Azure Cloud Infrastructure Project (GUI + ARM Template Deployment)

An organisation needs a cloud-based infrastructure on Microsoft Azure to support high availability, secure communication, and resilient storage. This deployment uses the **Azure Portal (GUI)** with an **ARM template** to provision and verify all resources step-by-step.

Project Github : https://github.com/sagarpatilbox/upgrad-azure-project

Project Drive :

https://drive.google.com/drive/folders/1gQr4-znjALOK8J5izXt4GY0J68CTxOSK

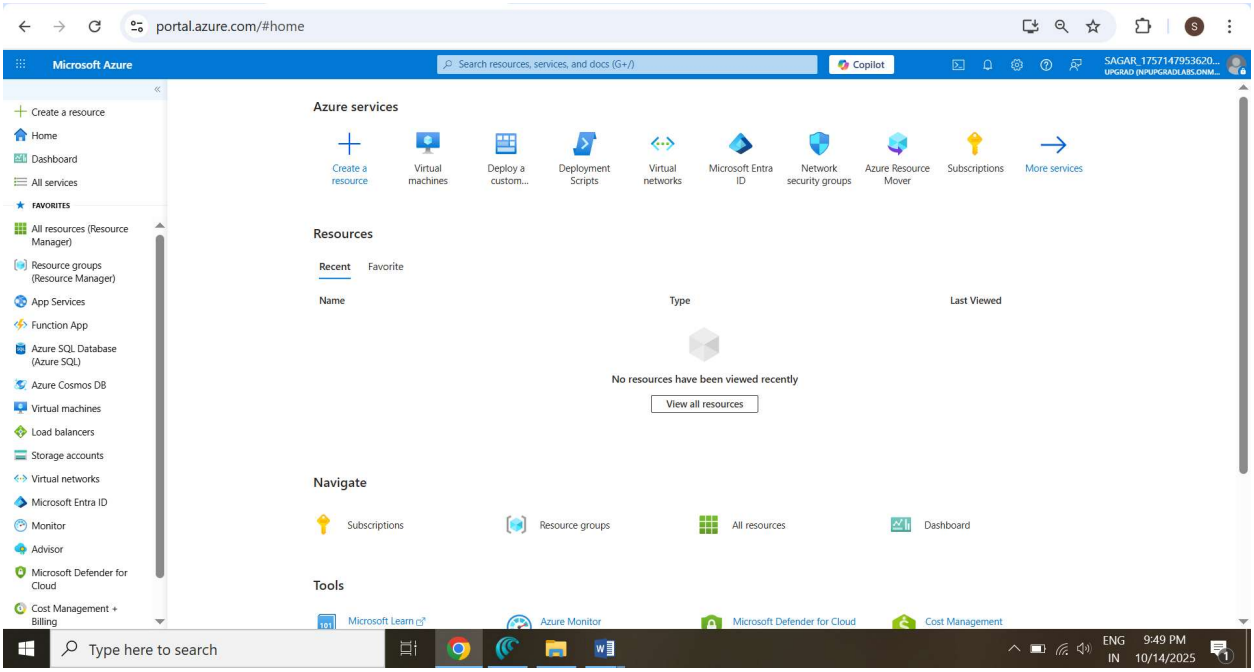
Task I — Setup Virtual Networks

Objective

Create two virtual networks for resource isolation and establish secure communication between them.

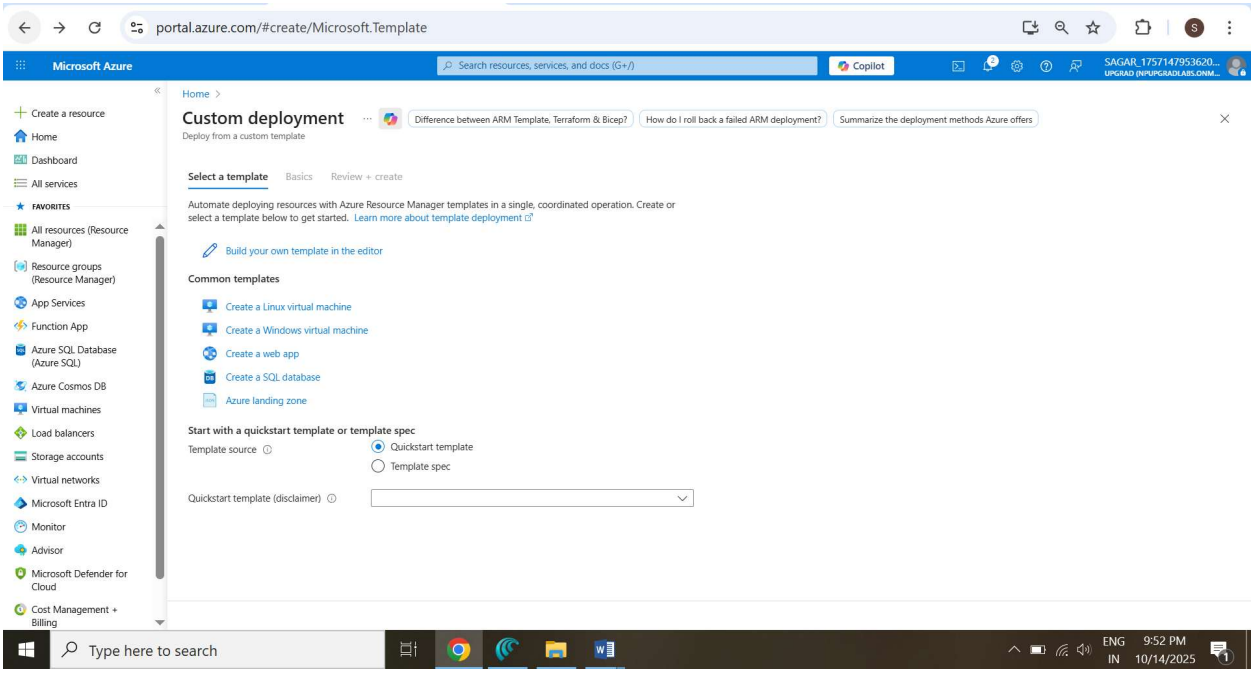
- Steps (in Azure Portal GUI)

1. **Login to Azure Portal**
    - Go to https://portal.azure.com.
    - Sign in using your Azure credentials.



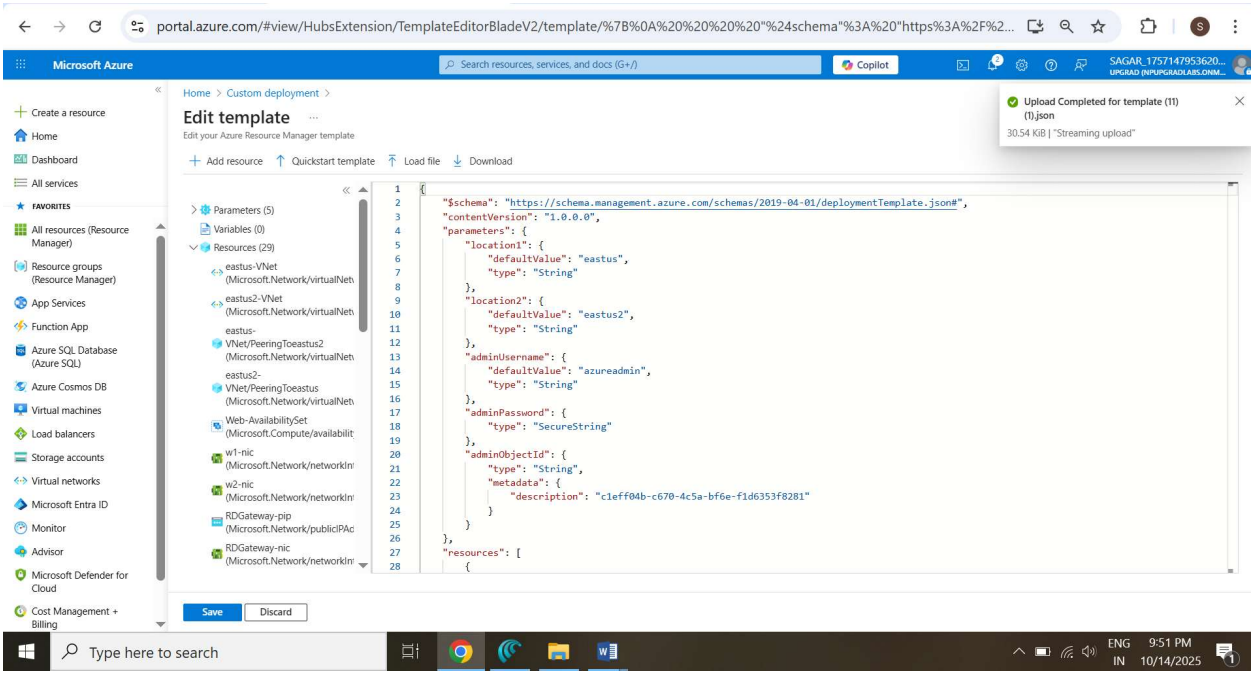
2. Start Custom Deployment

- Search → **“Deploy a custom template”** → click **Build your own template in the editor**.

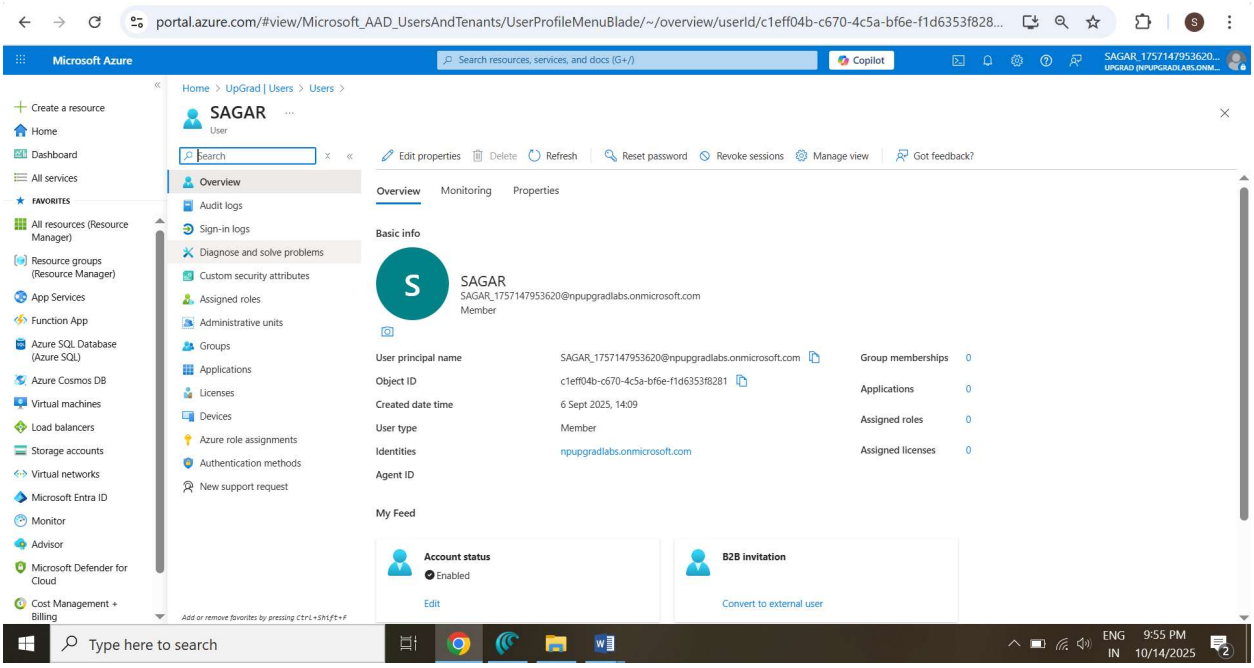


3. Upload ARM Template File

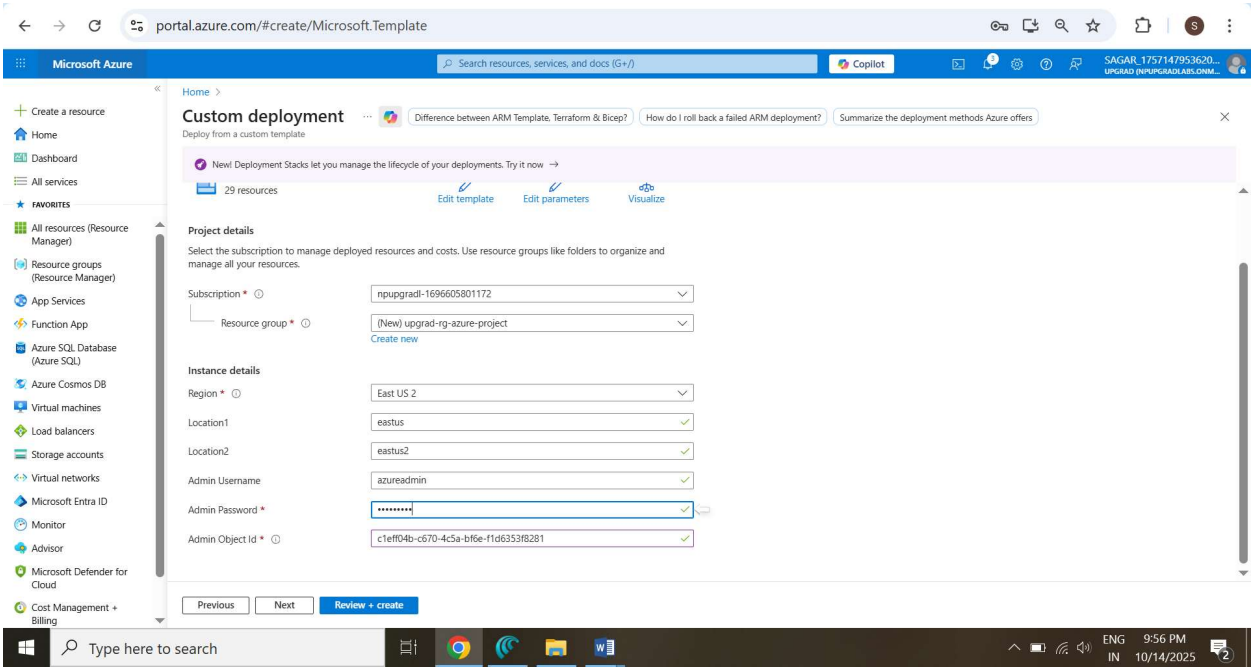
- Click **Load file** → **select** `Azure_Project_ARM.json`
- Click **Save**.



4. **Configure Deployment Parameters**
    - **Subscription** → choose active subscription.
    - **Resource Group** → “Create new” → `upgrad-rg-azure-project`.
    - Region → **West US**.
    - **Fill parameters** (adminUsername, adminPassword, adminObjectId).
    - Copy Object ID From Below :

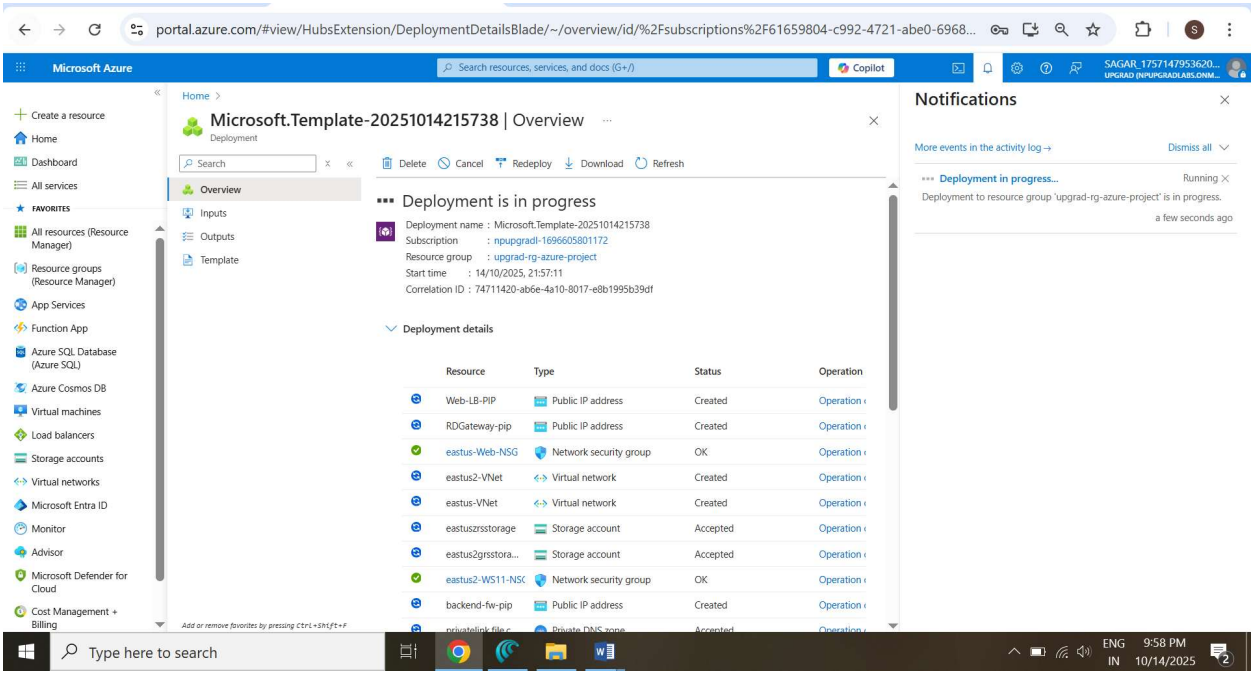


- Click **Review + Create** → **Create**.



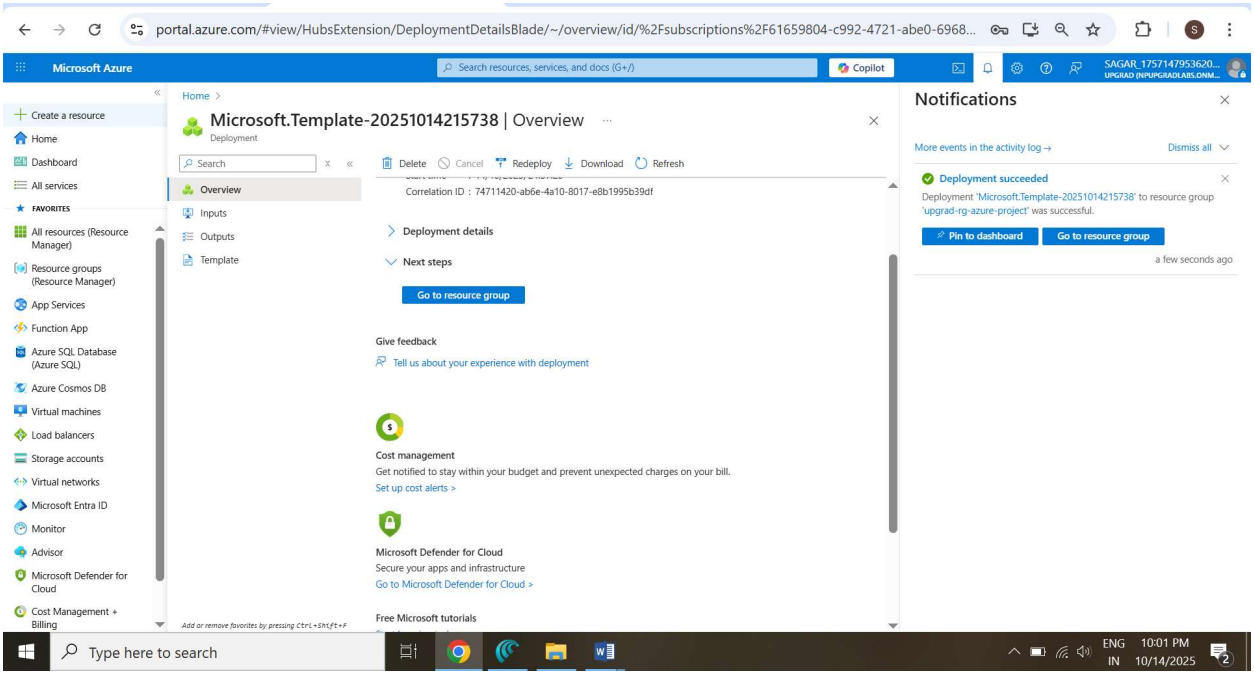
5. Monitor Deployment Progress

- Portal → **Notifications** → view progress.



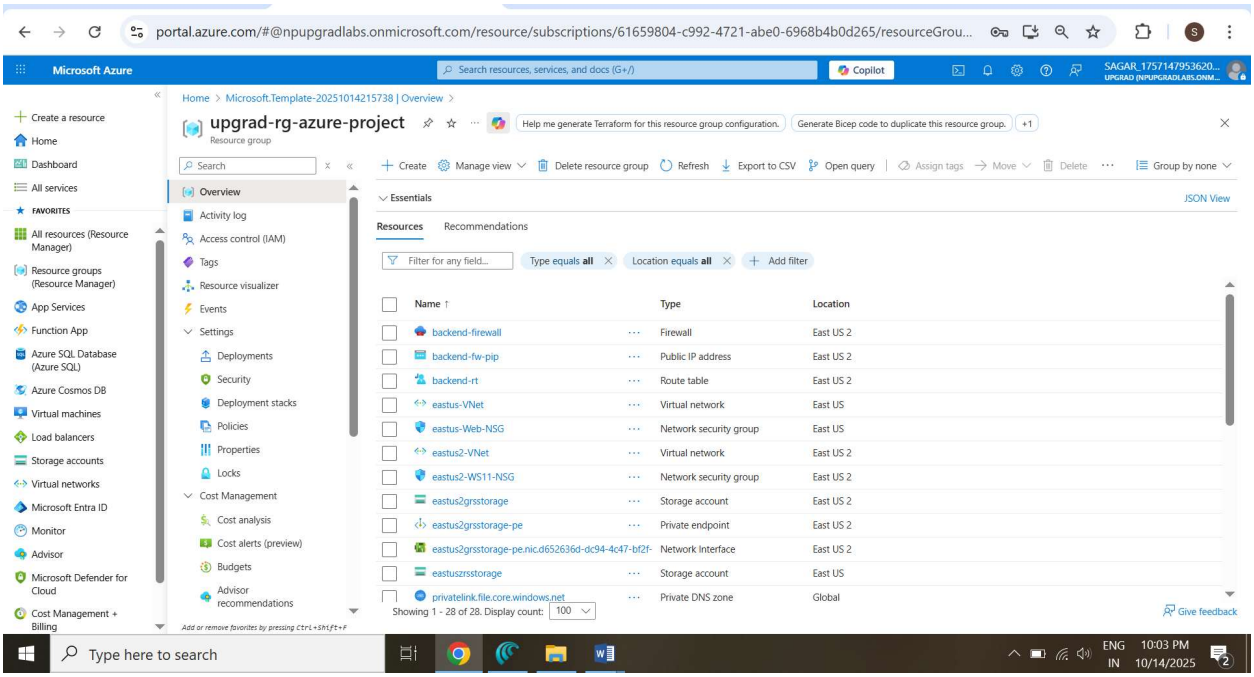
6. Deployment Outputs

- Once succeeded → **Go to Resource Group**

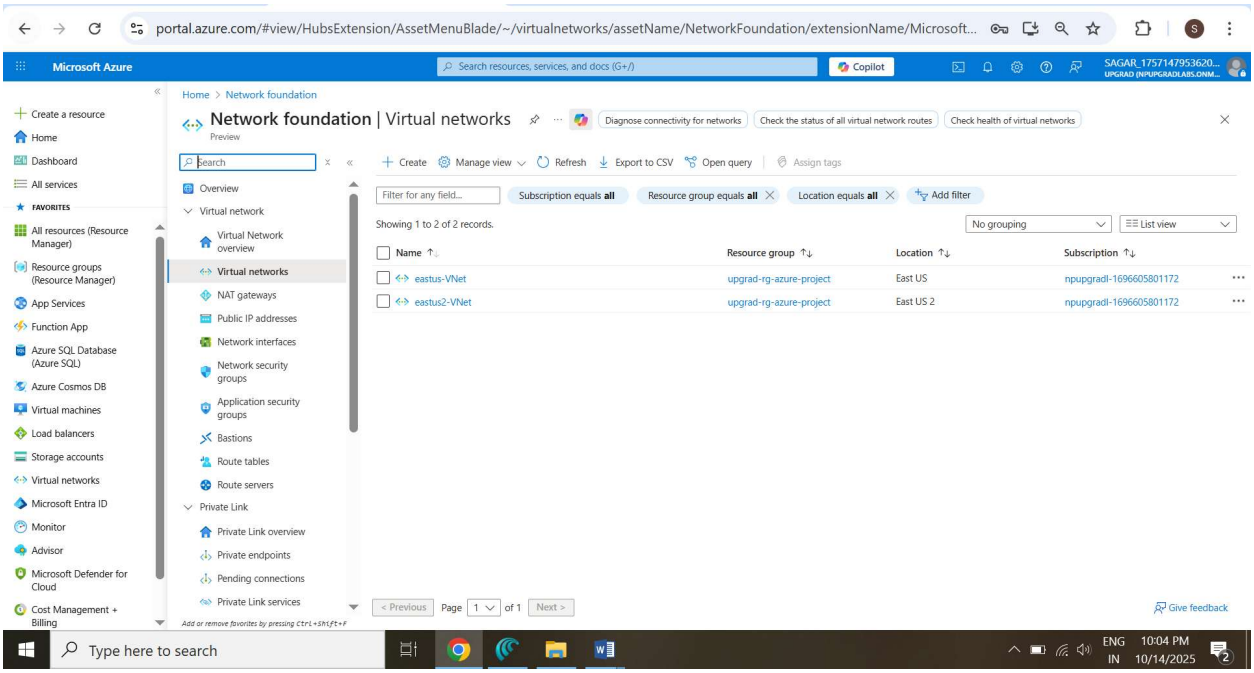


Verification

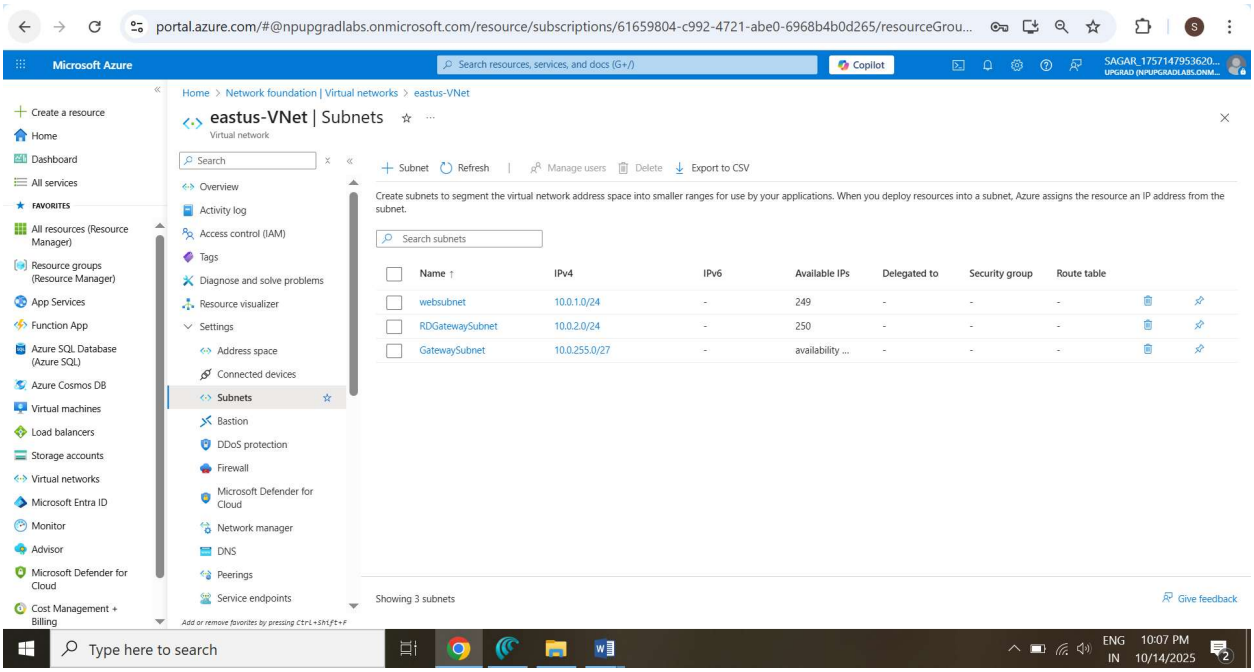
- **Resource Group** upgrad-rg-azure-project created.



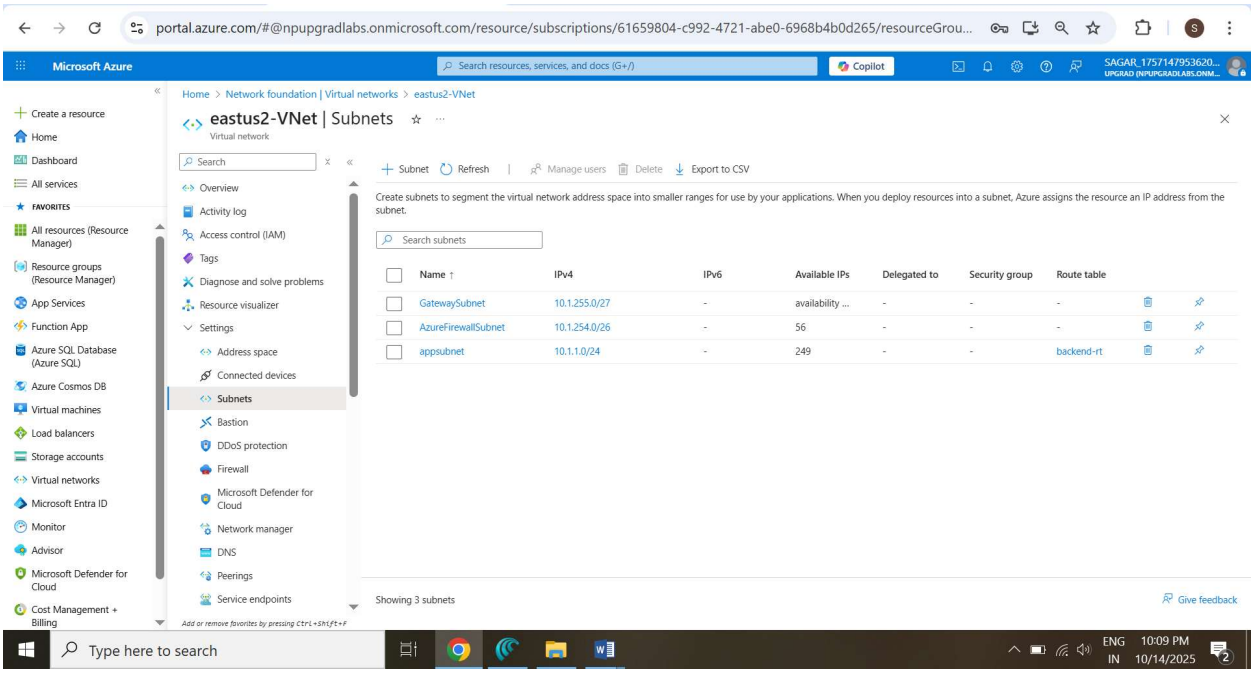
- **VNets** `eastus-VNet` and `eastus2-VNet` visible under **Virtual networks**.



- **Subnets** (`websubnet`, `RDGatewaySubnet`, `GatewaySubnet`) correctly created in **eastus-VNet**



- **Subnets** (`appsubnet`, `AzureFirewallSubnet`, `GatewaySubnet`) correctly created in **eastus2-VNet**



Task II — Deploy Virtual Machines

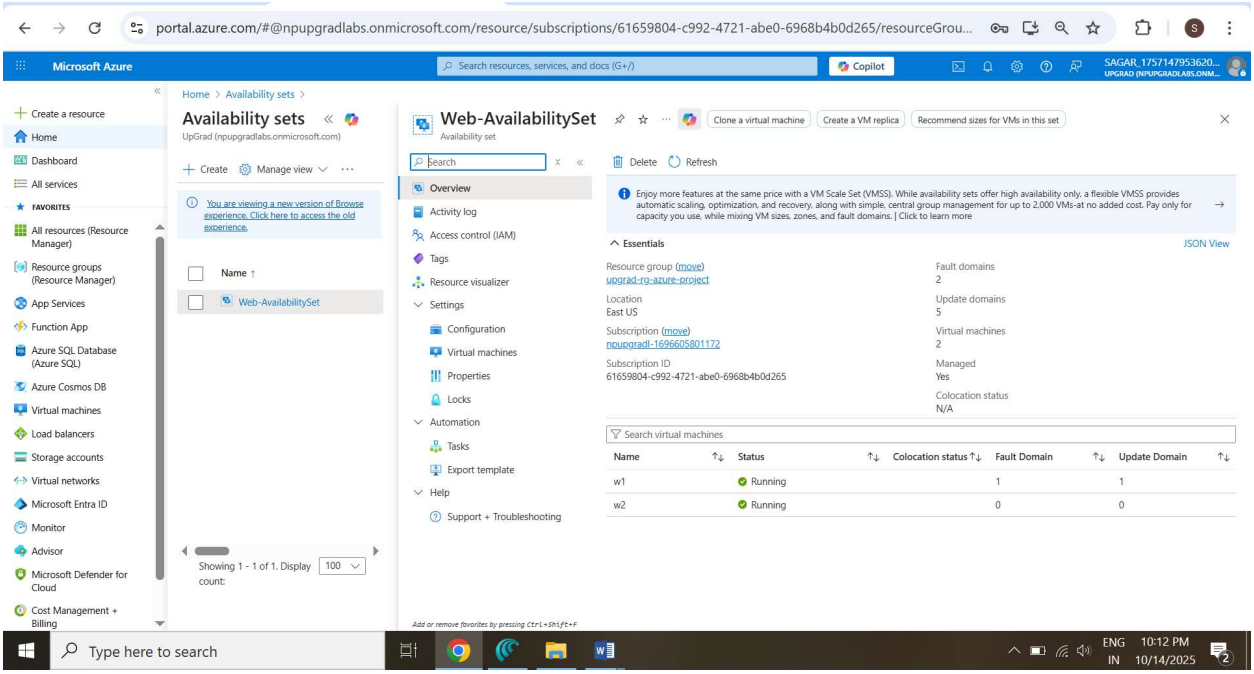
Objective

Deploy web servers, RD Gateway, and WS11 backend server with high availability.

Steps

1. **Verify Availability Set**

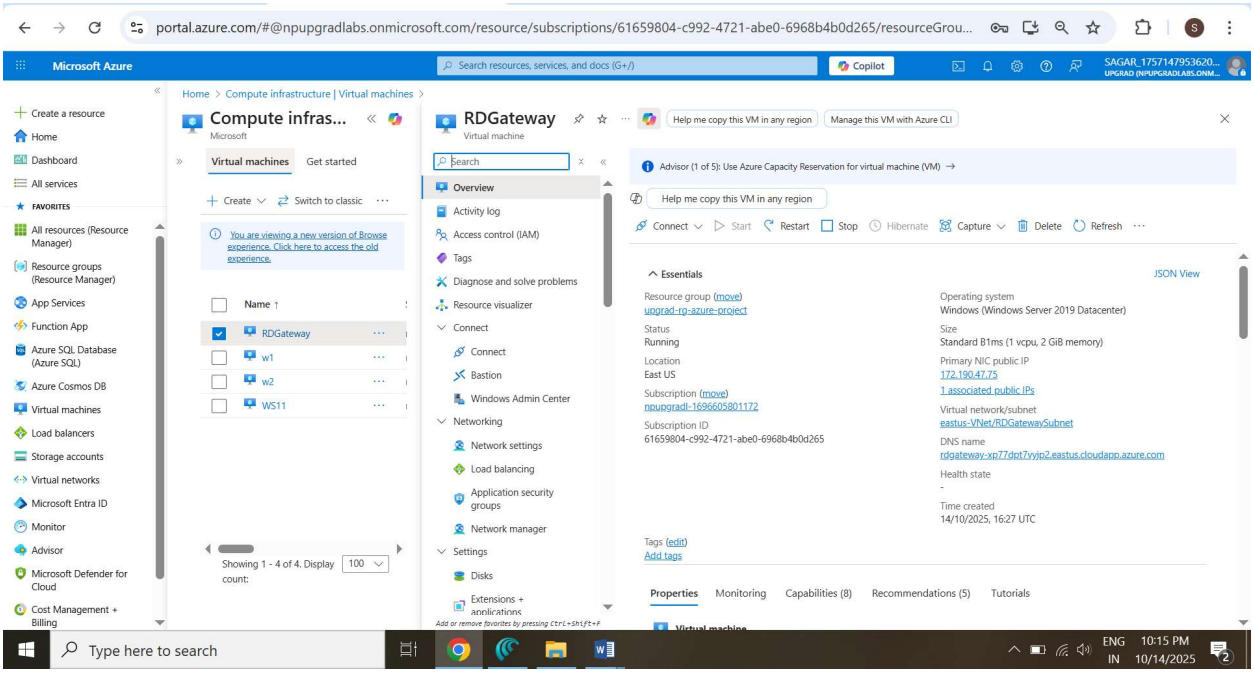
    - **Portal → Availability sets →** `Web-AvailabilitySet`.

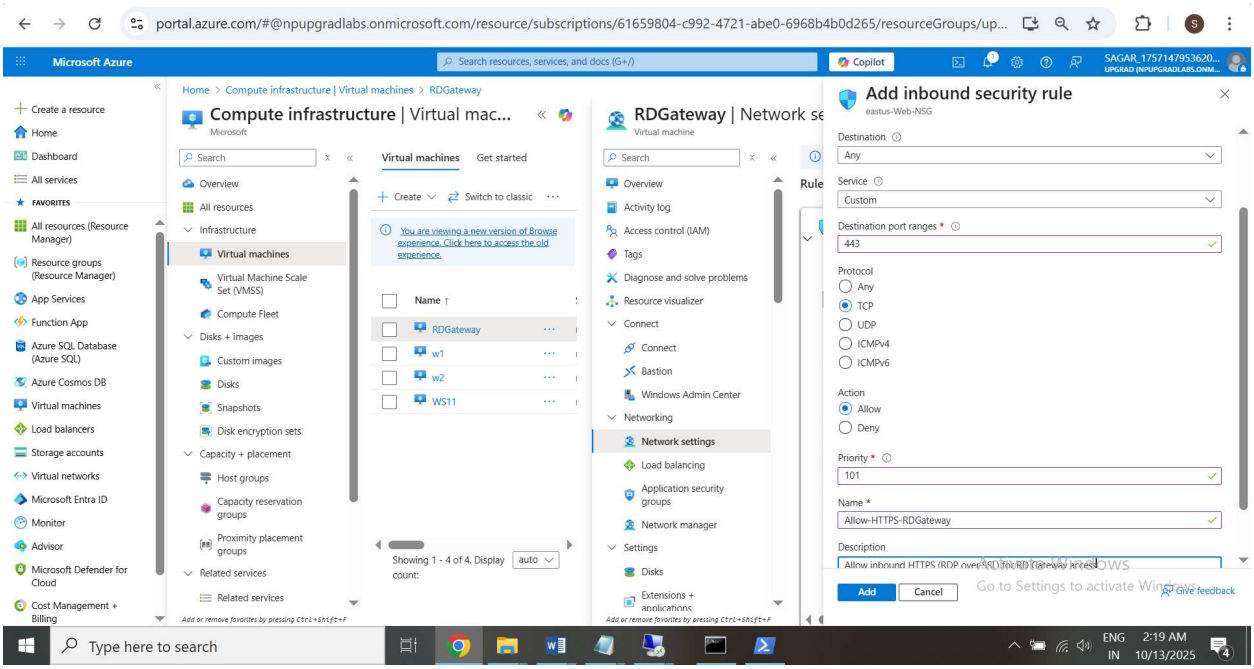


2. Confirm Virtual Machines

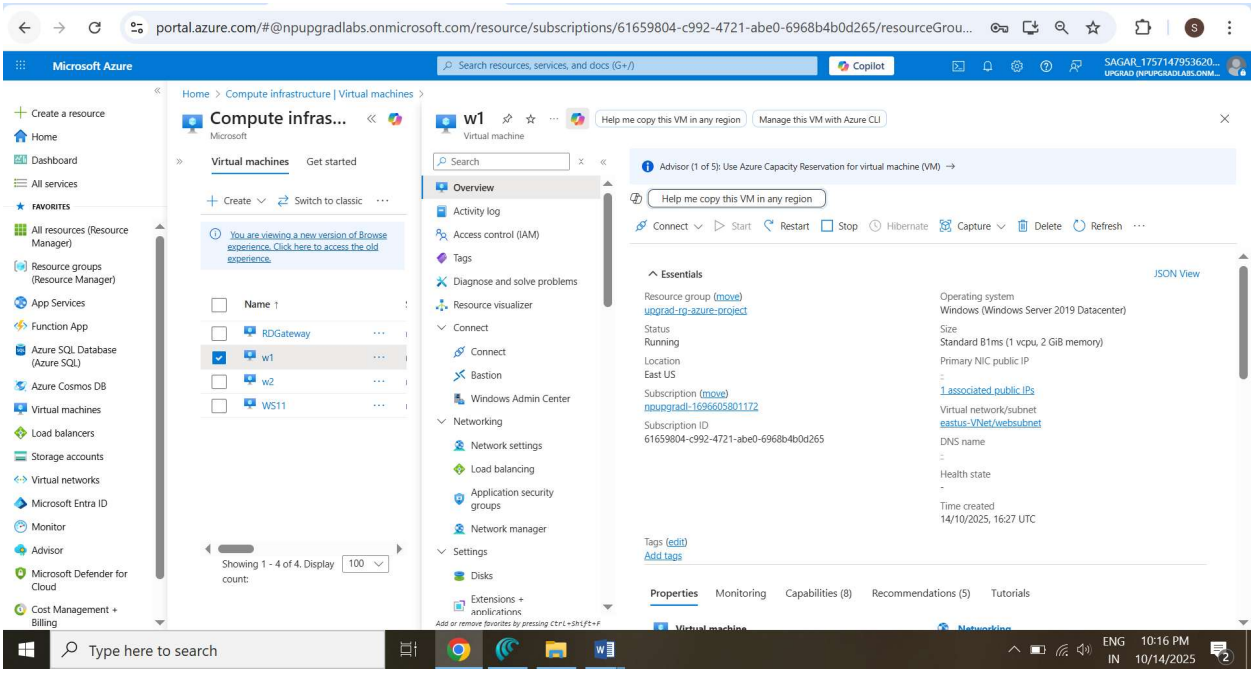
- `w1`, `w2`, `RDGateway`, `WS11` visible in **Virtual Machines**.

RDGateway →

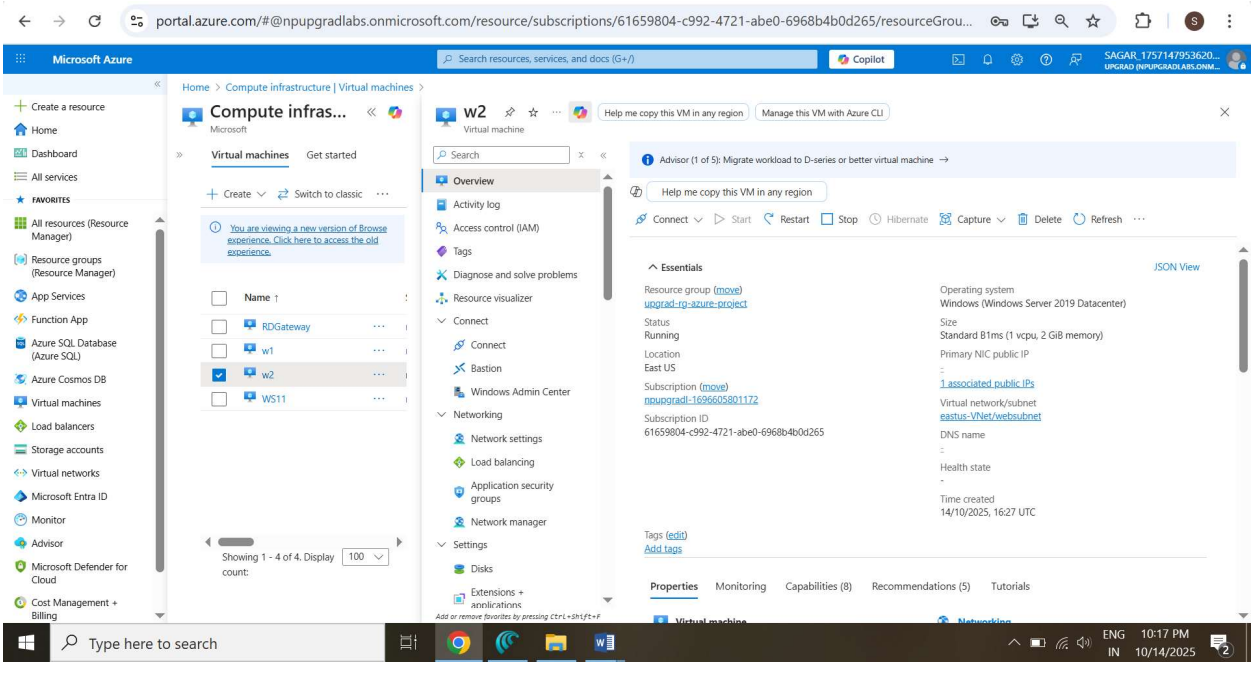




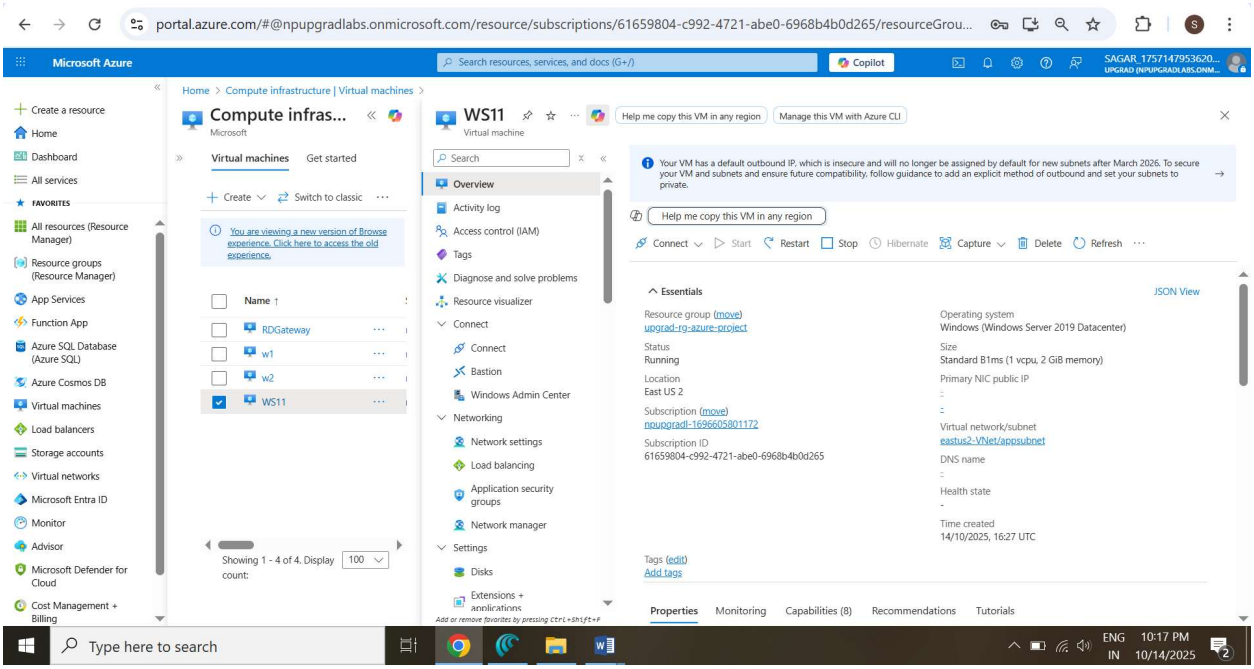
W1 →


portal.azure.com/#@npupgradlabs.onmicrosoft.com/resource/subscriptions/61659804-c992-4721-abe0-6968b4b0d265/resourceGrou...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Compute infrastructure | Virtual machines >
Compute infras...
Microsoft
Virtual machines
Get started
Create
Switch to classic
You are viewing a new version of Browse experience. Click here to access the old experience.
Name
RDGateway
w1
w2
WS11
Showing 1 - 4 of 4. Display count:
100
w1
Virtual machine
Help me copy this VM in any region
Manage this VM with Azure CLI
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Connect
Bastion
Windows Admin Center
Networking
Network settings
Load balancing
Application security groups
Network manager
Settings
Disks
Advisor (1 of 5): Use Azure Capacity Reservation for virtual machine (VM)
Connect
Start
Restart
Stop
Hibernate
Capture
Delete
Refresh
Essentials
JSON View
Resource group (move)
upgrad-rg-azure-project
Status
Running
Location
East US
Subscription (move)
npupgradl-1696605801172
Subscription ID
61659804-c992-4721-abe0-6968b4b0d265
Operating system
Windows (Windows Server 2019 Datacenter)
Size
Standard B1ms (1 vcpu, 2 GiB memory)
Primary NIC public IP
1 associated public IPs
Virtual network/subnet
eastus-VNet/websubnet
DNS name
Health state
Time created
14/10/2025, 16:27 UTC
Tags (edit)
Add tags
Properties
Monitoring
Capabilities (8)
Recommendations (5)
Tutorials
Type here to search
ENG
IN
10:16 PM
10/14/2025

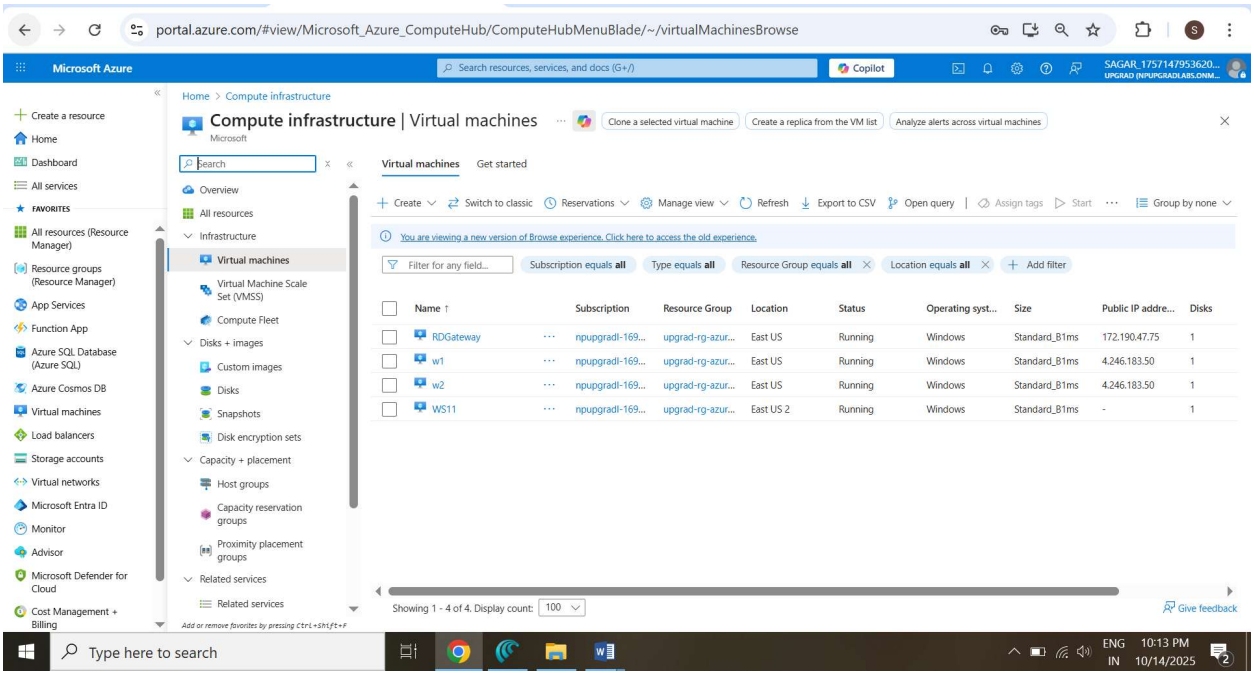
W2 →


portal.azure.com/#@npupgradlabs.onmicrosoft.com/resource/subscriptions/61659804-c992-4721-abe0-6968b4b0d265/resourceGrou...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > Compute infrastructure | Virtual machines >
Compute infras...
Microsoft
Virtual machines
Get started
Create
Switch to classic
You are viewing a new version of Browse experience. Click here to access the old experience.
Name
RDGateway
w1
w2
WS11
Showing 1 - 4 of 4. Display count:
100
w2
Virtual machine
Help me copy this VM in any region
Manage this VM with Azure CLI
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Resource visualizer
Connect
Bastion
Windows Admin Center
Networking
Network settings
Load balancing
Application security groups
Network manager
Settings
Disks
Advisor (1 of 5): Migrate workload to D-series or better virtual machine
Connect
Start
Restart
Stop
Hibernate
Capture
Delete
Refresh
Essentials
JSON View
Resource group (move)
upgrad-rg-azure-project
Status
Running
Location
East US
Subscription (move)
npupgradl-1696605801172
Subscription ID
61659804-c992-4721-abe0-6968b4b0d265
Operating system
Windows (Windows Server 2019 Datacenter)
Size
Standard B1ms (1 vcpu, 2 GiB memory)
Primary NIC public IP
1 associated public IPs
Virtual network/subnet
eastus-VNet/websubnet
DNS name
Health state
Time created
14/10/2025, 16:27 UTC
Tags (edit)
Add tags
Properties
Monitoring
Capabilities (8)
Recommendations (5)
Tutorials
Type here to search
ENG
IN
10:17 PM
10/14/2025

WS11 →

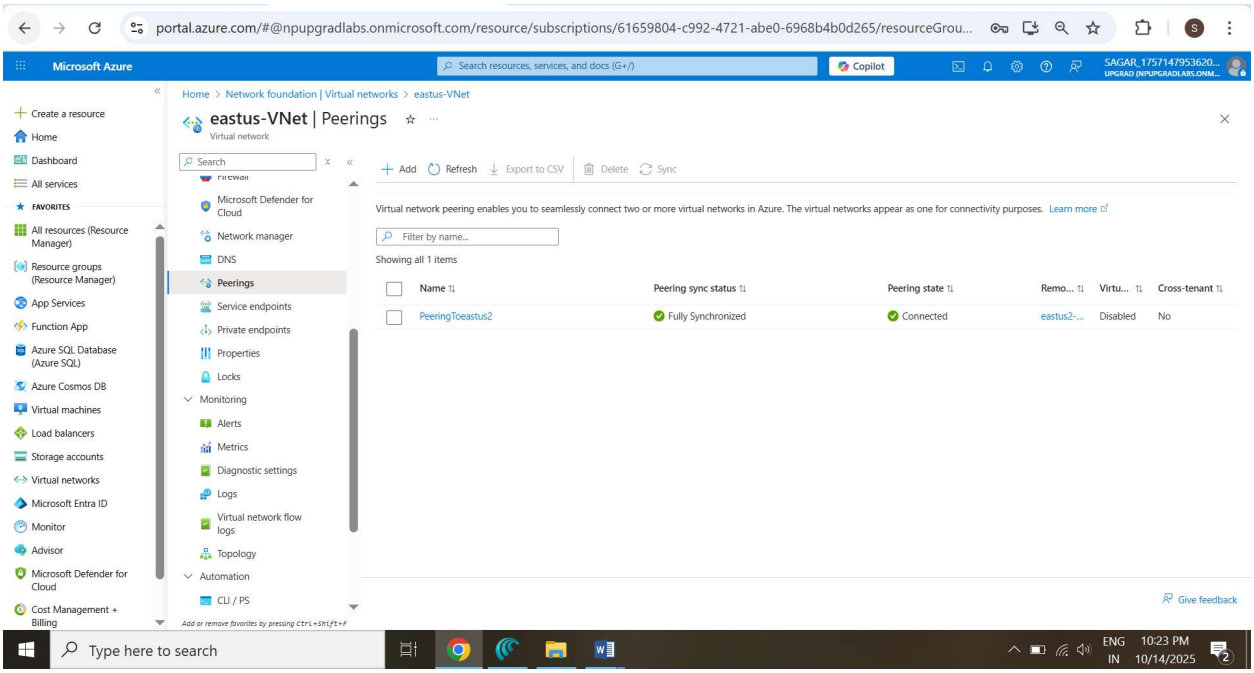


Virtual Machines →

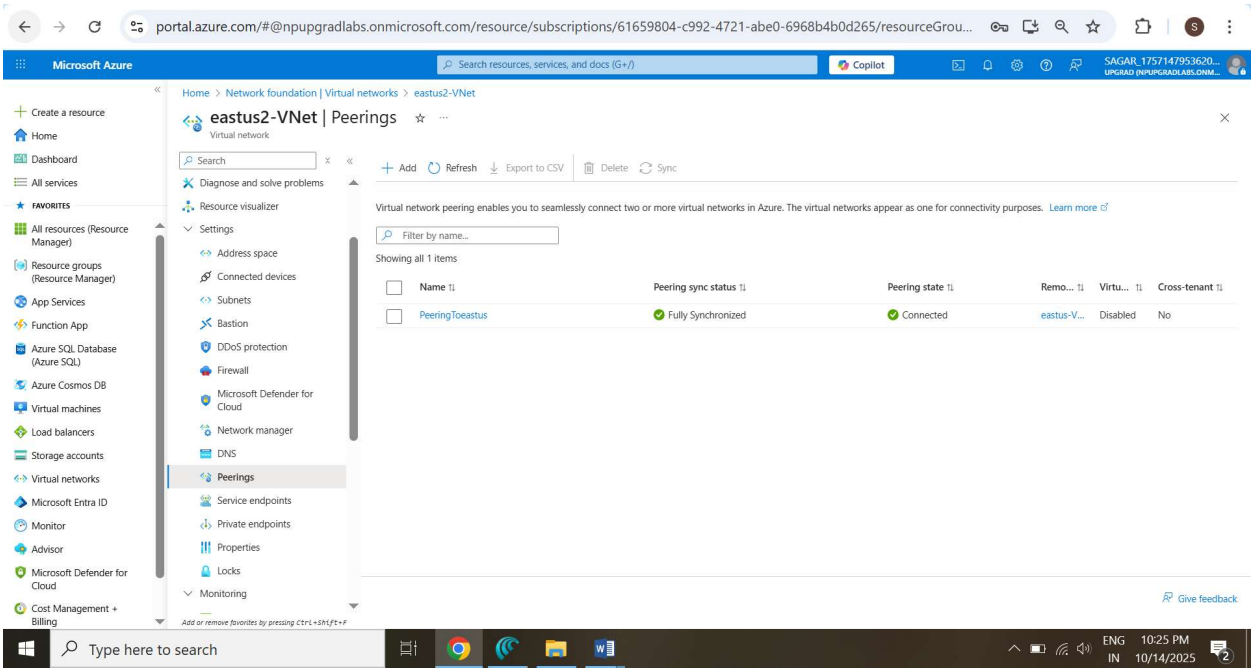


3. Check VNet Peering

- `eastUS-VNet` → **Peerings** → Confirm “Connected” to `eastUS2-VNet`.



- Vice versa **`eastUS2-VNet`** → **Peerings** → Confirm “Connected” **to** **`eastUS-VNet`**



Verification

- W1 and W2 deployed in EastUS under availability set. → Done above
- RDGateway has public IP. → Done above
- WS11 deployed in EastUS2 (private IP only). → Done above
- Peering between VNets connected. → Done above

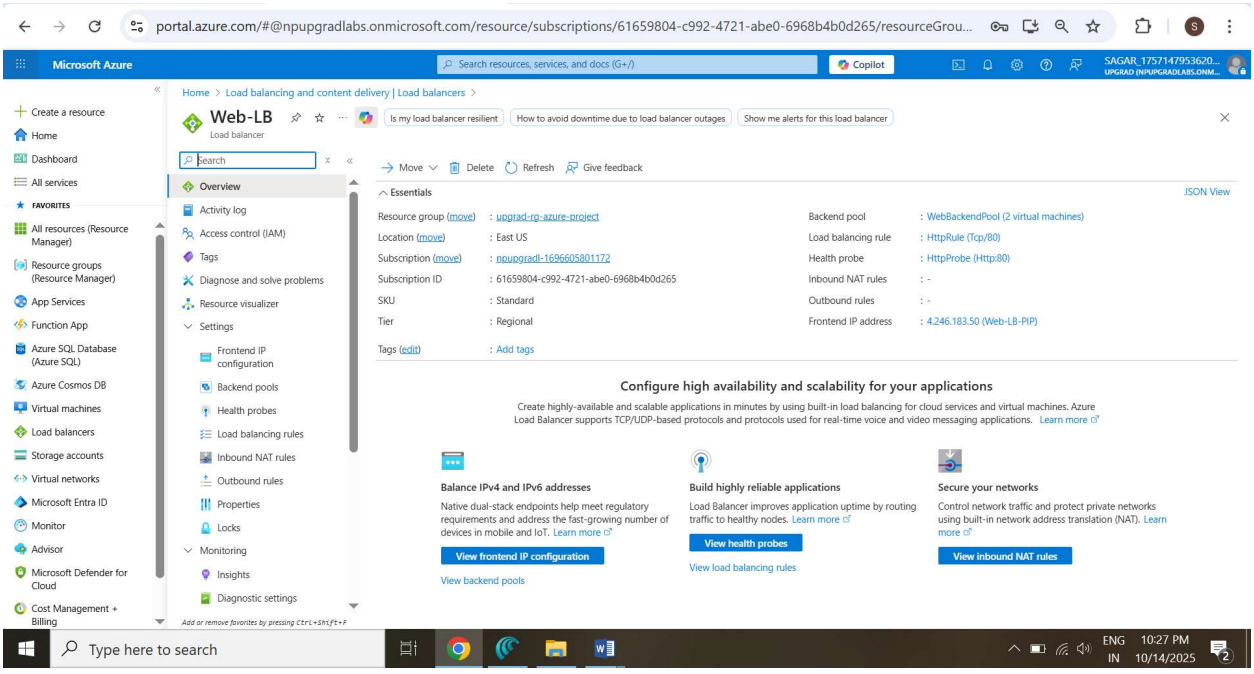
Task III — Implement Secure Connectivity

Objective

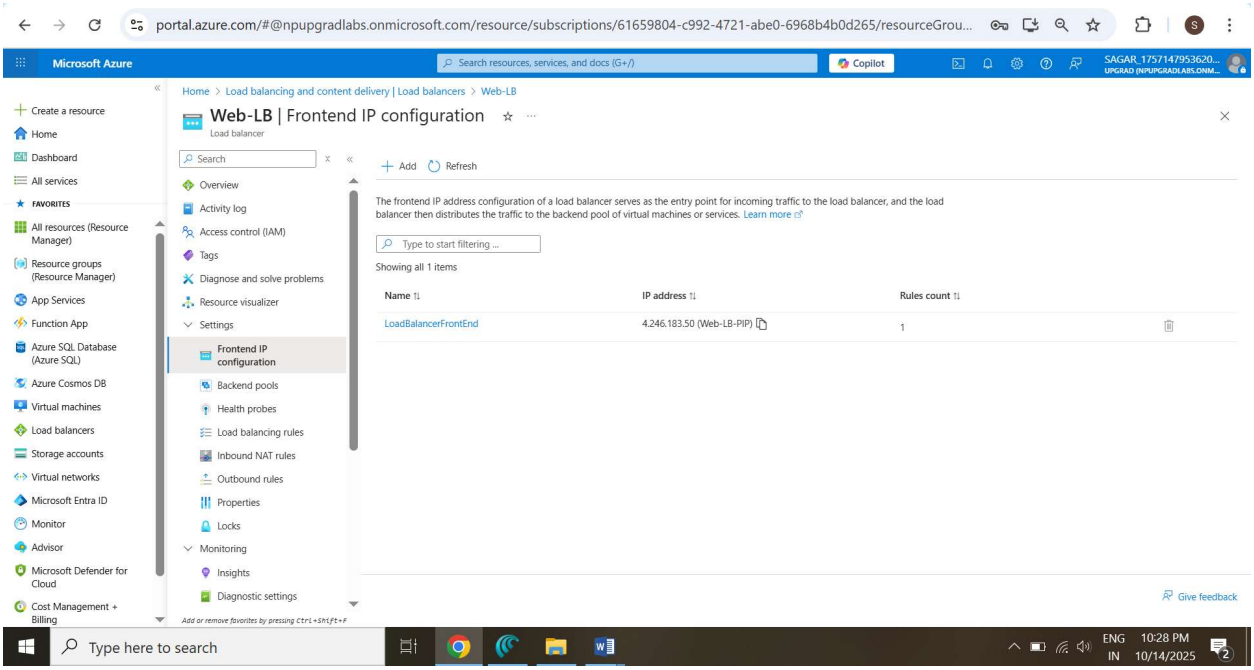
Establish secure access and restricted communication using NSGs, VPN/Private Link, and Load Balancer.

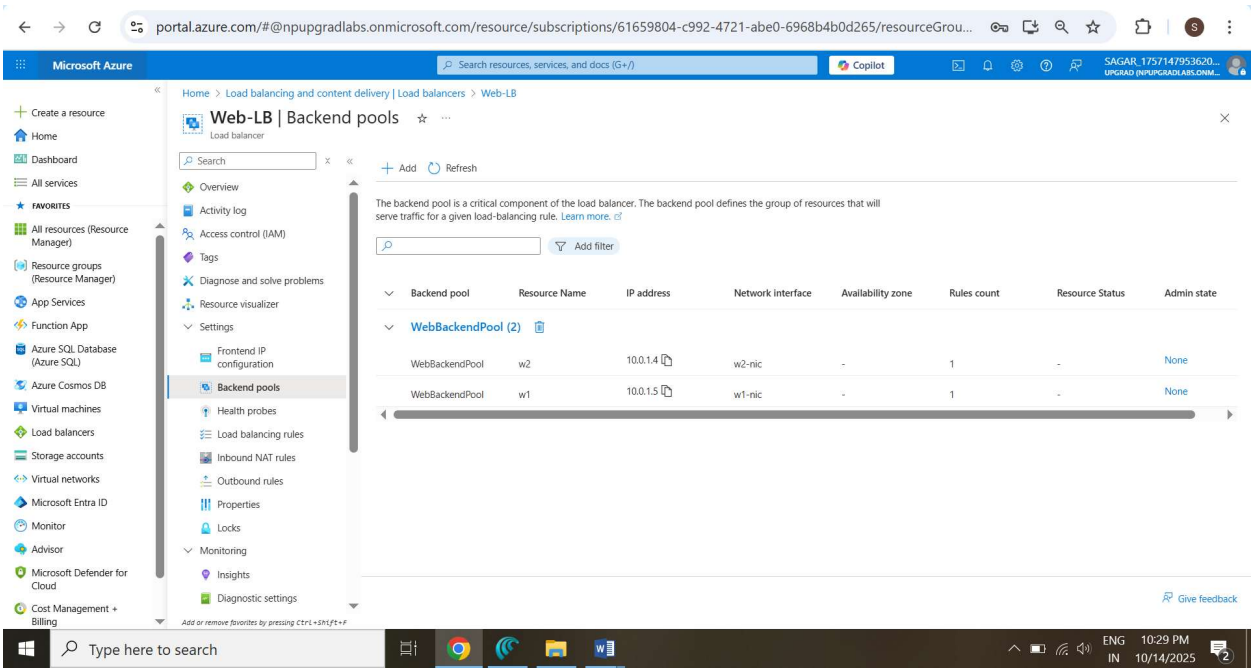
Steps

1. **Load Balancer Configuration**
    - Portal → Load Balancers → **`Web-LB.`**

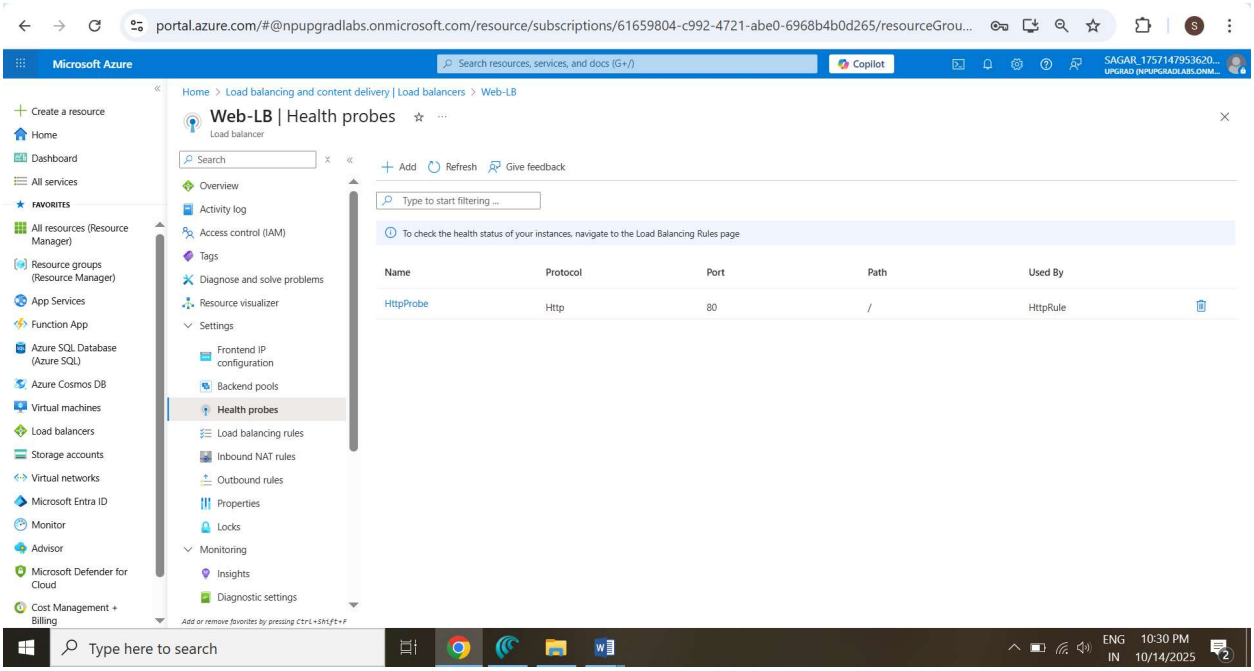


- Verify Frontend IP, Backend Pool, and Health Probe.



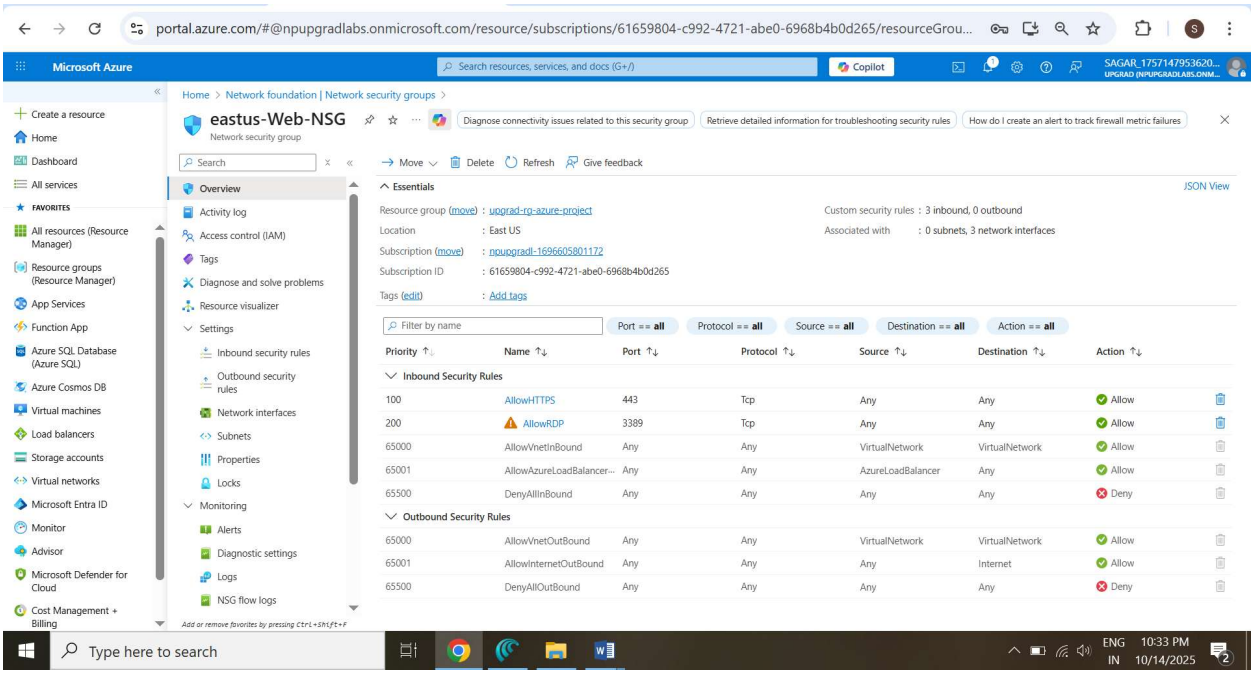


Health Probe :

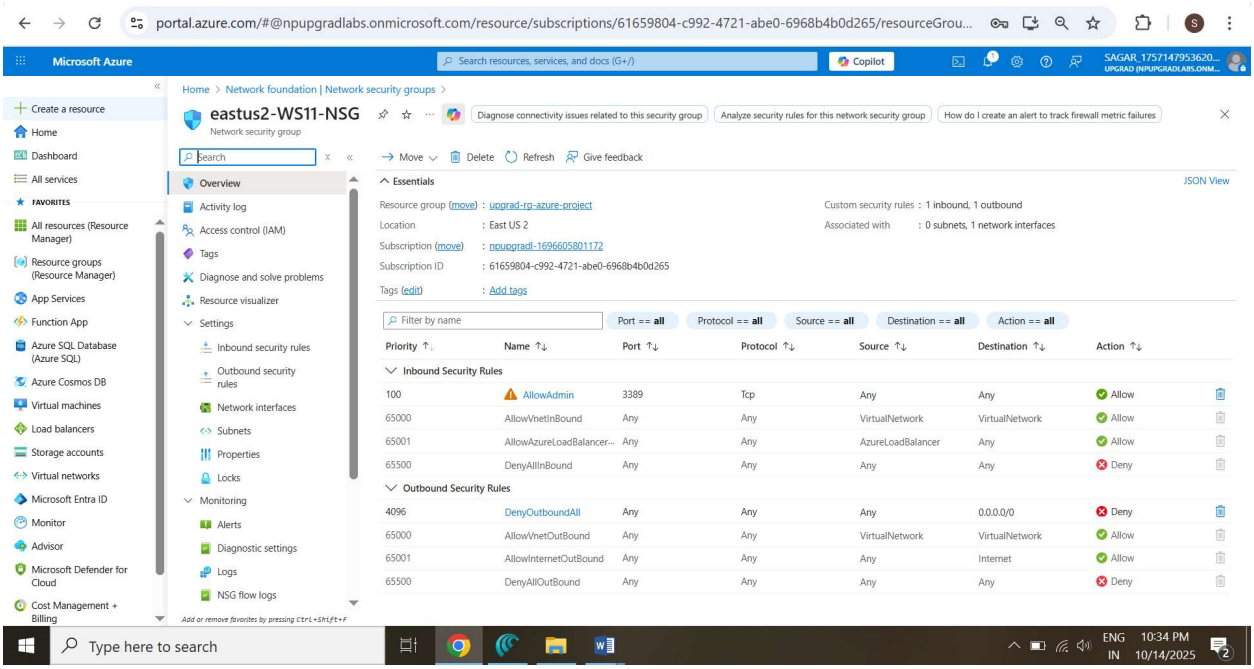


2. Network Security Groups (NSGs)

- `EastUS-Web-NSG` allow **HTTPS (443), RDP (3389).**

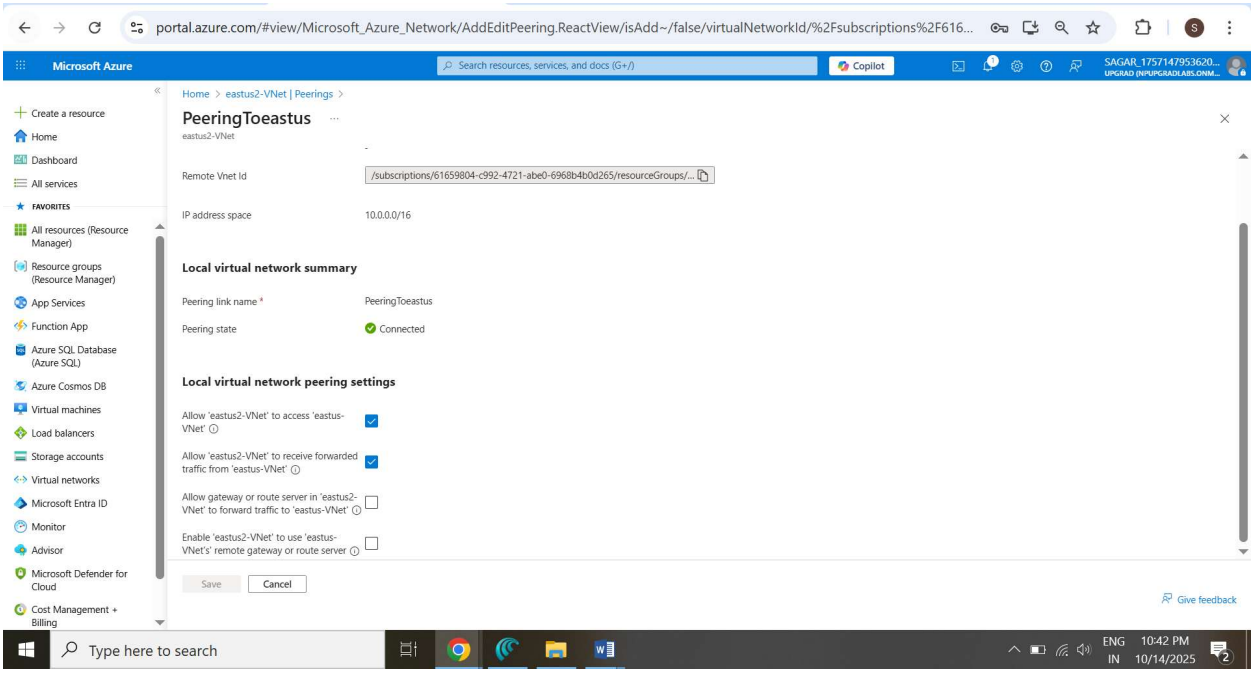


- `EastUS2-WS11-NSG` deny **outbound to social media** sites.

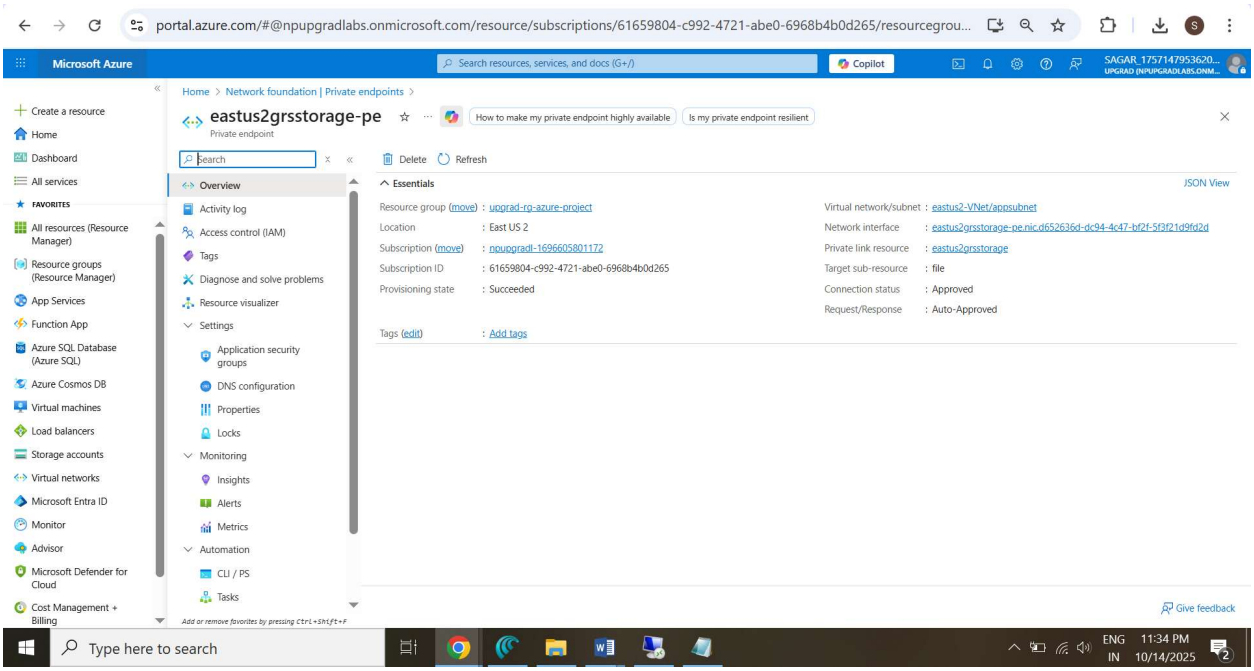


3. Secure Communication

- Confirm encrypted traffic **between VNets.**

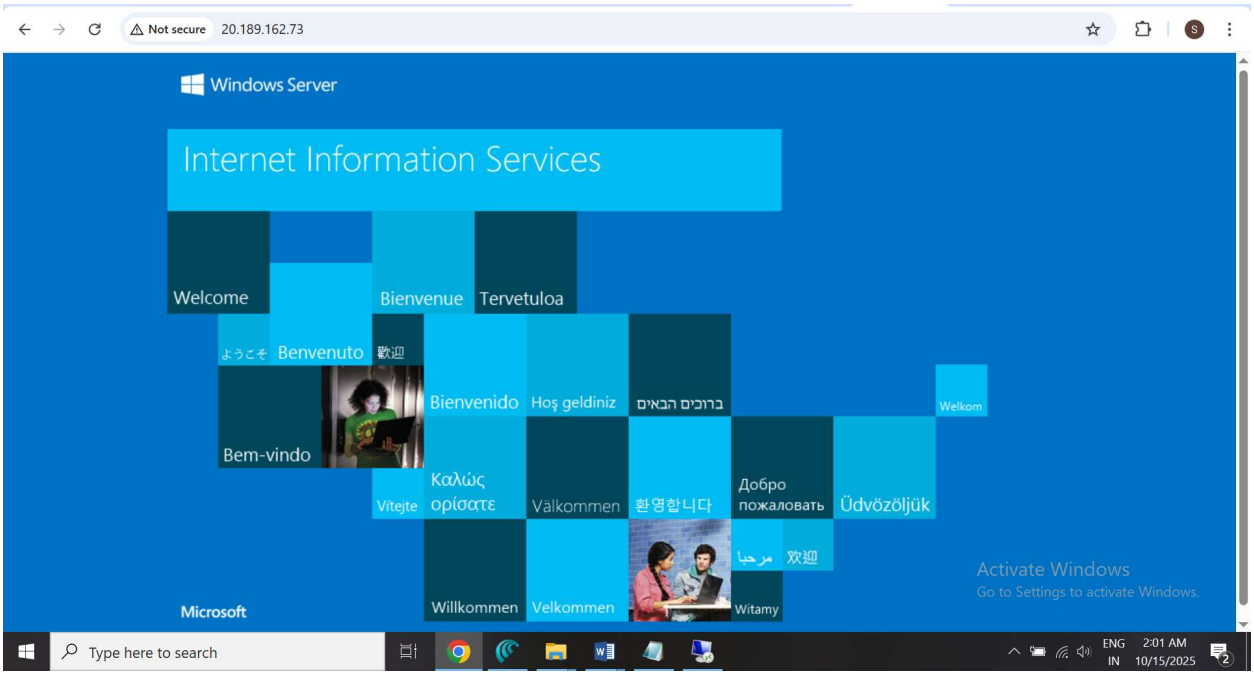


4. Azure Private Link Verification

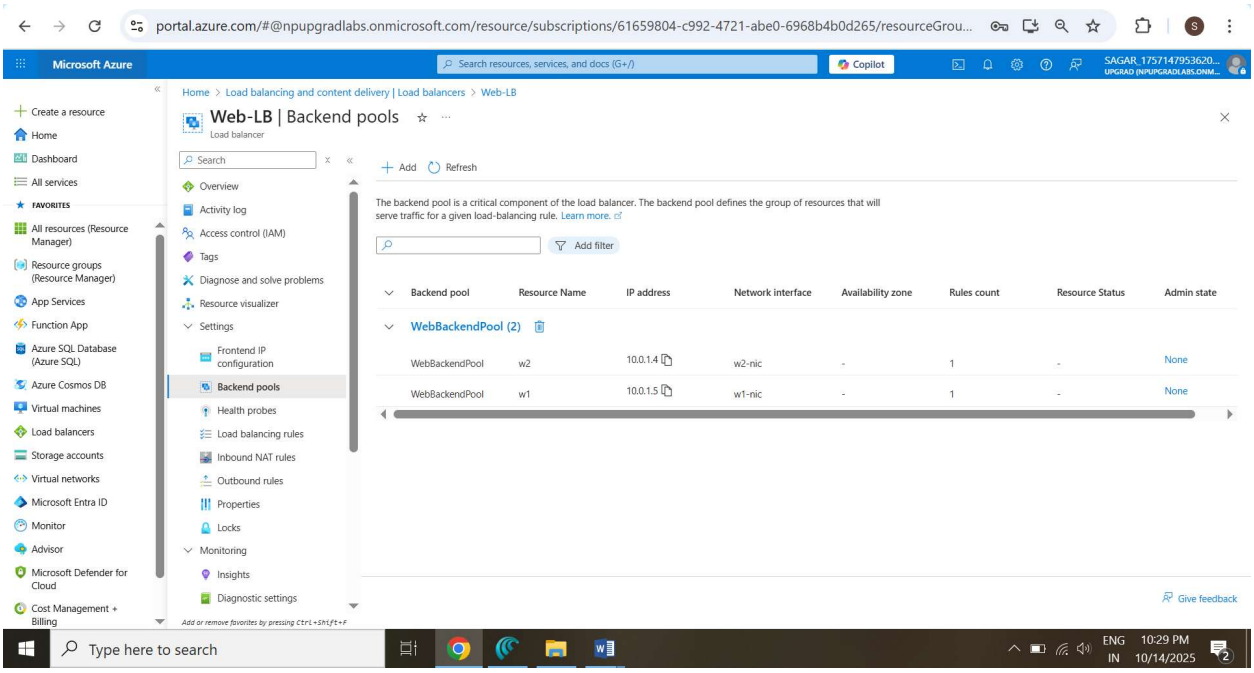


Verification

- LB public IP reachable.



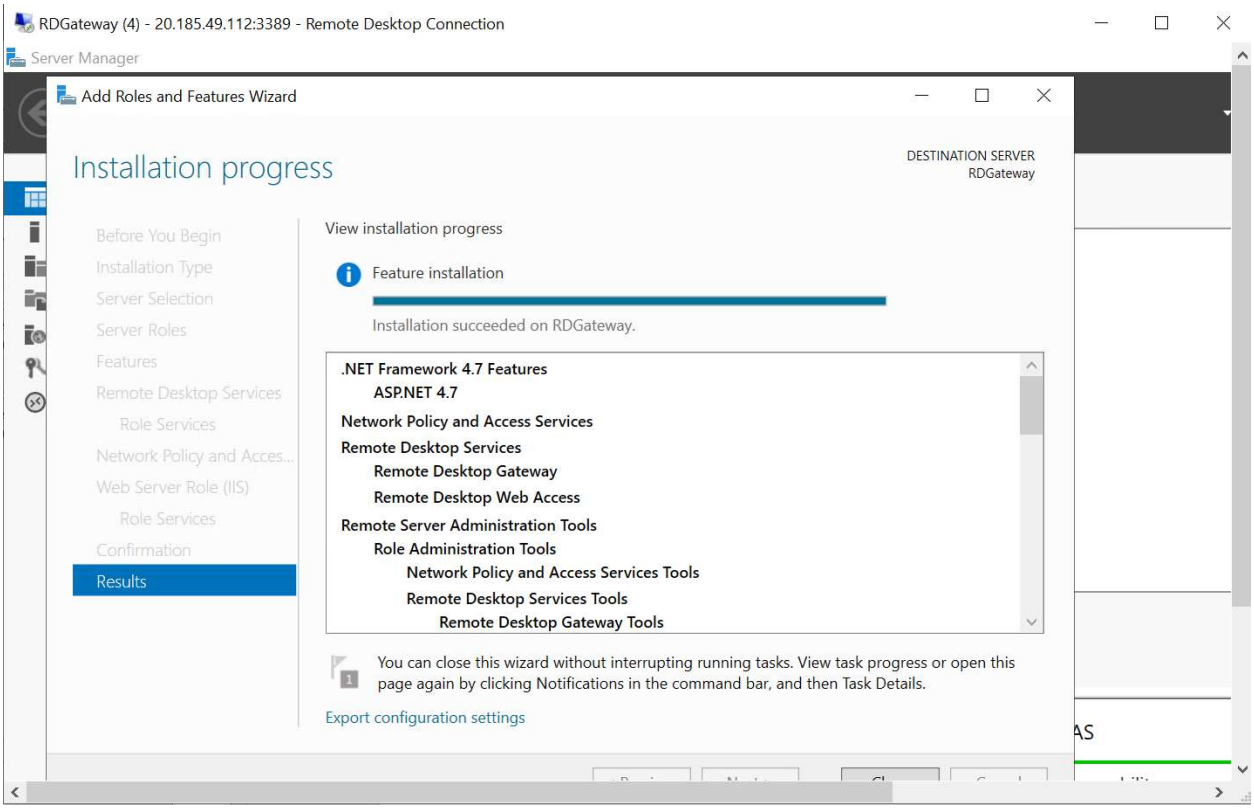
- w1 & w2 appear in backend pool.



RD Gateway & RD Web Access — GUI-based Setup (Windows Server 2022/2025)

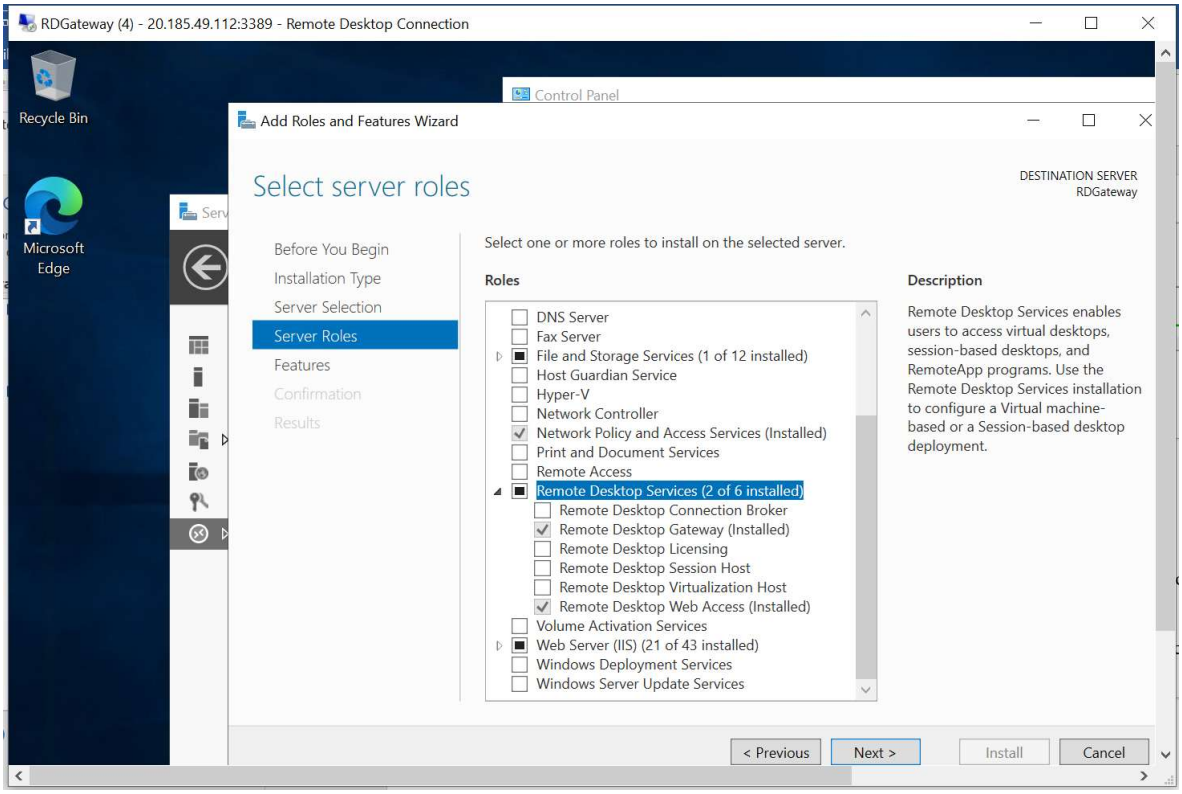
- ## STEP 1 — Install Roles via Server Manager

1. Open **Server Manager → Manage → Add Roles and Features**.
2. Choose **Role-based or feature-based installation**, click **Next**.
3. Select **This server**.
4. In **Server Roles**, check:
    - **Remote Desktop Gateway**
    - **Remote Desktop Web Access**
    - IIS and Management Tools will auto-select → keep them.
5. Click **Next** until **Install**.
6. Wait for installation to complete → **Close**.



- ## STEP 2 — Verify Roles Installed

- Open **Server Manager → Dashboard → Remote Desktop Services**.
- You should see **RD Gateway** and **RD Web Access** listed.
- Optionally, open **Programs and Features → Turn Windows features on/off** to confirm.



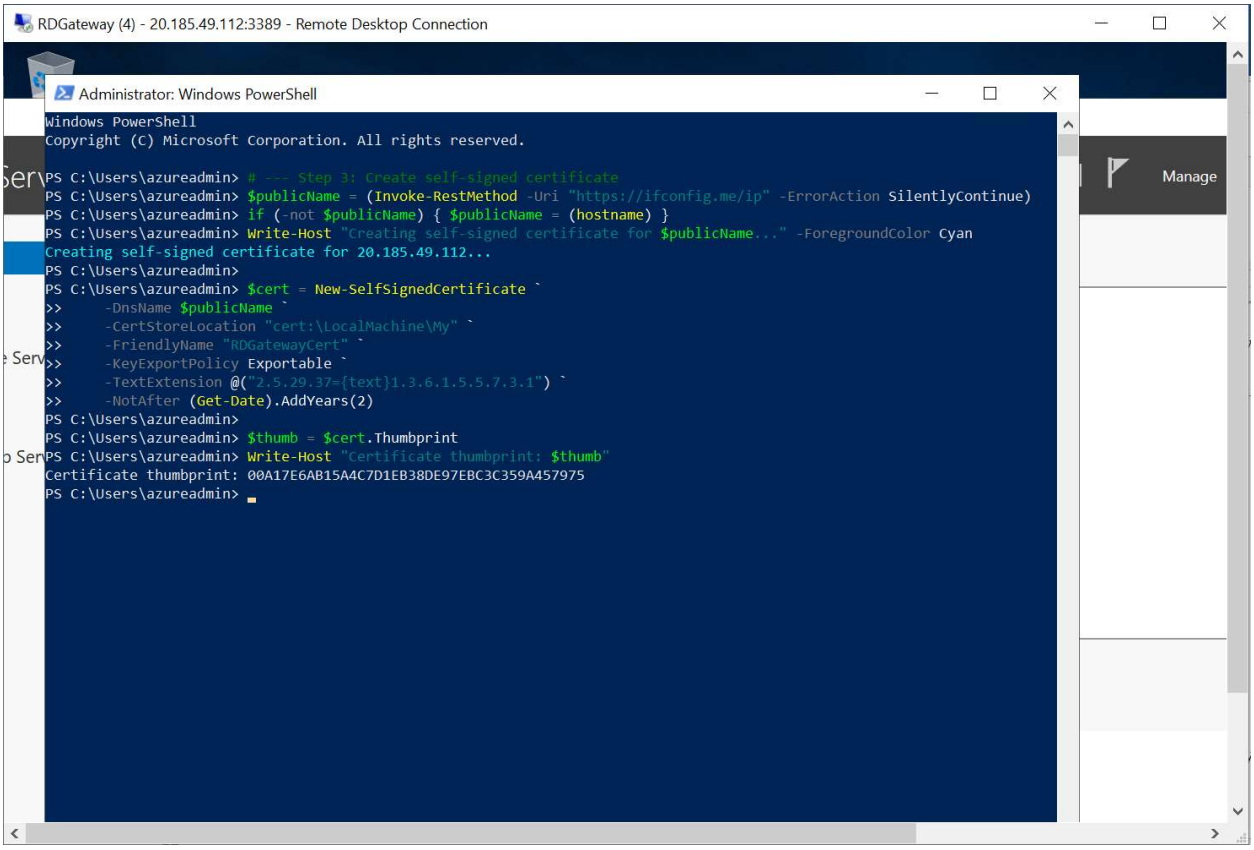
- **STEP 3 — Create a Valid HTTPS Certificate**

You can use **IIS Manager** or **Certificates MMC**. Using **Powershell**

```
# --- Step 3: Create self-signed certificate

$publicName = (Invoke-RestMethod -Uri "https://ifconfig.me/ip" -ErrorAction
SilentlyContinue)

if (-not $publicName) { $publicName = (hostname) }
Write-Host "Creating self-signed certificate for $publicName..." -ForegroundColor Cyan
$cert = New-SelfSignedCertificate `
    -DnsName $publicName `
    -CertStoreLocation "cert:\LocalMachine\My" `
    -FriendlyName "RDGatewayCert" `
    -KeyExportPolicy Exportable `
    -TextExtension @("2.5.29.37={text}1.3.6.1.5.5.7.3.1") `
    -NotAfter (Get-Date).AddYears(2)
$thumb = $cert.Thumbprint
Write-Host "Certificate thumbprint: $thumb"
```

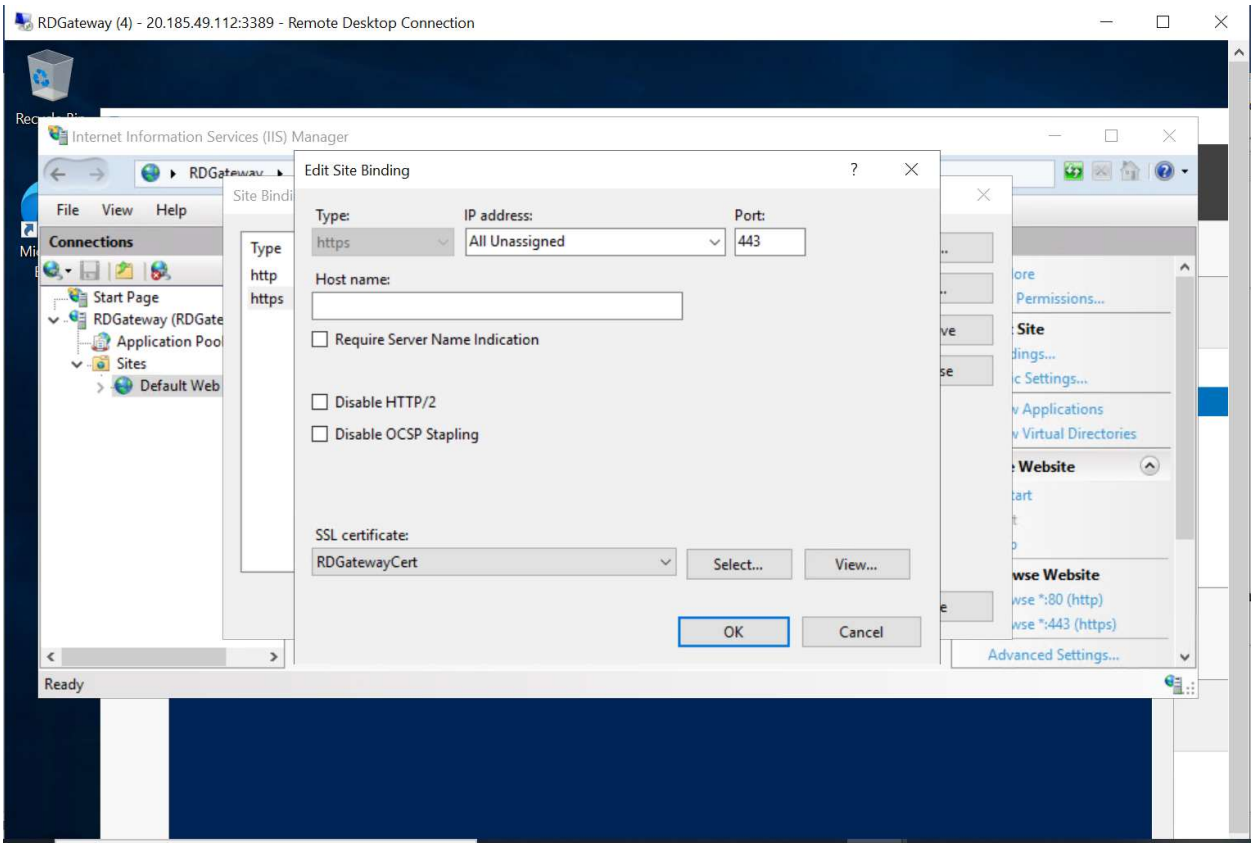


- **IIS Manager**

1. Open **IIS Manager** → click server name (left pane).
2. Double-click **Server Certificates** (center).
3. In right Actions pane → click **Create Self-Signed Certificate**.
4. Name it: `RDGW-SelfSigned-20.185.49.112`.
5. Choose **Personal** store → OK.

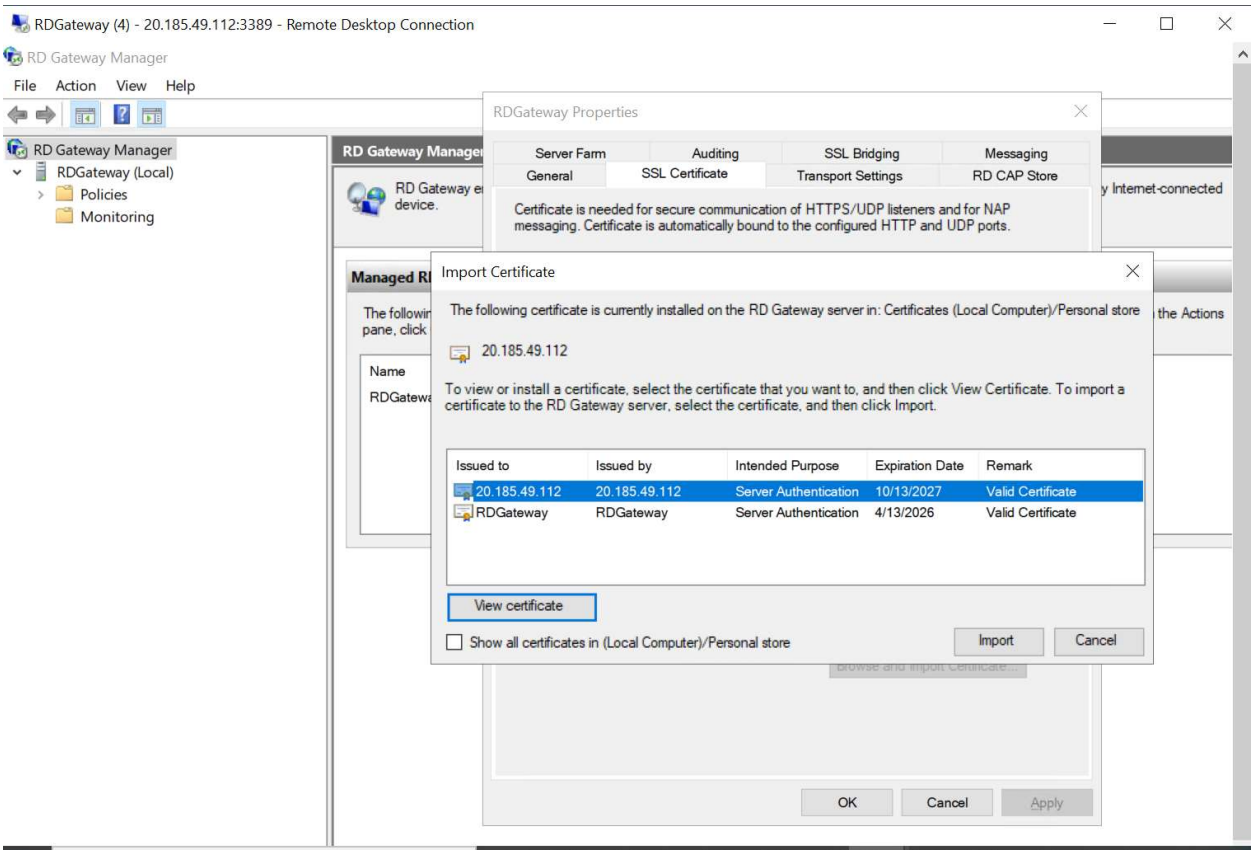
- **STEP 4 — Bind Certificate to IIS (HTTPS 443)**

1. In **IIS Manager** → expand **Sites** → **Default Web Site**.
2. Right pane → click **Bindings…**.
3. Click **Add… →**
    - Type: `https`
    - IP address: `All Unassigned` or your public IP
    - Port: `443`
    - SSL certificate: choose the cert you just created
    - Click **OK** → then **Close**.



- **STEP 5 — Bind Certificate to RD Gateway**

1. Open **RD Gateway Manager** (`tsgateway.msc`).
2. Right-click server name → **Properties**.
3. Go to **SSL Certificate** tab.
4. Click **Import a certificate into the personal store**.
5. Select your certificate → click **Import**.
6. Once imported, select it and click **Apply**.



- **STEP 6 — Configure Firewall Rules (if needed)**

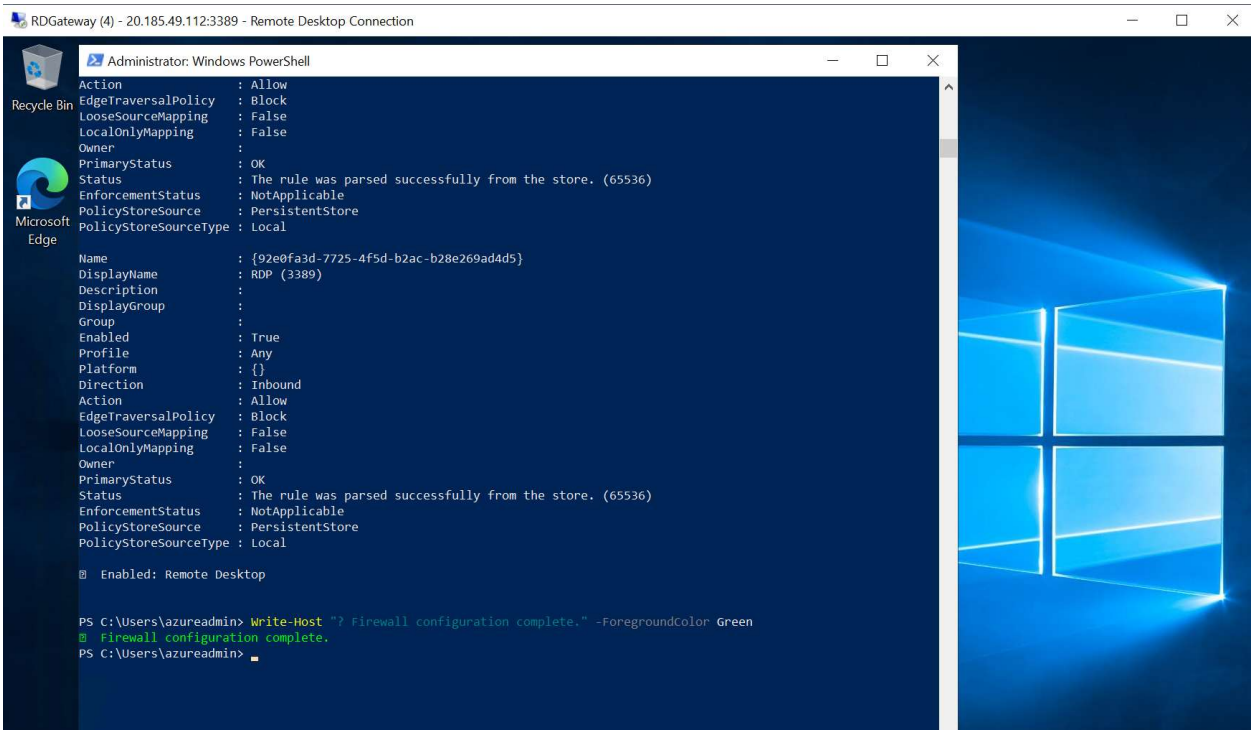
Windows usually enables these automatically.
To verify:

1. Open **Windows Defender Firewall → Advanced settings**.
2. Click **Inbound Rules** → scroll to:
    - *World Wide Web Services (HTTP and HTTPS)*
    - *Remote Desktop Gateway*
3. Make sure **Enabled = Yes** and **Action = Allow**.

If not → right-click → **Enable Rule**.

Powershell Script →

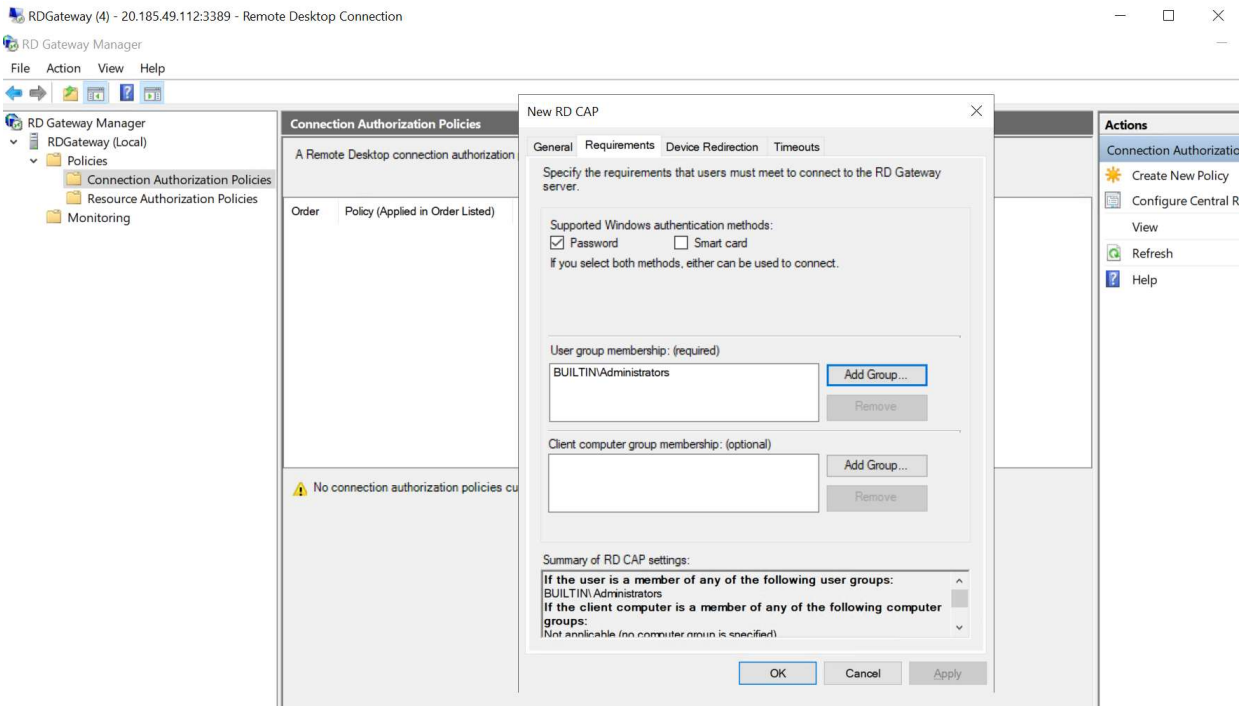
```
Write-Host "=== Enabling RD Gateway Firewall Rules ===" -ForegroundColor Cyan
$rules=@("World Wide Web Services (HTTP)","World Wide Web Services (HTTPS)","Remote Desktop Gateway","Remote Desktop")
foreach($r in $rules){
    try{Enable-NetFirewallRule -DisplayGroup $r -ErrorAction Stop;Write-Host "✓ Enabled: $r"}
    catch{if($r -match "HTTP" -and $r -notmatch "HTTPS"){New-NetFirewallRule -DisplayName "HTTP (80)" -Dir In -Protocol TCP -LocalPort 80 -Action Allow}
        elseif($r -match "HTTPS"){New-NetFirewallRule -DisplayName "HTTPS (443)" -Dir In -Protocol TCP -LocalPort 443 -Action Allow}
        elseif($r -match "Desktop"){New-NetFirewallRule -DisplayName "RDP (3389)" -Dir In -Protocol TCP -LocalPort 3389 -Action Allow}}}
Write-Host "✓ Firewall configuration complete." -ForegroundColor Green
```



- # STEP 7 — Configure CAP and RAP (Access Policies)

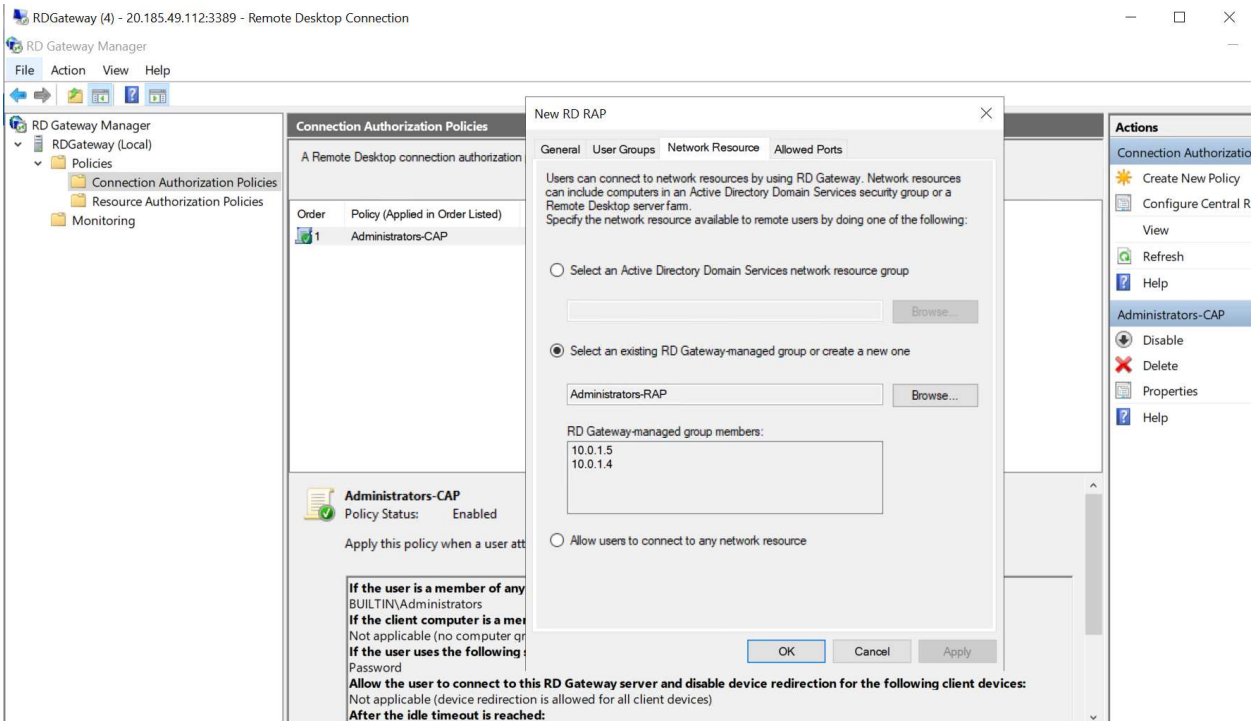
1. In **RD Gateway Manager**, expand your server → **Policies** → **Connection Authorization Policies** (CAP).

    - Right-click → **Create New Policy.**
    - Name: **Administrators-CAP.**
    - Choose **User Group: Administrators.**
    - Click **Finish.**



2. **Expand Resource Authorization Policies (RAP) →**

    - Right-click → **Create New Policy**.
    - Name: `Administrators-RAP`.
    - Add **Computer Group** → add `w1`, `w2`, `WS11` or any internal servers.
    - Click Finish.

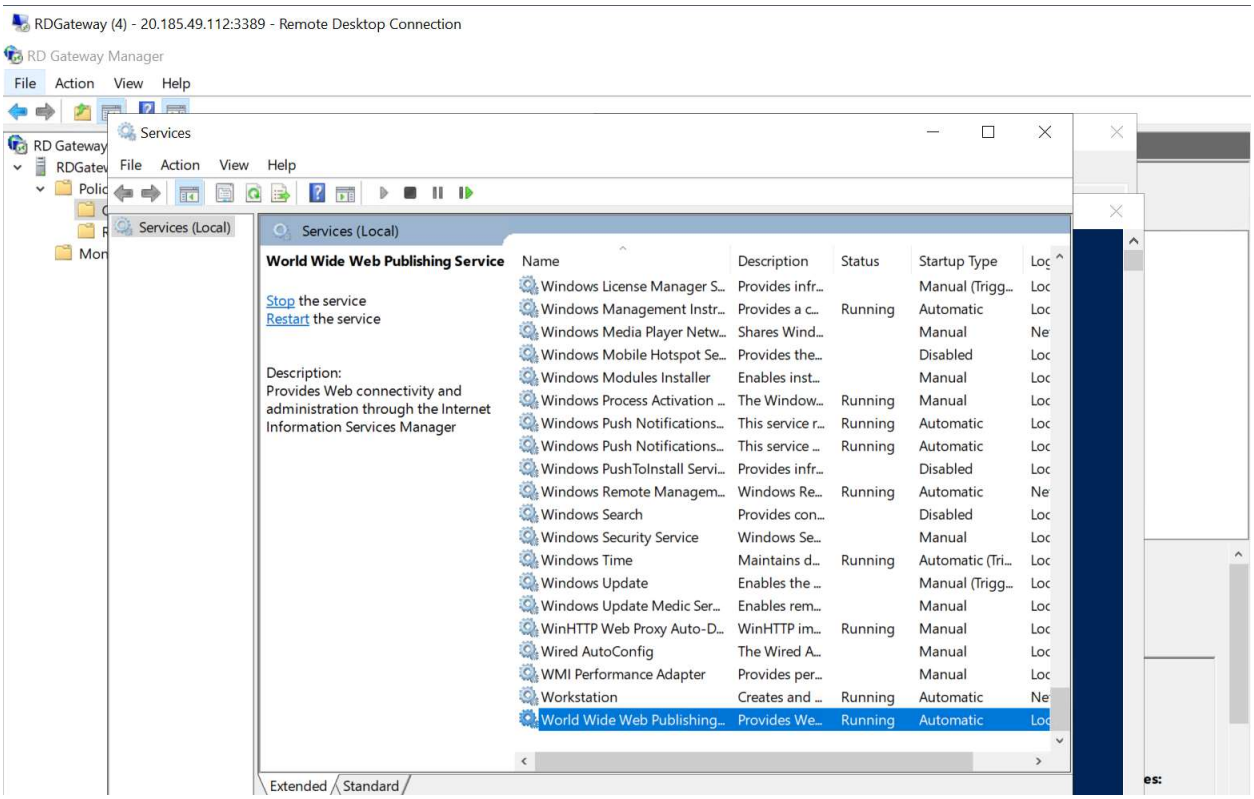


- # STEP 8 — Restart Services

1. Open **Services.msc**.
2. Locate **and restart:**

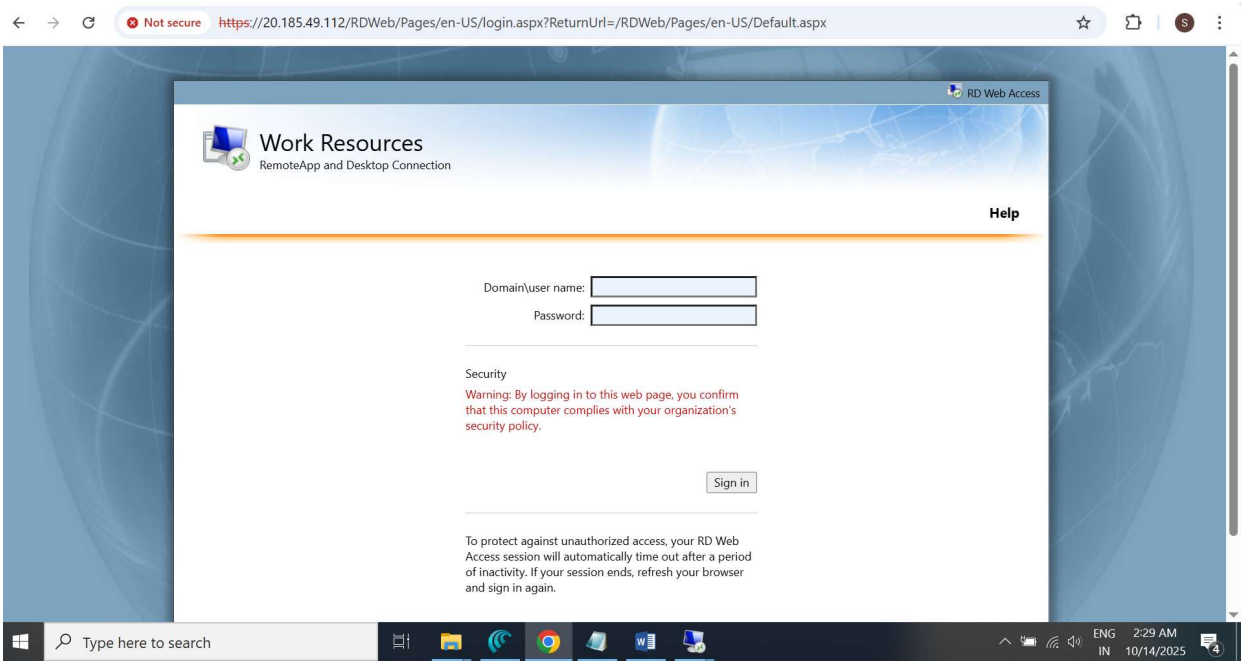
    - *World Wide Web Publishing Service (W3SVC)*
    - *Remote Desktop Gateway (TSGateway)*

3. Wait until **Status** = **Running**.



- **STEP 9 — Test the RD Web Portal**

1. Open a browser on the same server or remote client.
2. Navigate to
3. `https://20.185.49.112/RDWeb`
4. If it shows a certificate warning → click **Continue to site** (it's self-signed).
5. You should see **RD Web Access login page**.


Not secure
https://20.185.49.112/RDWeb/Pages/en-US/login.aspx?ReturnUrl=/RDWeb/Pages/en-US/Default.aspx
RD Web Access
Work Resources
RemoteApp and Desktop Connection
Help
Domain\user name:
Password:
Security
Warning: By logging in to this web page, you confirm that this computer complies with your organization's security policy.
Sign in
To protect against unauthorized access, your RD Web Access session will automatically time out after a period of inactivity. If your session ends, refresh your browser and sign in again.
Type here to search
ENG
IN
2:29 AM
10/14/2025

- LOGIN →

Task IV — Setup Storage Solutions & Verification

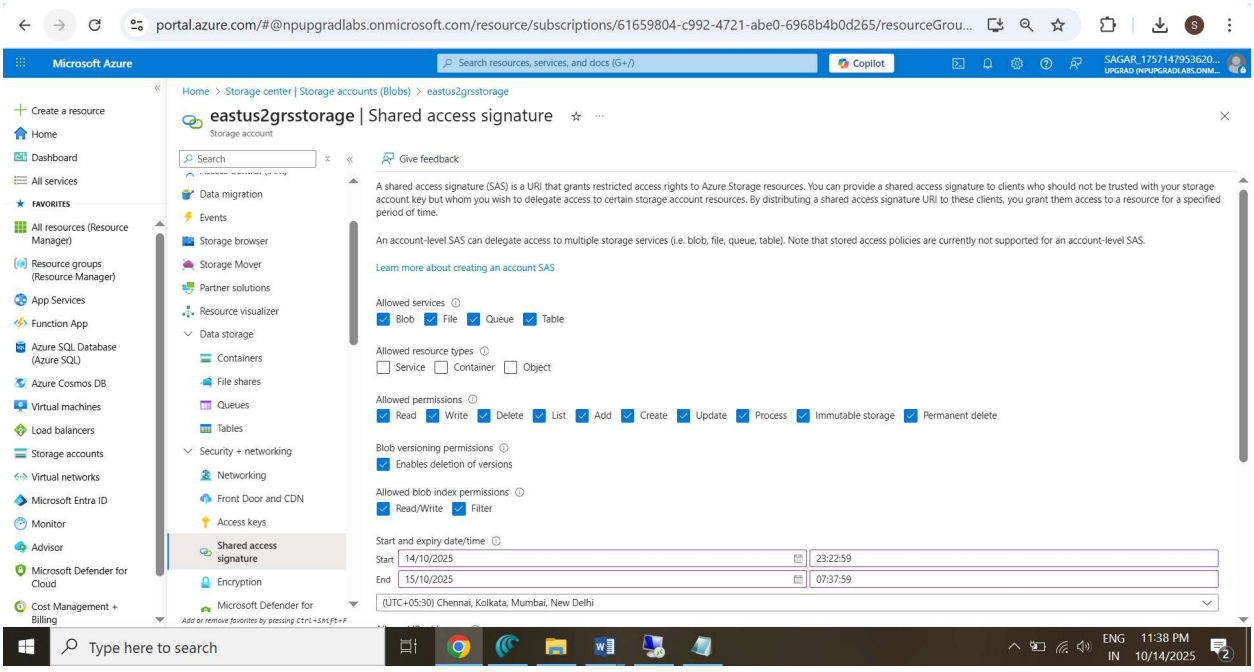
Objective

Deploy resilient storage, map file share, and verify connectivity.

Steps

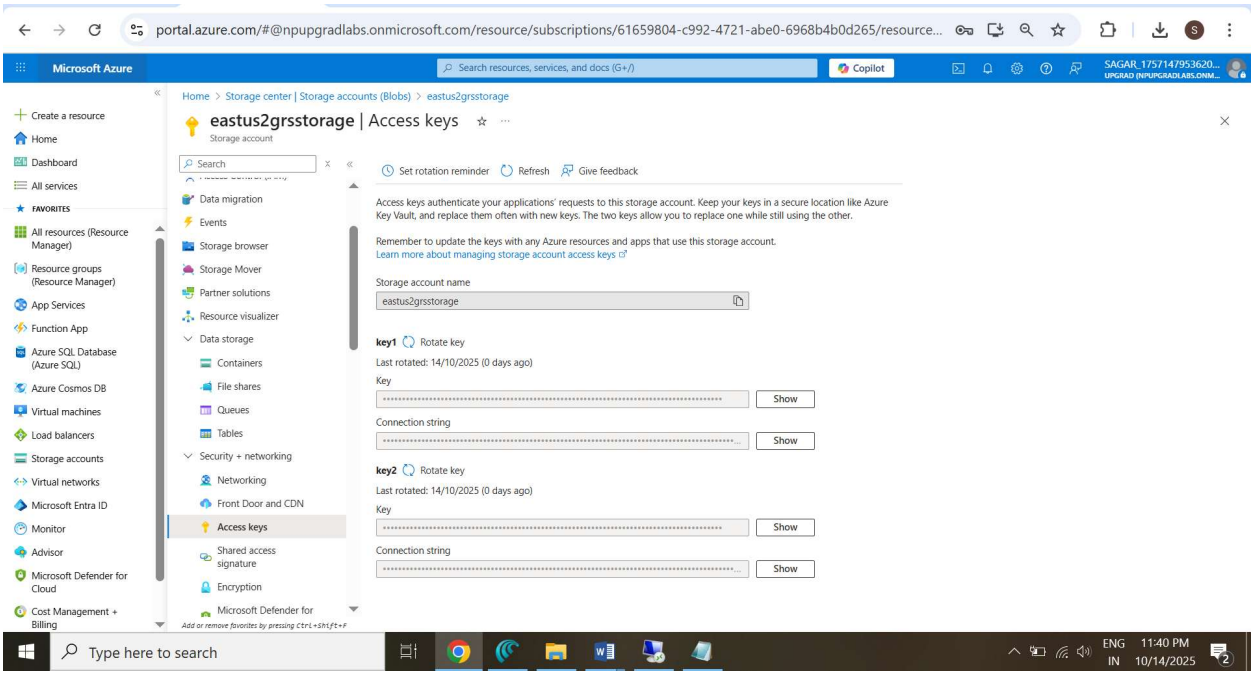
- Shared Access Signature (SAS)

- **SAS** = temporary, restricted access token generated from your storage account.



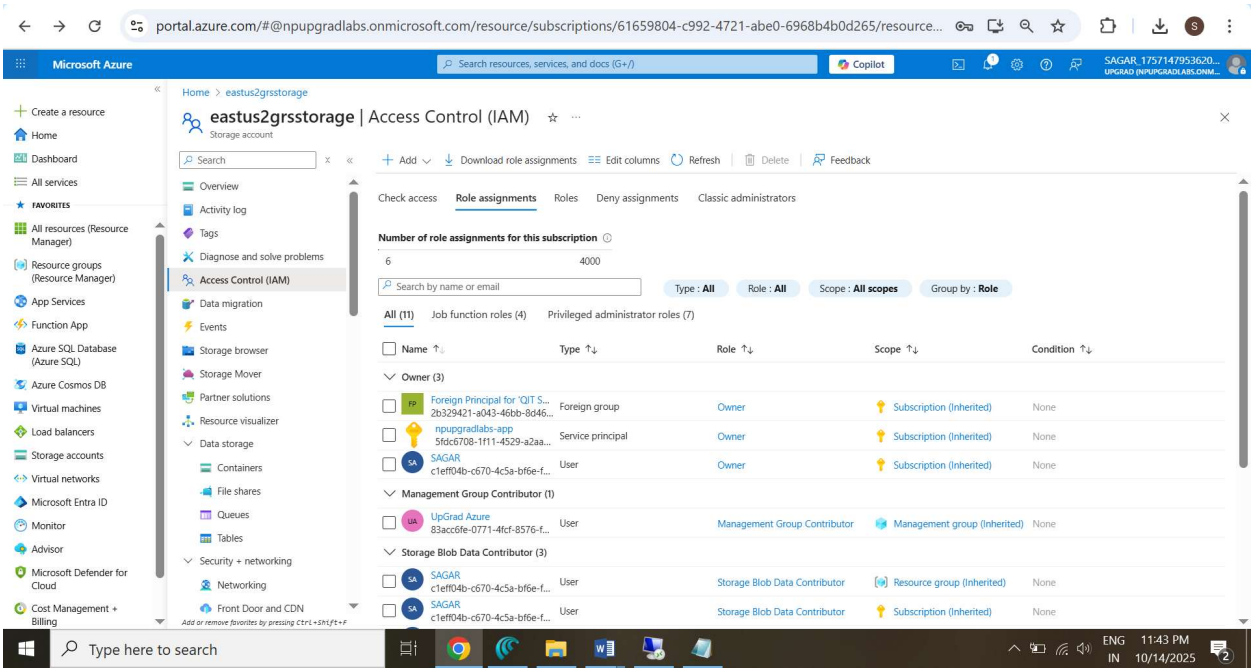
- Access Keys Verification

- **Access keys** = full admin access for storage



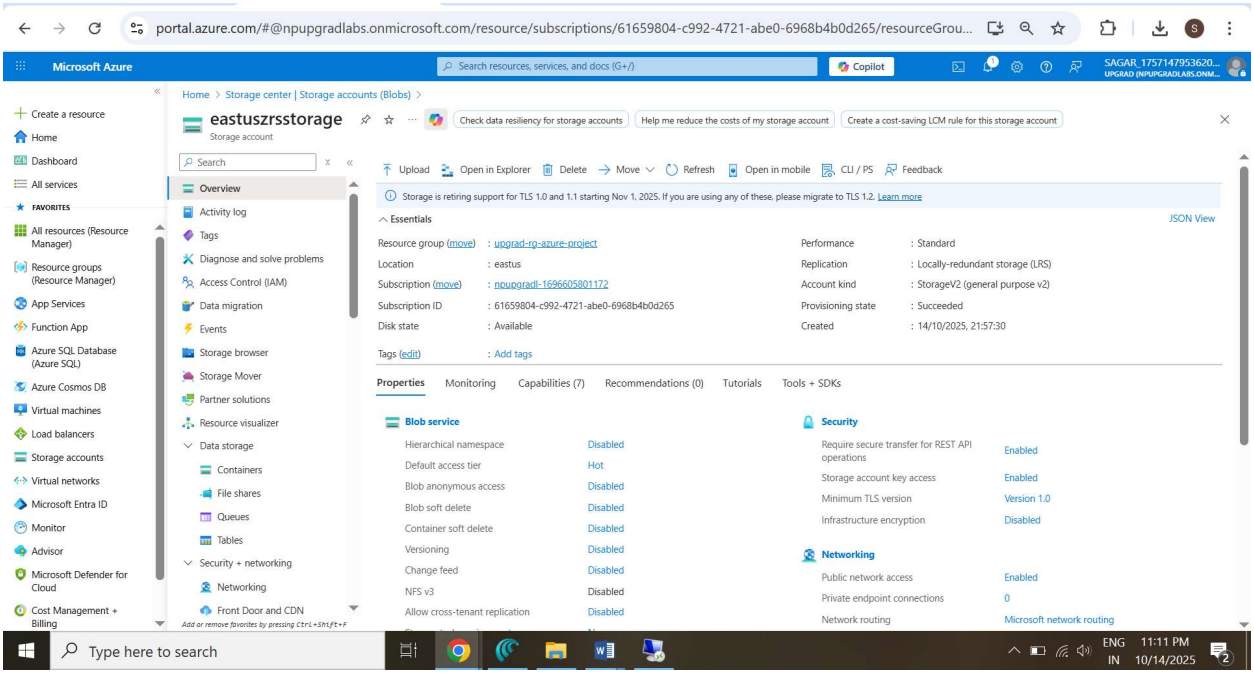
- Role-Based Access Control (RBAC) Verification

- **RBAC** = identity-based access via Azure AD roles.

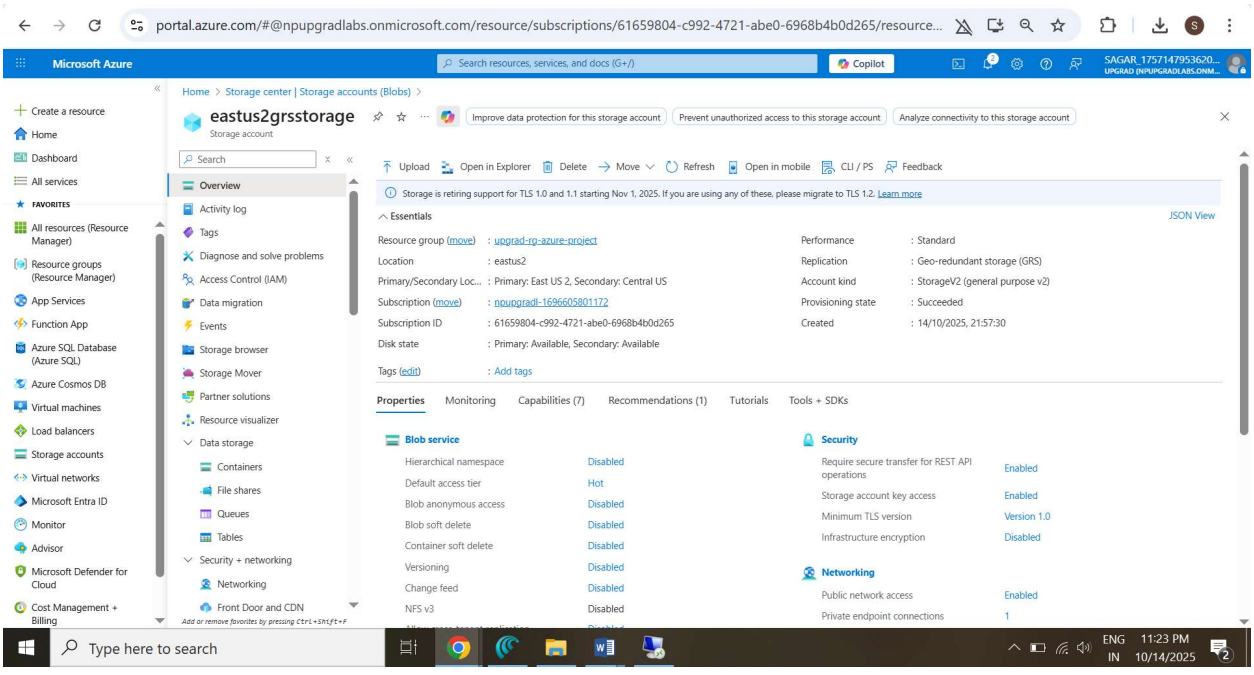


1. Verify Storage Accounts

- `eastuszrsstorage` → SKU = LRS.

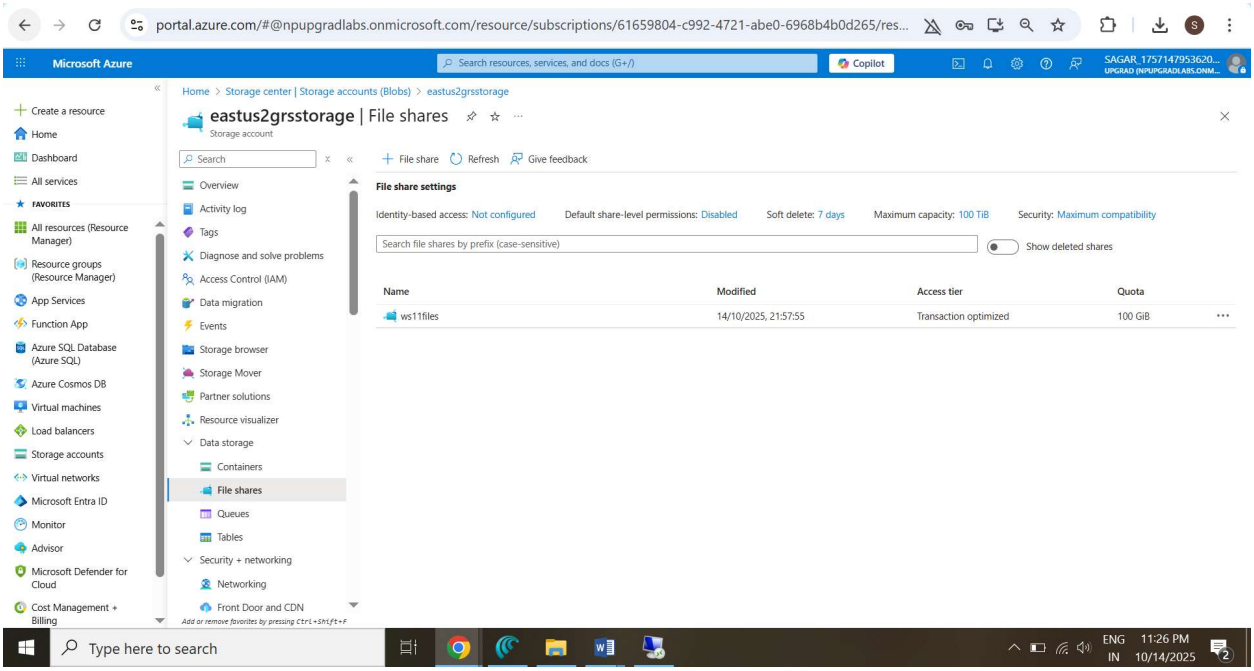


- `eastus2grsstorage` → SKU = GRS.



2. Create File Share

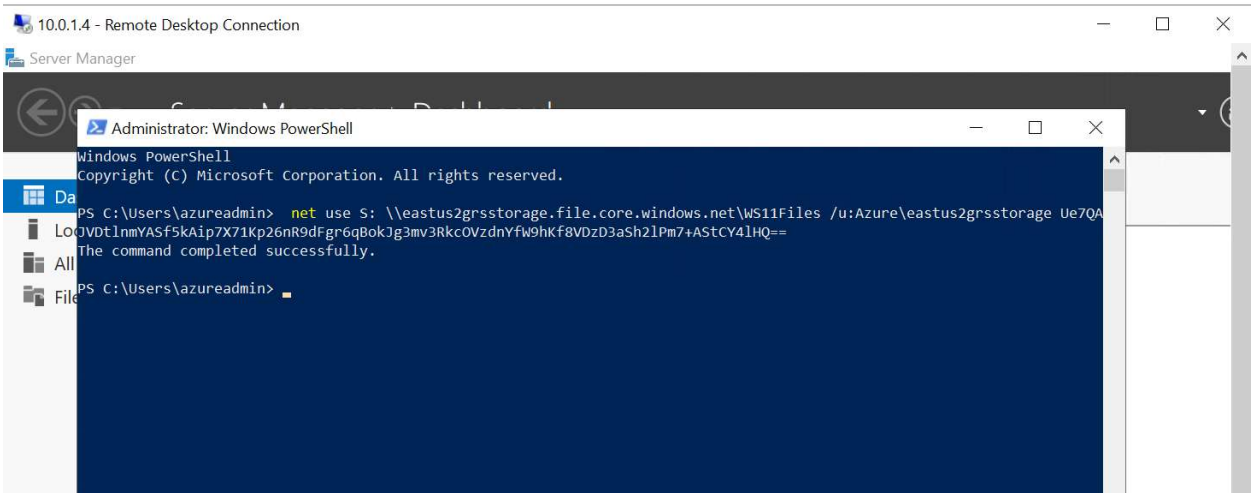
- In `eastus2grsstorage` → File shares → `WS11Files`.



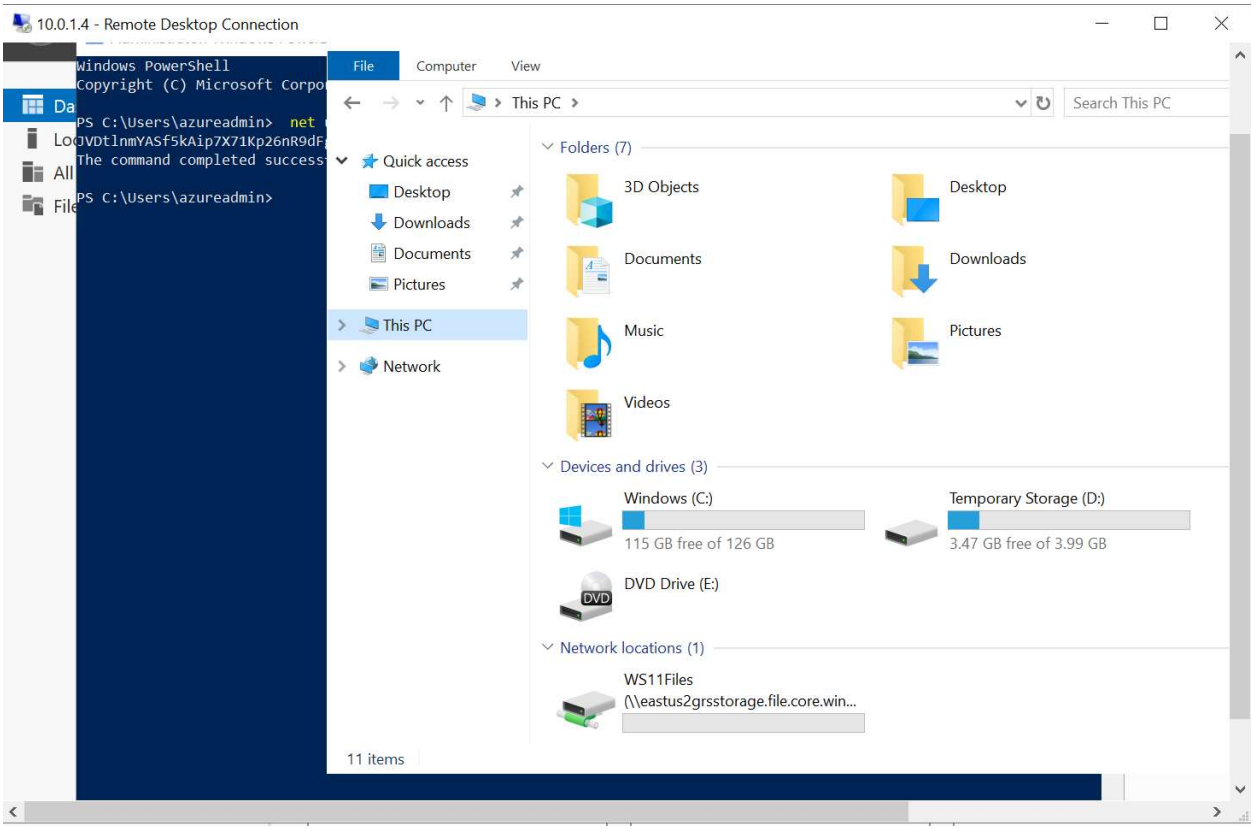
3. Map S: Drive on WS11

- **RDP** to `RDGateway` → connect to **WS11.**
- Run **PowerShell** →

```
net use S: \\eastus2grsstorage.file.core.windows.net\WS11Files /u:Azure\eastus2grsstorage Ue7QAJVDtlnmYASf5kAip7X71Kp26nR9dFgr6qBokJg3mv3RkcOVzdnYfW9hKf8VDzD3aSh2lPm7+AStCY4lHQ==
```

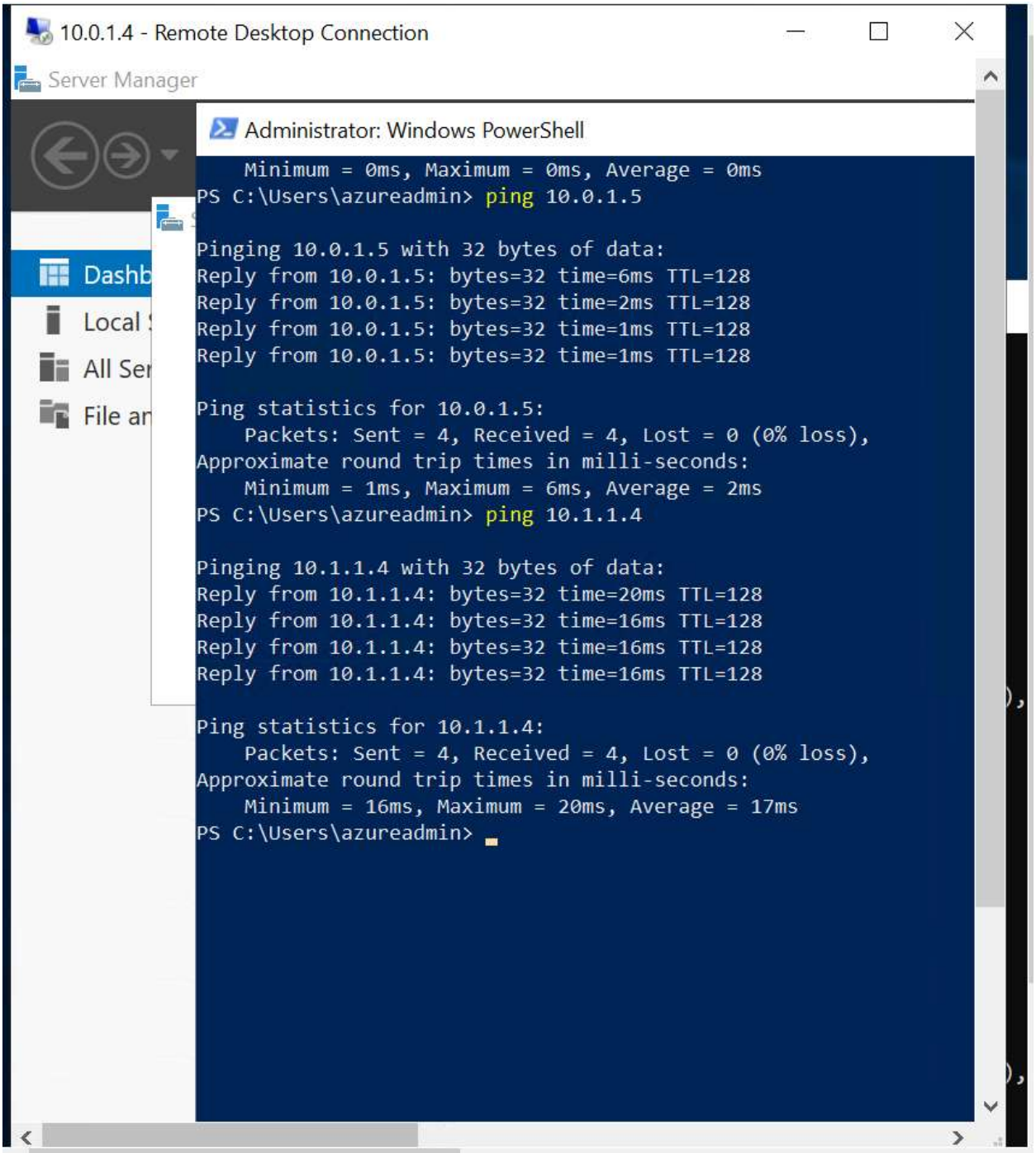


- Verify in File Explorer → **`S:` drive.**

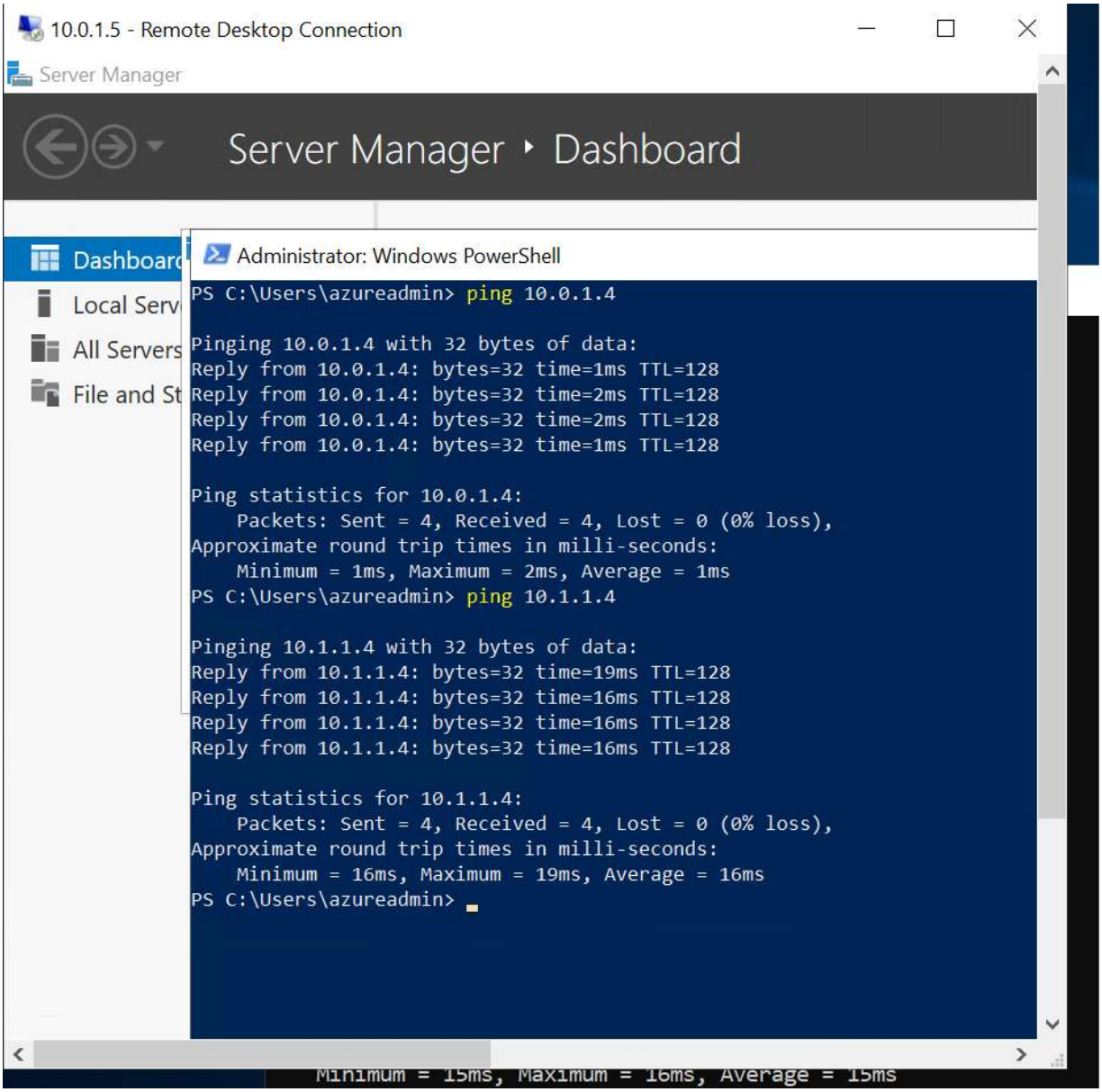


4. **Connectivity Tests**
    - **Ping between `w1`, `w2`, and `WS11`.**

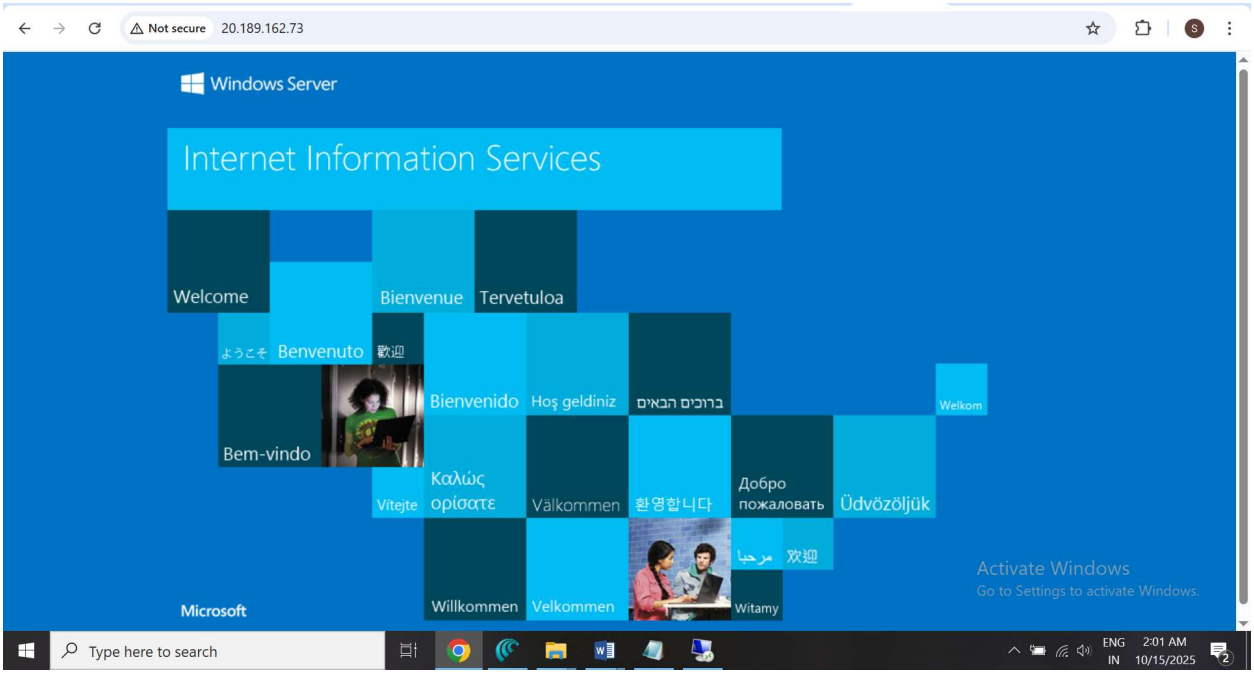
From W1 →



From W2 →

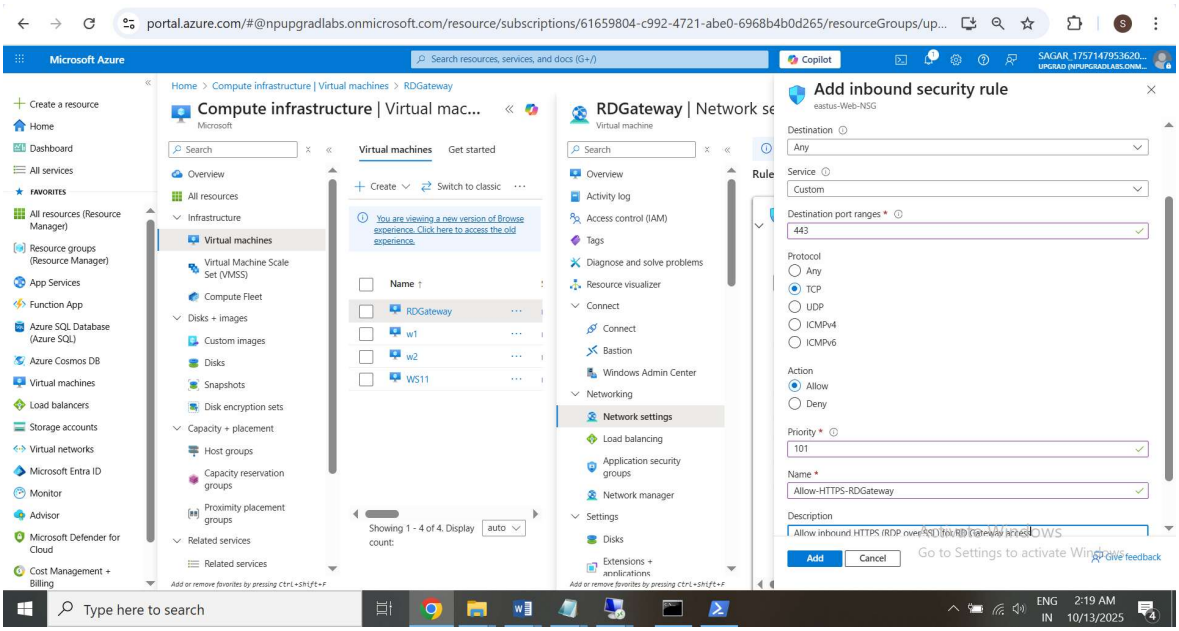

10.0.1.5 - Remote Desktop Connection
Server Manager
Server Manager ‣ Dashboard
Dashboard
Local Serv
All Servers
File and St
Administrator: Windows PowerShell
PS C:\Users\azureadmin> ping 10.0.1.4
Pinging 10.0.1.4 with 32 bytes of data:
Reply from 10.0.1.4: bytes=32 time=1ms TTL=128
Reply from 10.0.1.4: bytes=32 time=2ms TTL=128
Reply from 10.0.1.4: bytes=32 time=2ms TTL=128
Reply from 10.0.1.4: bytes=32 time=1ms TTL=128
Ping statistics for 10.0.1.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 1ms, Maximum = 2ms, Average = 1ms
PS C:\Users\azureadmin> ping 10.1.1.4
Pinging 10.1.1.4 with 32 bytes of data:
Reply from 10.1.1.4: bytes=32 time=19ms TTL=128
Reply from 10.1.1.4: bytes=32 time=16ms TTL=128
Reply from 10.1.1.4: bytes=32 time=16ms TTL=128
Reply from 10.1.1.4: bytes=32 time=16ms TTL=128
Ping statistics for 10.1.1.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 16ms, Maximum = 19ms, Average = 16ms
PS C:\Users\azureadmin>
Minimum = 15ms, Maximum = 16ms, Average = 15ms

- Access **Load Balancer HTTP** from browser.



□ Verification

Web page accessible



Visual Diagram:

- **Project Architecture Diagram→**