CELEBAL TECHNOLOGY INTERNSHIP (CSI)

Name: Harsh Tongariya

College: Arya College of Engineering Information Technology

Domain: Cloud Infra & Security

Student ID: CT_CSI_CI_1160

Research & Implementation Document

Title: Implementation of Secure Three-Tier Architecture in Azure

---

1. Introduction

This document provides a comprehensive R&D guide to implement a secure three-tier architecture in Microsoft Azure, following the n-tier architectural style. The deployment consists of three logically separated subnets: Web Tier, Application (App) Tier, and Database (DB) Tier. Each subnet will host two Virtual Machines (VMs): one running Linux (Apache) and another running Windows (IIS) to demonstrate cross-platform deployment.

The architecture enforces strict network isolation policies to follow zero-trust principles:

- Web Tier → Internet (Allowed)
- Web Tier → App Tier (Allowed)
- App Tier → DB Tier (Allowed)
- DB Tier → Any Other Tier (Denied)

2. Goal Recap

Create:

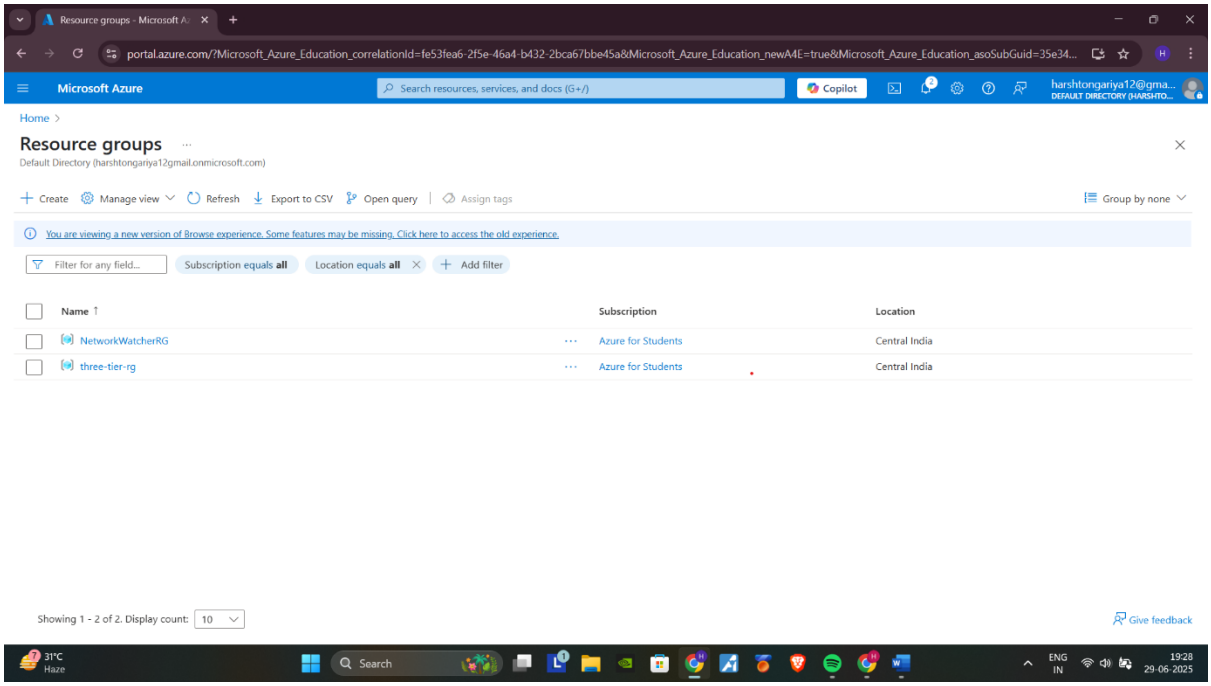
- 3 Subnets: Web, App, and DB
- Deploy 6 VMs: 2 per subnet (1 Linux + 1 Windows)
- Configure:
    - Apache on Linux VMs
    - IIS on Windows VMs
- Setup Network Security Groups (NSGs) to control access as follows:
    - Web Tier: Only talks to App Tier, and can go to Internet
    - App Tier: Talks to DB and Web Tiers

- DB Tier: No communication with Web or App Tier

3. Step-by-Step Deployment

Step 1: Create Resource Group

1. Go to Azure Portal
2. Search “Resource groups”
3. Click Create
4. Name: three-tier-rg
5. Region: Central India
6. Click Review + Create

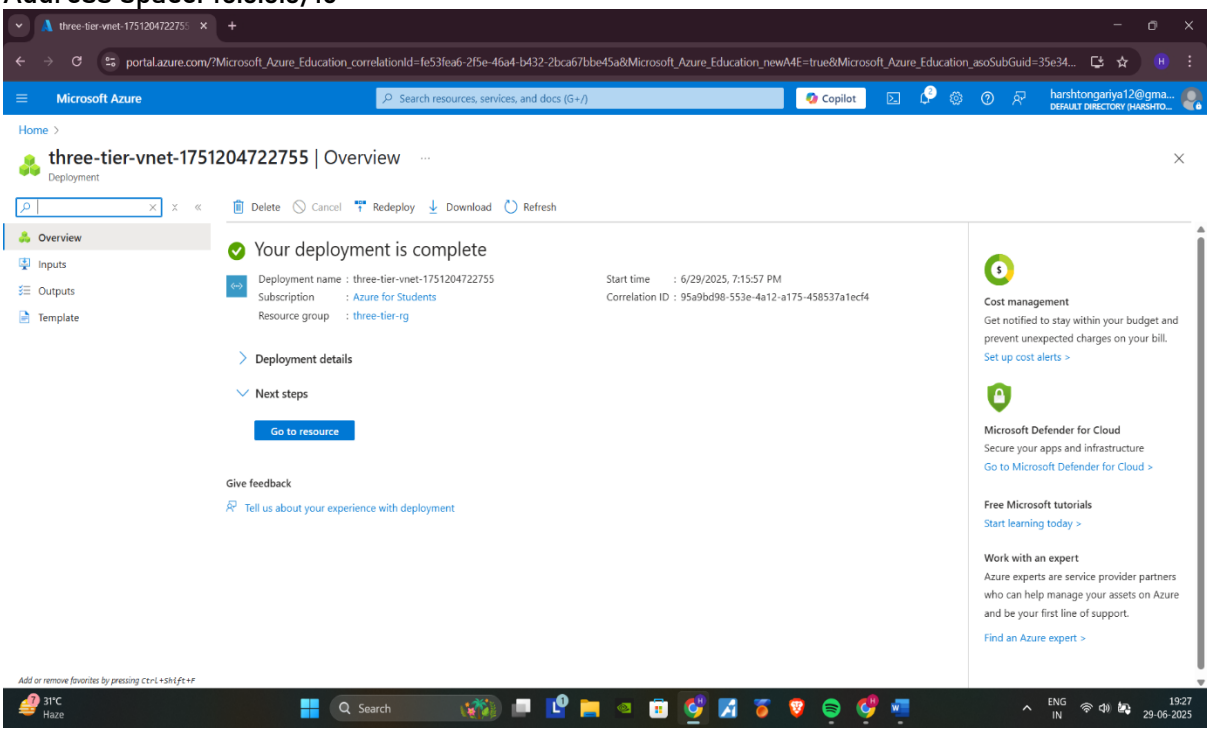


Step 2: Create Virtual Network + Subnets

1. Go to “Virtual Networks” > Create
2. Name: three-tier-vnet

3. Resource Group: three-tier-rg

4. Address space: 10.0.0.0/16



Now add 3 subnets:

- WebSubnet → 10.0.1.0/24
- AppSubnet → 10.0.2.0/24
- DBSubnet → 10.0.3.0/24

5. Click Review + Create

Step 3: Create Network Security Groups (NSGs)

You'll create 3 NSGs, one per subnet.

3.1 Create NSG for Web Tier:

1. Go to "Network Security Groups" > Create
2. Name: web-nsg
3. Add Inbound Rule:

- Allow HTTP (Port 80) from Internet
- Source: Any | Destination: Any | Port: 80 | Action: Allow

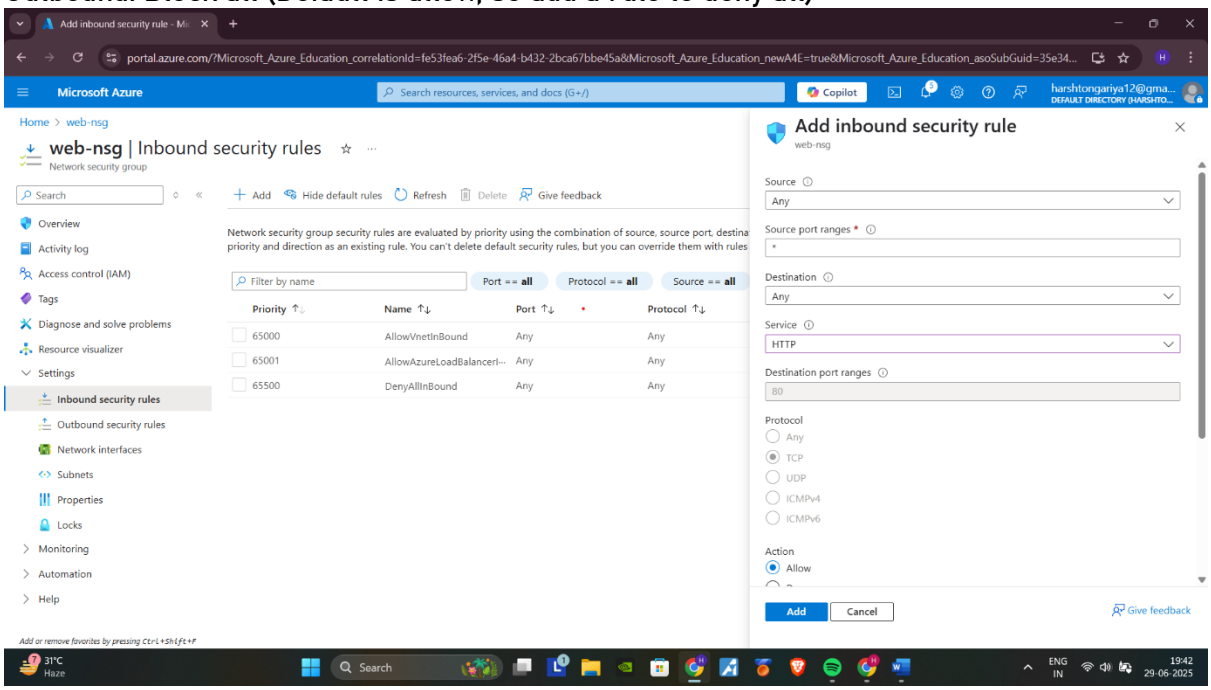
4. Outbound Rule:
    - Allow to AppSubnet 10.0.2.0/24 on required ports

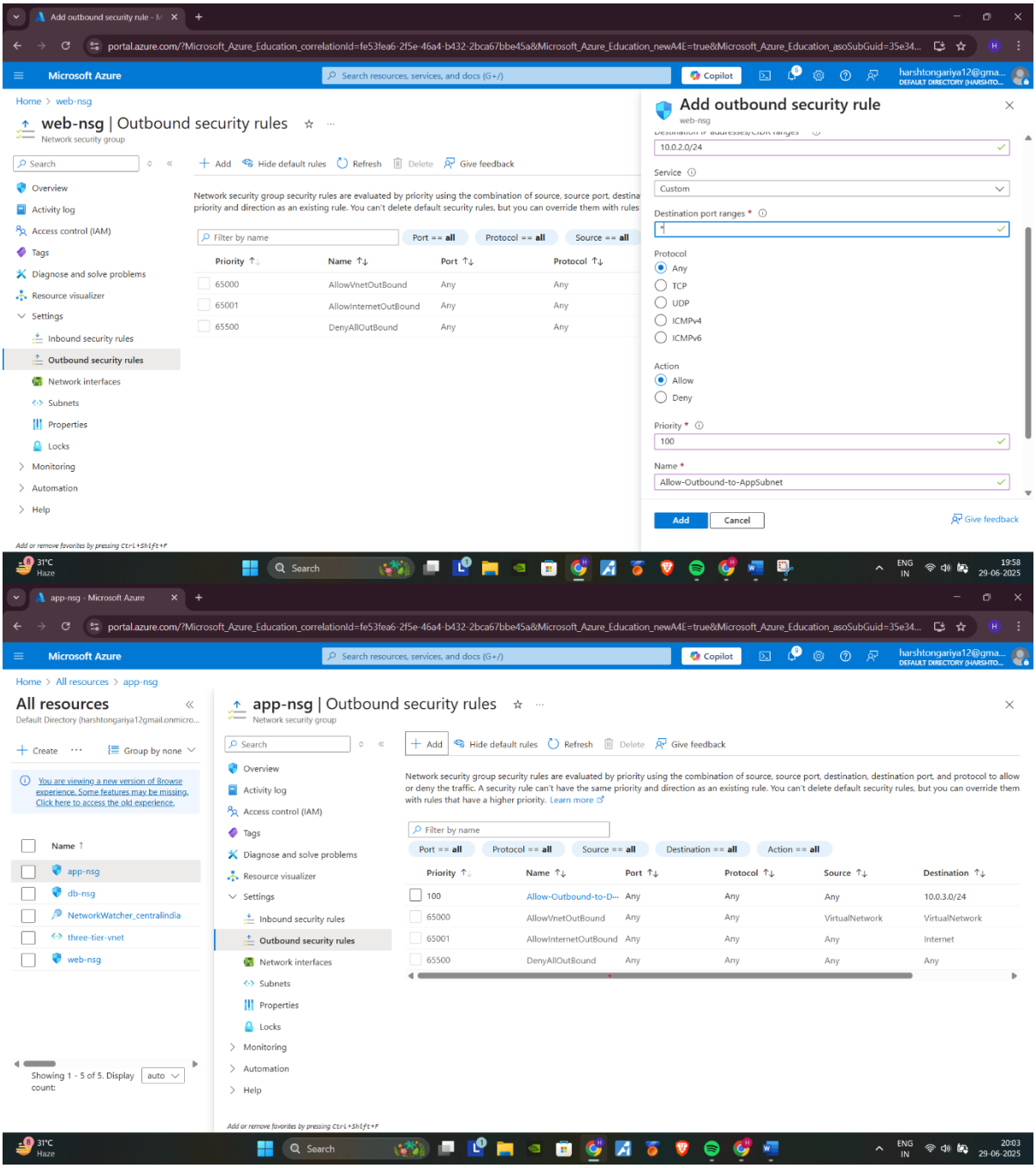
3.2 NSG for App Tier:

- Inbound: Allow from WebSubnet 10.0.1.0/24
- Outbound: Allow to DBSubnet 10.0.3.0/24

3.3 NSG for DB Tier:

- Inbound: Only allow from AppSubnet 10.0.2.0/24
- Outbound: Block all (Default is allow, so add a rule to deny all)

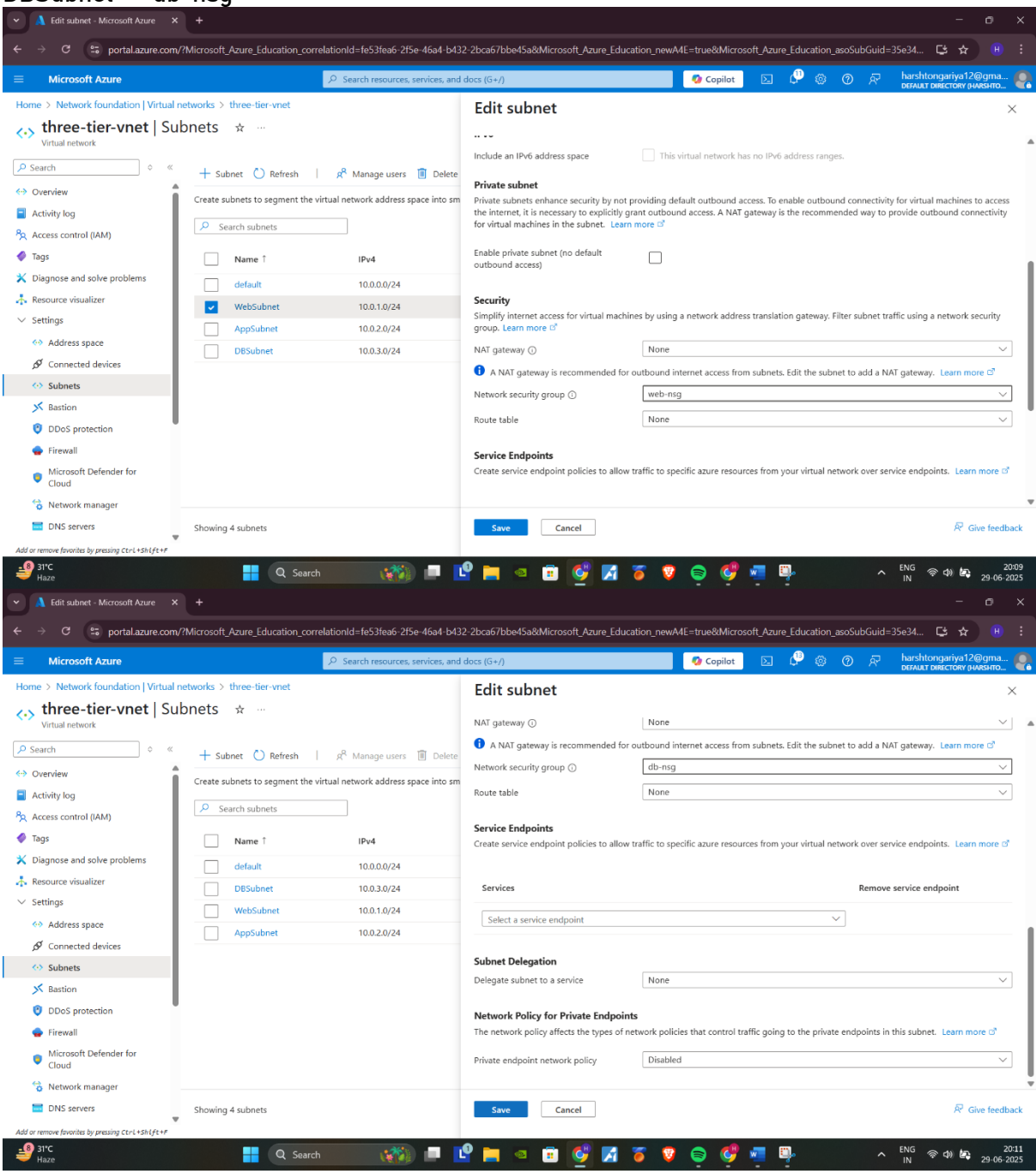




Step 4: Associate NSGs to Subnets

1. Go to your Virtual Network → Subnets
2. Click each subnet → Associate the right NSG:
   - WebSubnet → web-nsg
   - AppSubnet → app-nsg

- DBSubnet → db-nsg



Step 5: Create Virtual Machines (6 Total)

Repeat below steps 6 times (2 per subnet; 1 Linux + 1 Windows)

For Linux VM (Apache):

1. Go to Virtual Machines > Create VM

2. Name: web-linux, app-linux, db-linux
3. Image: Ubuntu LTS
4. Size: B1s (free tier)
5. Username: azureuser
6. Subnet: Select the right subnet (Web/App/DB)
7. Public IP:
    - Web Tier: Yes
    - App & DB Tiers: No
8. Custom data (scroll to Advanced tab → Custom data):

Paste this (auto installs Apache):

- bash

sudo apt update

sudo apt install apache2 -y

sudo systemctl start apache2

sudo systemctl enable apache2

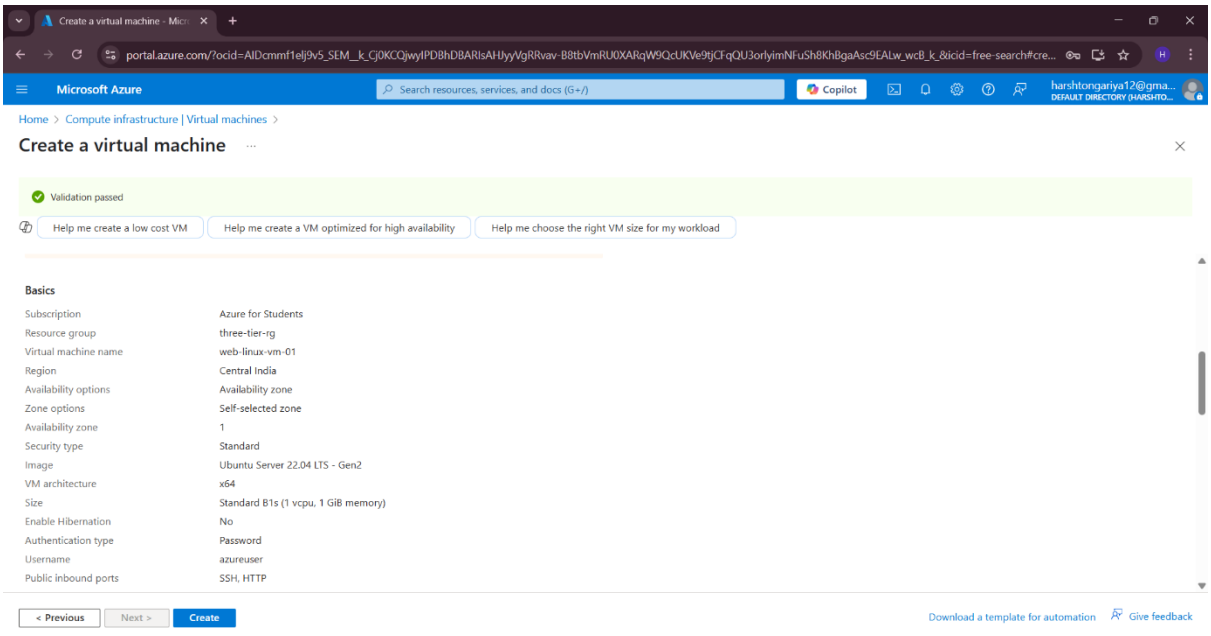
9. Click Review + Create

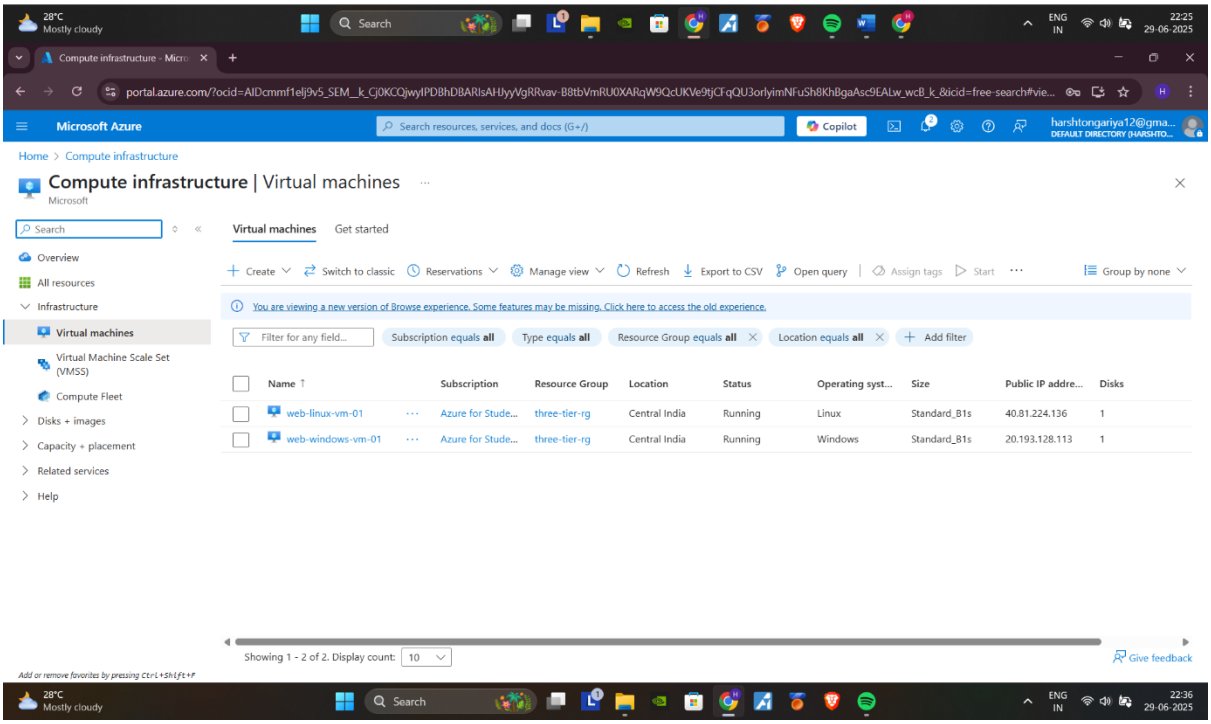
For Windows VM (IIS):

1. Same as above but choose Windows Server 2019/2022
2. Subnet: Same logic
3. Public IP:
    - Only Web Tier VMs should have Public IP
4. After login (via RDP):
    - Open PowerShell:

Install-WindowsFeature -name Web-Server -IncludeManagementTools

Start-Service W3SVC





Step 6: Test Network Connectivity

- Use ping/curl from:
    - Web → App: ☑

- App → DB: ☑
- DB → Any: ❌
- Web → DB: ❌
- Web → Internet: ☑
- App → Internet: ❌

4.Conclusion

The described architecture follows Microsoft's recommended n-tier design pattern, enabling:

- Security isolation between functional layers.
- Cross-platform application support (Linux + Windows).
- Scalable deployment of services.
- Tight access controls using NSGs and routing.

This structure is ideal for production-grade applications requiring modularity, maintainability, and robust security.