Prepared by: Nandini Goyal
Intern at Celebal Technologies
Department: Cloud Infra & Security
Date: 13 July 2025

R&D Document

Tiered Application Deployment in Azure with Subnetting, NSG Configuration, and Server Setup

Table of Contents

Serial Number	Content	Page Number
1	Objective	3
2	Azure Resources Overview 2.1 Overview of NSG 2.2 VNet and Subnet Creation	3
3	Network Design	4
4	VM Deployment and Configuration 4.1 WebLinuxVM (Ubuntu) 4.2 AppLinuxVM (Ubuntu) 4.3 DBLinuxVM (Ubuntu) 4.4 WebWindowsVM (Windows) 4.5 AppWindowsVM (Windows) 4.6 DBWindowsVM (Windows)	4
5	Network Security Group (NSG) Rules 5.1 WebNSG 5.2 AppNSG 5.3 DBNSG	11
6	Connectivity Testing	13
7	Apache and IIS Server Setup 7.1 Apache Installation on Linux VMs 7.2 IIS Installation on Windows VMs	15
8	Observations and Deviations	16
9	Conclusion	17
10	References	17

1. Objective

The objective of this week's task is to implement a 3-tier architecture in Azure by:

- Creating three subnets: Web, App, and DB.
- Deploying 2 VMs in each subnet (1 Linux + 1 Windows per tier).
- Configuring NSGs to control inter-tier communication.
- Installing Apache on Linux VMs and IIS on Windows VMs.
- Validating setup through ping and HTTP requests.

2. Azure Resources Overview

2.1 Overview of NSG

A resource group `Week6-TieredAppRG` was created to group and manage all Azure resources related to this deployment.

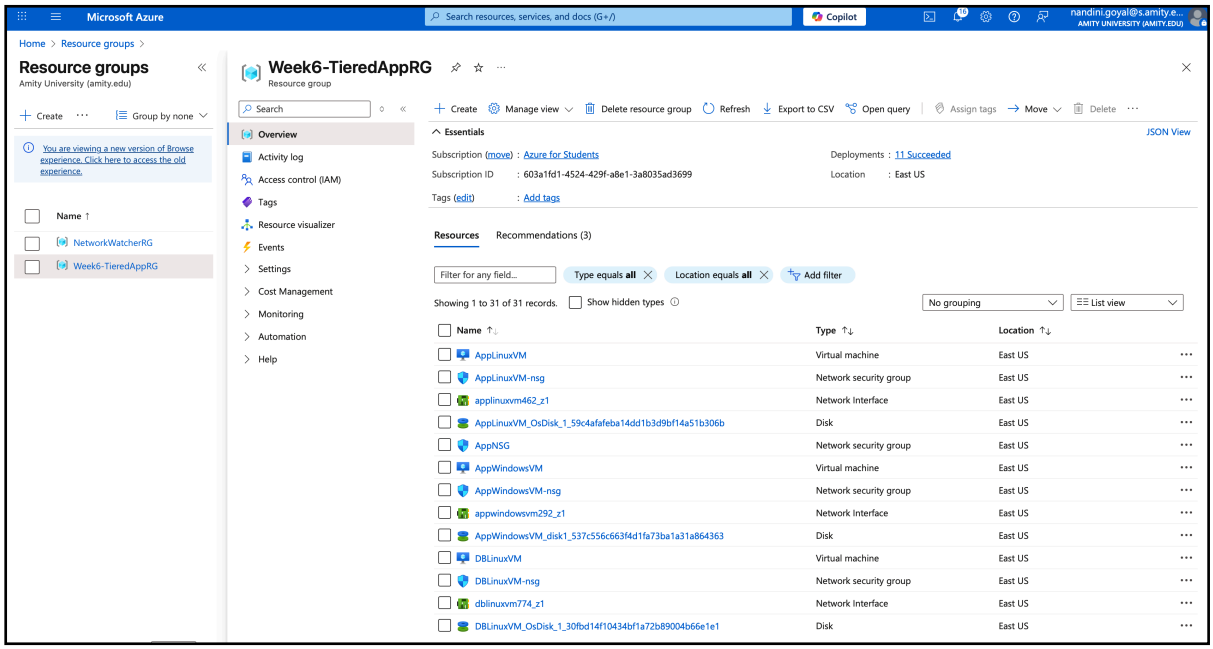


Figure 1: Week6-TieredAppRG Resource Group

2.2 VNet and Subnet Creation

A Virtual Network `Week6VNet` with address space `10.0.0.0/16` was created, and three subnets were added:

- WebSubnet: `10.0.1.0/24`

- AppSubnet: `10.0.2.0/24`
- DBSubnet: `10.0.3.0/24`

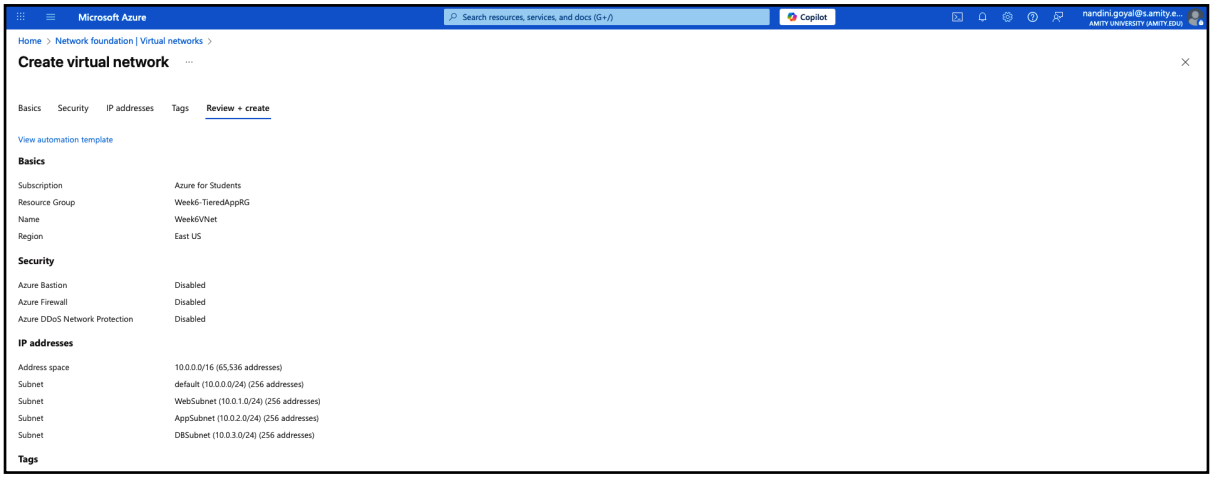


Figure 2: VNet view showing all subnets

3. Network Design

Details about tiered subnet layout and VMs.

- WebTier
- AppTier
- DBTier

4. VM Deployment and Configuration

Each subnet contains one Linux and one Windows VM, making a total of 6 VMs.

4.1 WebLinuxVM (Ubuntu)

Public IP enabled for SSH and internet access.

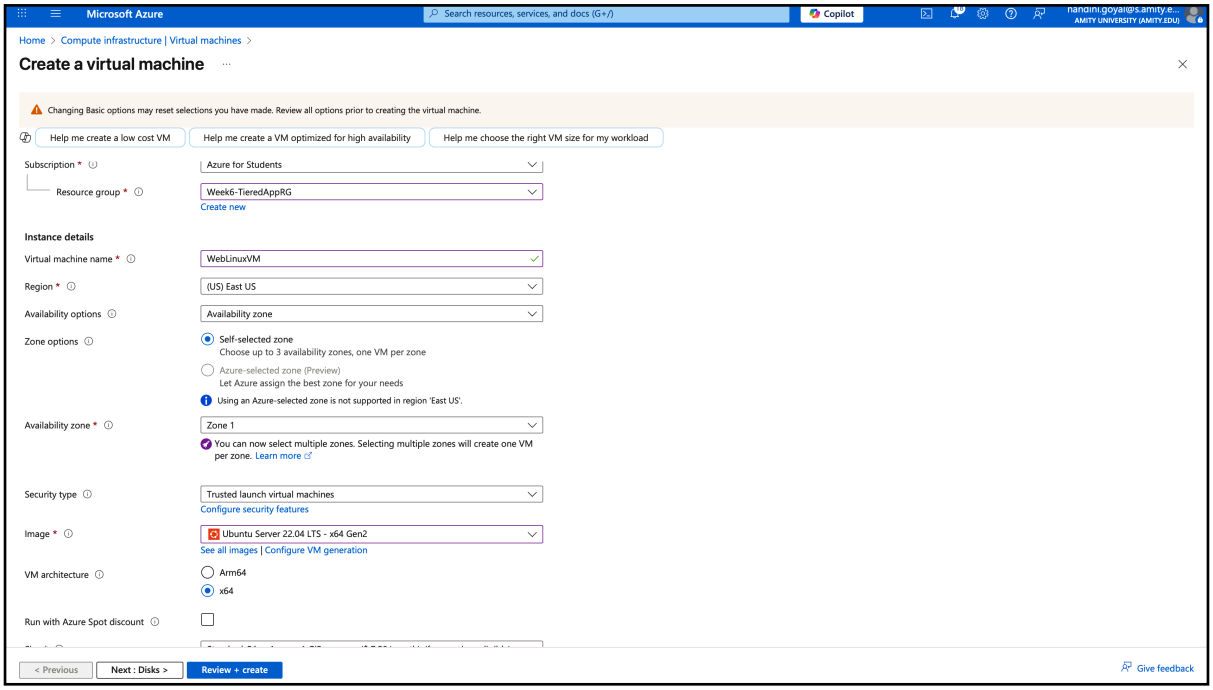


Figure 3: WebLinux VM Basics Tab

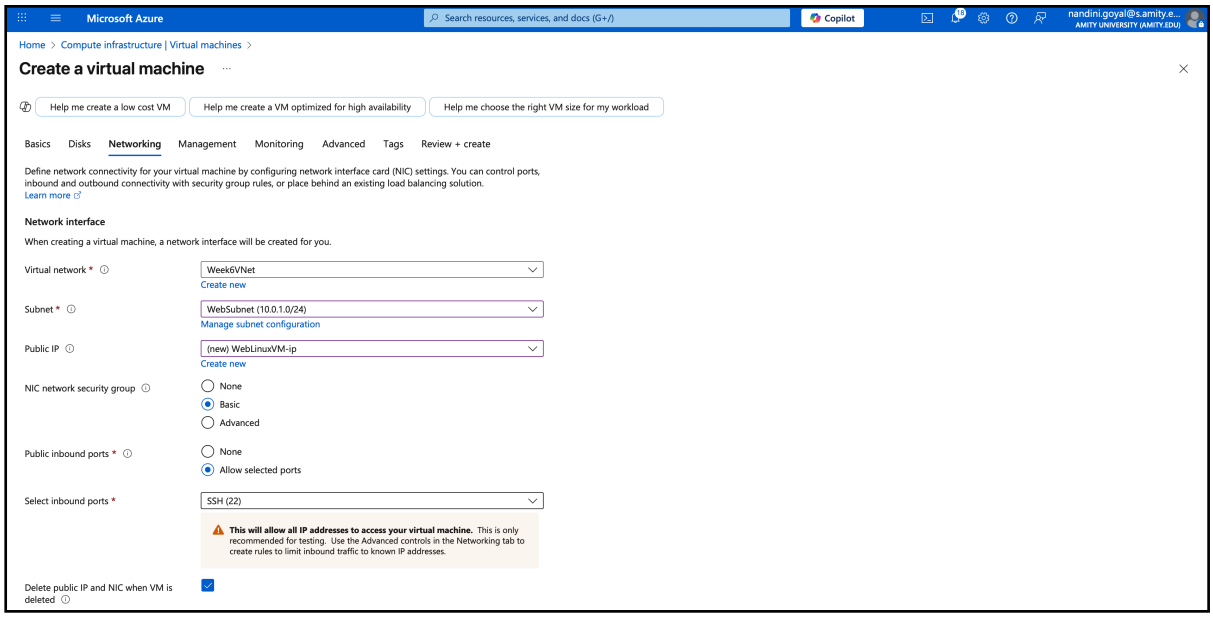


Figure 4: WebLinux VM Networkings Tab

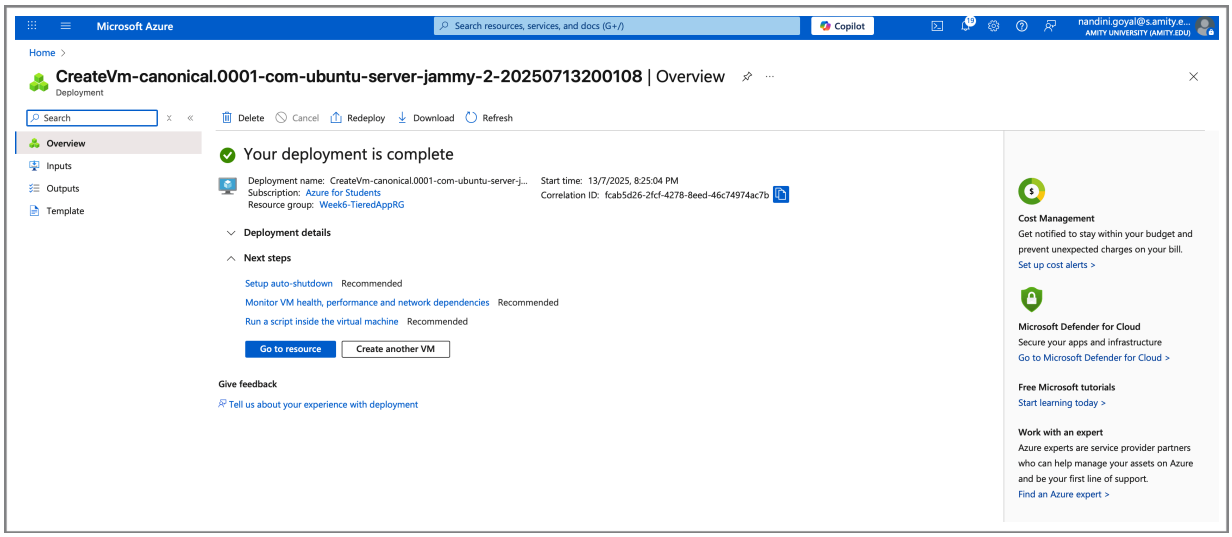


Figure 5: WebLinux VM Deployed

4.2 AppLinuxVM (Ubuntu)

Deployed in AppSubnet without a public IP. Connected via WebLinuxVM.

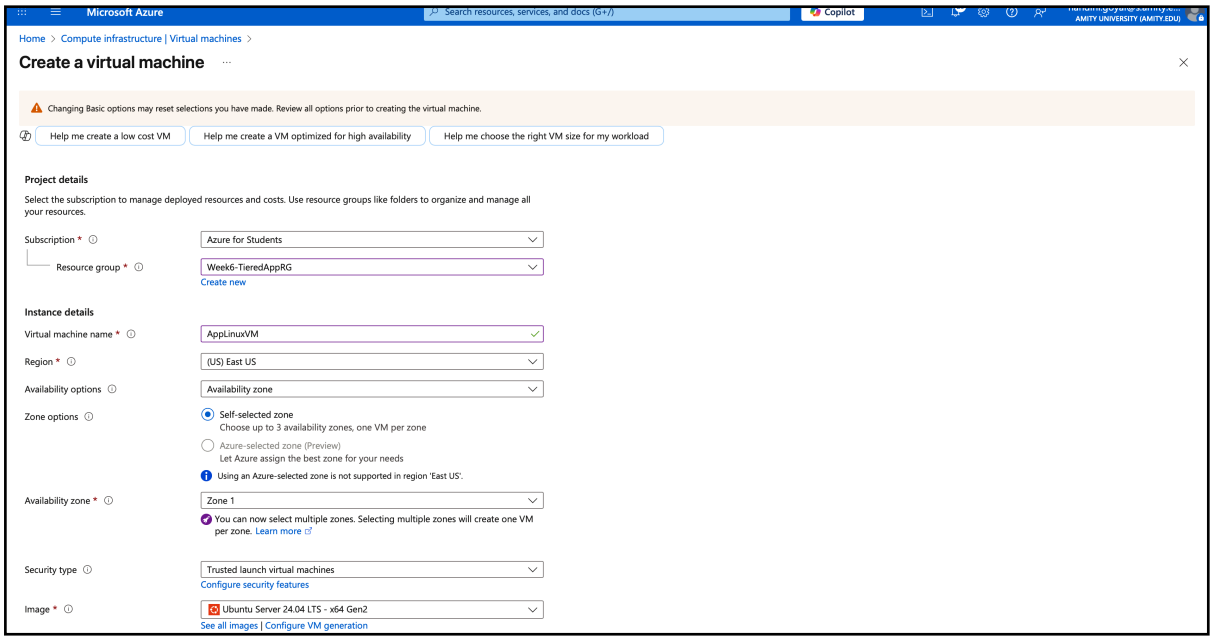


Figure 6: AppLinux VM Basics Tab

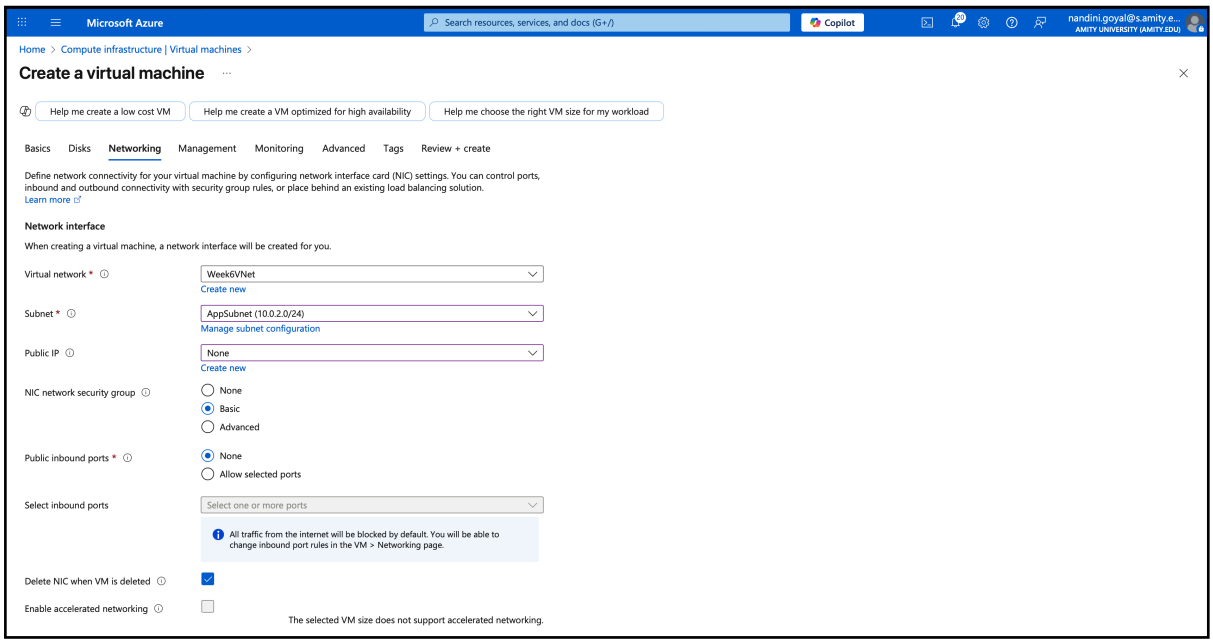


Figure 6: AppLinux VM Networkings Tab

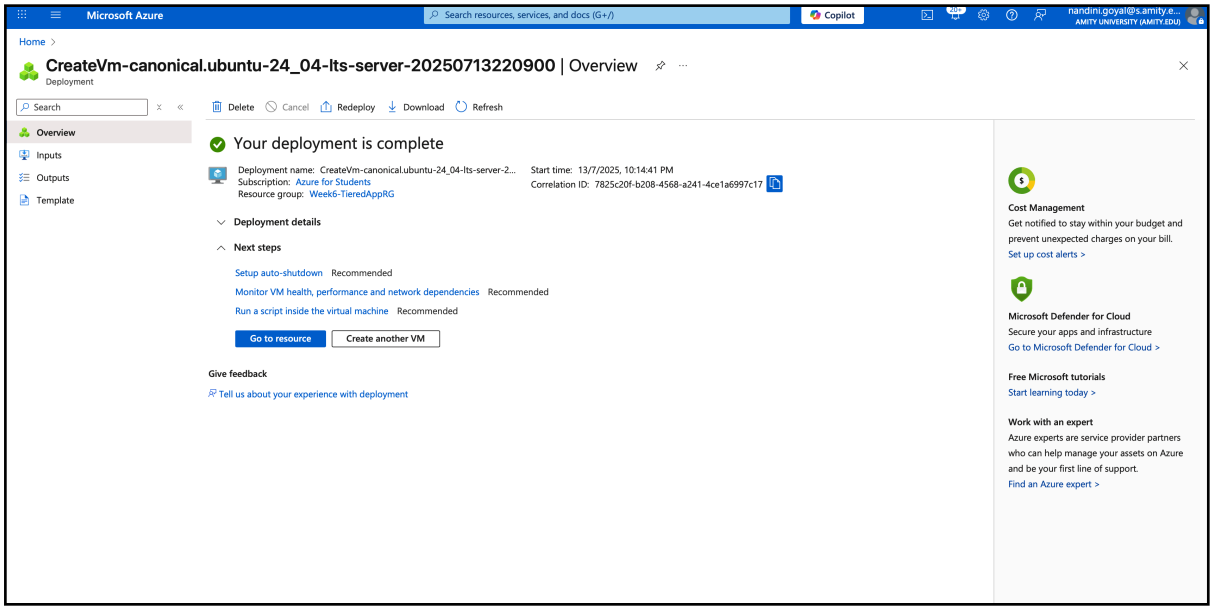


Figure 7: AppLinux VM Deployed

4.3 DBLinuxVM (Ubuntu)

Deployed in DBSubnet without public IP. Connected via AppLinuxVM.

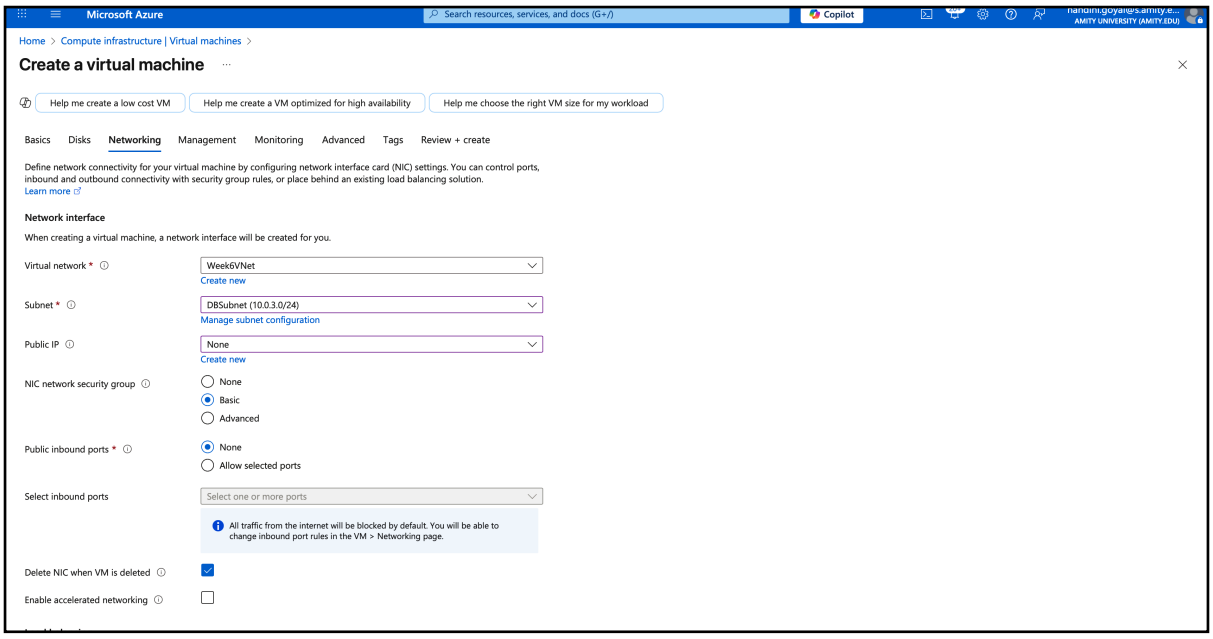


Figure 8: DBLinux VM Networkings Tab

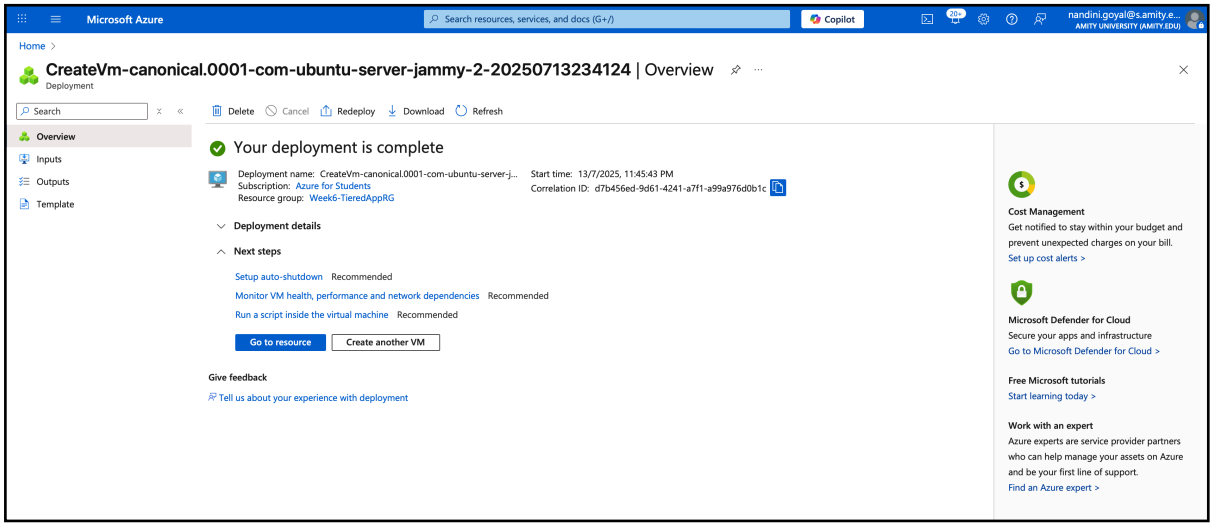


Figure 9: DBLinux VM Deployed

4.4 WebWindowsVM (Windows)

Public IP enabled for RDP and internet access.

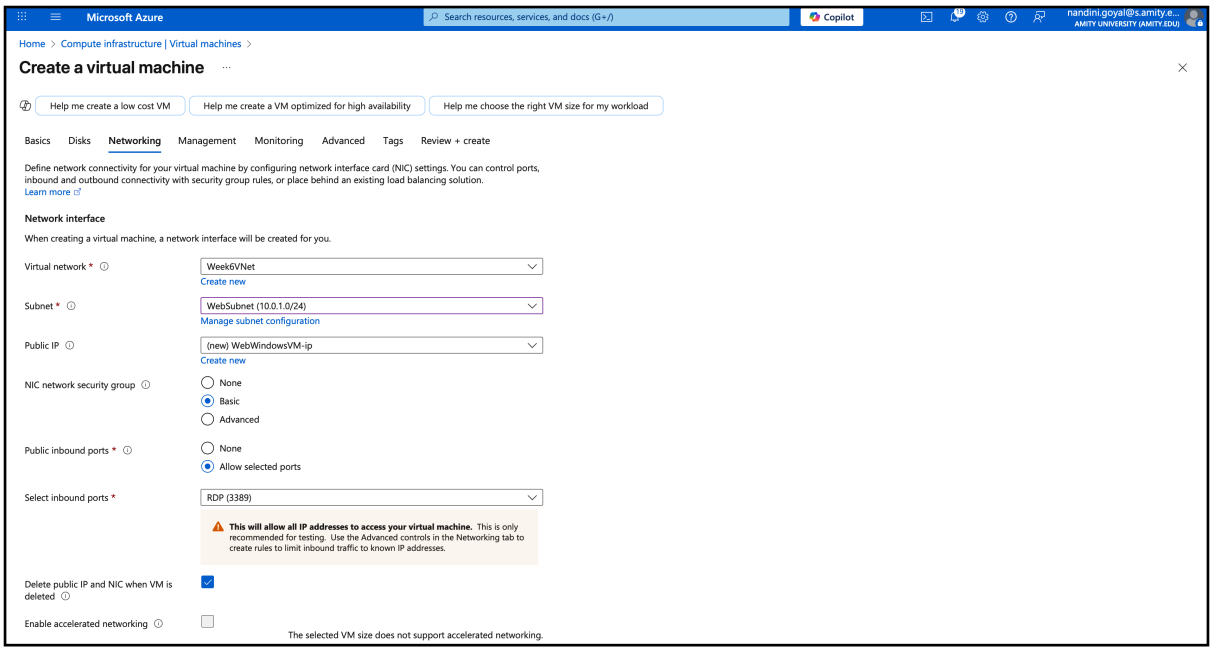


Figure 10: WebWindows VM Networkings Tab

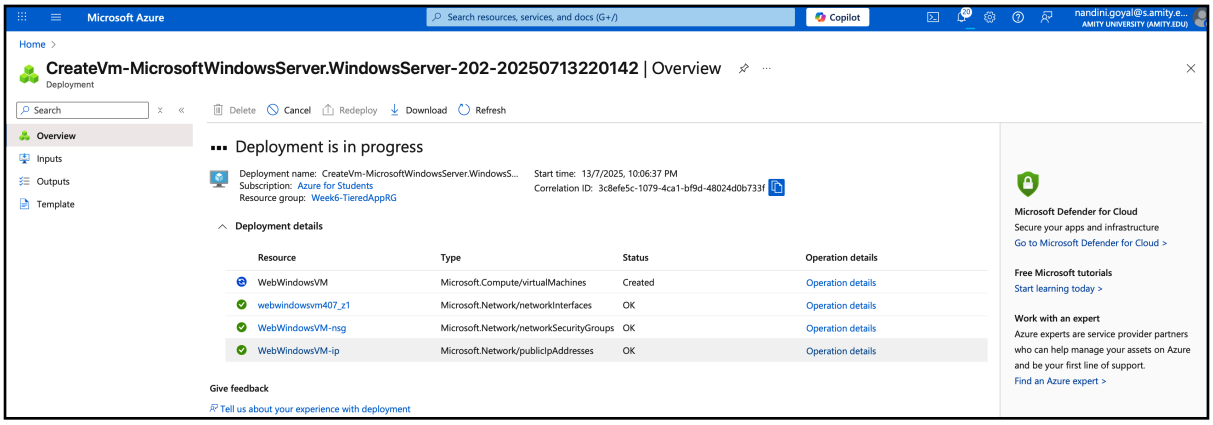


Figure 11: WebWindows VM Deployed

4.5 AppWindowsVM (Windows)

Deployed in AppSubnet without a public IP.

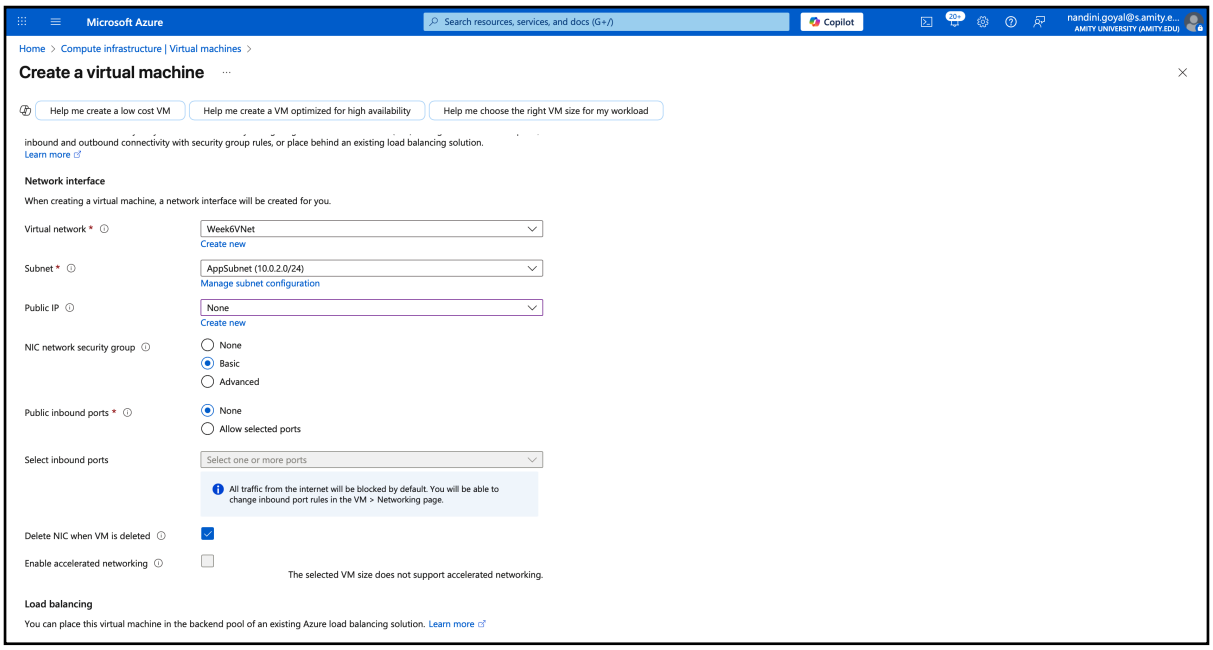


Figure 12: AppWindows VM Networkings Tab

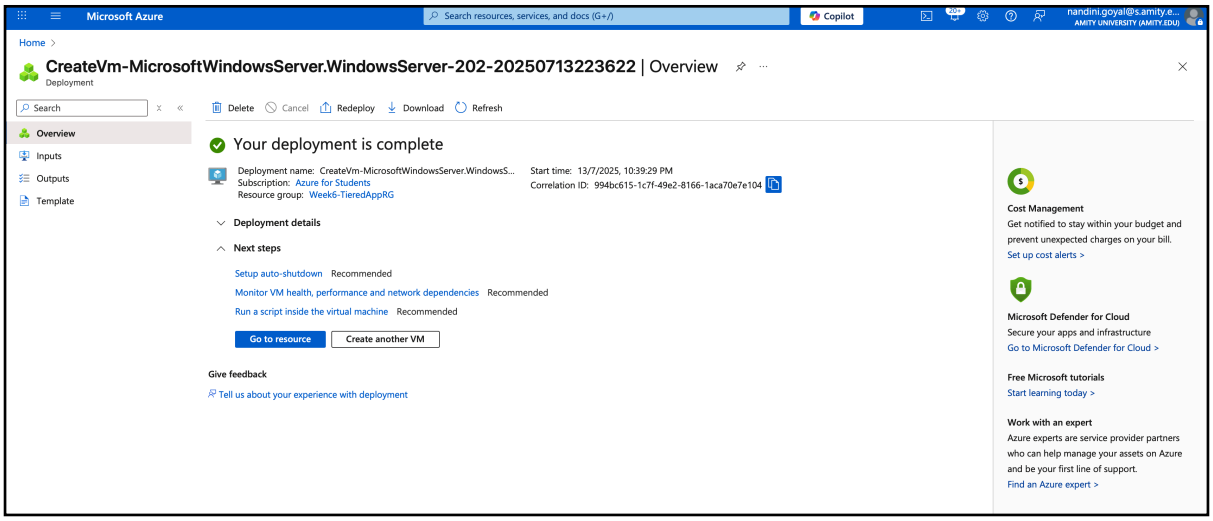


Figure 13: AppWindows VM Deployed

4.6 DBWindowsVM (Windows)

Deployed in DBSubnet without a public IP.

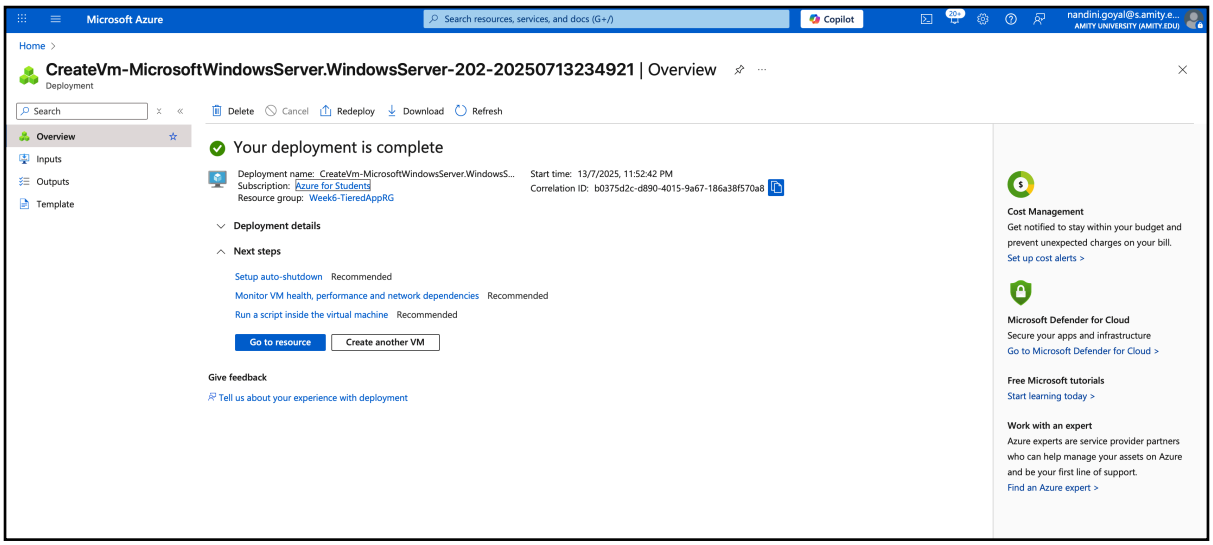


Figure 14: DBWindows VM Deployed

5. Network Security Group (NSG) Rules

5.1 WebNSG

Configured to allow internet access and communication with App tier.

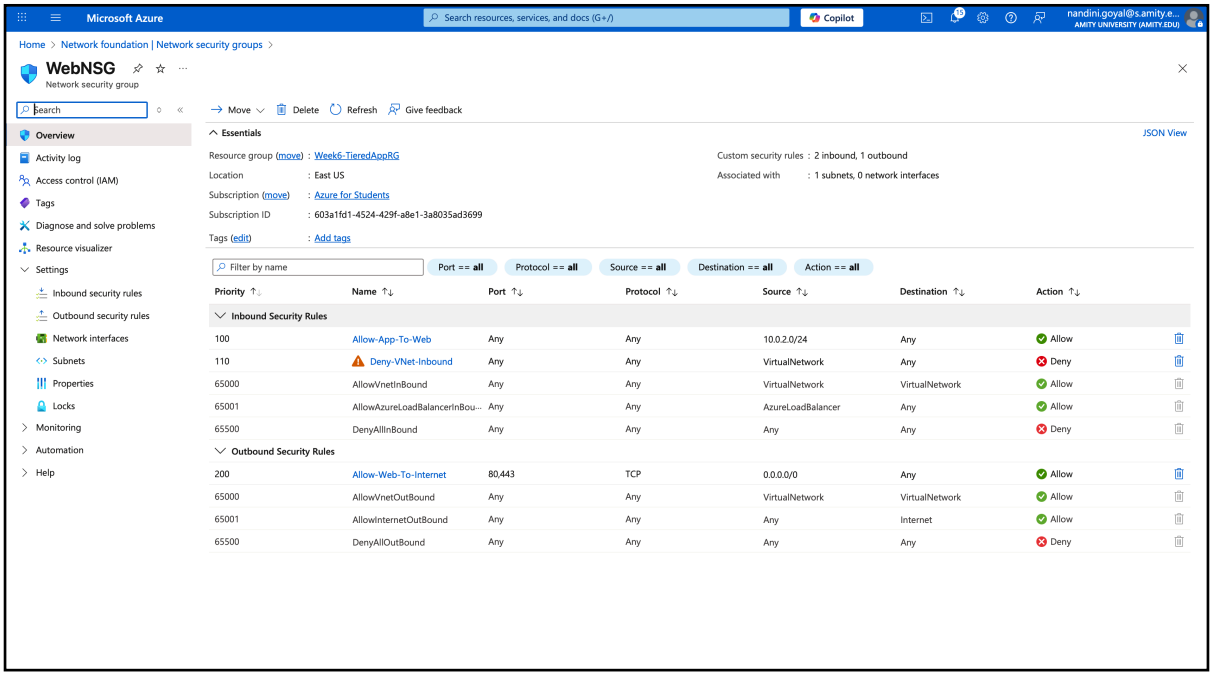


Figure 15: WebNSG Rules

5.2 AppNSG

Allows communication with both Web and DB tiers. Blocks DB from initiating connection.

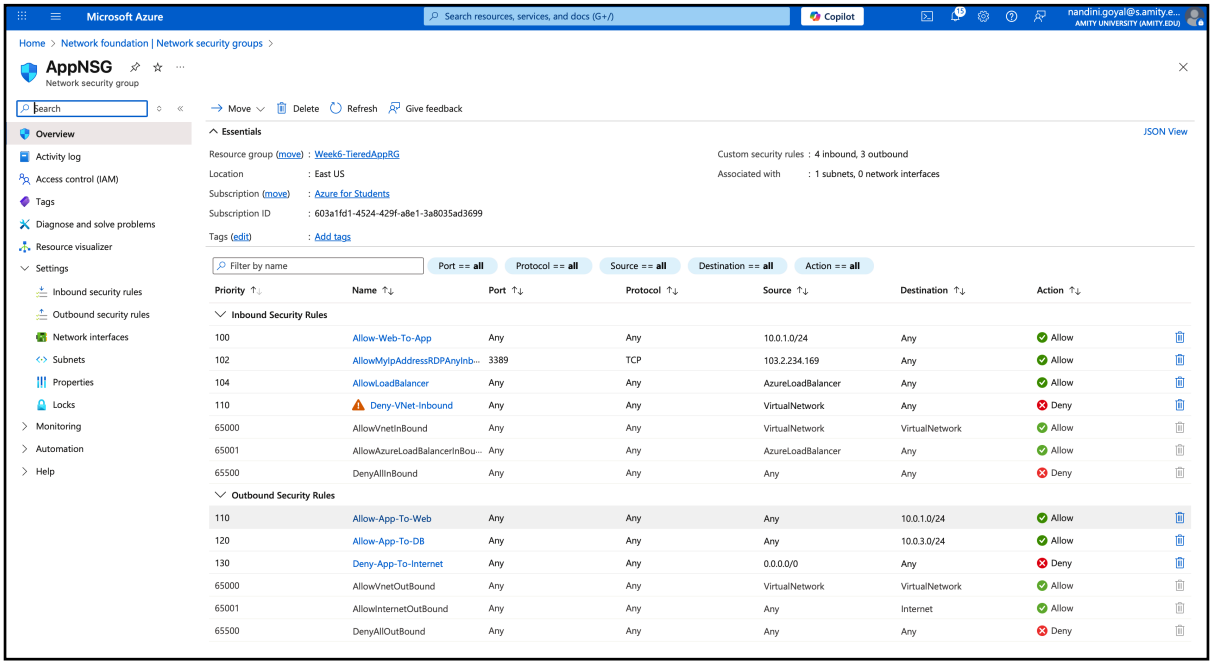
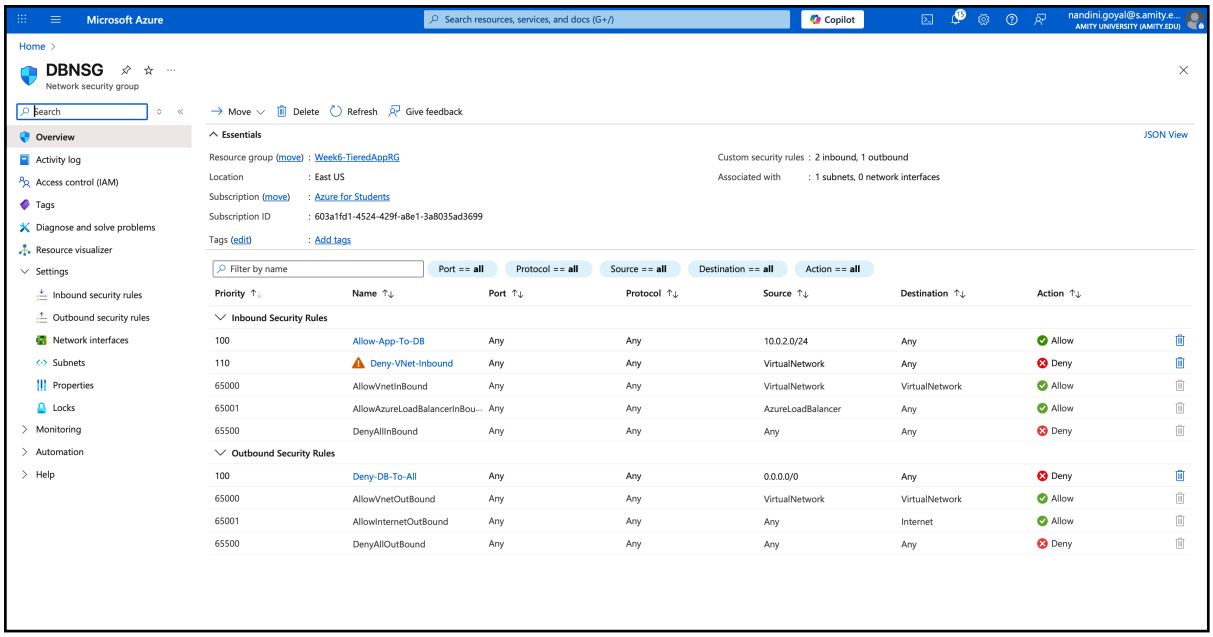


Figure 16: AppNSG Rules

5.3 DBNSG

Blocks all inbound and outbound traffic except from App tier to DB.



.**Figure 17:** DBNSG Rules

6. Connectivity Testing

Connectivity was verified using `ping` commands between Linux VMs as per the NSG rules.

Source VM	Destination VM	Expected	Result
WebLinuxVM	AppLinuxVM	Yes	Success
WebLinuxVM	DBLinuxVM	No	Fail
AppLinuxVM	WebLinuxVM	Yes	Success
AppLinuxVM	DBLinuxVM	Yes	Success
DBLinuxVM	AppLinuxVM	No	Fail
DBLinuxVM	WebLinuxVM	No	Fail

```
linuxuser1@WebLinuxVM:~$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=1.10 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=1.92 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=0.969 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=2.47 ms
^C
--- 10.0.2.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 0.969/1.613/2.465/0.610 ms
```

.**Figure 18:** Ping WebLinux VM to AppLinuxVM

```
linuxuser1@WebLinuxVM:~$ ping 10.0.3.4
PING 10.0.3.4 (10.0.3.4) 56(84) bytes of data.
^C
--- 10.0.3.4 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10225ms
```

.**Figure 19:** Ping WebLinux VM to DBLinuxVM

```
linuxuser2@AppLinuxVM:~$ ping 10.0.1.4
PING 10.0.1.4 (10.0.1.4) 56(84) bytes of data.
64 bytes from 10.0.1.4: icmp_seq=1 ttl=64 time=0.914 ms
64 bytes from 10.0.1.4: icmp_seq=2 ttl=64 time=1.19 ms
64 bytes from 10.0.1.4: icmp_seq=3 ttl=64 time=2.39 ms
64 bytes from 10.0.1.4: icmp_seq=4 ttl=64 time=1.46 ms
64 bytes from 10.0.1.4: icmp_seq=5 ttl=64 time=1.26 ms
^C
--- 10.0.1.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4034ms
rtt min/avg/max/mdev = 0.914/1.441/2.394/0.507 ms
```

Figure 20: Ping AppLinux VM to WebLinuxVM

```
linuxuser2@AppLinuxVM:~$ ping 10.0.3.4
PING 10.0.3.4 (10.0.3.4) 56(84) bytes of data.
64 bytes from 10.0.3.4: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 10.0.3.4: icmp_seq=2 ttl=64 time=1.11 ms
64 bytes from 10.0.3.4: icmp_seq=3 ttl=64 time=1.36 ms
^C
--- 10.0.3.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.106/1.281/1.380/0.124 ms
```

Figure 21: Ping AppLinux VM to DBLinuxVM

```
linuxuser3@DBLinuxVM:~$ ping 10.0.1.4
PING 10.0.1.4 (10.0.1.4) 56(84) bytes of data.
^C
--- 10.0.1.4 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1020ms
```

Figure 22: Ping DBLinux VM to WebLinuxVM

```
linuxuser3@DBLinuxVM:~$ ping 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
^C
--- 10.0.2.4 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3050ms
```

Figure 22: Ping DBLinux VM to AppLinuxVM

7. Apache and IIS Server Setup

7.1 Apache Installation on Linux VMs

Apache was installed using the following commands:

```
sudo apt update
sudo apt install apache2 -y
curl http://localhost
```

The output from `curl` confirmed successful installation.

```
linuxuser1@WebLinuxVM:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Sun 2025-07-13 18:59:23 UTC; 1h 41min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 5138 (apache2)
      Tasks: 55 (limit: 1009)
     Memory: 5.2M
        CPU: 323ms
     CGroup: /system.slice/apache2.service
             ├─5138 /usr/sbin/apache2 -k start
             ├─5140 /usr/sbin/apache2 -k start
             └─5141 /usr/sbin/apache2 -k start

Jul 13 18:59:23 WebLinuxVM systemd[1]: Starting The Apache HTTP Server...
Jul 13 18:59:23 WebLinuxVM systemd[1]: Started The Apache HTTP Server.
```

Figure 23: Apache Server

```
linuxuser1@WebLinuxVM:~$ curl https://10.0.1.4
curl: (7) Failed to connect to 10.0.1.4 port 443 after 0 ms: Connection refused
linuxuser1@WebLinuxVM:~$ curl http://10.0.1.4
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2022-03-22
    See: https://launchpad.net/bugs/1966004
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
  * {
    margin: 0px 0px 0px 0px;
    padding: 0px 0px 0px 0px;
  }

  body, html {
    padding: 3px 3px 3px 3px;

    background-color: #D8DBE2;

    font-family: Ubuntu, Verdana, sans-serif;
    font-size: 11pt;
    text-align: center;
  }

  div.main_page {
    position: relative;
    display: table;

    width: 800px;

    margin-bottom: 3px;
    margin-left: auto;
    margin-right: auto;
    padding: 0px 0px 0px 0px;

    border-width: 2px;
```

Figure 24: Apache Curl Output

7.2 IIS Installation on Windows VMs

Steps performed:

1. Connected via RDP.
2. Opened PowerShell as Admin.
3. Installed IIS:

Install-WindowsFeature -name Web-Server -IncludeManagementTools

4. Verified by visiting http://localhost.

8. Observations and Deviations

- Apache was installed and tested via terminal due to browser access issues.
- IIS was installed via PowerShell.
- Ping tests helped confirm the proper application of NSG rules.
- Couldn’t assign a public IP to App/DB VMs — used jump-box method instead.

9. Conclusion

The 3-tier architecture was successfully implemented using Azure VNets, subnets, and NSGs. VM connectivity was properly segmented with Apache and IIS servers installed on the respective operating systems. Network security and access were thoroughly tested using ping and HTTP requests. This setup simulates real-world tiered deployments and strengthens understanding of Azure infrastructure management.

10. References

- Azure NSG Documentation
- Apache Installation on Ubuntu
- IIS Installation Guide
- Azure VM SSH Access