Azure Multi-Subnet Configuration

Get in touch

📧ikuesan1503@gmail.com

📞07561 735514

Skill Set

🛜Azure Virtual Network

🧱Subnets

📈Subnet IP Ranges

📩IP Addressing

🛑NSG’s

🖥️Azure Virtual Machine

🌐VM Network Interface

📶Connectivity Testing

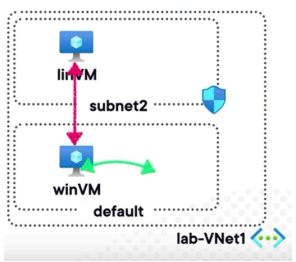
Certifications

Microsoft Azure Fundamentals (AZ-900)

Working towards the Microsoft Azure Administration (AZ-104)

Scenario

Your company wants to block communication between two VMs. They currently reside on the same network and on the same subnet. Ensure they are separated into different subnets and communication is blocked using a network security group.



About the project

This lab demonstrates how to **create multiple subnets in Azure** to *isolate virtual machines* and *control communication* between them using **network security groups (NSGs)**.

The goal was to **separate two VMs onto different subnets** and **block traffic** between them. Key tasks included **creating a new subnet**, **updating VM network interfaces**, and **configuring inbound and outbound NSG rules**.

Successful completion was supposed to be validated by **testing connectivity between the VMs**, however that was not the case. **Command Prompt** could receive a ping reading, maybe because:

- They were still on the *same subnet*, or
- They were on *different subnets* but no **Network Security Group (NSG) rules** were blocking ICMP traffic between them (which was *not the case*).

I tried to troubleshoot the issue by **verifying subnet values**, **confirming the correct subnet**, **checking the NSG**, **deallocating the VM**, and **reviewing the NIC settings**. These steps restored functionality, but the project’s goal —to **block communication between the VMs using a Network Security Group**—was not fully achieved before the lab timed out.

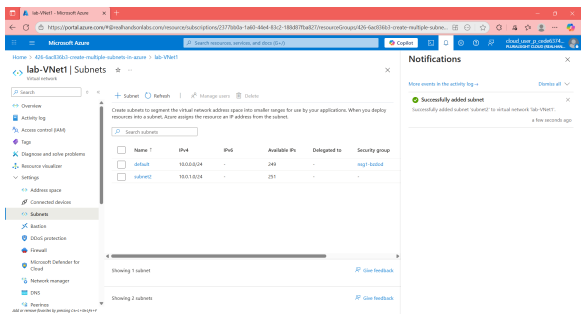
https://github.com/Ikedrew?tab=repositories

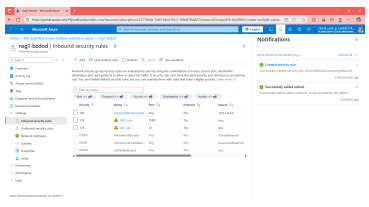
www.linkedin.com/in/andrew-ikuesan-525654183

Configuring Subnets and Network Security Groups to Isolate VM Communication in Azure

From the resource group, I opened the pre-deployed virtual network and navigated to **Subnets** under *settings*.

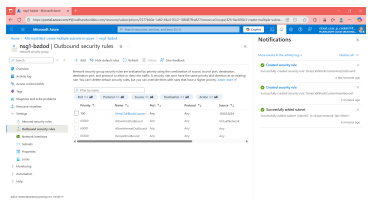
I created a new subnet named **subent2**, copied its IP range for later use, save the changes and noted the default subnet’s IP range before returning to the resource group.

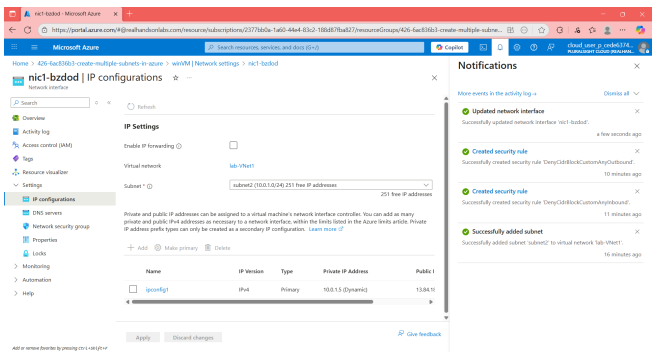




In the pre-deployed Network Security Group, I navigated to **Inbound security rules** and added a new rule. I set the source to **10.0.1.0/24**, the destination to **10.0.0.0/24**, allowed all ports, set the action to **Deny**, and assigned a priority of **100** to ensure it took precedence over existing rules before saving the rule.

I navigated to **Outbound security rules** in the Network Security Group and added a new rule. The source was set to **10.0.0.0/24**, the destination to **10.0.1.0/24**, all ports were included, the action was **Deny**, and the priority was **100** to override existing rules before saving and returning to the resource group.



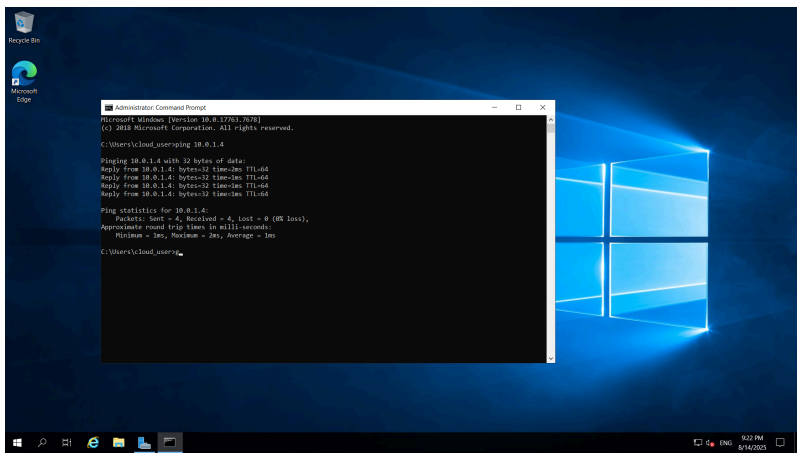


I opened **linVM**, went to **Network Settings**, and accessed its NIC through the provided hyperlink.

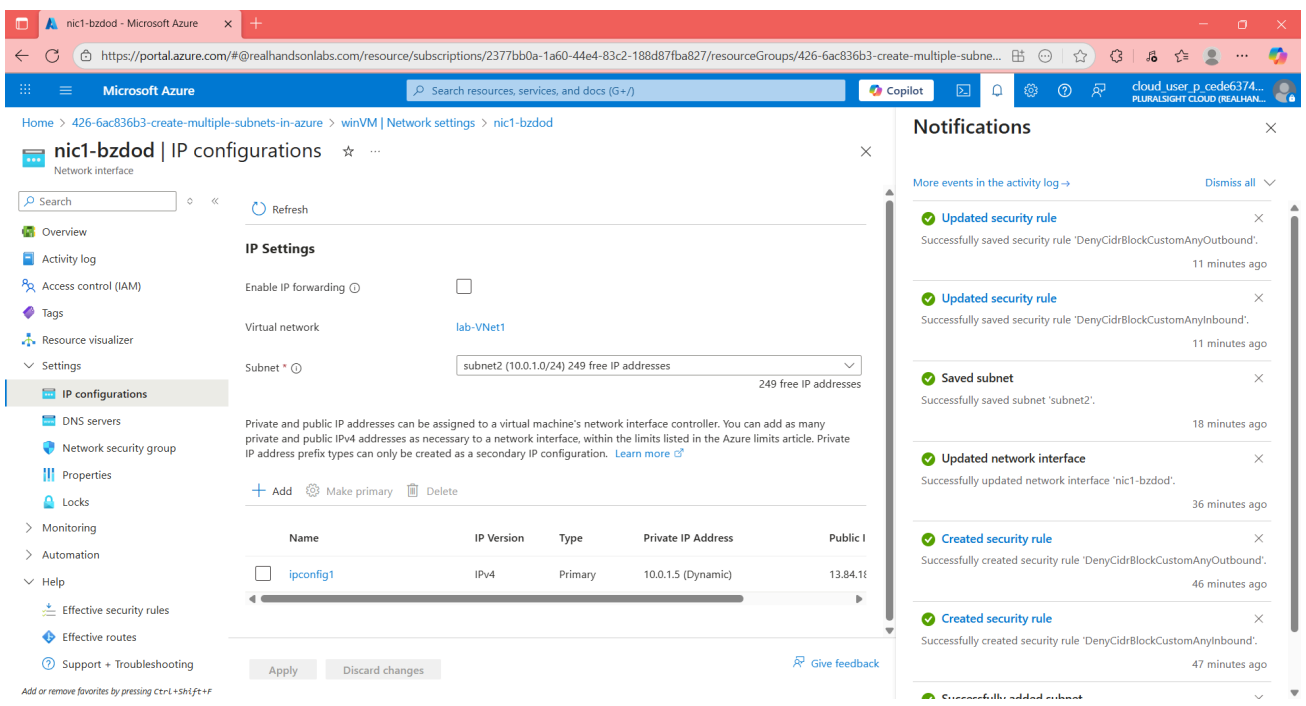
Under **IP configurations**, I updated the subnet to **subnet2** and clicked **Apply**, which restarted the VM, then returned to the resource group.

I navigated to **winVM** and used the **Connect** option to download the RDP file, then connected locally using an RDP client. After allowing network discoverability, I opened a command prompt, ran `ping 10.0.1.4`, and confirmed it failed, verifying that the lab configuration was successful

Completion of the lab was intended to be validated by testing connectivity between the VMs, but the expected results were not observed. The Command Prompt was able to receive a ping response, which could have been caused by either the VMs still residing on the same subnet or the NSG rules not effectively blocking ICMP traffic between them. In this case, the latter was not the issue, indicating a possible misconfiguration or delay in applying the subnet changes or NSG rules.



To troubleshoot, I verified the subnet values, confirmed the VMs were assigned to the correct subnets, checked the NSG configuration, deallocated and restarted the VM, and reviewed the NIC settings. These steps restored expected functionality, but the main objective of the lab—blocking communication between the VMs using a Network Security Group—was not fully achieved before the lab session timed out. Which looked something like this.



💡Key Takeaways

- Creating separate subnets is essential to isolate VMs and control communication within a virtual network.
- Network Security Groups (NSGs) allow granular inbound and outbound traffic control between subnets.
- Proper configuration of NSG rules, including source, destination, ports, and priority, is critical to achieving the intended network restrictions.
- VM network interface cards (NICs) must be updated to the correct subnet for NSG rules to take effect.
- Connectivity testing, such as using `ping`, is necessary to validate network isolation and troubleshoot any configuration issues.
- Lab timing or misapplied settings can prevent intended outcomes, highlighting the importance of verification and troubleshooting steps.
- Unexpected connectivity can occur even after configuring subnets and NSG rules, highlighting the need to verify changes and allow time for them to propagate.
- Troubleshooting steps such as checking subnet assignments, reviewing NSG rules, and restarting VMs are essential to diagnose and resolve network configuration issues.