Three Tier Architecture

In modern enterprise cloud deployments, the three-tier architecture is a foundational design pattern used to ensure scalability, fault isolation, and security across distributed systems. This architecture decouples an application into three distinct layers: Web, Application, and Database, each hosted in its own isolated subnet within a Virtual Network (VNet).

This separation of concerns allows for easier maintenance, enhanced security control, and independent scaling of each tier based on specific workload requirements

1. Web Tier (Presentation/Client Layer)

The Web tier functions as the front door to the application. It is the only layer exposed to the public internet, typically via a load balancer or application gateway. Its core responsibilities include:

- Serving static assets such as HTML, CSS, JavaScript, and images.
- Handling user-facing content and API entry points.
- Forwarding requests to the Application tier for business logic processing.
- Communicating with the internet for updates, integrations, or user requests.

To support external access, this tier requires a public IP and allows inbound HTTP/HTTPS traffic from the internet. Minimal outbound access is also required for package updates and logging integrations.

2. Application Tier (Logic Layer)

The Application tier hosts the core business logic and services that process inputs from the frontend and interact with the backend database. It is completely private, not exposed to the internet, and is only accessible by the Web tier within the VNet.

This layer may include web services, APIs, microservices, or containerized workloads. Key characteristics:

- Accepts traffic only from the Web tier subnet.
- Initiates connections to the Database tier for data queries or transactions.
- Requires fine-grained access control to ensure secure internal communication.
- Internet access is explicitly blocked or tightly restricted for outbound requests.

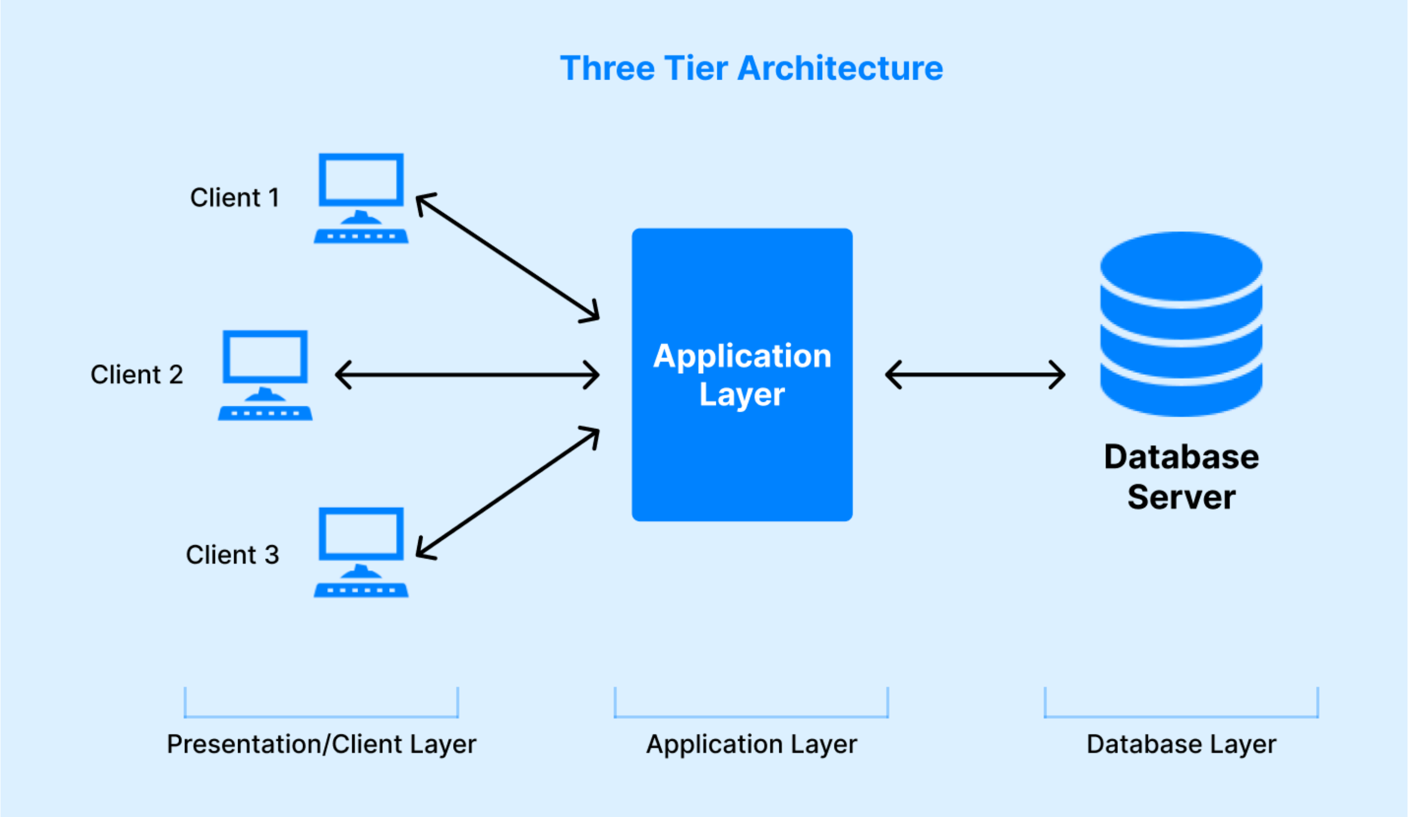
The isolation of this tier mitigates the risk of direct exposure and lateral movement from public-facing components.

3. Database Tier (Data Layer)

The Database tier contains persistent data storage systems such as SQL databases, NoSQL engines, or data warehouses. It is the most sensitive part of the architecture and requires the strictest access controls.

- Only the Application tier is allowed to access this layer.
- No public IPs or internet access are permitted.
- Firewall and NSG rules are applied to deny any traffic from Web tier or unauthorized sources.
- May leverage Azure Private Link or service endpoints for secure connectivity.

This guarantees that all data transactions originate from trusted internal services, reducing the attack surface significantly.



Security Posture and Traffic Control

To enforce network-level segmentation and traffic flow, each subnet is secured using **Azure Network Security Groups (NSGs)**. These NSGs are configured with specific inbound and outbound rules to reflect the intended communication flow:

- **Web Tier**:
  - Inbound: Allowed from Internet (HTTP, HTTPS, SSH/RDP for admin access)
  - Outbound: Allowed to Application tier and Internet
- **Application Tier**:
  - Inbound: Allowed only from Web tier
  - Outbound: Allowed only to Database tier
  - Internet access is restricted
- **Database Tier**:
  - Inbound: Allowed only from Application tier
  - Outbound: Blocked or strictly controlled

Additional safeguards include:

- **Role-Based Access Control (RBAC)** to limit who can manage or access resources.
- **No Public IPs** on Application and Database VMs.
- **Jumpbox or Azure Bastion** used to manage private-tier virtual machines securely.

By combining subnet isolation, directional NSG rules, and role-based permissions, this architecture achieves a hardened security boundary that aligns with zero-trust principles and enterprise compliance standards.

Implementation Steps on Azure

Objective : Create three subnets : 1. Web tier 2. App tier 3. DB tier DB Tier should not access any tier(Web & App tier) App tier should access the DB tier and Web tier as well, Web tier should access only App tier. Only Web tier is allowed to connect to the internet.Deploy two VM's in each tier(One VM should be Linux & another should be Windows). Configure Apache Server on Linux VM's And IIS Server on Windows.

1. Create the Resource Group and Virtual Network (VNet)

a. Create Resource Group (if not already created)

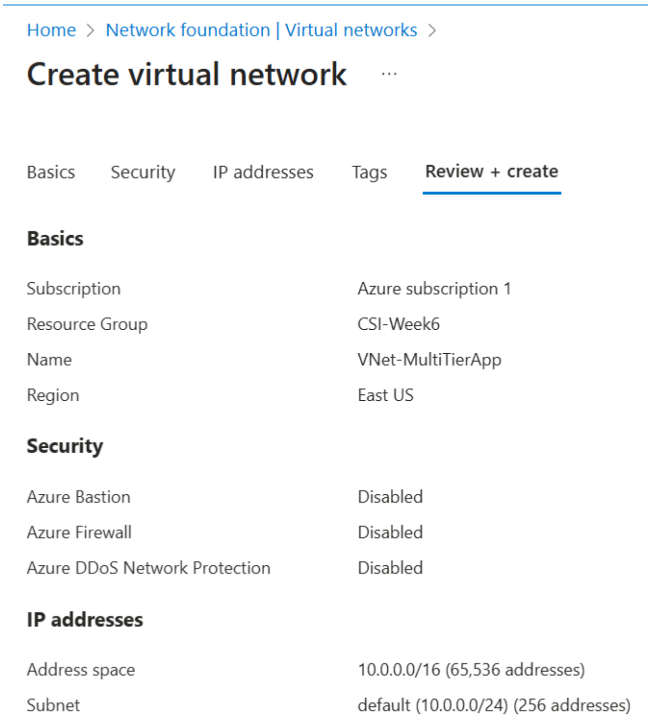
b. Create Virtual Network

- Name: `VNet-MultiTierApp`
- Address Space: `10.0.0.0/16`

c. Create Subnets within VNet

- **Web Tier Subnet**
  - Name: `Subnet-Web`
  - Address range: `10.0.1.0/24`
- **Application Tier Subnet**
  - Name: `Subnet-App`
  - Address range: `10.0.2.0/24`
- **Database Tier Subnet**
  - Name: `Subnet-DB`
  - Address range: `10.0.3.0/24`

Each subnet must be logically isolated but reside in the same VNet for routing and integration simplicity.



Create virtual network

Basics

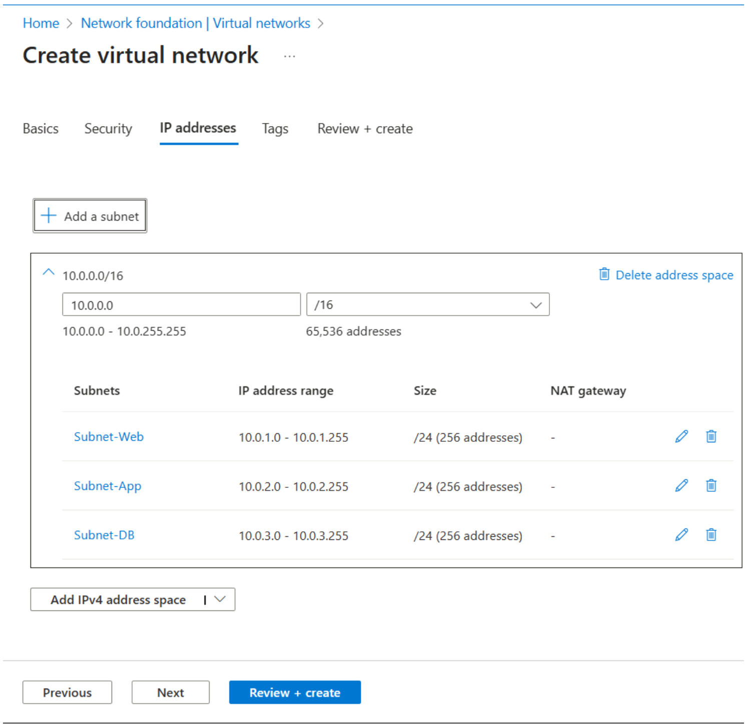
Subscription	Azure subscription 1
Resource Group	CSI-Week6
Name	VNet-MultiTierApp
Region	East US

Security

Azure Bastion	Disabled
Azure Firewall	Disabled
Azure DDoS Network Protection	Disabled

IP addresses

Address space	10.0.0.0/16 (65,536 addresses)
Subnet	default (10.0.0.0/24) (256 addresses)



Create virtual network

10.0.0.0/16

10.0.0.0 /16

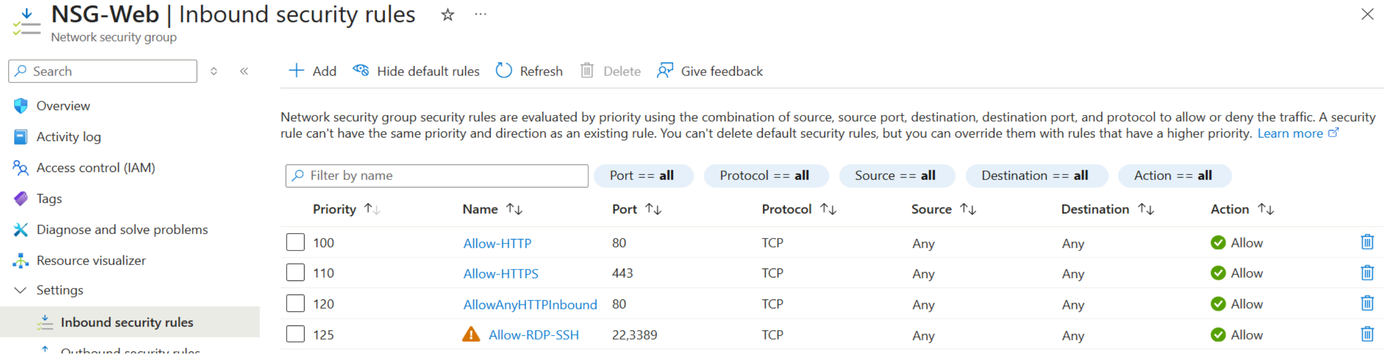
10.0.0.0 - 10.0.255.255 65,536 addresses

Subnets	IP address range	Size	NAT gateway
Subnet-Web	10.0.1.0 - 10.0.1.255	/24 (256 addresses)	-
Subnet-App	10.0.2.0 - 10.0.2.255	/24 (256 addresses)	-
Subnet-DB	10.0.3.0 - 10.0.3.255	/24 (256 addresses)	-

2. Create and Configure Network Security Groups (NSGs)

a. NSG for Web Tier (`NSG-Web`)

- **Inbound Rules**
  - Allow HTTP (TCP/80) from Internet
  - Allow HTTPS (TCP/443) from Internet
  - Allow RDP (TCP/3389) or SSH (TCP/22) from specific admin IPs only
- **Outbound Rules**
  - Allow all traffic to App Tier (`10.0.2.0/24`)
  - Allow outbound to Internet (Azure default, or explicitly allow `0.0.0.0/0`)

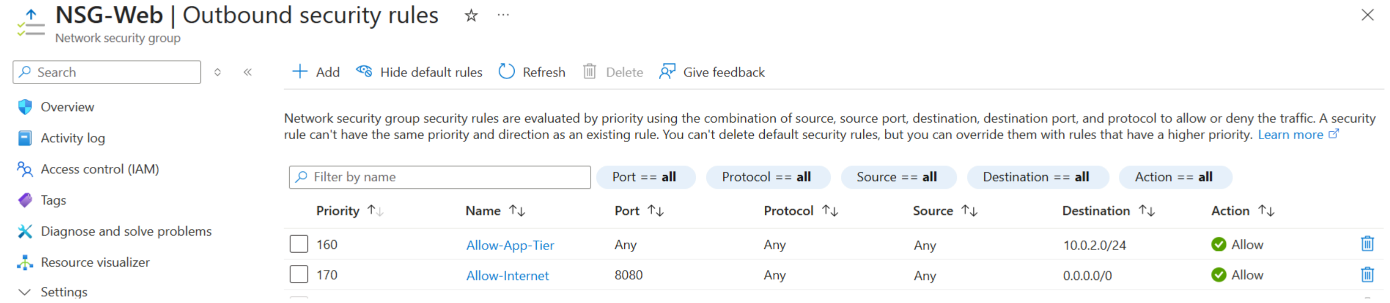


NSG-Web | Inbound security rules

Network security group

Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

	Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
□	100	Allow-HTTP	80	TCP	Any	Any	✓ Allow
□	110	Allow-HTTPS	443	TCP	Any	Any	✓ Allow
□	120	AllowAnyHTTPInbound	80	TCP	Any	Any	✓ Allow
□	125	⚠ Allow-RDP-SSH	22,3389	TCP	Any	Any	✓ Allow



NSG-Web | Outbound security rules

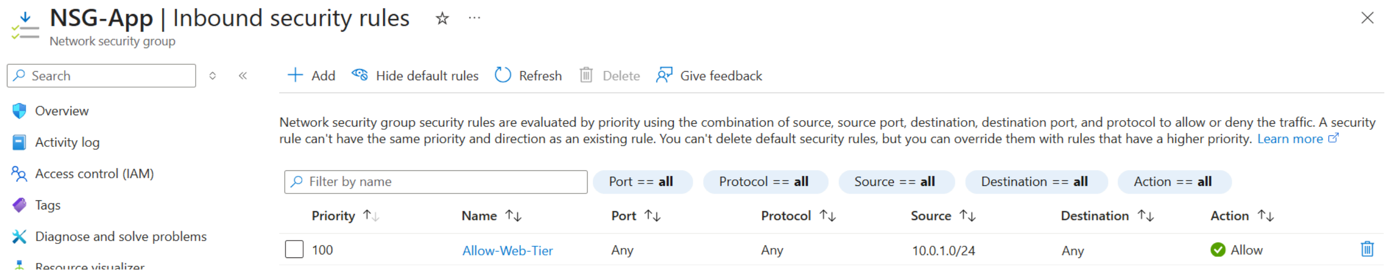
Network security group

Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

	Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
□	160	Allow-App-Tier	Any	Any	Any	10.0.2.0/24	✓ Allow
□	170	Allow-Internet	8080	Any	Any	0.0.0.0/0	✓ Allow

b. NSG for App Tier (`NSG-App`)

- **Inbound Rules**
  - Allow traffic from Web Tier subnet (`10.0.1.0/24`)
- **Outbound Rules**
  - Allow traffic to Database Tier (`10.0.3.0/24`)
  - Deny outbound to Internet (add high-priority Deny rule to `0.0.0.0/0`)

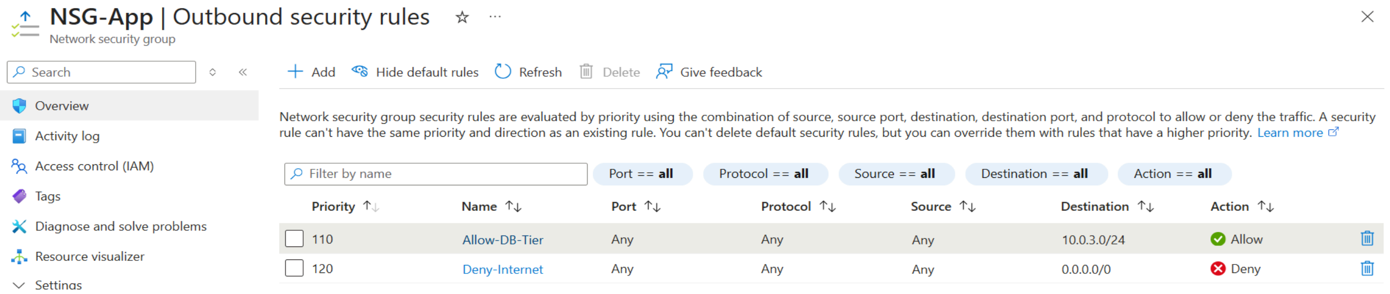


NSG-App | Inbound security rules

Network security group

Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

	Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
□	100	Allow-Web-Tier	Any	Any	10.0.1.0/24	Any	✓ Allow



NSG-App | Outbound security rules

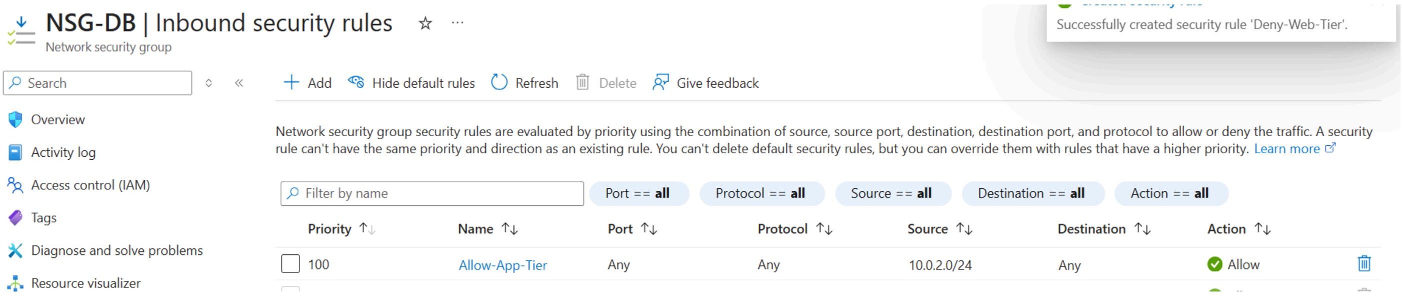
Network security group

Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

	Priority ↑↓	Name ↑↓	Port ↑↓	Protocol ↑↓	Source ↑↓	Destination ↑↓	Action ↑↓
□	110	Allow-DB-Tier	Any	Any	Any	10.0.3.0/24	✓ Allow
□	120	Deny-Internet	Any	Any	Any	0.0.0.0/0	✗ Deny

c. NSG for DB Tier (`NSG-DB`)

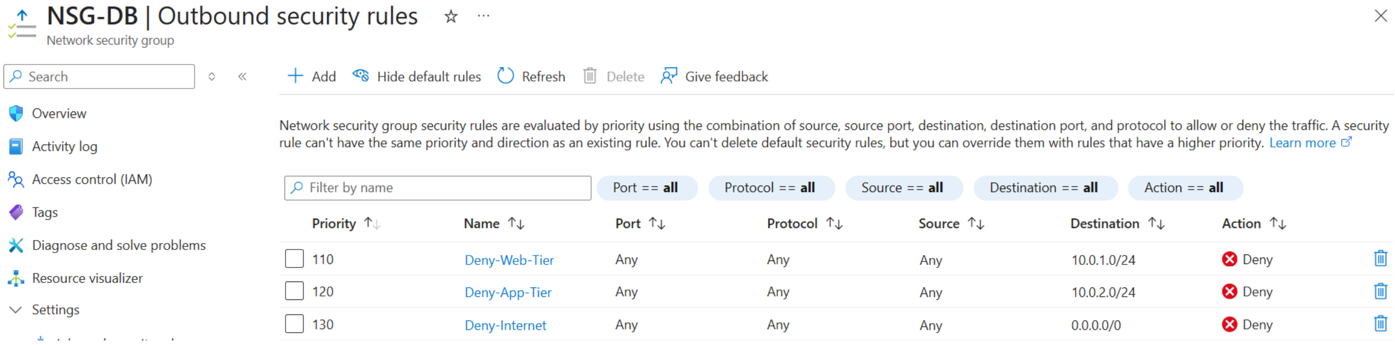
- **Inbound Rules**
  - Allow traffic from App Tier subnet (`10.0.2.0/24`)
- **Outbound Rules**
  - Deny all outbound traffic to any subnet or Internet (deny to `10.0.1.0/24`, `10.0.2.0/24`, and `0.0.0.0/0`)



Successfully created security rule 'Deny-Web-Tier'.

Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

Priority	Name	Port	Protocol	Source	Destination	Action
□ 100	Allow-App-Tier	Any	Any	10.0.2.0/24	Any	✓ Allow

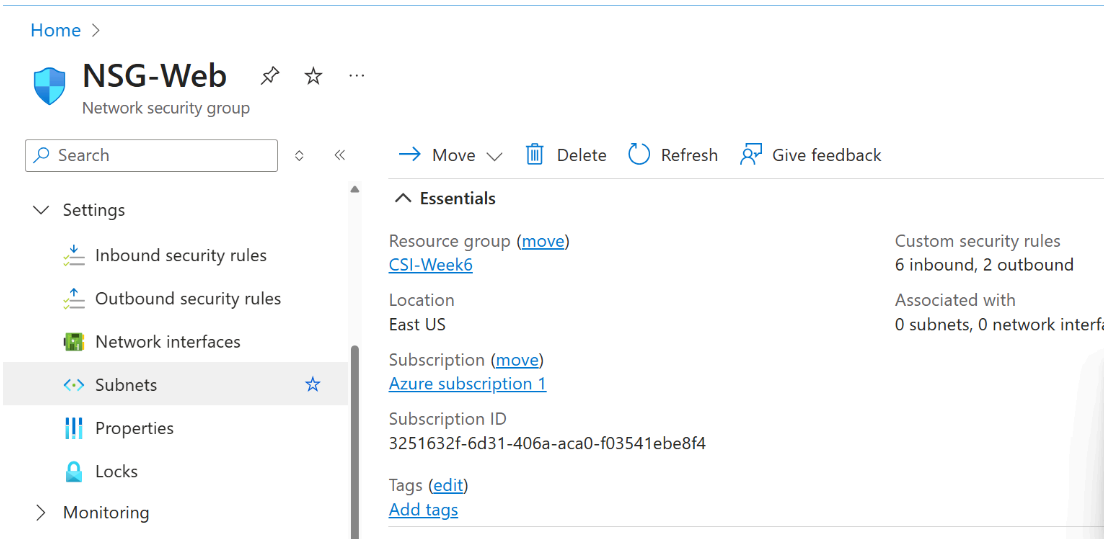


Network security group security rules are evaluated by priority using the combination of source, source port, destination, destination port, and protocol to allow or deny the traffic. A security rule can't have the same priority and direction as an existing rule. You can't delete default security rules, but you can override them with rules that have a higher priority. Learn more

Priority	Name	Port	Protocol	Source	Destination	Action
□ 110	Deny-Web-Tier	Any	Any	Any	10.0.1.0/24	☒ Deny
□ 120	Deny-App-Tier	Any	Any	Any	10.0.2.0/24	☒ Deny
□ 130	Deny-Internet	Any	Any	Any	0.0.0.0/0	☒ Deny

Associate NSGs

- Bind `NSG-Web` to `Subnet-Web`
- Bind `NSG-App` to `Subnet-App`
- Bind `NSG-DB` to `Subnet-DB`



Resource group (move)
CSI-Week6

Location
East US

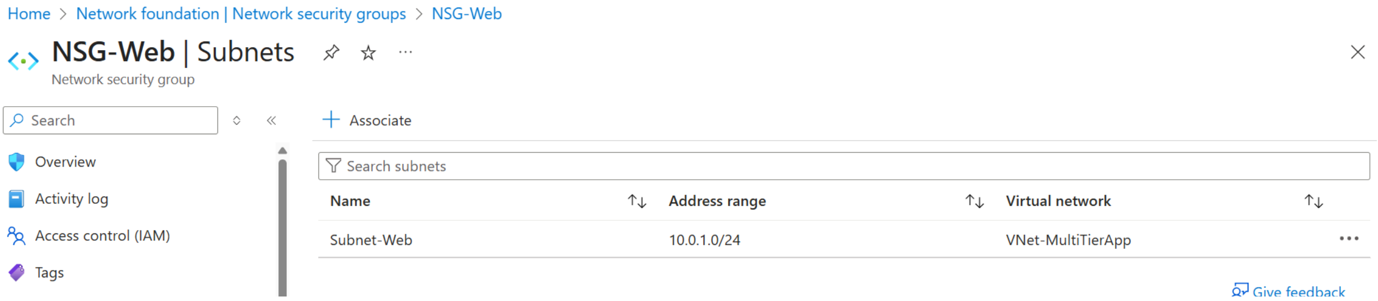
Subscription (move)
Azure subscription 1

Subscription ID
3251632f-6d31-406a-aca0-f03541ebe8f4

Tags (edit)
Add tags

Custom security rules
6 inbound, 2 outbound

Associated with
0 subnets, 0 network interf



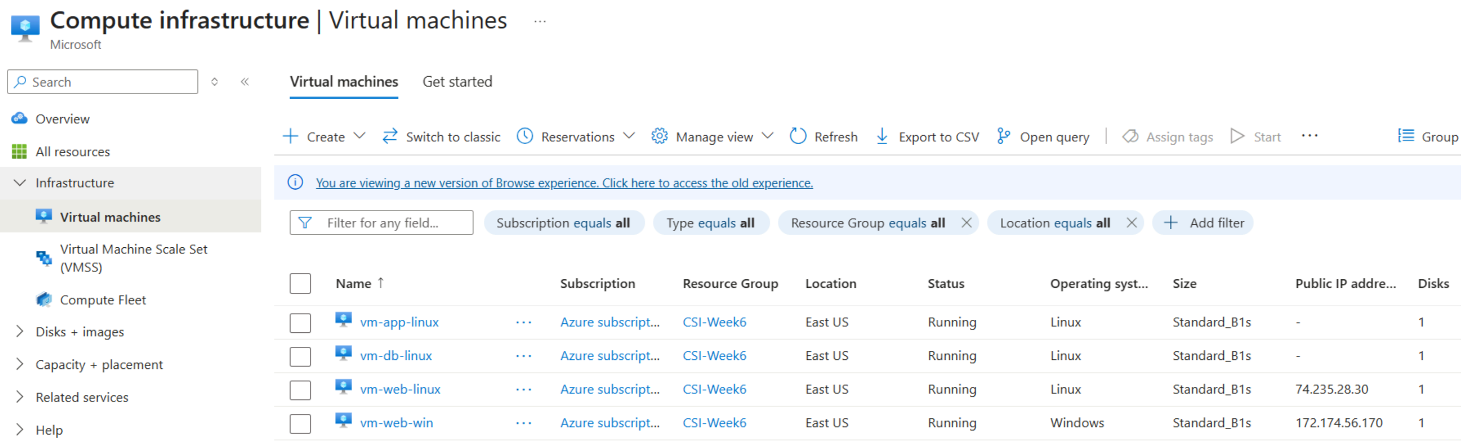
Name	Address range	Virtual network
Subnet-Web	10.0.1.0/24	VNet-MultiTierApp

3. Deploy Virtual Machines (Two Per Tier)

For each tier, deploy one Windows VM and one Linux VM to support cross-platform application deployment and compatibility testing.

VM Naming Convention:

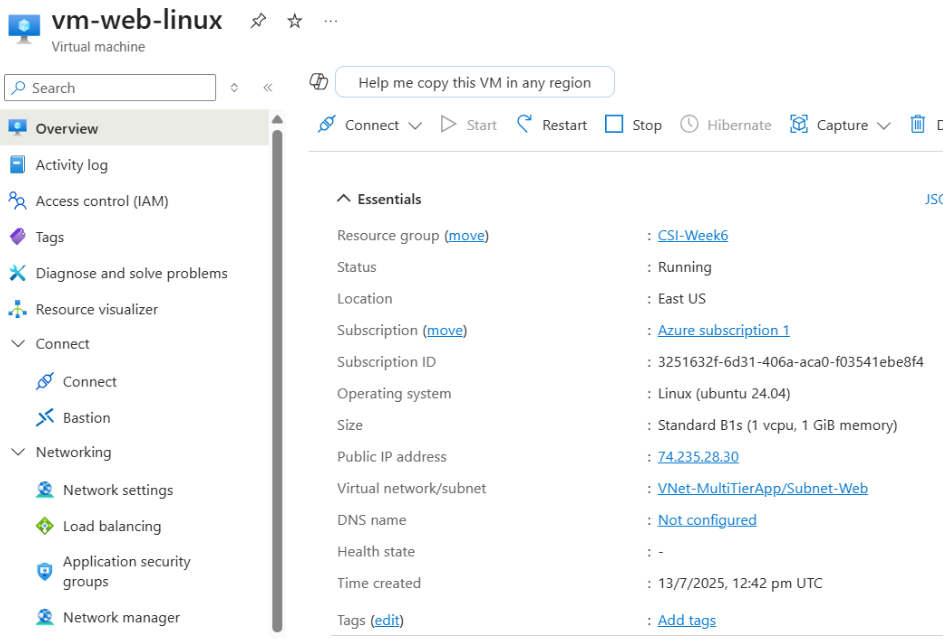
- **Web Tier**: `vm-web-linux`, `vm-web-win`
- **App Tier**: `vm-app-linux`, `vm-app-win`
- **DB Tier**: `vm-db-linux`, `vm-db-win`



Name ↑	Subscription	Resource Group	Location	Status	Operating syst...	Size	Public IP addre...	Disks
vm-app-linux	Azure subscript...	CSI-Week6	East US	Running	Linux	Standard_B1s	-	1
vm-db-linux	Azure subscript...	CSI-Week6	East US	Running	Linux	Standard_B1s	-	1
vm-web-linux	Azure subscript...	CSI-Week6	East US	Running	Linux	Standard_B1s	74.235.28.30	1
vm-web-win	Azure subscript...	CSI-Week6	East US	Running	Windows	Standard_B1s	172.174.56.170	1

VM Configuration:

- **Size**: `Standard_B2s` (2 vCPU, 4 GB RAM)
- **OS Images**:
  - Linux: Ubuntu Server 22.04 LTS
  - Windows: Windows Server 2022 Datacenter
- **Network Settings**:
  - Deploy VMs into the appropriate subnets
  - Only Web VMs should have Public IP addresses
  - App and DB VMs should be private only



Resource group (move)	: CSI-Week6
Status	: Running
Location	: East US
Subscription (move)	: Azure subscription 1
Subscription ID	: 3251632f-6d31-406a-aca0-f03541ebe8f4
Operating system	: Linux (ubuntu 24.04)
Size	: Standard B1s (1 vcpu, 1 GiB memory)
Public IP address	: 74.235.28.30
Virtual network/subnet	: VNet-MultiTierApp/Subnet-Web
DNS name	: Not configured
Health state	: -
Time created	: 13/7/2025, 12:42 pm UTC
Tags (edit)	: Add tags

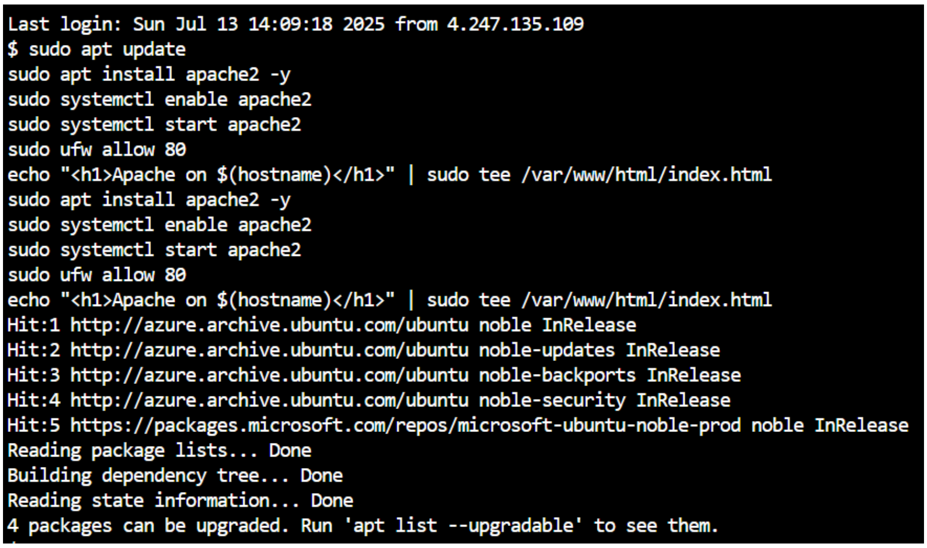
4. Configure Apache and IIS Servers

On Linux VMs:

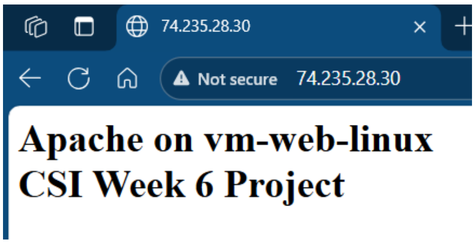
Use the following script after connecting via SSH or Azure Cloud Shell:

```
sudo apt update
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2
sudo ufw allow 80
echo "<h1>Apache on $(hostname)</h1>" | sudo tee /var/www/html/index.html
```

Verify Apache is running by accessing the VM's private IP (or public IP for web tier) on port 80.



```
Last login: Sun Jul 13 14:09:18 2025 from 4.247.135.109
$ sudo apt update
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2
sudo ufw allow 80
echo "<h1>Apache on $(hostname)</h1>" | sudo tee /var/www/html/index.html
sudo apt install apache2 -y
sudo systemctl enable apache2
sudo systemctl start apache2
sudo ufw allow 80
echo "<h1>Apache on $(hostname)</h1>" | sudo tee /var/www/html/index.html
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease
Hit:5 https://packages.microsoft.com/repos/microsoft-ubuntu-noble-prod noble InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
4 packages can be upgraded. Run 'apt list --upgradable' to see them.
```



Apache on vm-web-linux
CSI Week 6 Project

On Windows VMs:

Run the following commands in PowerShell as Administrator:

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
Start-Service W3SVC
```

Optionally create a test web page:

```
echo "<h1>IIS on $env:COMPUTERNAME</h1>" > C:\inetpub\wwwroot\index.html
```

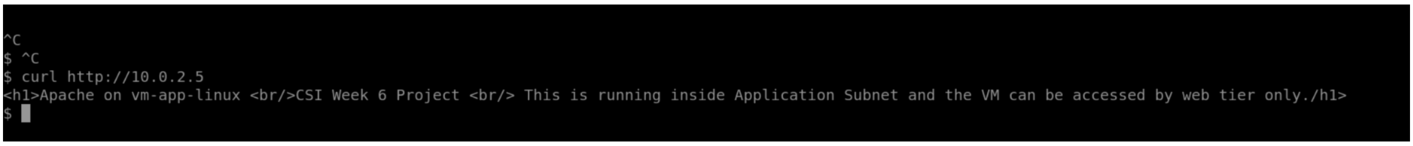
Access the VM via browser using IP address on port 80 to verify IIS is serving the page.

5. Validate the Network Communication

Perform the following validation tests:

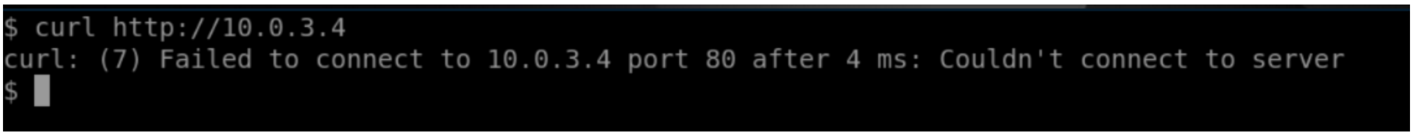
Web Tier:

- Should **access App tier** VMs via internal IP (Linux: use `curl`, Windows: use browser or PowerShell)



```
^C
$ ^C
$ curl http://10.0.2.5
<h1>Apache on vm-app-linux <br/>CSI Week 6 Project <br/> This is running inside Application Subnet and the VM can be accessed by web tier only./h1>
$ █
```

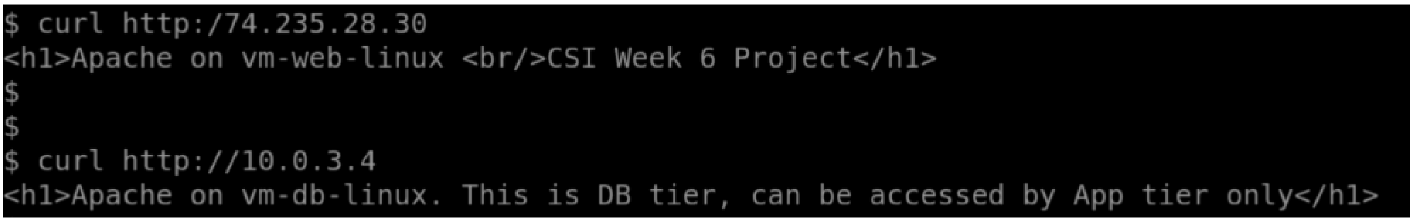
- Should **not access DB tier** (blocked by NSG)



```
$ curl http://10.0.3.4
curl: (7) Failed to connect to 10.0.3.4 port 80 after 4 ms: Couldn't connect to server
$ █
```

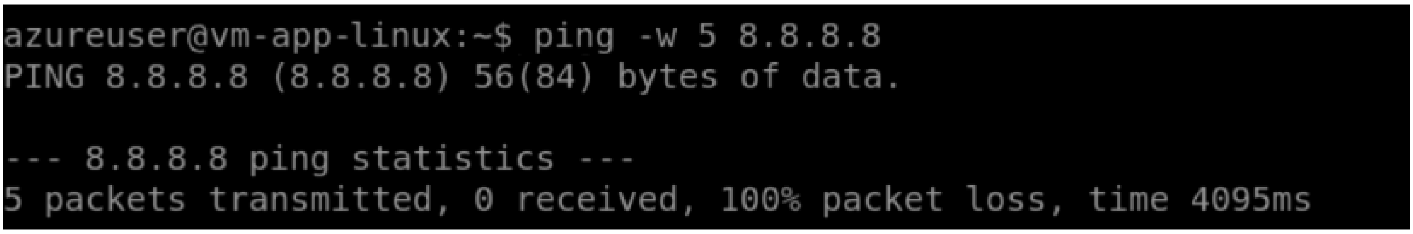
App Tier:

- Should **access both Web tier and DB tier**



```
$ curl http:/74.235.28.30
<h1>Apache on vm-web-linux <br/>CSI Week 6 Project</h1>
$
$
$ curl http://10.0.3.4
<h1>Apache on vm-db-linux. This is DB tier, can be accessed by App tier only</h1>
```

- Should **no access Internet** (validate by running `ping 8.8.8.8` or `curl google.com` on Linux, which should fail)



```
azureuser@vm-app-linux:~$ ping -w 5 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4095ms
```

DB Tier:

- Should **only respond to traffic from App tier**
- Should **not initiate any outbound traffic**, not even to App or Web tier

Use Azure Network Watcher tools like **Connection Troubleshoot** or **NSG Flow Logs** for deeper inspection.