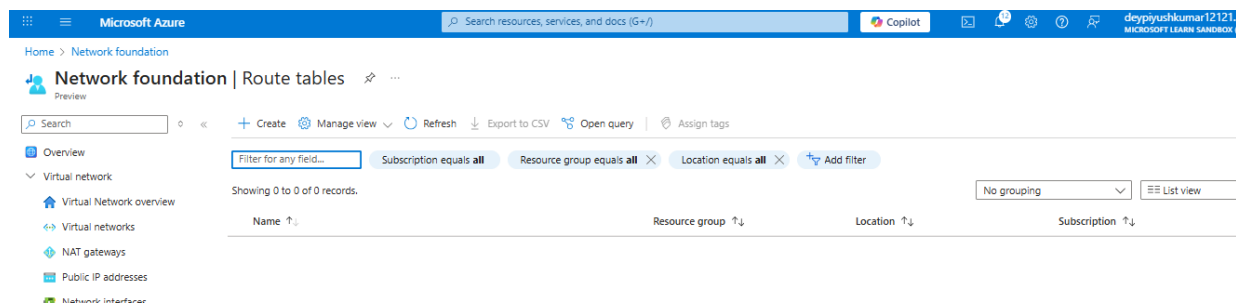
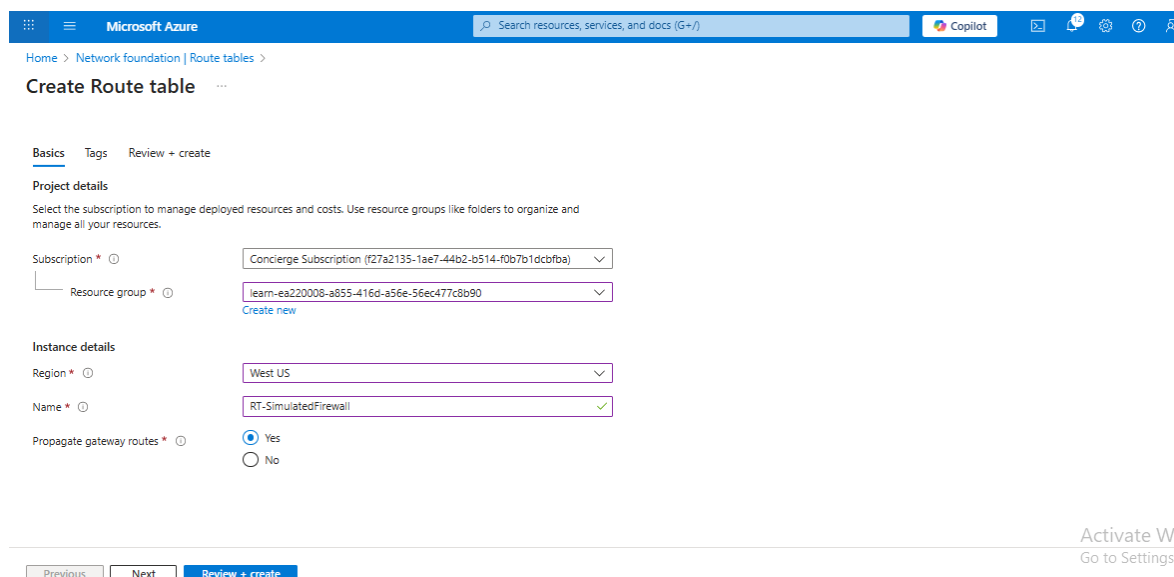


Figure: 6

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.

The screenshot shows the Azure portal interface. On the left, the 'Routes' menu is selected under the 'RT-SimulatedFirewall' route table. The main area displays the 'Add route' dialog. The dialog contains the following fields:

- Route name:** AllTrafficToFirewall
- Destination type:** IP Addresses
- Destination IP addresses/CIDR ranges:** 0.0.0.0/0
- Next hop type:** Virtual appliance
- Next hop address:** 10.0.0.4

At the bottom of the dialog, there is an 'Add' button. A note at the bottom of the dialog states: "Ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings."

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.

The screenshot shows the Azure portal home page. The 'Virtual machines' service is selected, and the 'Create' button is clicked, opening a dropdown menu. The dropdown menu contains the following options:

- Create:** A button to create a new virtual machine.
- Learn more with Copilot:** A link to learn more about Copilot.
- Free training from Microsoft:** A section with links to various training resources:
 - Introduction to Azure virtual machines (8 units - 1 hr 7 min)
 - Create a Windows virtual machine in Azure (9 units - 51 min)
 - Create a Linux virtual machine in Azure (7 units - 1 hr 26 min)

The 'Resources' section on the left shows a list of recent resources, including 'RT-SimulatedFirewall', 'SimulatedFirewall-NSG', 'WebVM-NSG', 'Hub-VNet', 'Spoke-Storage-VNet', 'Spoke-Web-VNet', 'NSG-westus', and 'Concierge Subscription'.

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.

Microsoft Azure

Home > Compute infrastructure | Virtual machines >

Create a virtual machine

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload

Basics | Disks | Networking | Management | Monitoring | Advanced | Tags | Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. [Learn more](#)

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group * [Create new](#)

Instance details

Virtual machine name *

Region *

Availability options

Security type [Configure security features](#)

Image * [See all images](#) | [Configure VM generation](#)

VM architecture ☐ Arm64 ☒ x64
Arm64 is not supported with the selected image.

Run with Azure Spot discount ☐

Size * [See all sizes](#)
Item(s) availability based on policy assignment(s) for the selected scope.
1ba712d0-8089-7ba5-e106-1e759d0da658/Microsoft.Authorization/vm-assignment (Policy details)

Enable Hibernation ☐
Hibernate is not supported by the size that you have selected. Choose a size that is compatible with hibernation to enable this feature. [Learn more](#)

Administrator account

Username *

Password *

Confirm password *

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ☐ None ☒ Allow selected ports

Select inbound ports *

⚠ This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

[Go to Settings to activate Windows.](#) [Give feedback](#)

[< Previous](#) [Next : Disks >](#) [Review + create](#)

Figure: 7

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

Microsoft Azure

Home > Compute infrastructure | Virtual machines >

Create a virtual machine

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload

Basics | Disks | **Networking** | Management | Monitoring | Advanced | Tags | Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * [Create new](#)

Subnet * [Manage subnet configuration](#)

Public IP [Create new](#)

NIC network security group ☒ None ☐ Basic ☐ Advanced

⚠ The selected subnet 'WebSubnet (10.1.0.0/24)' is already associated to a network security group 'NSG-WebSubnet'. We recommend managing connectivity to this virtual machine via the existing network security group instead of creating a new one here.

[Go to Settings to activate Windows.](#) [Give feedback](#)

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.

Name	Subscription	Resource Group	Location	Status	Operating system	Size	Public IP address	Disks	Update status
SimulatedFirewallVM	Concierge Subscr...	learn-ea220008-a...	West US	Running	Windows	Standard_DS1_v2	20.253.201.32	1	Enable periodic a...
WebVM	Concierge Subscr...	learn-ea220008-a...	West US	Running	Windows	Standard_DS1_v2	57.154.173.36	1	Enable periodic a...

Figure: 8

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

WebVM Virtual machine

Help me copy this VM in any region

Connect Start Restart Stop Hibernate Capture Delete Refresh Open in mobile Feedback CU / PS

Essentials

Resource group (move) : learn-ea220008-a855-416d-a56e-56ec477cd690

Status : Running

Location : West US

Subscription (move) : Concierge Subscription

Subscription ID : f27a2135-1ae7-44b2-b514-40b7b1dcfbfa

Operating system : Windows (Windows Server 2019 Datacenter)

Size : Standard_DS1_v2 (1 vcpu, 3.5 GiB memory)

Public IP address : 57.154.173.36

Virtual network/subnet : Spoke-Web-VNet/WebSubnet

DNS name : Not configured

Health state : -

Time created : 7/25/2025, 4:30 AM UTC

Tags (edit) : Add tags

Properties Monitoring Capabilities (8) Recommendations Tutorials

Virtual machine

Computer name : WebVM

Operating system : Windows (Windows Server 2019 Datacenter)

VM generation : V2

VM architecture : x64

Agent status : Ready

Agent version : 2.7.41491.1149

Hibernation : Disabled

Networking

Public IP address : 57.154.173.36 (Network interface webvm188)

Public IP address (IPv6) : -

Private IP address : 10.1.0.4

Private IP address (IPv6) : -

Virtual network/subnet : Spoke-Web-VNet/WebSubnet

DNS name : Configure

Activate Windows
Go to Settings to activate Windows.

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

WebVM | Connect Virtual machine

Now view, configure, and even save your connection settings — all in one place. Have comments or suggestions for our new Connect experience? Provide feedback

Refresh Reset password or keys Troubleshoot Feedback

Native RDP MOST POPULAR LOCAL MACHINE

Source machine

Source machine OS : Windows

Source IP address : Local IP range | 157.34.0.0/16

Destination VM

VM IP address : Public IP | 57.154.173.36

VM port : 3389

Connection prerequisites

Just-in-time (JIT) access : Unable to determine if JIT access is enabled for this subscription. Please attempt to configure. More error details

VM access : Check inbound NSG rules

Configure + Check access

Connect using RDP file

Download and open file to connect

Download RDP file

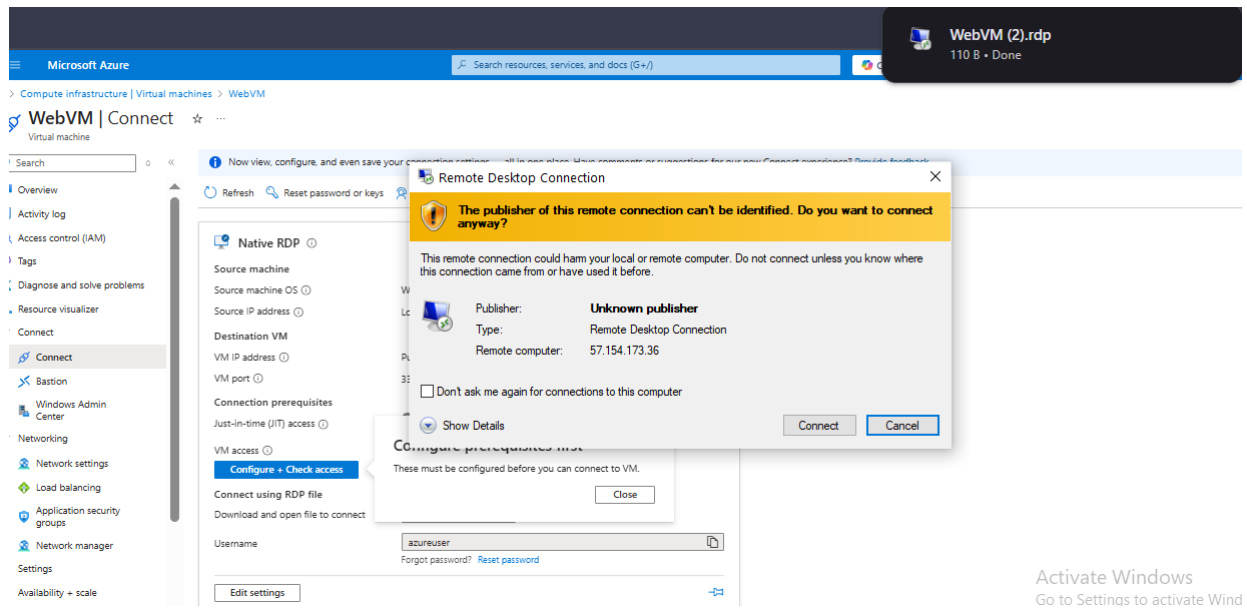
Username : azureuser

Forgot password? Reset password

Edit settings

Activate Windows
Go to Settings to activate Windows.

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

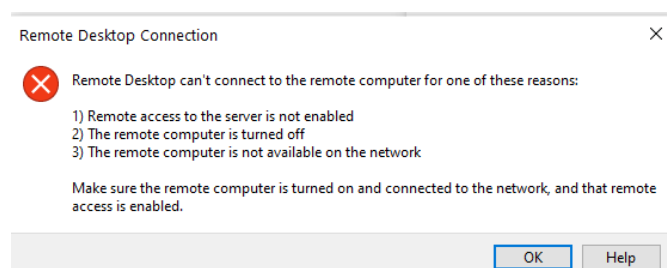


Remote Desktop Limitation (Microsoft Learn Sandbox)

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](#)