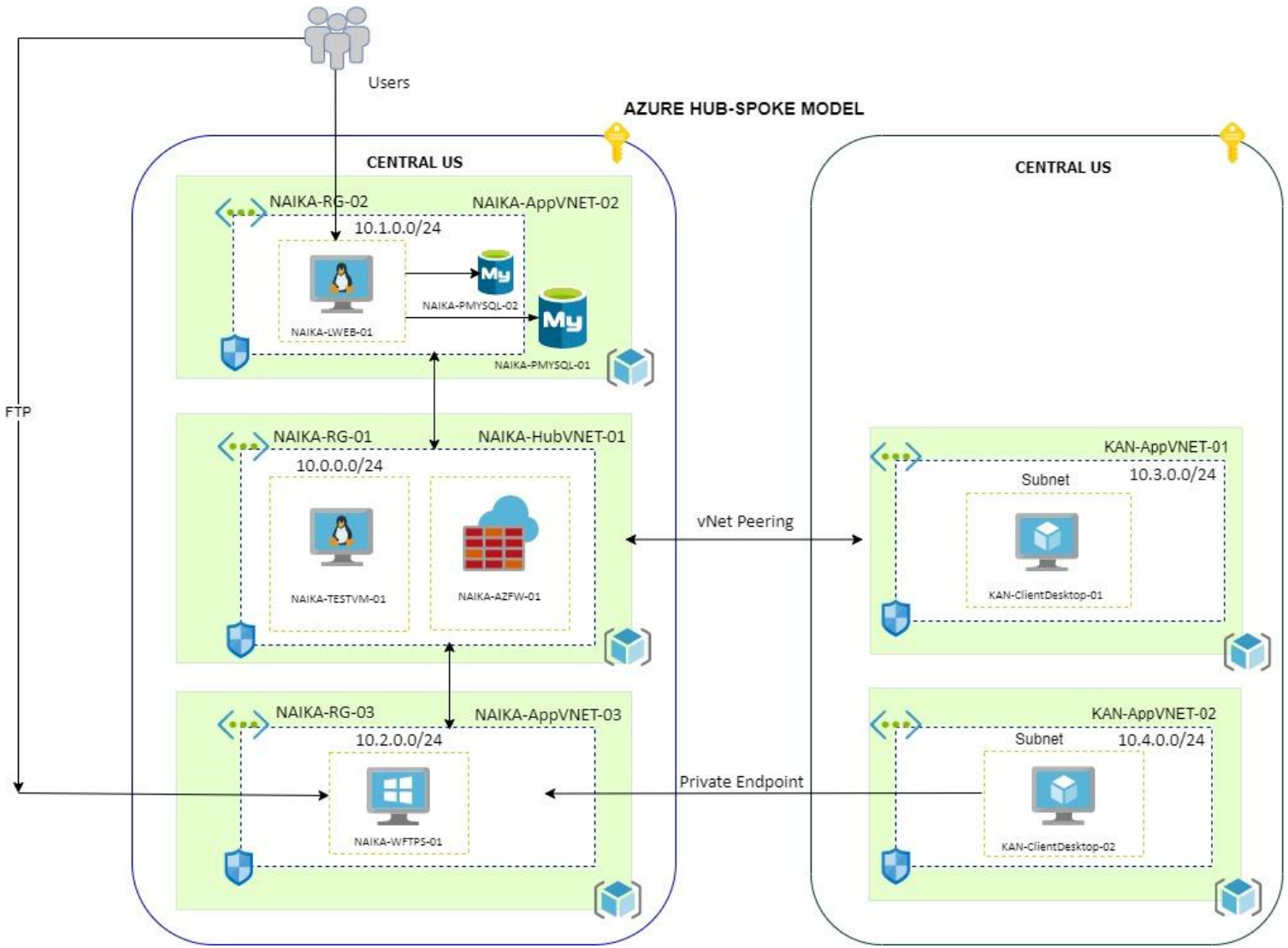
LAB CONFIGURATION – CAPSTONE PROJECT

Group A
Nitika Gupta - 8788824
Kanan Garach - 7923477
Anjali Kumari - 8731767
Thanh Trung Nguyen - 8635130

AZURE FOUNDATION DESIGN

AZURE CLOUD HUB-SPOKE MODEL



Azure Services Used

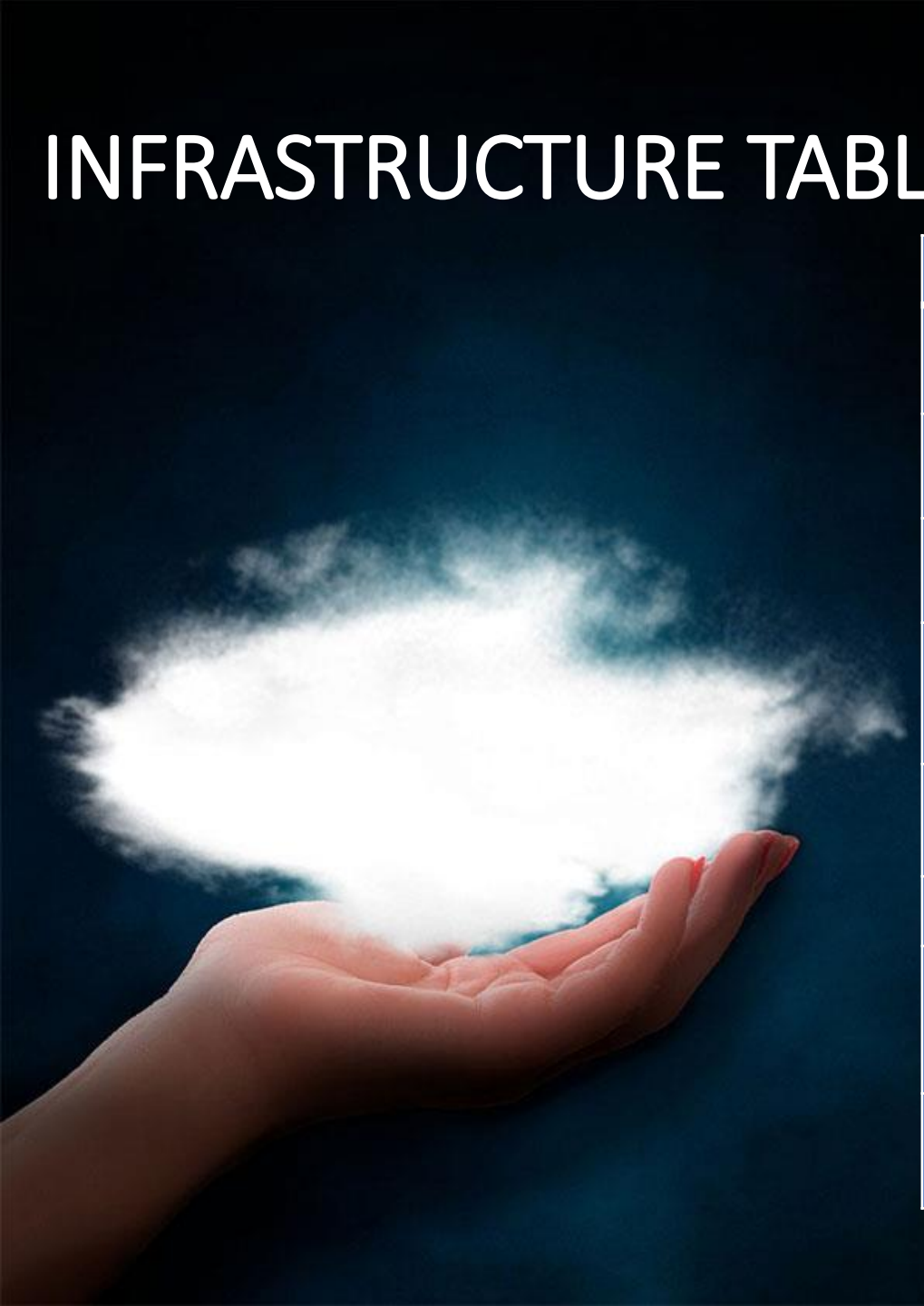
- Subscriptions
- Resource Groups
- vNets and Subnets
- Network Security Groups
- vNet Peering
- Azure Firewall
- Azure MySQL Database
- Azure Virtual Machines
- Azure Storage Containers
- Private Endpoint

INFRASTRUCTURE DETAILS

NETWORK TABLE

Resource Group	vNet Name	IP Address Space	Subnet Name	Subnet IP Address Space
AZURE SUBSCRIPTION 1				
NAIKA-RG-01	NAIKA-HubVNET-01	10.0.0.0/16	default	10.0.0.0/24
NAIKA-RG-02	NAIKA-AppVNET-02	10.1.0.0/16	default	10.1.0.0/24
NAIKA-RG-03	NAIKA-AppVNET-03	10.2.0.0/16	default	10.2.0.0/24
AZURE SUBSCRIPTION 2				
KAN-RG-01	KAN-AppVNET-01	10.3.0.0/16	default	10.3.0.0/24
KAN-RG-02	KAN-AppVNET-02	10.4.0.0/16	default	10.4.0.0/24

INFRASTRUCTURE TABLE



Resource Group	Server Role	Server Name	IP Address	Operating System	Application Software
NAIKA-RG-03	FTP Server (Windows)	NAIKA-WFTPS-01	10.2.0.4/24	Windows server 2019 Std. Edition	FTP
NAIKA-RG-02	MySQL Server	NAIKA-PMYSQL-01	N/A	Windows server 2019 Std. Edition	MySQL Server 8
NAIKA-RG-02	MySQL Server	NAIKA-PMYSQL-02	N/A	Windows server 2019 Std. Edition	MySQL Server 8
KAN-RG-01	Windows Client / User machine	KAN-ClientDesktop-01	10.3.0.4/24	Windows 10 Pro	MySQL Workbench, Firefox, FileZilla, Putty
KAN-RG-02	Windows Client / User machine	KAN-ClientDesktop-02	10.4.0.4/24	Windows 10 Pro	Firefox
NAIKA-RG-01	Azure Firewall	NAIKA-AZFW-01	10.0.1.4/24	NA	NA
NAIKA-RG-01	Client Desktop	NAIKA-TestVM1	10.0.0.4/24	Windows 10 Pro	MySQL Workbench, FileZilla, Firefox, Putty
NAIKA-RG-03	Azure Storage Container	naikastr01	N/A	N/A	N/A

vNET Peering in same AD Tenant between Hub and Spoke – Azure Network Information

The below table shows the network details of Hub and Spoke networks which will be vNet Peered.

	HUB	SPOKE 1	SPOKE 2
vNET Name	NAIKA-HubVNET-01	NAIKA-AppVNET-02	NAIKA-AppVNET-03
Address space	10.0.0.0/16	10.1.0.0/16	10.2.0.0/16
Subnet name	Default	Default	Default
Subnet address range	10.0.0.0/24	10.1.0.0/24	10.2.0.0/24
Subscription	Azure Subscription 1	Azure Subscription 1	Azure Subscription 1
Resource group	NAIKA-RG-01	NAIKA-RG-02	NAIKA-RG-03
Location	Central US	Central US	Central US

vNET Peering in same AD Tenant between Hub and Spoke

vNet Peering is setup as:

- HubVNET-01 - peered to – AppVNET-02
- HubVNET-01 - peered to – AppVNET-03
- AppVNET-02- peered to – HubVNET-01
- AppVNET-03- peered to – HubVNET-01

vNet Peering Between Hub vNet and AppVNET-03

- Peering link name: vNetPeer-Hub-App03
- Remote Peering Link Name: vNetPeer-App03-hub
- Remote Virtual Network: NAIKA-AppVNET-03

vNet Peering Between Hub vNet and AppVNET-02

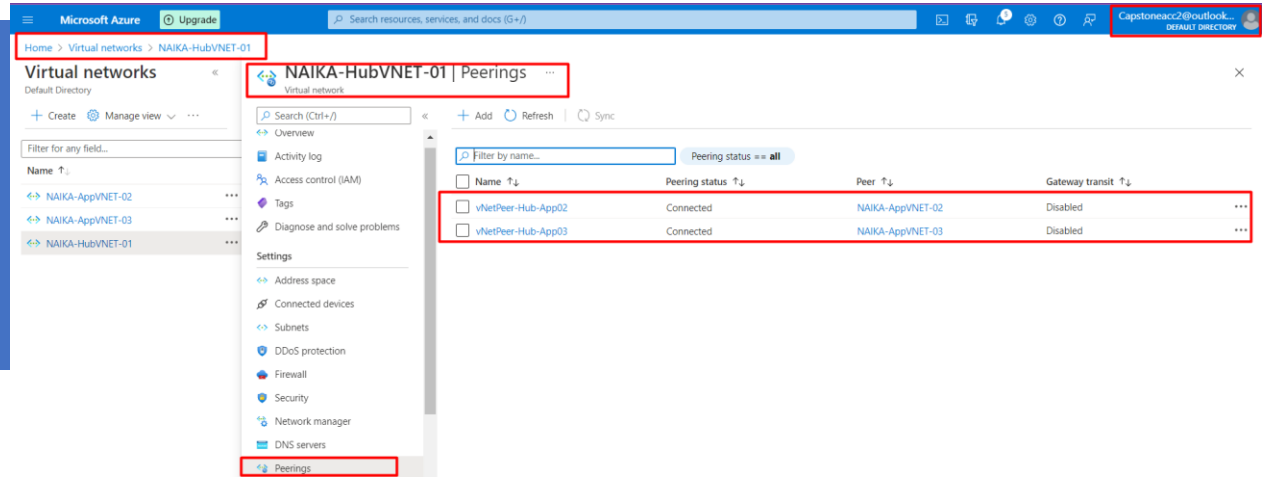
- Peering link name: vNetPeer-Hub-App02
- Remote Peering Link Name: vNetPeer-App02-hub
- Remote Virtual Network: NAIKA-AppVNET-02

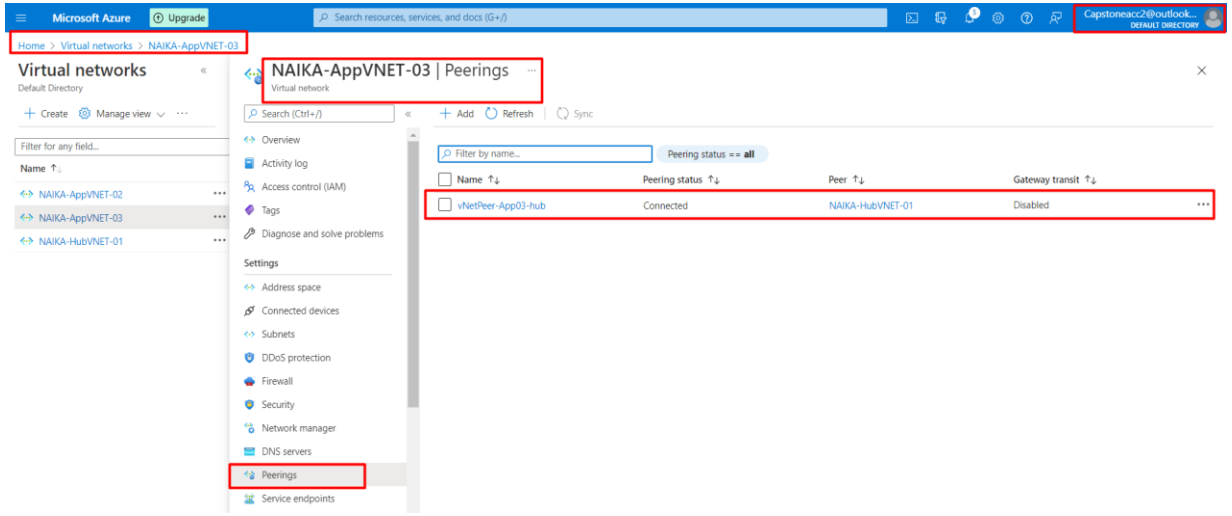
- Peering is supported for non overlapping IP address space.
- Table in Slide 6 shows the network details and CIDR used for setting up vNet peering between Hub and Spoke
- In peered networks, the resources communicated with each other with the same latency as they were in same vLAN / virtual network. This solution is helpful to connect latency sensitive applications across or within subscriptions.
- “Connected” state of vNet peering shows successful connection establishment.
- The subscription where peering resource is created must also be registered with the Microsoft network resource provider.

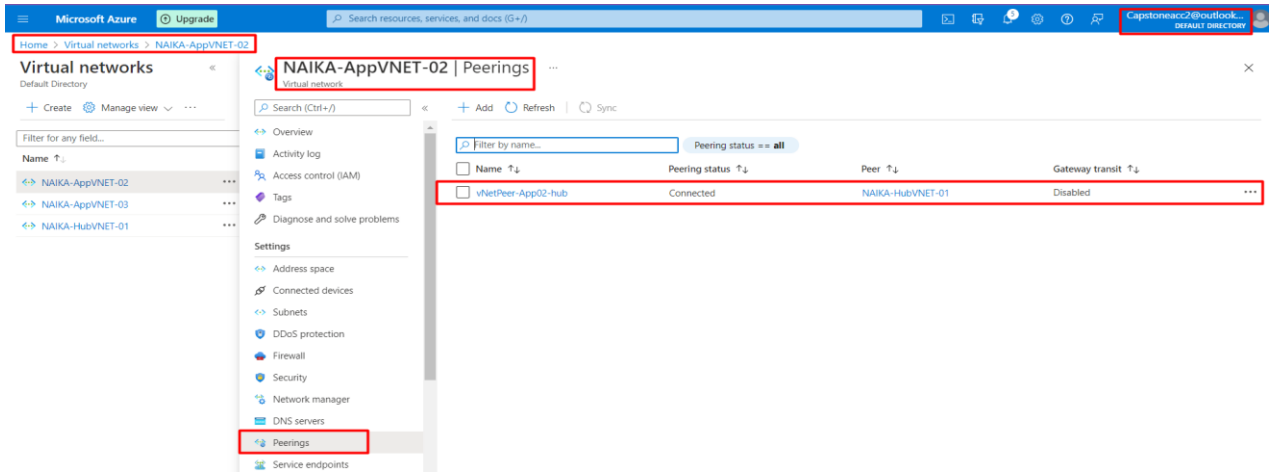
Bi-directional peering is successfully setup. Connectivity Testing was done using RDP protocol over private network between two subscriptions after updating the NSG rules:

- Disassociate public IP from KAN-ClientDesktop-01 machines
- RDP from TestVM1 in NAIKA domain to KAN-ClientDesktop-01 in KAN domain - Successful

vNET Peering in same AD Tenant between Hub and Spoke







NSG rules created on Virtual Machine in Hub vNet for testing across Networks

Home > TestVM1

TestVM1 | Networking

Virtual machine

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Networking
- Connect
- Disks
- Size
- Security
- Advisor recommendations
- Extensions + applications
- Continuous delivery
- Availability + scaling
- Configuration
- Identity
- Properties
- Locks

Operations

- Bastion
- Auto-shutdown
- Backup
- Disaster recovery

Attach network interface Detach network interface

Network Interface: testvm130 Effective security rules Troubleshoot VM connection issues Topology

Virtual network/subnet: NAIKA-HubVNET-01/default NIC Public IP: **40.122.198.15** NIC Private IP: **10.0.0.5** Accelerated networking: **Disabled**

Inbound port rules Outbound port rules Application security groups Load balancing

Network security group TestVM1-nsg (attached to subnet: default)
Impacts 1 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
300	RDP	3389	TCP	Any	Any	Allow
310	Port_80	80	TCP	Any	Any	Allow
320	azuredns	Any	Any	168.63.129.16	VirtualNetwork	Allow
330	ICMP_IN	Any	ICMP	10.3.0.0/24,40.86.15.71	10.0.0.5	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Network security group TestVM1-nsg (attached to network interface: testvm130)
Impacts 1 subnets, 1 network interfaces

Add inbound port rule

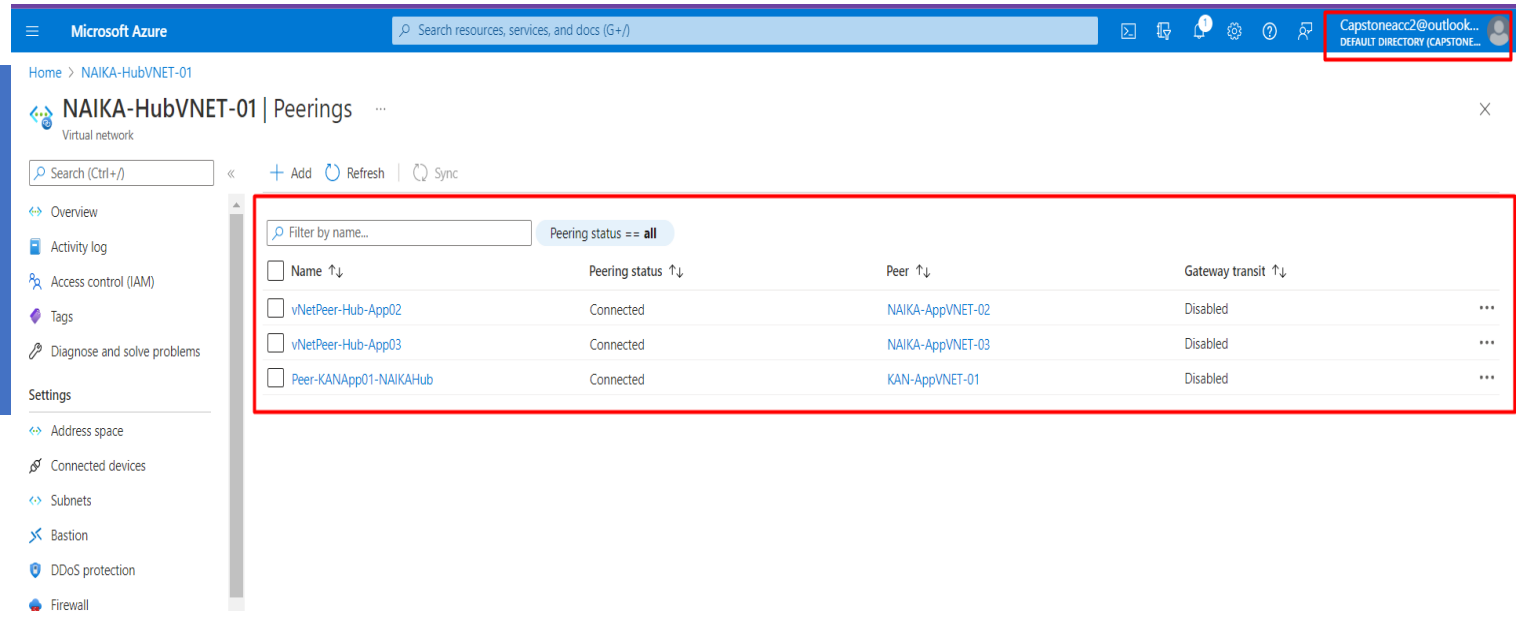
Priority	Name	Port	Protocol	Source	Destination	Action
300	RDP	3389	TCP	Any	Any	Allow
310	Port_80	80	TCP	Any	Any	Allow
320	azuredns	Any	Any	168.63.129.16	VirtualNetwork	Allow
330	ICMP_IN	Any	ICMP	10.3.0.0/24,40.86.15.71	10.0.0.5	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

vNet Peering between different AD tenants – Azure Account information

The below table shows the network details across two AD tenants which will be vNet Peered.

	AD TENANT 1	AD TENANT 2
vNET Name	NAIKA-HubVNET-01	KAN-AppVNET-01
Address space	10.0.0.0/16	10.3.0.0/16
Subnet name	Default	Default
Subnet address range	10.0.0.0/24	10.3.0.0/24
Subscription	Azure Subscription 1	Azure Subscription 1
Resource group	NAIKA-RG-01	KAN-RG-01
Location	Central US	Central US

Cross AD Tenant vNet Peering



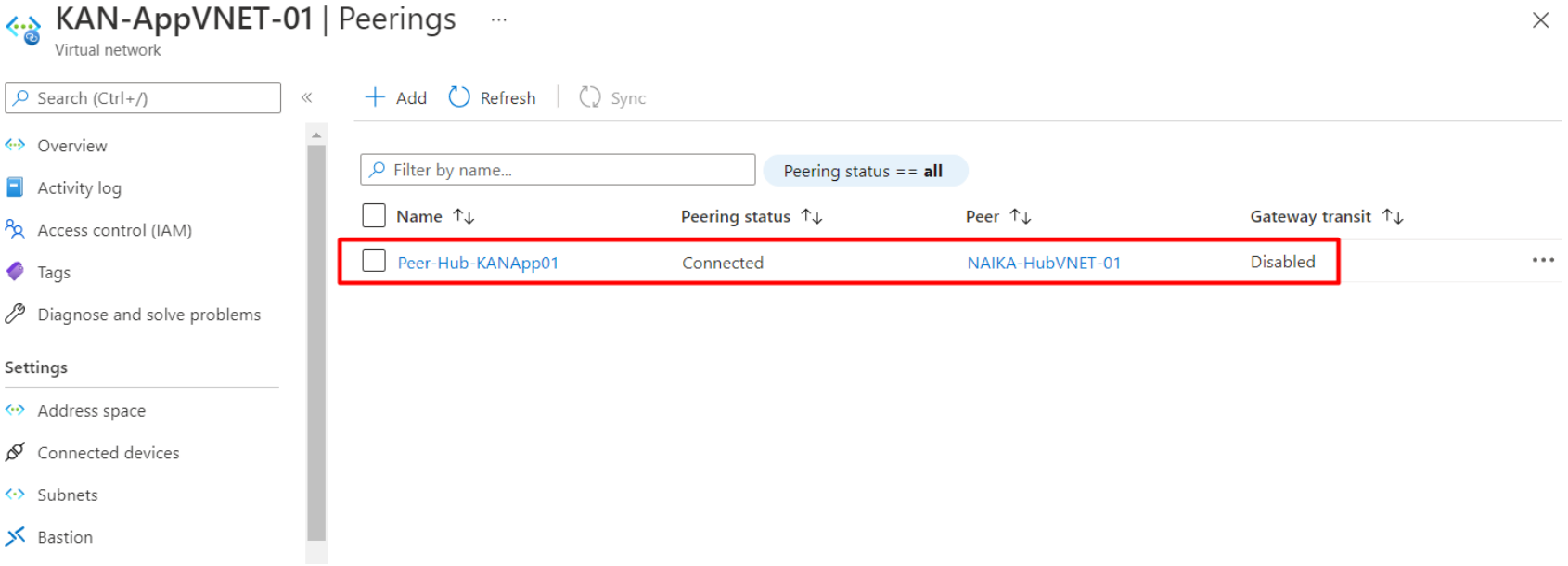
vNet Peering from NAIKA to KAN is setup as:

- **NAIKA-HubVNET-01 - peered to – KAN-AppVNET-02 (Capstoneacc2 to Syst8200Proj)**

- **Peering across AD tenants is unidirectional, so must be set from both accounts to be peered if bi-directional connectivity is required. Cross-AD vNet Peering requires authorization across the AD tenants.**

Resource ID used for peering:
KAN-AppVNET-01 resource ID: ***/subscriptions/45459a97-366d-479e-871b-c82607eda9c0/resourceGroups/KAN-RG-01/providers/Microsoft.Network/virtualNetworks/KAN-AppVNET-01***

Cross AD Tenant vNet Peering Contd…

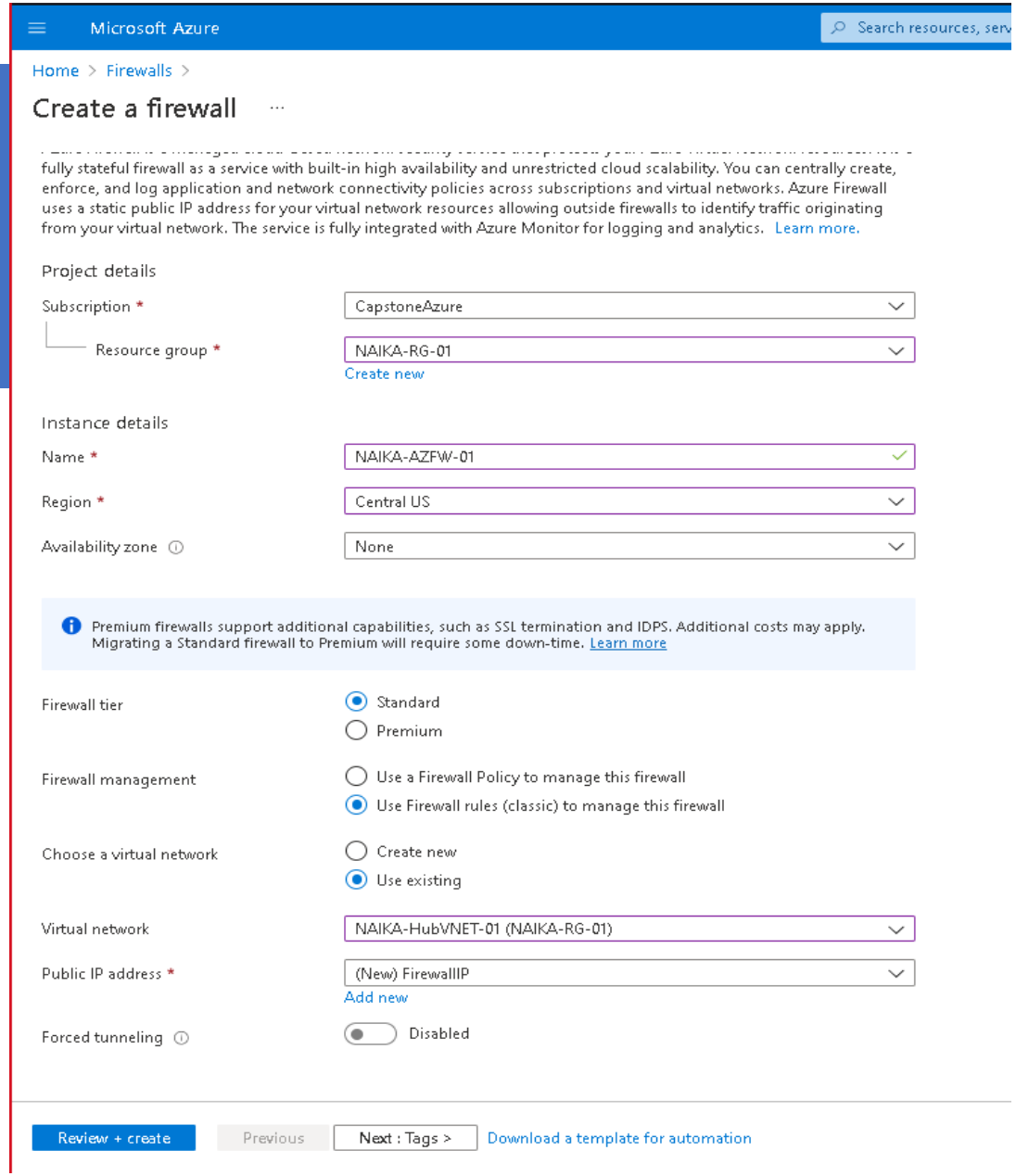


vNet Peering from KAN to NAIKA is setup as:

- **KAN-AppVNET-02- peered to – NAIKA-HubVNET-01 (Syst8200Proj to Capstoneacc2)**
- **Network Contributor Role are required to the accounts across the AD tenants with which peering is established.**

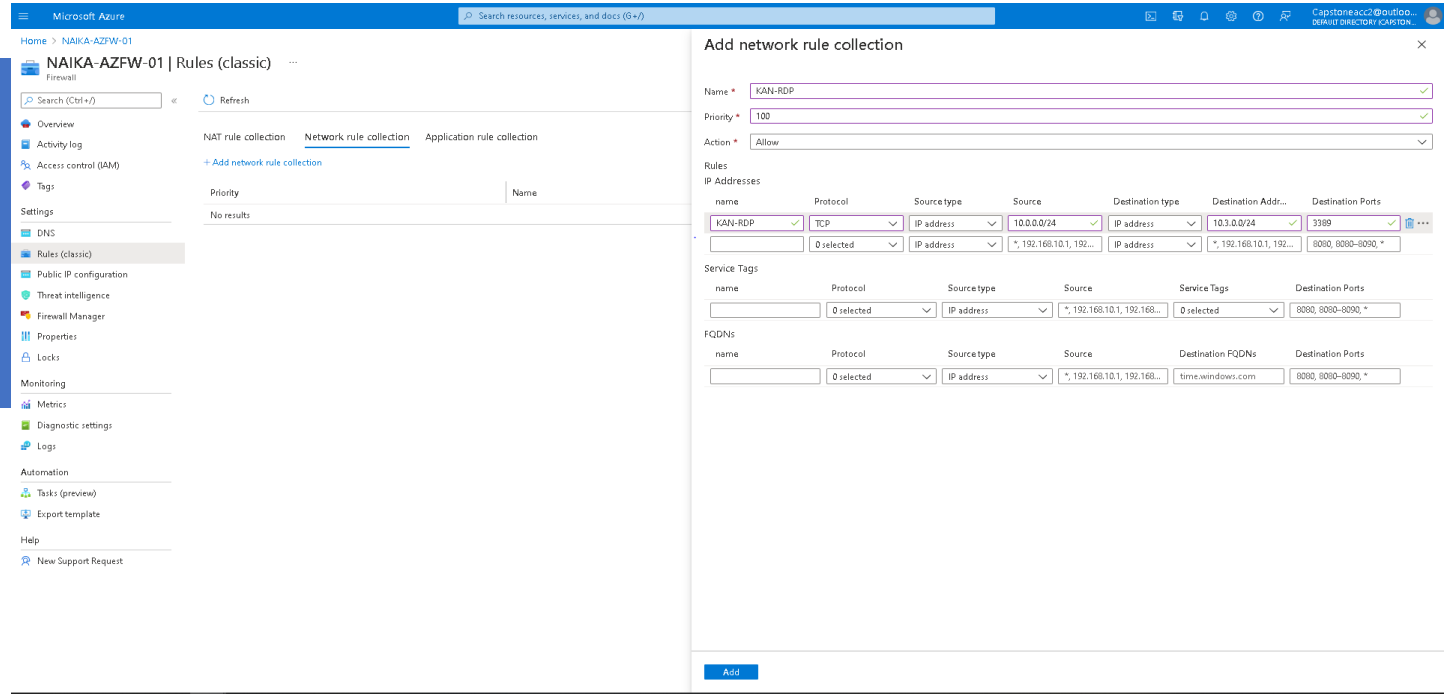
Resource ID used for peering:
NAIKA-HubVNET-01 Resource ID: ***/subscriptions/e27b2238-1465-409d-9d2a-824d64998a1f/resourceGroups/NAIKA-RG-01/providers/Microsoft.Network/virtualNetworks/NAIKA-HubVNET-01***

Azure Firewall Setup Between Peered Networks



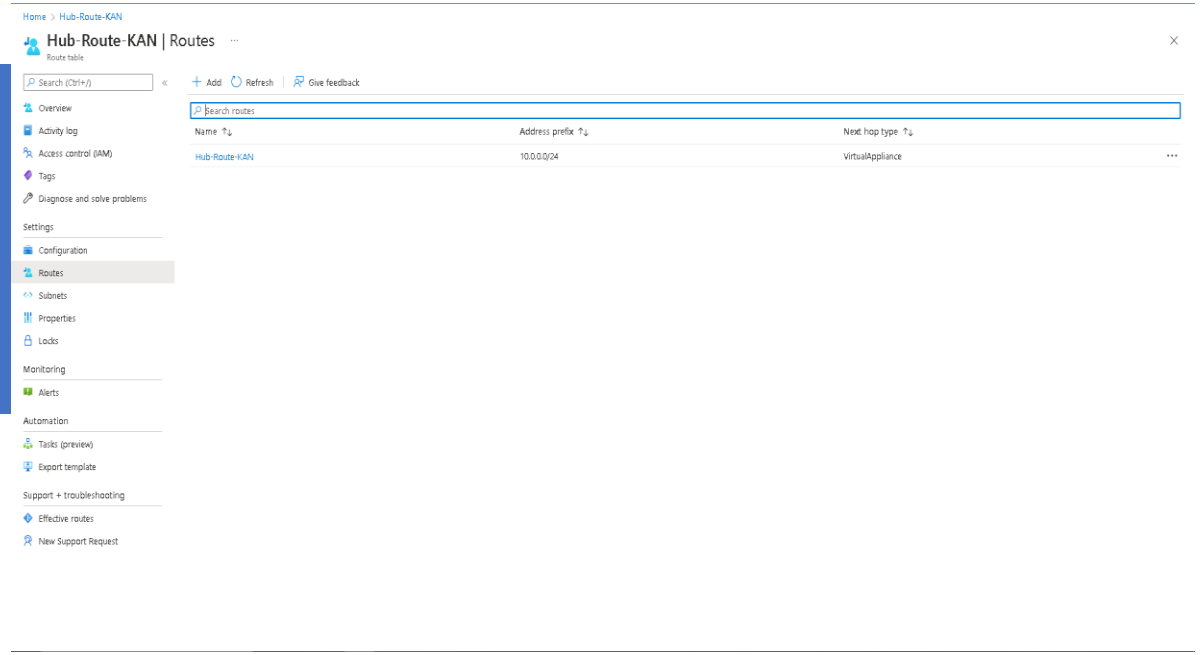
- Azure Firewall is used in this solution to secure the private connectivity between two AD tenants. Security is paramount in cloud and connections from a third-party network are not secure by default. Hence, solutions like Azure Firewall are used.
- Firewall has been created in the NAIKA-RG-01 resource group and has a Public IP assigned to it from the Firewall IP resource
- Firewall service resource and all associated components have been built in the Central US region where other resources reside

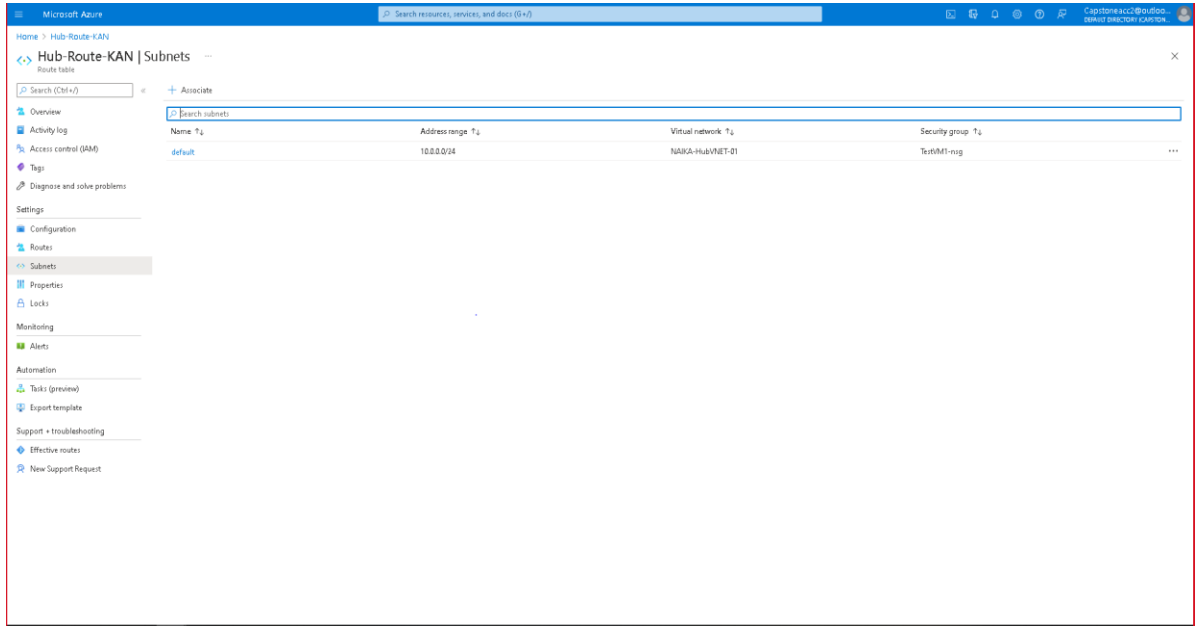
Azure Firewall Setup Between Peered Networks



- Network rule is configured during the Firewall creation to allow the connectivity between source and target network over permissible port and protocol
- In this configuration, we have set up RDP connection over NAIKA (10.0.0.0/24) and KAN (10.3.0.0/24) domain, securing it through Azure Firewall

Azure Firewall Setup Between Peered Networks





- In order to configure the Azure Firewall to route using the designated virtual networks, a Route within the Route Table service has been created
- This Route entry has been propagated with the designated source subnet and next hop type to designate the traffic routing
  - Next hop type set as Virtual Appliance(Firewall)
  - From the route table, we have associated the NAIKA default subnet so that the Firewall is the next hop for NAIKA
- After the Firewall setup, NSG rules for ClientDesktop-01 in KAN network were updated to allow RDP from NAIKA network via the Azure Firewall
  - These rules are captured in the subsequent slide (16)

Client Desktop (KAN) NSG rules to allow traffic from NAIKA network after Peering

Microsoft Azure | Search resources, services, and docs (G+/) | Syst8200Proj@outlook.c... DEFAULT DIRECTORY (SYST8200...

Home > Recent > KAN-RG-01 > KAN-ClientDesktop-01

KAN-ClientDesktop-01 | Networking

Virtual machine

Search (Ctrl+/)

- Overview
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems

Settings

- Networking
- Connect
- Disks
- Size
- Security
- Advisor recommendations
- Extensions + applications
- Continuous delivery
- Availability + scaling
- Configuration
- Identity
- Properties
- Locks

Operations

- Bastion
- Auto-shutdown
- Backup

Attach network interface | Detach network interface | Feedback

kan-clientdesktop878

IP configuration

ipconfig1 (Primary)

Network Interface: kan-clientdesktop878 | Effective security rules | Troubleshoot VM connection issues | Topology

Virtual network/subnet: KAN-AppVNET-01/KAN-APP-01 | NIC Public IP: - | NIC Private IP: **10.3.0.4** | Accelerated networking: **Disabled**

Inbound port rules | Outbound port rules | Application security groups | Load balancing

Network security group KAN-ClientDesktop-01-nsg (attached to network interface: kan-clientdesktop878)
Impacts 0 subnets, 1 network interfaces

Add inbound port rule

Priority	Name	Port	Protocol	Source	Destination	Action
300	RDP	3389	TCP	Any	Any	Allow
310	ICMP_IN	Any	ICMP	10.0.0.0/24,40.122.198.15	10.3.0.0/24	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

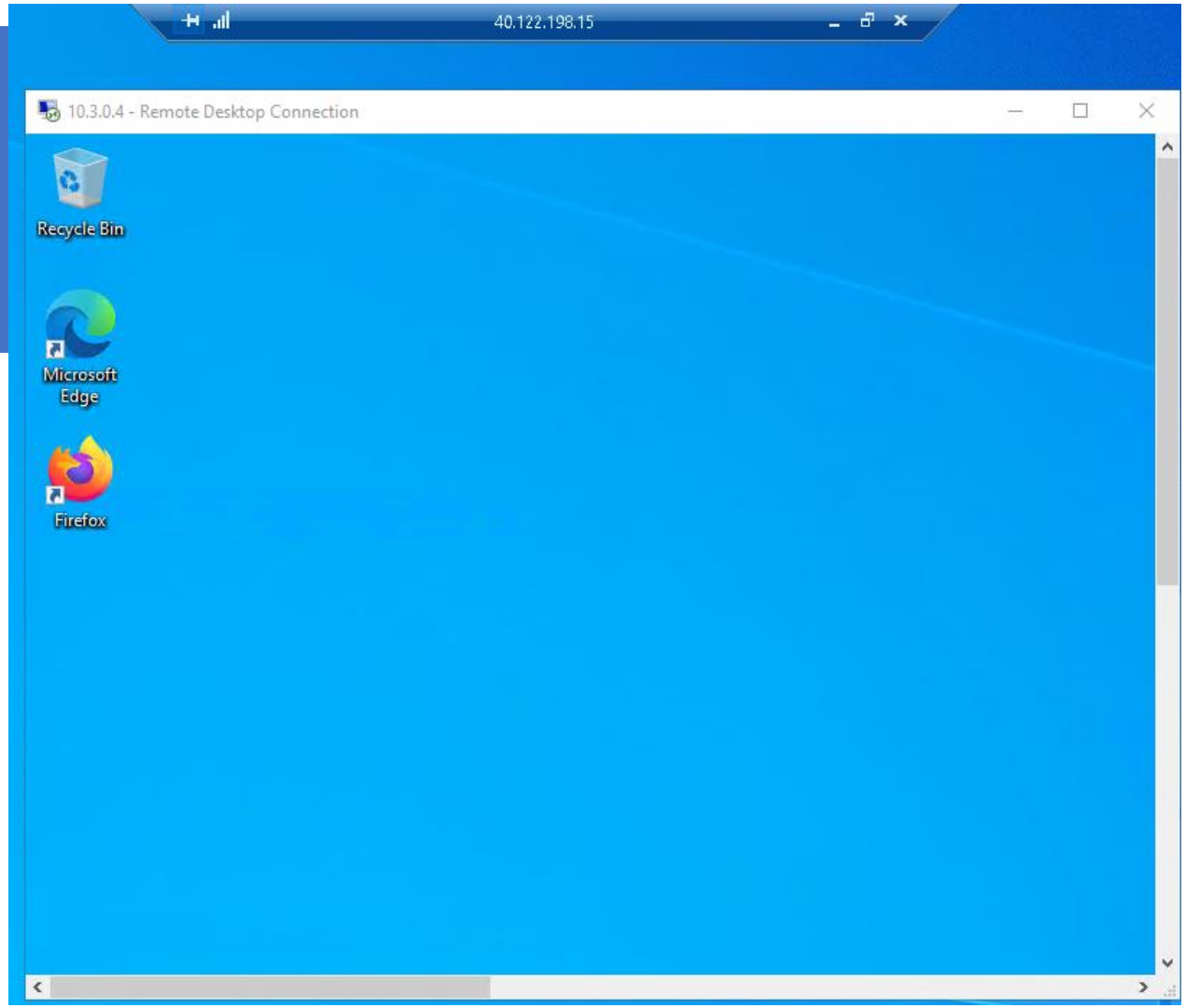
Need help?

Understand Azure load balancing

Quickstart: Create a public load balancer to load balance Virtual Machines

Quickstart: Direct web traffic with Azure Application Gateway

Azure Firewall Connectivity Test

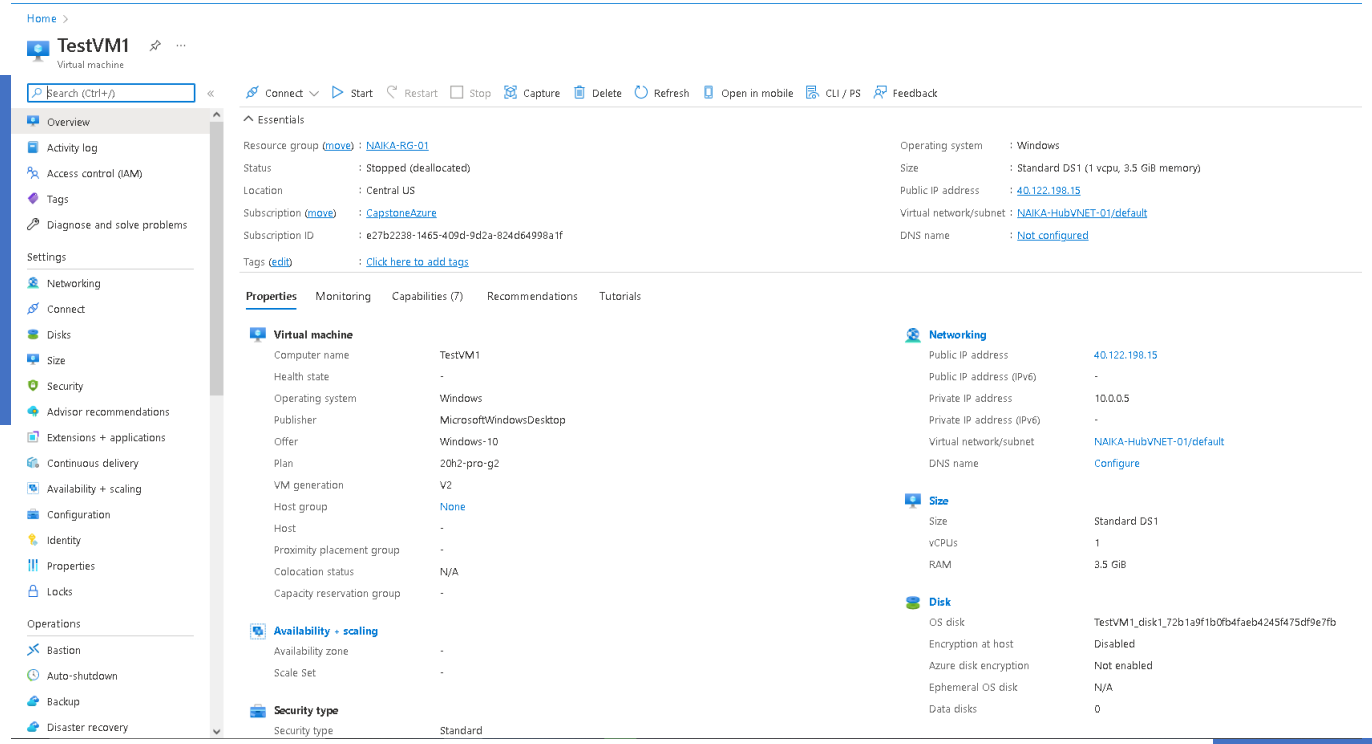


- Firewall connectivity has been tested by establishing a nested remote desktop connection from the NAIKA TestVM1 into the ClientDesktop-01 located in the KAN domain
- After configuration of Network Security Group and Route resources associated with the Firewall, the remote connection will be successful if all was configured correctly
- Image demonstrates successful connection through a Remote Desktop Connection from the NAIKA TESTVM1 via public IPv4 address into the KAN-ClientDesktop-01 through its private IPv4 address

CLIENT DESKTOP TABLE

Client Desktop Name	vNet Name	IP Address Space	Subnet Name	Subnet IP Address Space	Applications Installed
NAIKA-TestVM1	NAIKA-HubVNET-01	10.0.0.0/16	default	10.0.0.0/24	MySQL Workbench, Firefox, Putty, FileZilla
KAN-ClientDesktop-01	NAIKA-AppVNET-02	10.1.0.0/16	default	10.3.0.0/24	MySQL Workbench, Firefox, Putty, FileZilla
KAN-ClientDesktop-02	NAIKA-AppVNET-03	10.2.0.0/16	default	10.4.0.0/24	Firefox

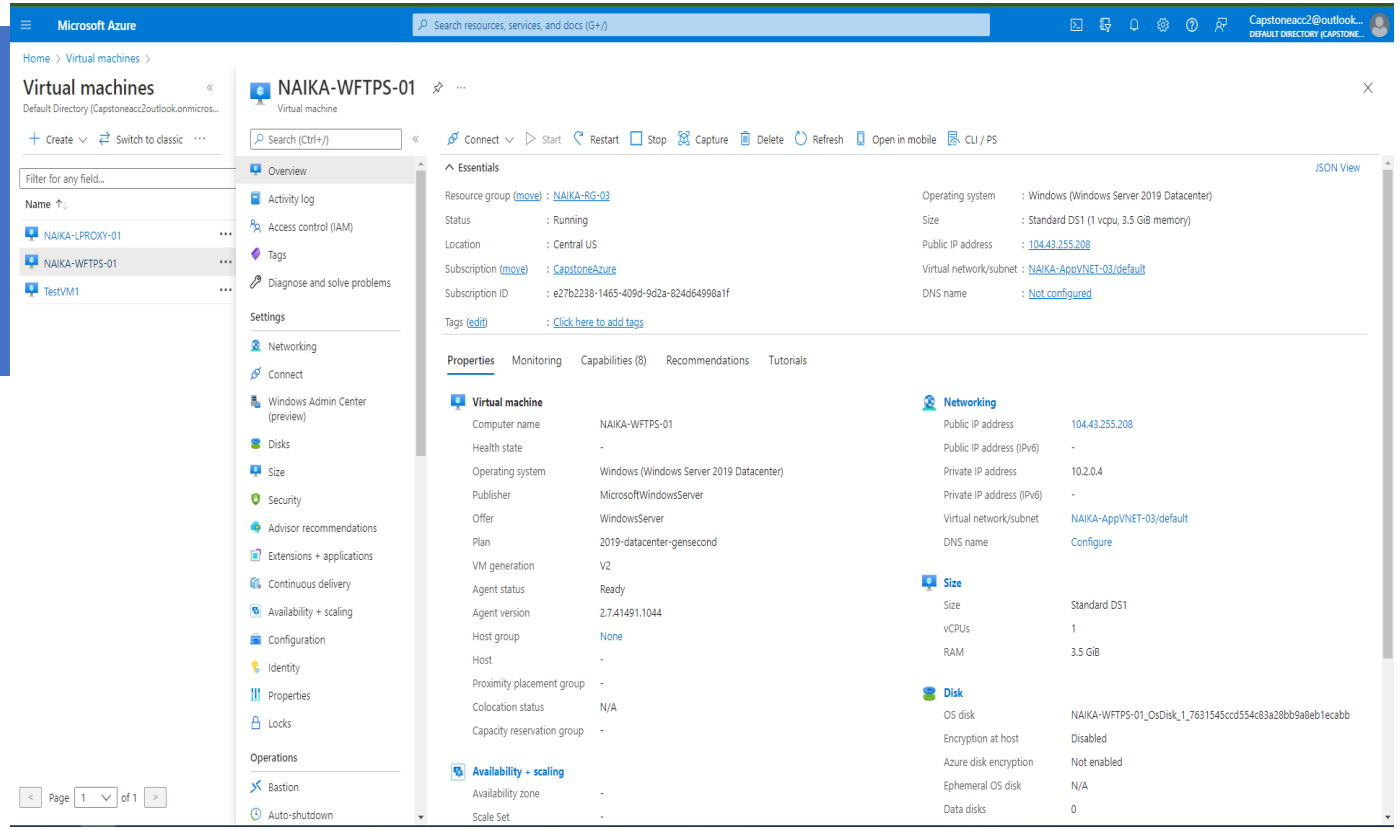
Client Desktop Setup



- Windows 10 Desktops have been created to serve as Client Desktops for all environments
- Client Desktop's have been set up in both domains to simulate an environment which incorporates workstations on different networks
- Client Desktops are also used to demonstrate connectivity between vNets and subscriptions for testing purposes

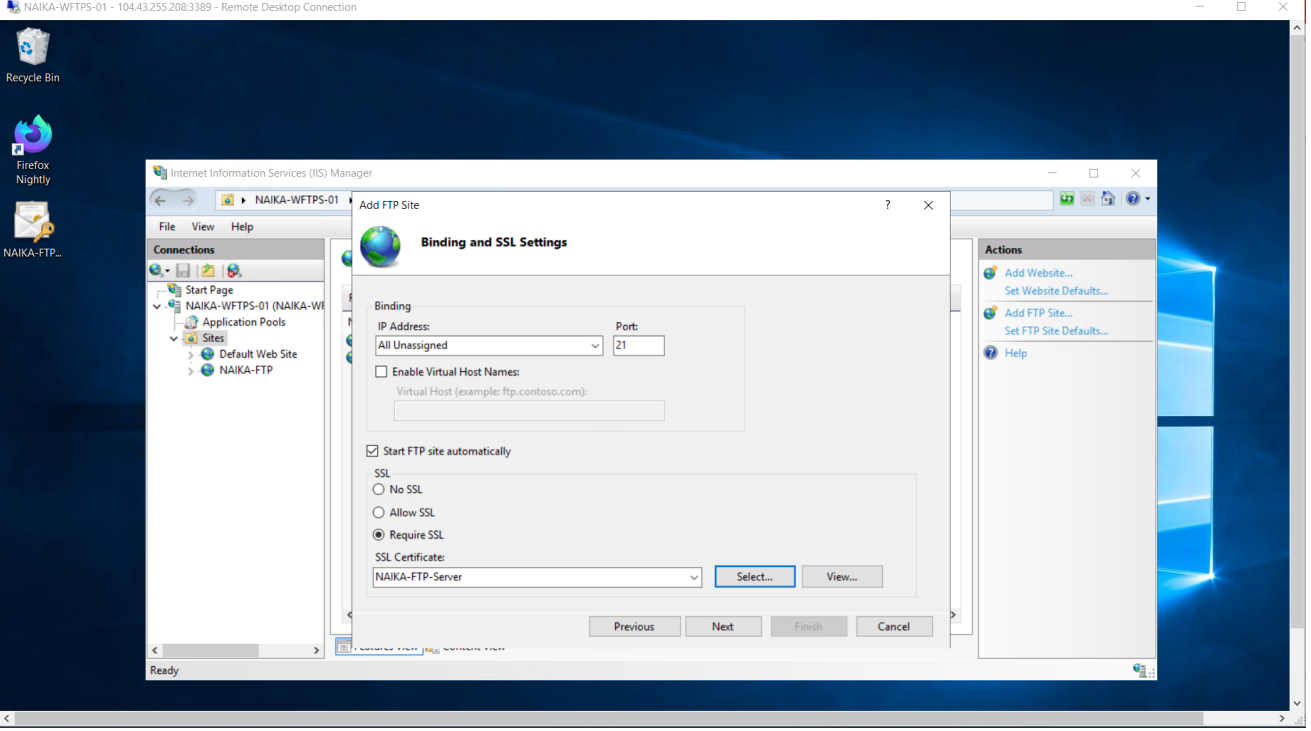
APPLICATIONS AND DATABASE SETUP

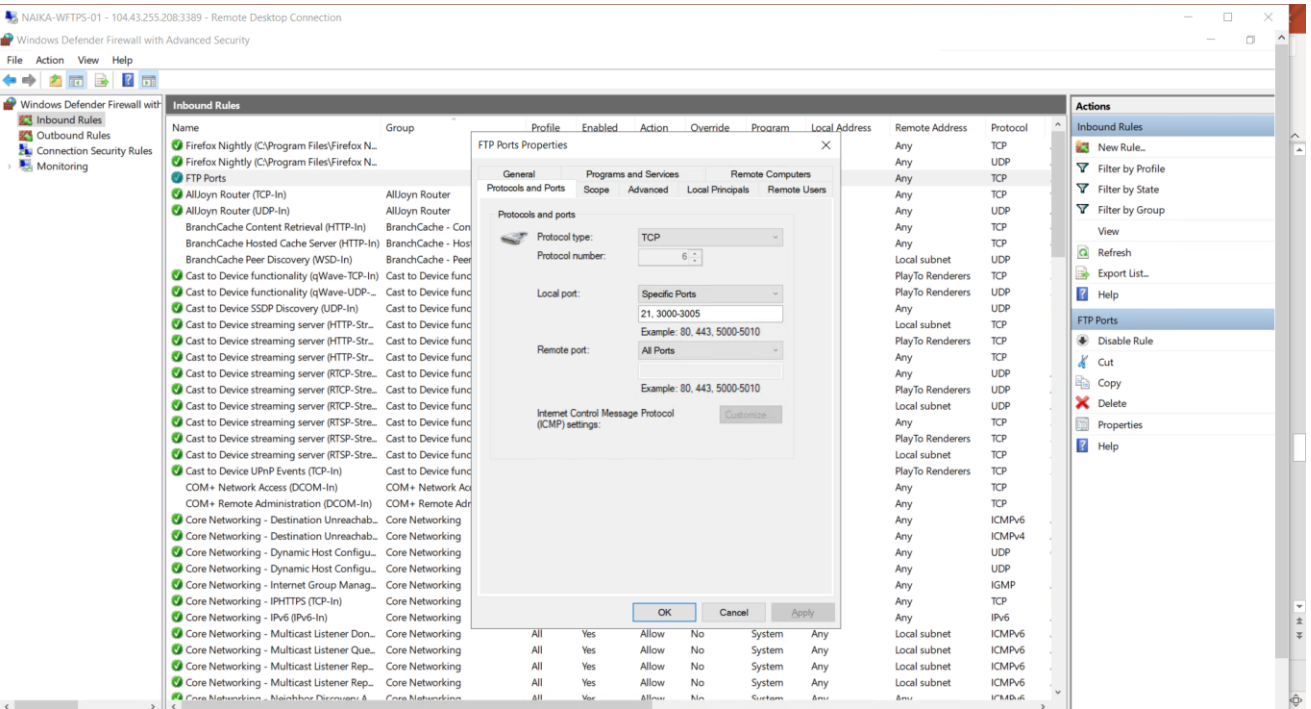
FTP Server Configuration



- FTP server is used to transfer files. We have used FTP server in our solution to demonstrate file transfer across networks.
- The FTP Server machine name: NAIKA-WFTPS-01 that is in the NAIKA-AppVNET-03/default (10.2.0.4/16).
- The machine allows FTPS connection for the Inbound rules. A self-signed SSL certificate has been used to secure FTP server for demo purpose.

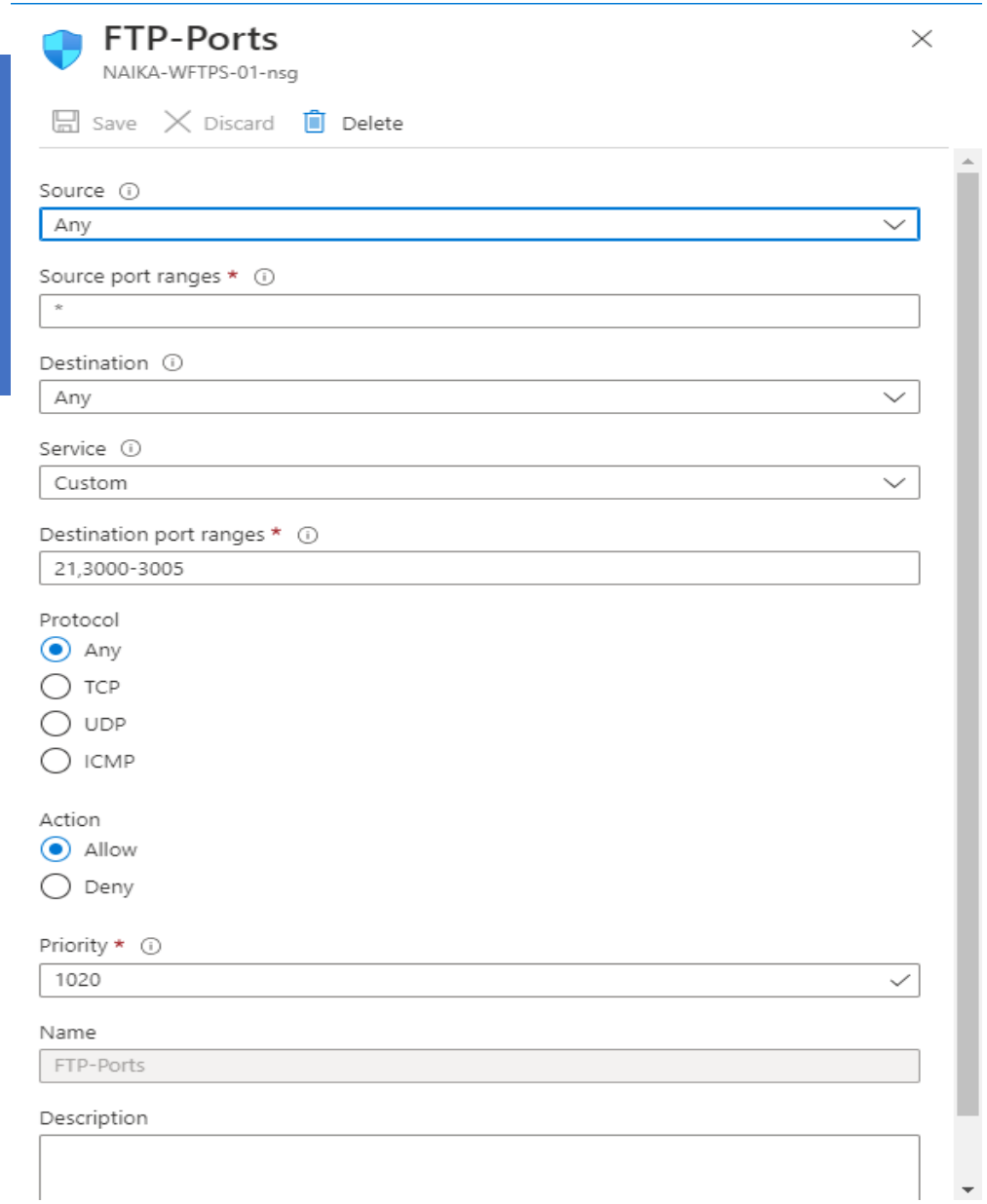
FTP Server Firewall Rules and Other Settings





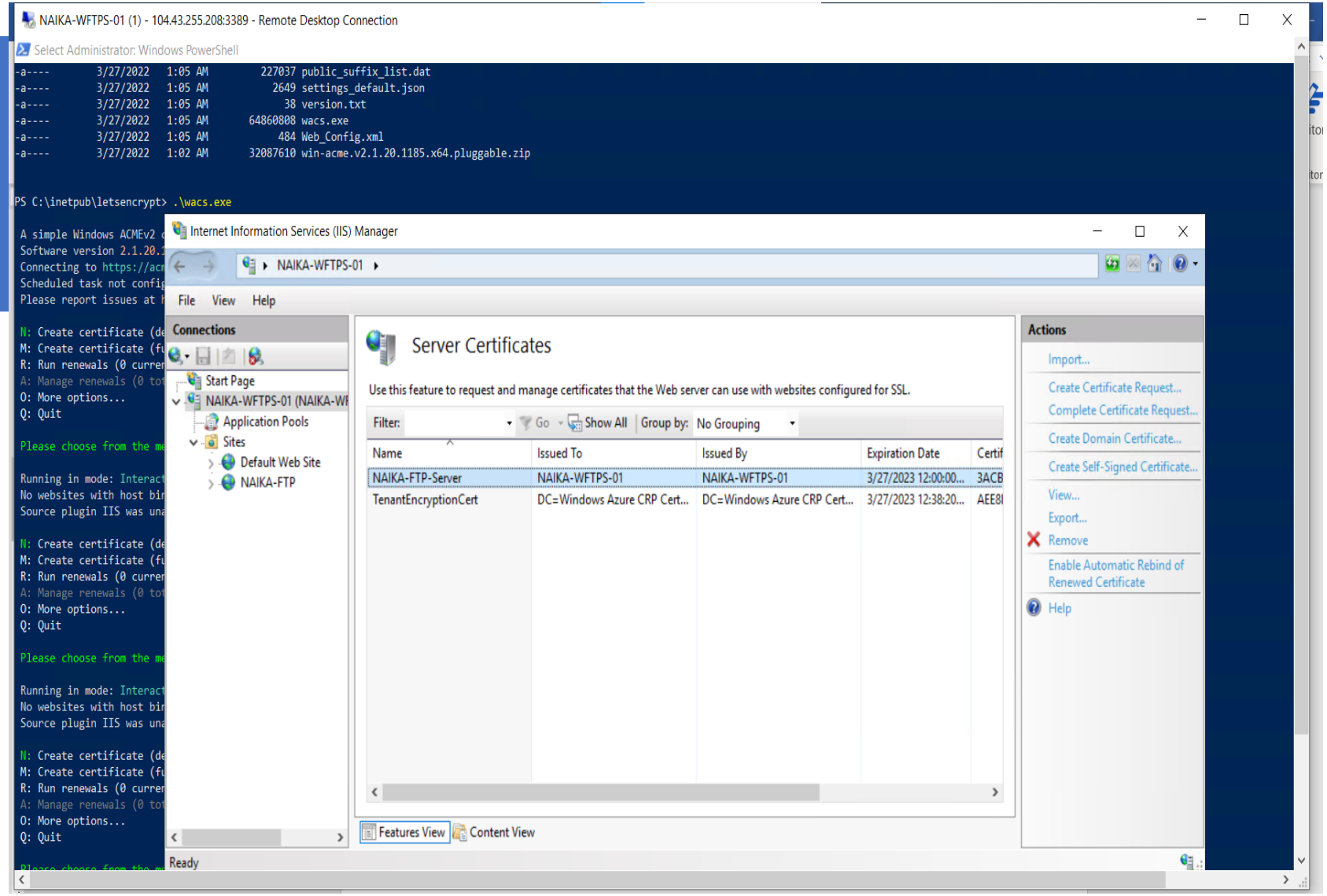
- Binding the certificate to the FTP Site requires the SSL certificate to establish connection between client and server.
- Define the Windows Firewall rules to allow FTP / FTPS Port (21) and destination range of the port for FTP Server to work on.

FTP Server Firewall Rules and Other Settings Contd..



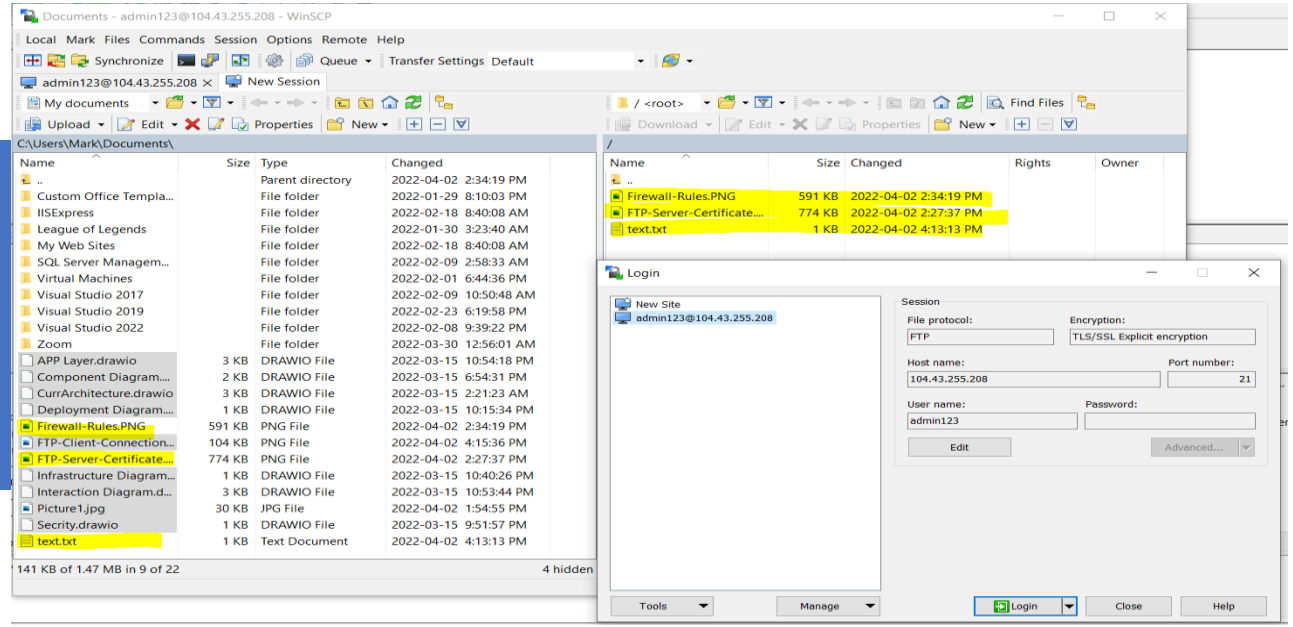
- **The NSG rules are updated at the VM level to allow Inbound connectivity on FTP port.**

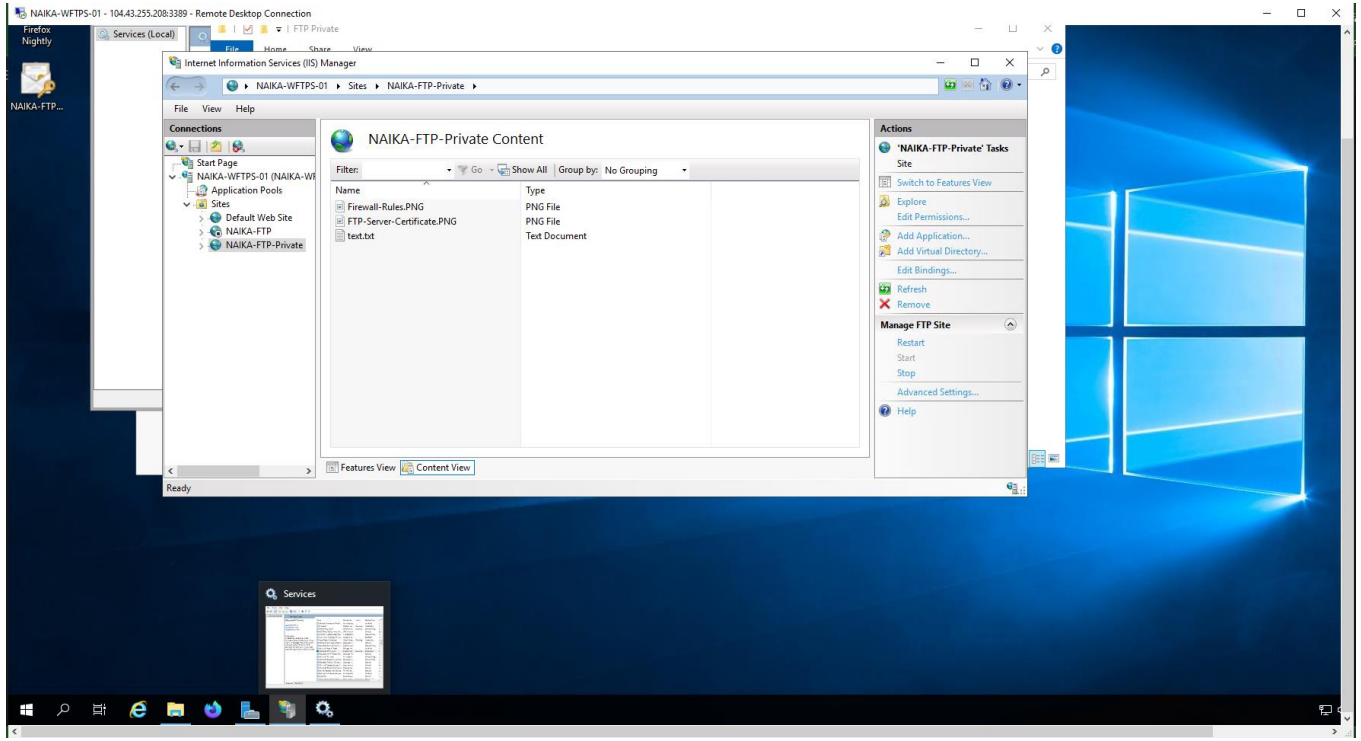
FTP Server certificates



- NAIKA-WFTPS-01 is the FTP website.
- Self-signed certificate is created on the FTP Server.
- It requires the explicit encryption to connect from FTP Client.
- Server certificates contain the certificate created for this website. The details of the SSL certificate can be viewed from this snap-in.

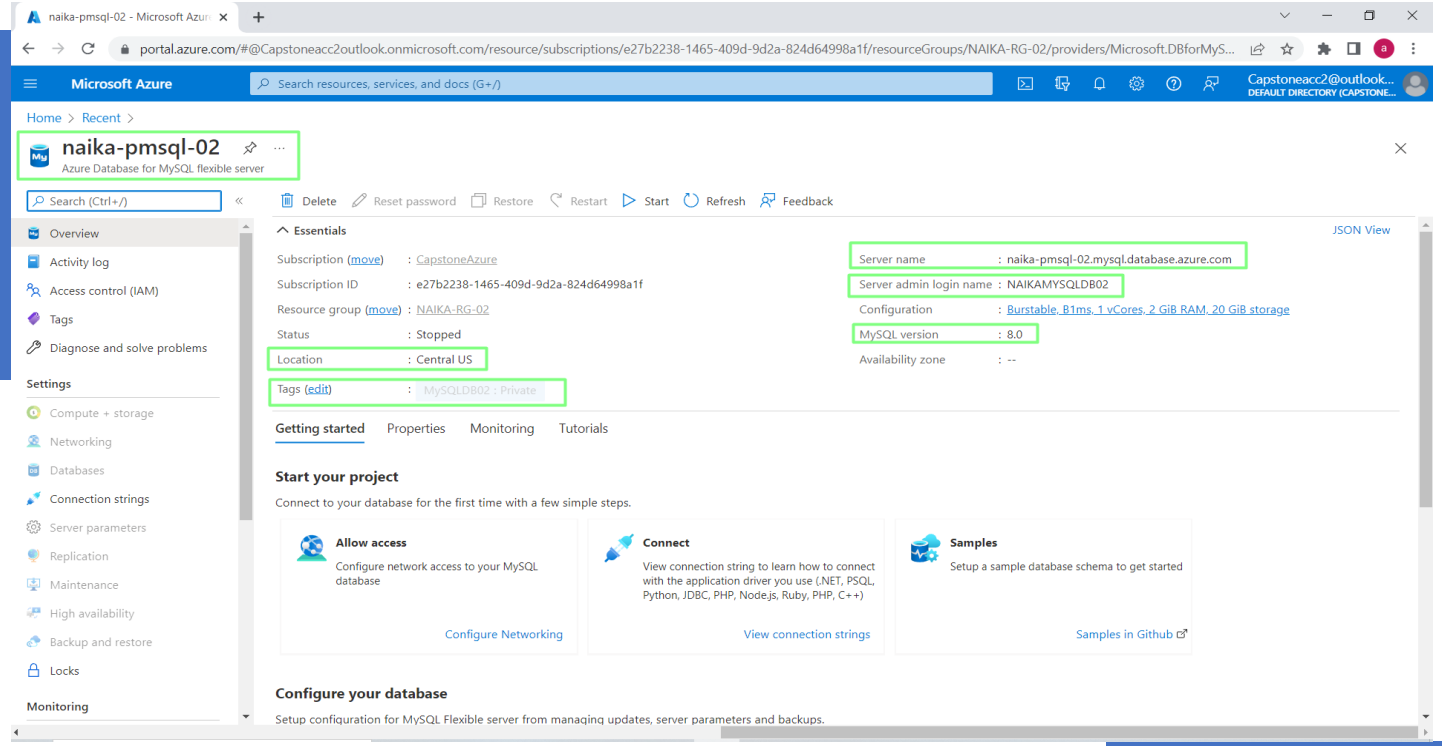
FTP Server File Transfer





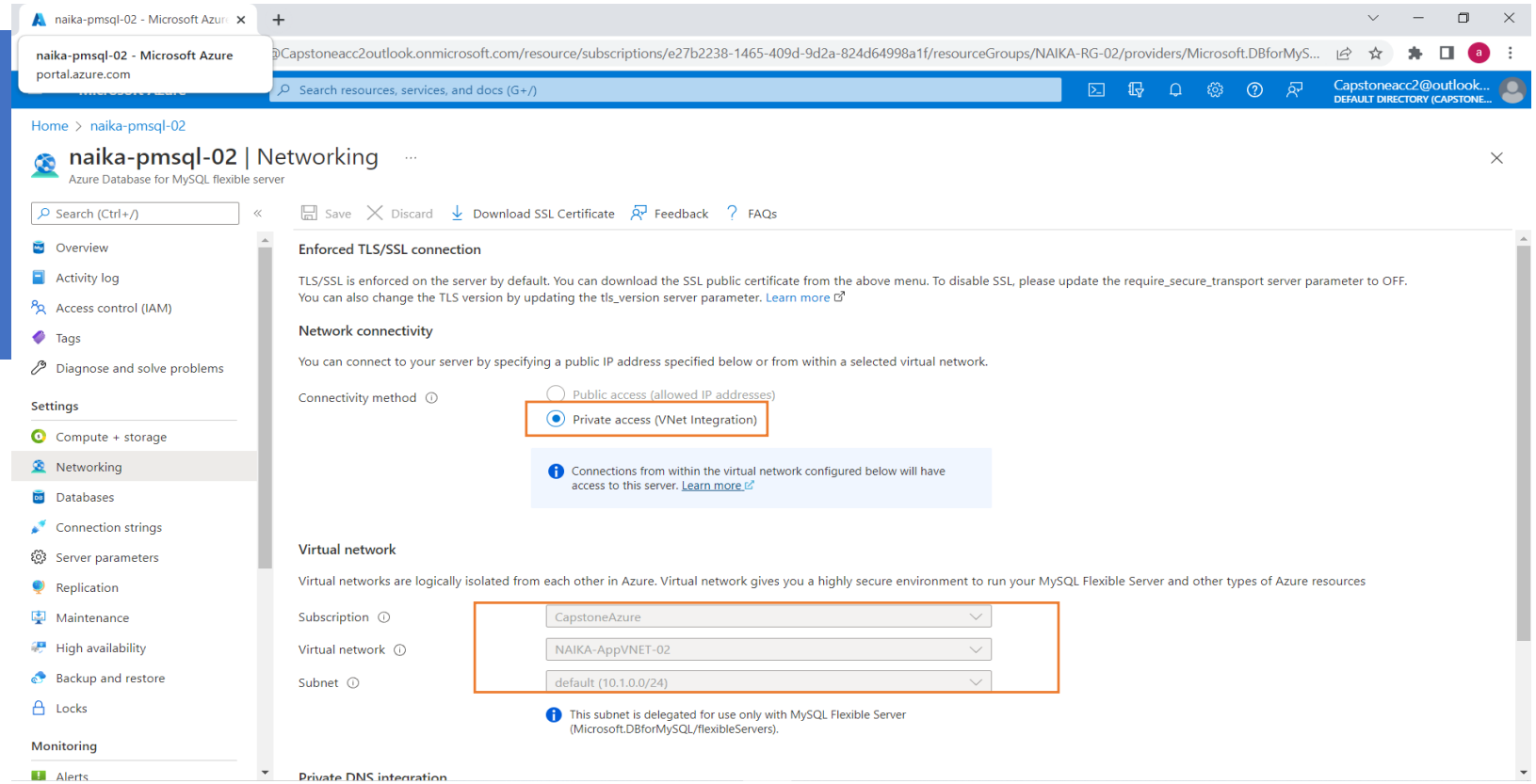
- WINSCP tool is installed on NAIKA-TestVM-01 which is used to test file transfer to the FTP server.
- For testing purpose, we have run a test from local machines (laptop) as FTPS is published over Internet.
- The connection to FTP Server requires the explicit encryption.
- This is the final test that the files transferred successful.

Private MySQL database



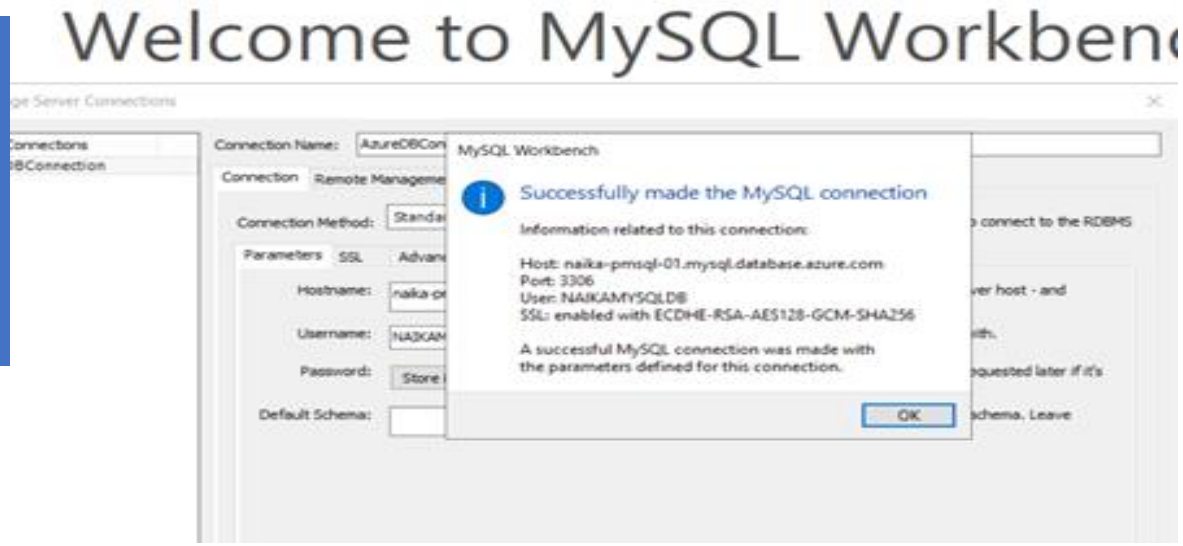
- In this solution, we have created Azure MySQL database, NAIKA-PMSQL-02 which is accessed privately in to order to provide the secure access from the application server.
- The DB is configured in the resource group NAIKA-RG-02, with private access enabled for the subnet 10.1.0.0/24
- This network is vNet peered to 10.0.0.0/24 where the application server is running. Hence, the end-to-end connectivity is over private networking enhancing the security of the database.
- Azure manages the DNS by default for private name resolution, we have used IP to connect to the DB.

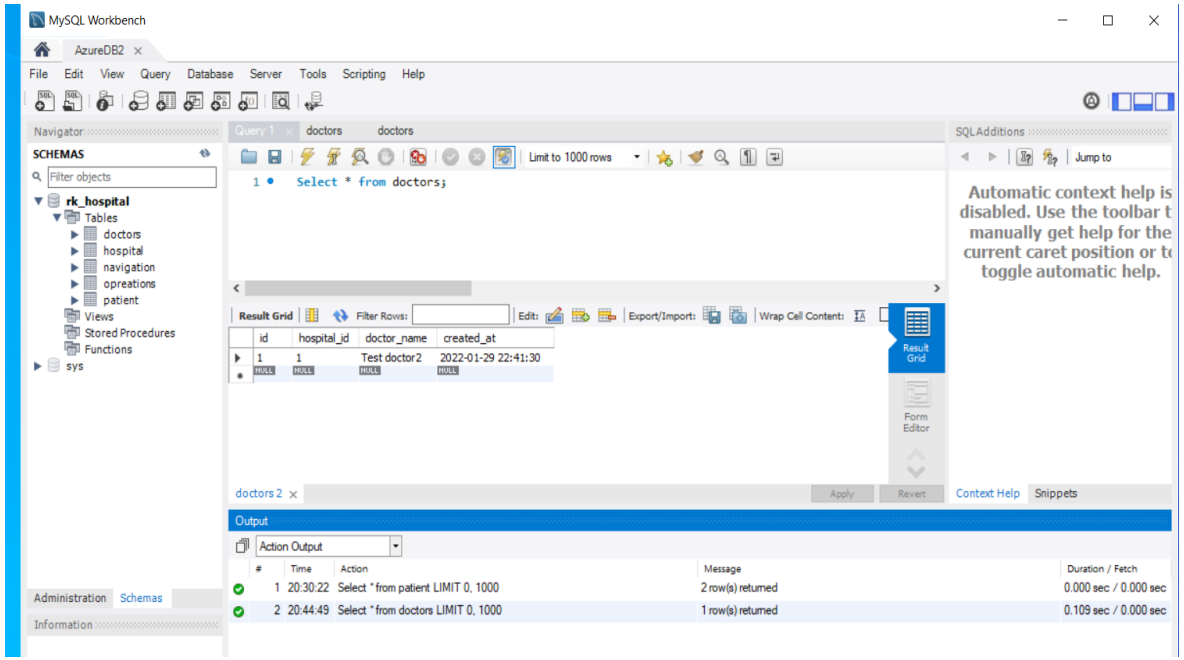
Private MySQL database



- **The screenshot shows Private access configuration done on one of the databases.**

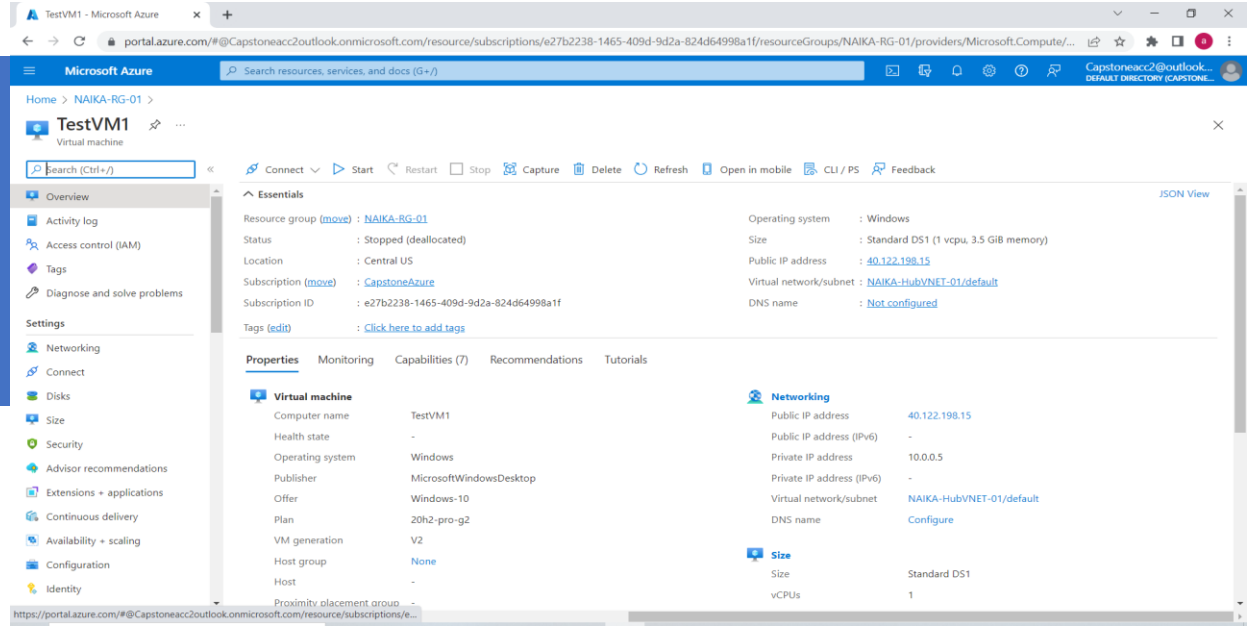
MySQL Workbench Connection to MySQL DB

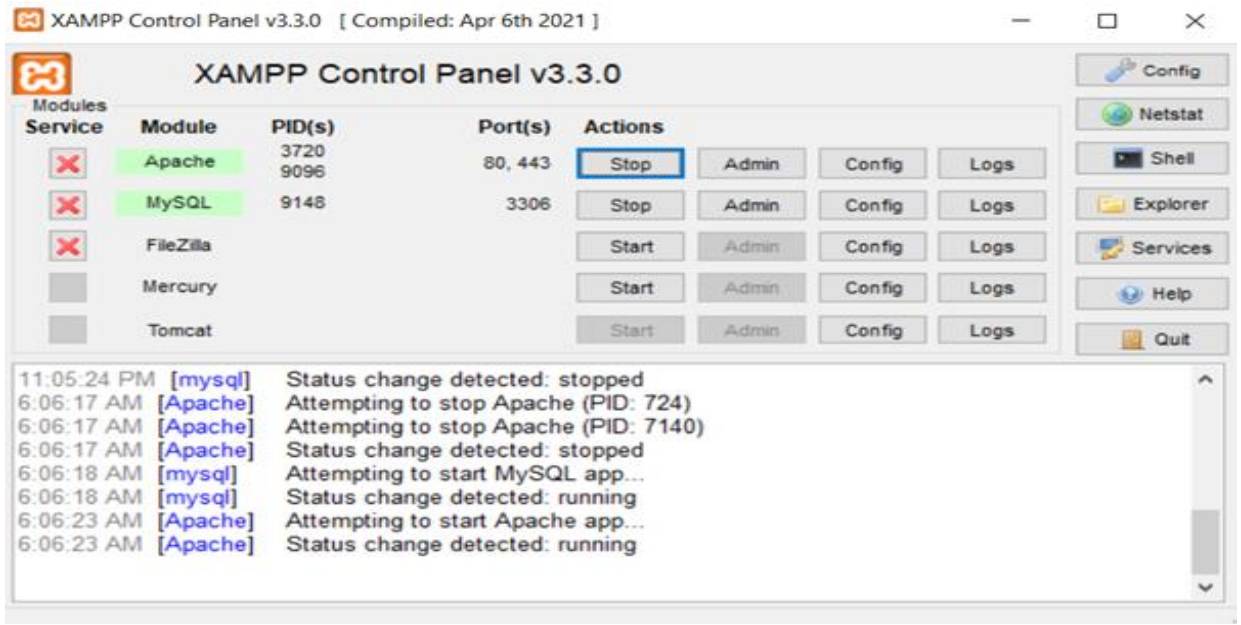




- We have used the My SQL workbench as DB client tool to verify the database connectivity, query the DB.
- The application is connected to this database and DB records can be modified from the application front end. The changes will reflect in real time on the backend database.
- MySQL Workbench provides ease of DB management for BAU activities.
- My SQL workbench is configured on the TESTVM1 which is windows 10 desktop.
- The connection settings of MySQL Workbench include DB server's name (endpoint), server admin login, Azure MySQL database name, port 3306.
- We have used an existing application's DB and schema for the demo purpose in this project. The schema name is rk_hospital , DB is imported from the rk_hopsital.sql file provided as part of App setup.
- After the successful import of the database records, run a query to check the records.

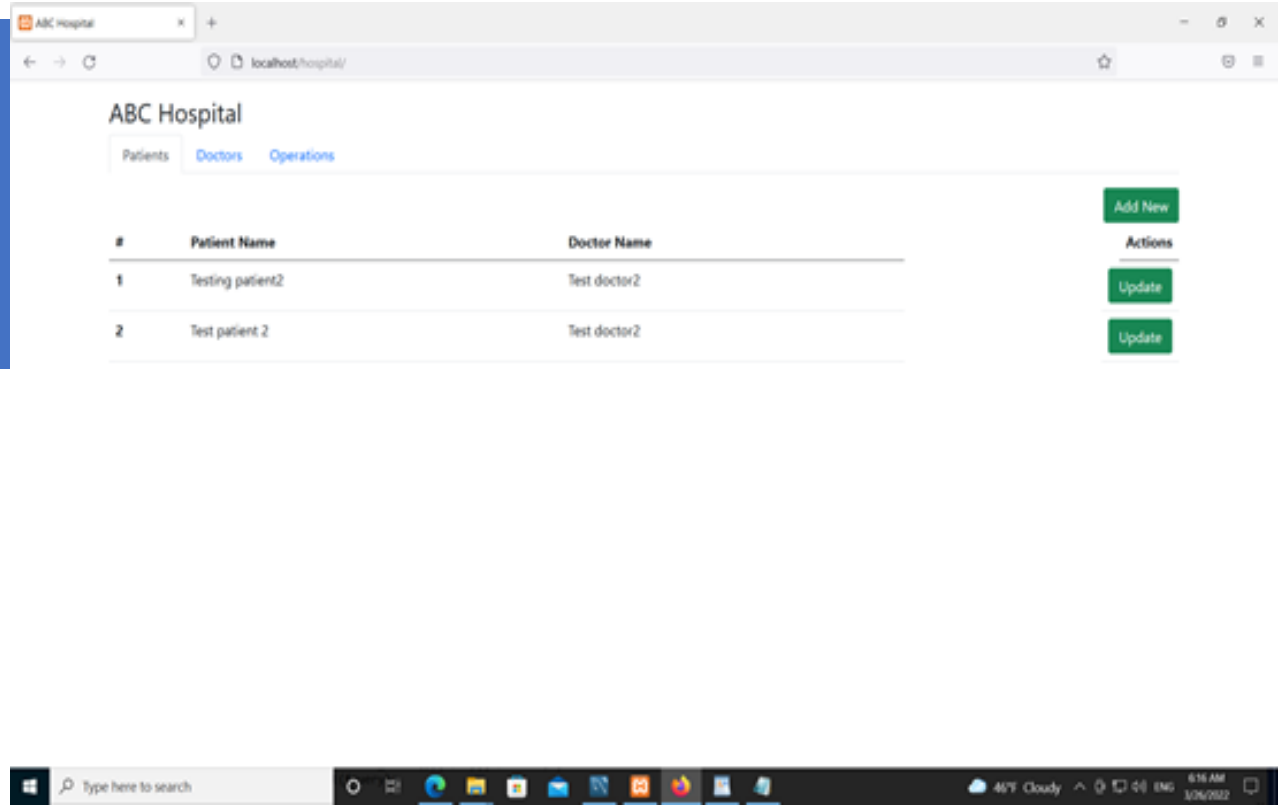
APACHE WEB Server Set-Up





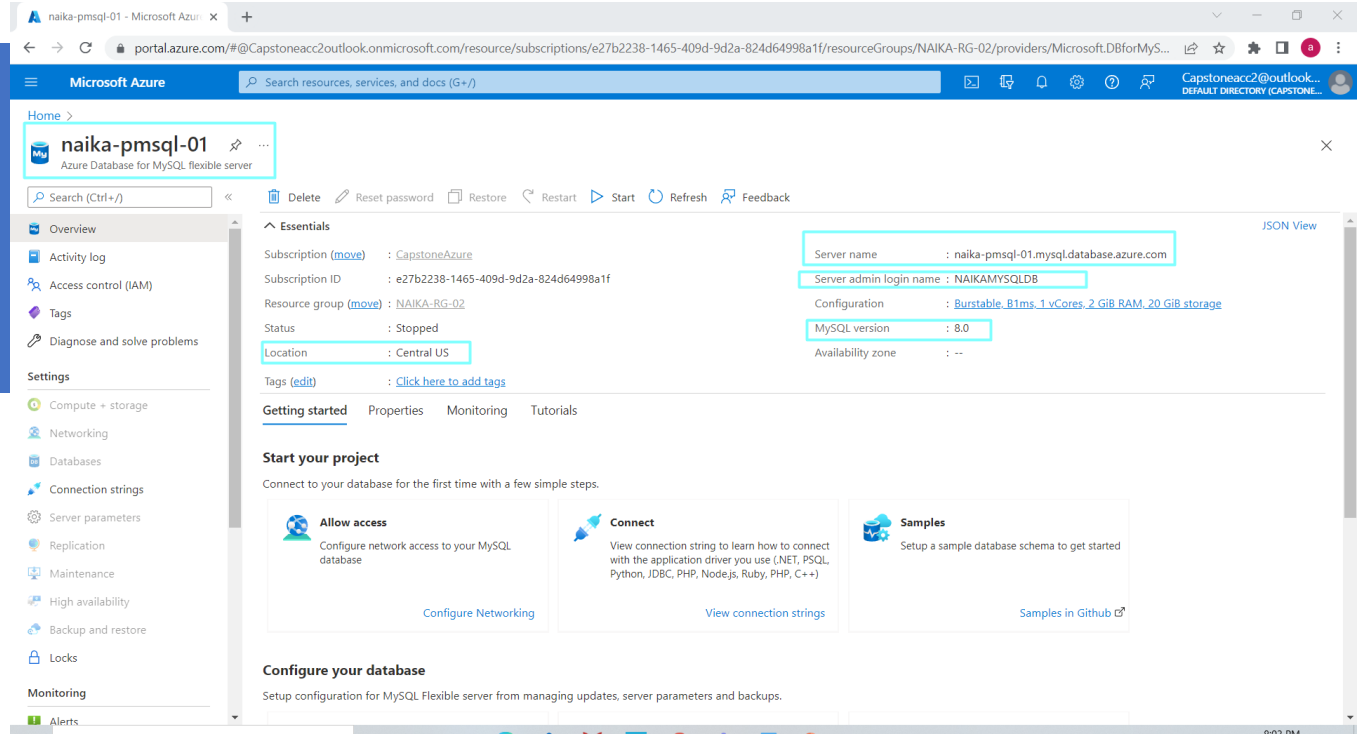
- **There is test application provided in one of the courses, it is referred to as Hospital application. For the demonstration of this project, we have re-used the same application stack.**
- **The application runs on XAMPP stack, which uses Apache is the web front-end.**
- **Configured the XAMPP on the VM – NAIKA-TestVM1**

Launch the Hospital Application



- The application web URL can be launched via http://localhost/hospital
- From this GUI, records can be added / updated .
- This demonstrates successful connectivity between subnet 10.0.0.0/24 and 10.1.0.0/24

Public MySQL database



- In this solution, we have created a public Azure SQL databases, NAIKA-PMSQL-01 to demonstrate the different between publicly and privately accessible databases and the difference in the configurations.
- The DB is configured in the resource group NAIKA-RG-02.
- The database server name (endpoint) and server admin login name is used to configure the connection with My SQL Workbench.
- With Publicly accessible DB, any server can connect to the endpoint with admin / SQL credentials if setup at DB level.

THANK YOU