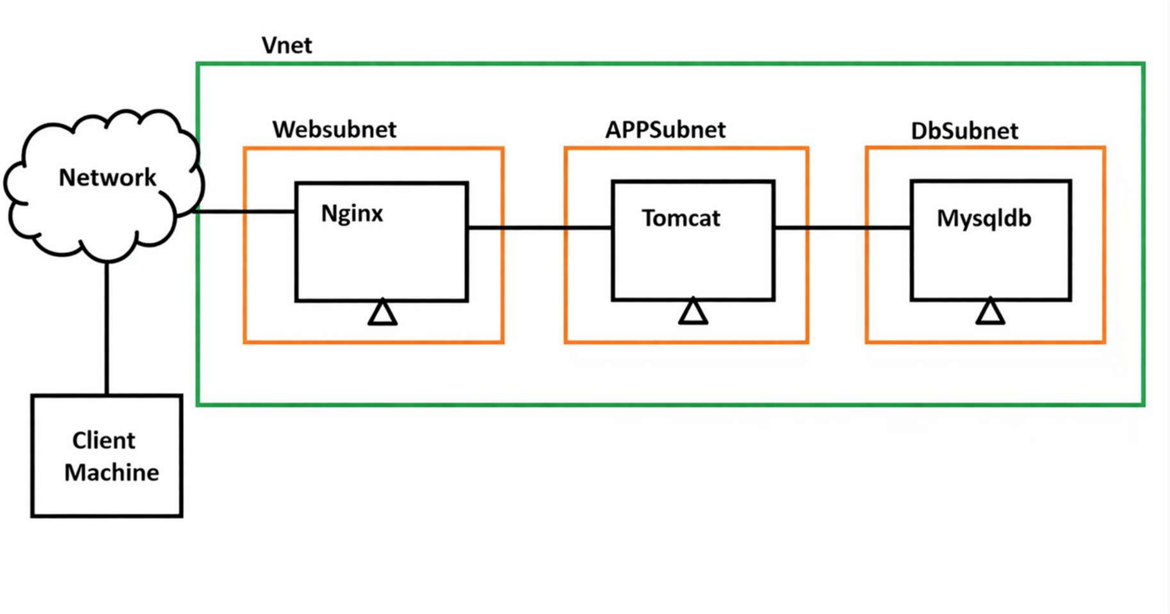
Name: Narendra Sai Manoj Chawakula

Batch: B-29

Mobile: 9110742799

Project -1

Building a 3-Tier Architecture On Azure



3-Tier Architecture

This project demonstrates the implementation of a **3-tier architecture on Microsoft Azure** to build a secure and scalable web application environment. The architecture is divided into three layers: a **Web Server layer** that handles client requests, an **Application Server layer** running **Apache Tomcat on port 8080**, and a **Database layer** using **MySQL on port 3306**. Each tier is deployed on separate Azure Virtual Machines to ensure proper isolation and manageability.

The Web Server acts as the entry point for users and forwards application requests to the internal application layer. The Application Server processes all business logic and communicates with the database server to store and retrieve data. The Database Server is placed in a private network to protect sensitive information and allow access only from the application tier.

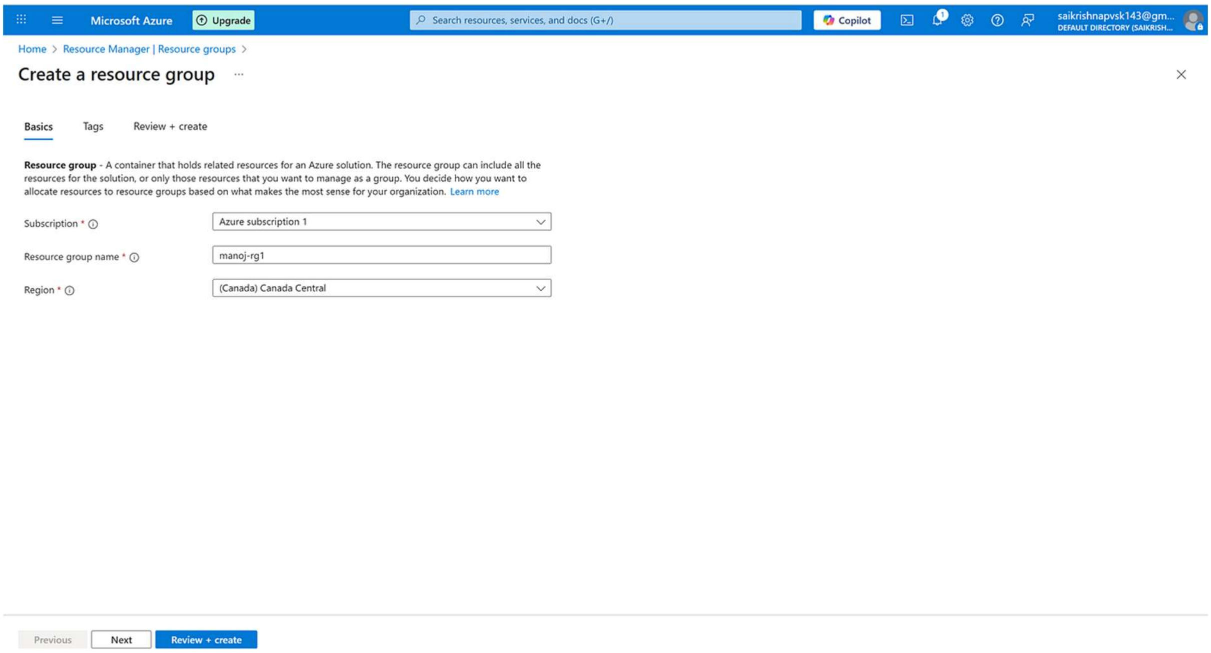
By using Azure Virtual Networks, subnets, and Network Security Groups, the architecture follows cloud best practices for security, availability, and scalability. This project represents a real-world enterprise-style deployment of a multi-tier application using Microsoft Azure.

Prerequisites:

- Azure Free Trail Subscription
- Resource Group
- Virtual Network (VNet)
- Subnets-WebServer (10.0.0.0), AppServer (172.16.0.0), DB server (172.16.1.0)
- Network Security Group (NSG)

Steps to Build 3-Tier Architecture:

Step 1: Create a Resource Group



Microsoft Azure
Upgrade
Search resources, services, and docs (G+/)
Copilot
saikrishnapvsk143@gm...
DEFAULT DIRECTORY (SAIKRISH...

Home > Resource Manager | Resource groups >

Create a resource group

Resource group - A container that holds related resources for an Azure solution. The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Learn more

Subscription * : Azure subscription 1

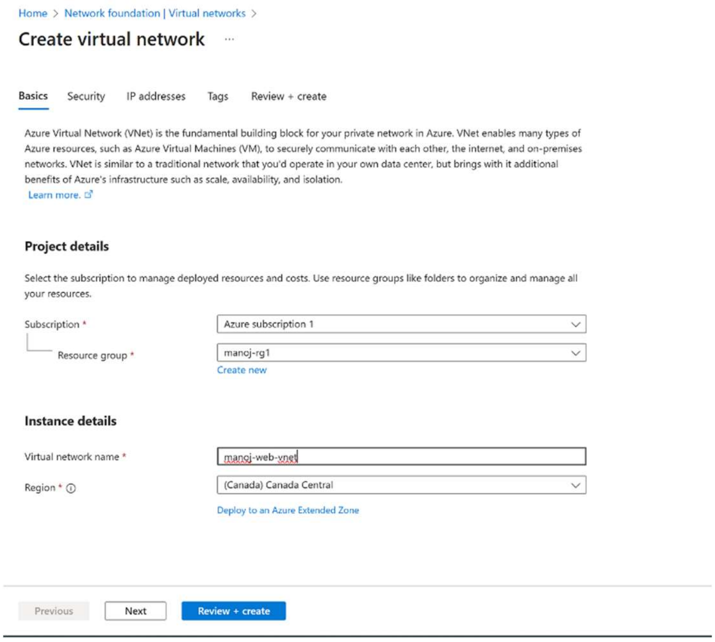
Resource group name * : manoj-rg1

Region * : (Canada) Canada Central

Select a region of your choice and assign a name to your Resource Group (RG) and then click on **Review + Create.**

Step 2: Create a Virtual Network (VNet):

Search for a “**Virtual Network”** and click on create.



Create virtual network

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure subscription 1

Resource group * : manoj-rg1

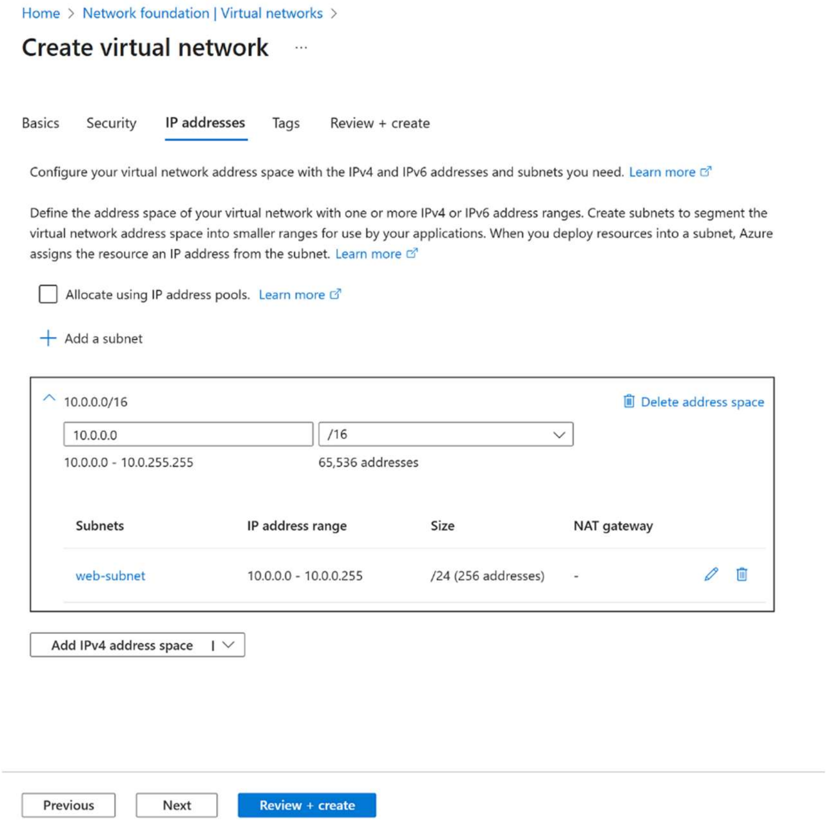
Instance details

Virtual network name * : manoj-web-vnet

Region * : (Canada) Canada Central

Select a Resource Group, where the VNet is deployed and give a name to the Virtual Network and select a region and click next and move to IP Address section.

Note: The region of Resource Group and VNet can be different as it is not mandatory to be same, and it does not effect our architecture.



Create virtual network

Configure your virtual network address space with the IPv4 and IPv6 addresses and subnets you need.

Define the address space of your virtual network with one or more IPv4 or IPv6 address ranges. Create subnets to segment the virtual network address space into smaller ranges for use by your applications. When you deploy resources into a subnet, Azure assigns the resource an IP address from the subnet.

□ Allocate using IP address pools.

10.0.0.0/16

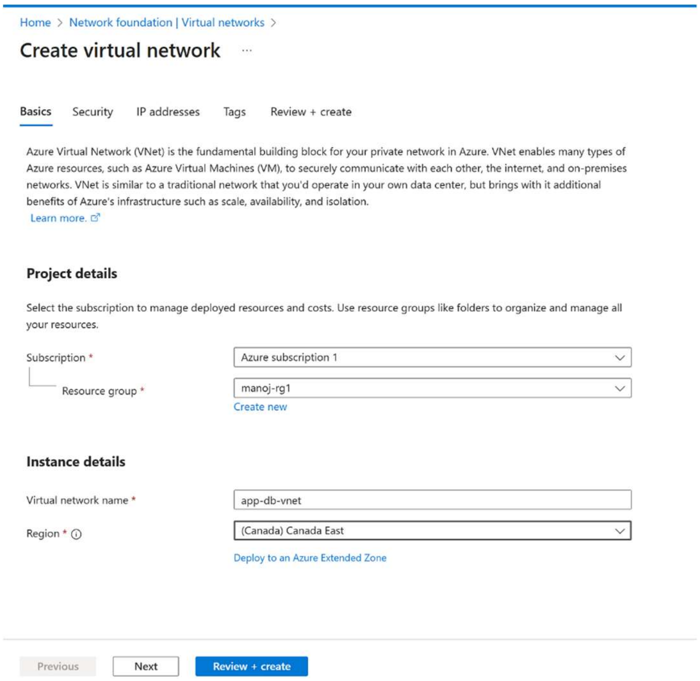
10.0.0.0 /16

10.0.0.0 - 10.0.255.255 65,536 addresses

Subnets	IP address range	Size	NAT gateway
web-subnet	10.0.0.0 - 10.0.0.255	/24 (256 addresses)	-

Here select **IP Address and CIDR Range** as required for your architecture and edit the name and subnet. Here we are going to use this Subnet for Web Server.

Create Another Virtual Network:


Home > Network foundation | Virtual networks >
Create virtual network ···
Basics
Security
IP addresses
Tags
Review + create
Previous
Next
Review + create

Azure Virtual Network (VNet) is the fundamental building block for your private network in Azure. VNet enables many types of Azure resources, such as Azure Virtual Machines (VM), to securely communicate with each other, the internet, and on-premises networks. VNet is similar to a traditional network that you'd operate in your own data center, but brings with it additional benefits of Azure's infrastructure such as scale, availability, and isolation.
Learn more.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * Azure subscription 1

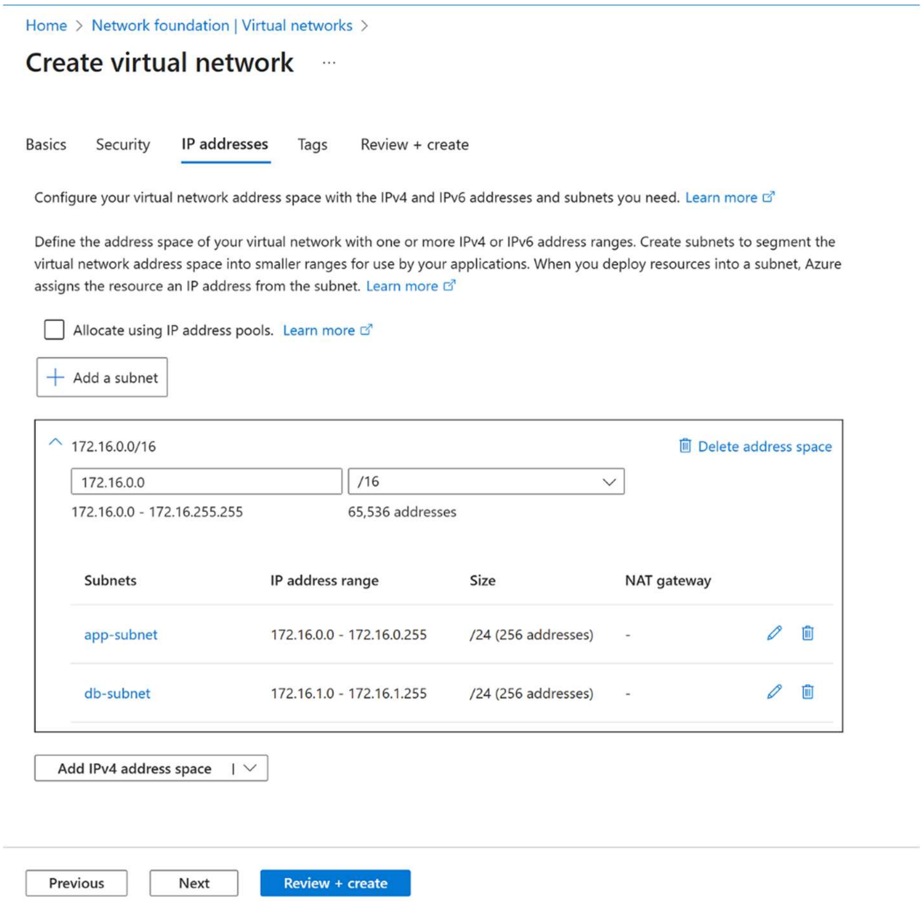
Resource group * manoj-rg1
Create new

Instance details

Virtual network name * app-db-vnet

Region * (Canada) Canada East
Deploy to an Azure Extended Zone

Here select a different regionand and click on next and move to IP Address page.



Configure your virtual network address space with the IPv4 and IPv6 addresses and subnets you need. Learn more

Define the address space of your virtual network with one or more IPv4 or IPv6 address ranges. Create subnets to segment the virtual network address space into smaller ranges for use by your applications. When you deploy resources into a subnet, Azure assigns the resource an IP address from the subnet. Learn more

□ Allocate using IP address pools. Learn more

172.16.0.0/16

172.16.0.0 /16
172.16.0.0 - 172.16.255.255 65,536 addresses

Subnets	IP address range	Size	NAT gateway
app-subnet	172.16.0.0 - 172.16.0.255	/24 (256 addresses)	-
db-subnet	172.16.1.0 - 172.16.1.255	/24 (256 addresses)	-

Here change the VNet range and Create two Subnets, and click on create.

172.16.0.0 - It will be assigned to App Server

172.16.1.0 - It Will be assigned to DB Server

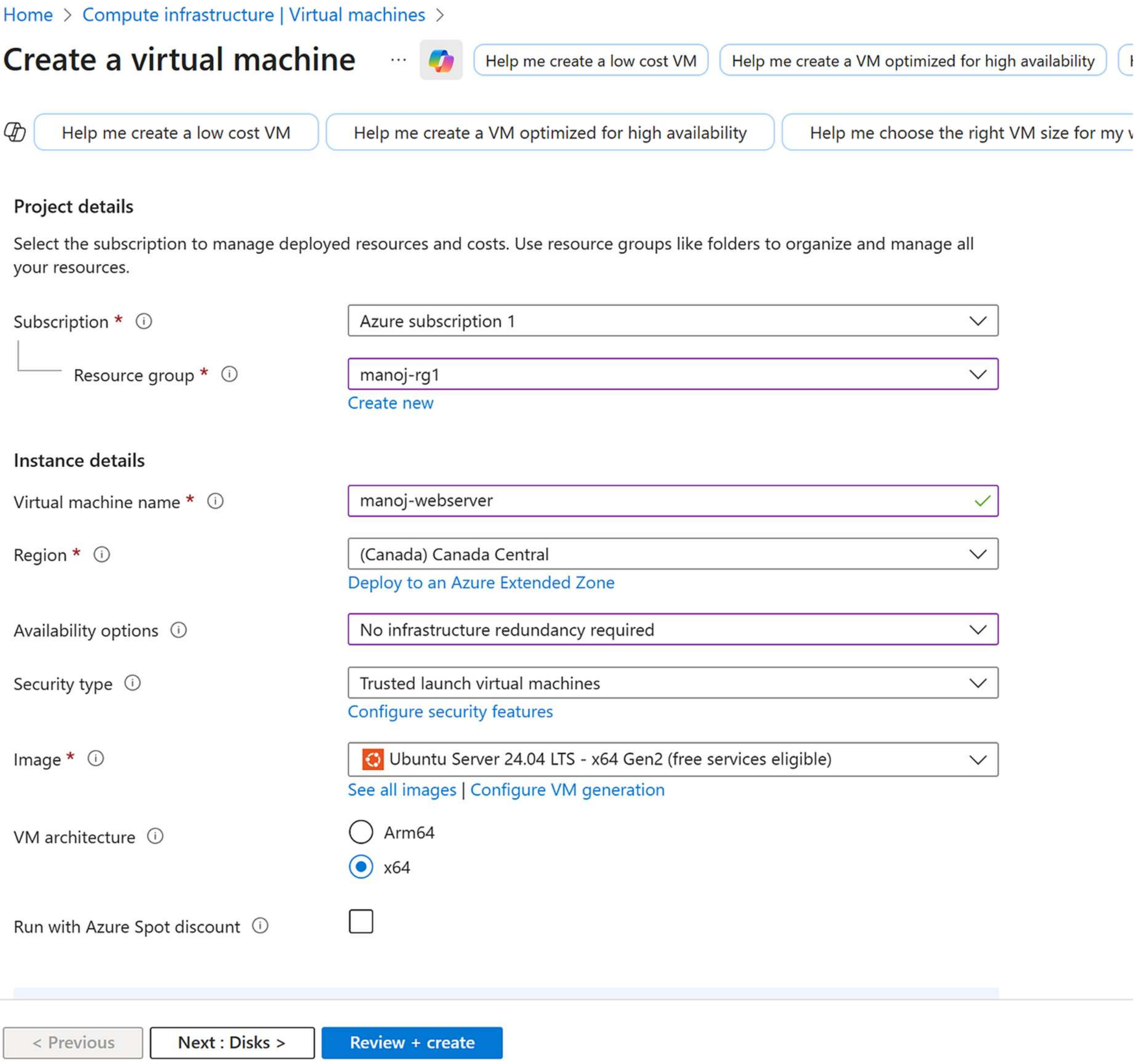
Note: We are creating to VNet's in two different region because, Azure Free trail doesn't provide **1vcpu** service (Total of 4vcpu's). And here we need three Servers so we use to different VNets in two different regions.

Step 3: Create Virtual Machines:

Now its time to create VM's or Servers.

Web Server:

Search for Virtual Machines and click on create and select first option (virtual machine).



Home > Compute infrastructure | Virtual machines >

Create a virtual machine

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *: Azure subscription 1

Resource group *: manoj-rg1

Instance details

Virtual machine name *: manoj-webserver

Region *: (Canada) Canada Central

Availability options: No infrastructure redundancy required

Security type: Trusted launch virtual machines

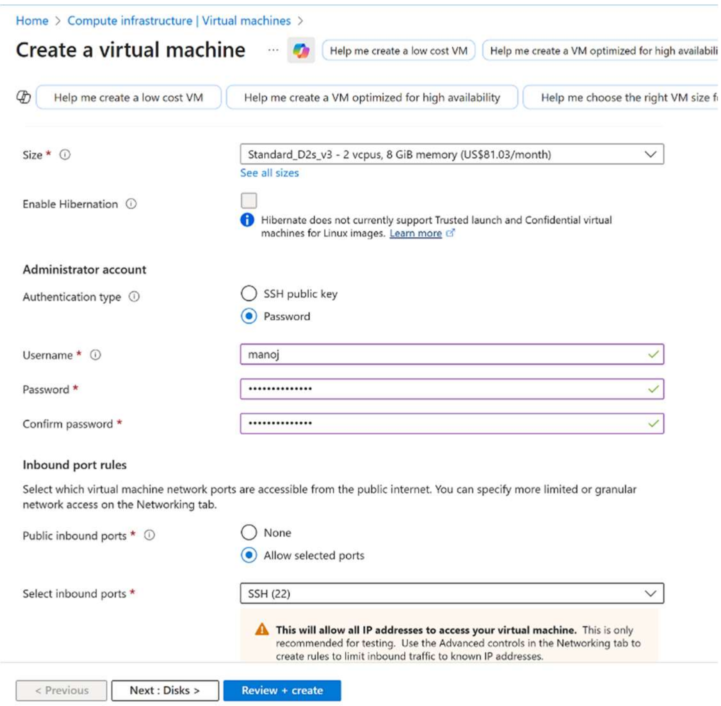
Image *: Ubuntu Server 24.04 LTS - x64 Gen2 (free services eligible)

VM architecture: ○ Arm64 ● x64

Run with Azure Spot discount: □

Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my v
Create new
Deploy to an Azure Extended Zone
Configure security features
See all images | Configure VM generation
< Previous
Next : Disks >
Review + create

Here select the RG and give a name to your VM, and Select the region where you created you first VNet i.e; **manoj-web-vnet.** and select availability options you need and the image (Ubuntu), and scroll down



Home > Compute infrastructure | Virtual machines >

Create a virtual machine ··· Help me create a low cost VM | Help me create a VM optimized for high availabili

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size f

Size * : Standard_D2s_v3 - 2 vcpus, 8 GiB memory (US$81.03/month)
See all sizes

Enable Hibernation : □
Hibernate does not currently support Trusted launch and Confidential virtual machines for Linux images. Learn more

Administrator account

Authentication type : ○ SSH public key ● Password

Username * : manoj

Password * : ••••••••••••••

Confirm password * : ••••••••••••••

Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

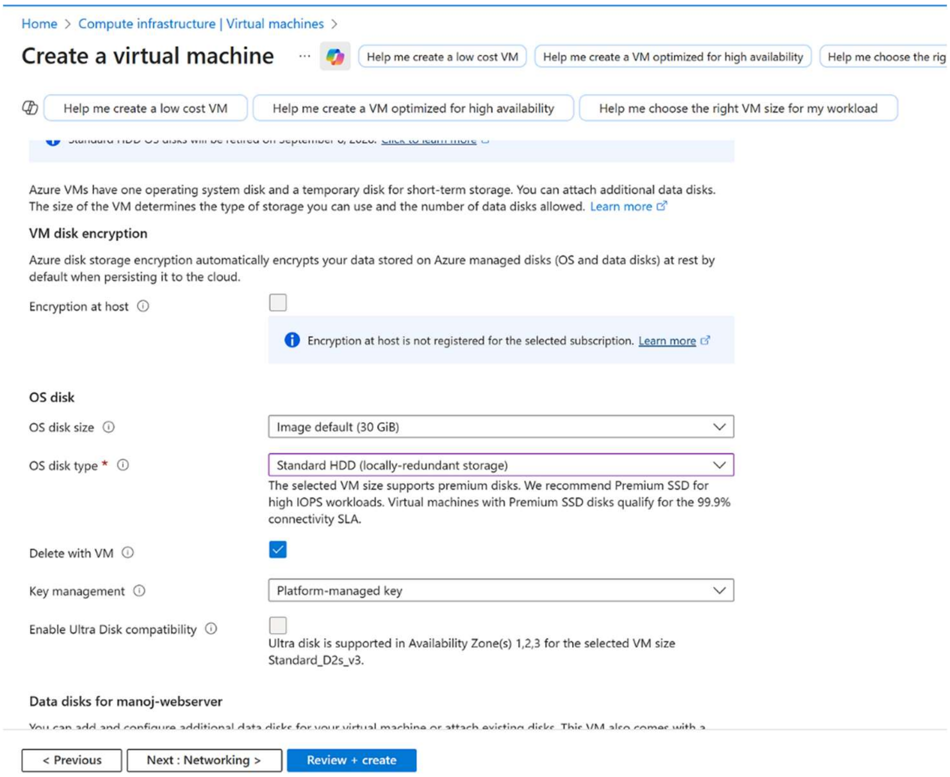
Public inbound ports * : ○ None ● Allow selected ports

Select inbound ports * : SSH (22)

⚠ **This will allow all IP addresses to access your virtual machine.** This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

< Previous | Next : Disks > | Review + create

Select the size (smallest size 2vcpus), and select SSH or password as you wish (here i selected password and gave username and password), and allow SSH port, because we need it to connect to the server, and go disks page.



Home > Compute infrastructure | Virtual machines >

Create a virtual machine ··· Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the rig

Help me create a low cost VM | Help me create a VM optimized for high availability | Help me choose the right VM size for my workload

Azure VMs have one operating system disk and a temporary disk for short-term storage. You can attach additional data disks. The size of the VM determines the type of storage you can use and the number of data disks allowed. Learn more

VM disk encryption

Azure disk storage encryption automatically encrypts your data stored on Azure managed disks (OS and data disks) at rest by default when persisting it to the cloud.

Encryption at host : □
Encryption at host is not registered for the selected subscription. Learn more

OS disk

OS disk size : Image default (30 GiB)

OS disk type * : Standard HDD (locally-redundant storage)
The selected VM size supports premium disks. We recommend Premium SSD for high IOPS workloads. Virtual machines with Premium SSD disks qualify for the 99.9% connectivity SLA.

Delete with VM : ☑

Key management : Platform-managed key

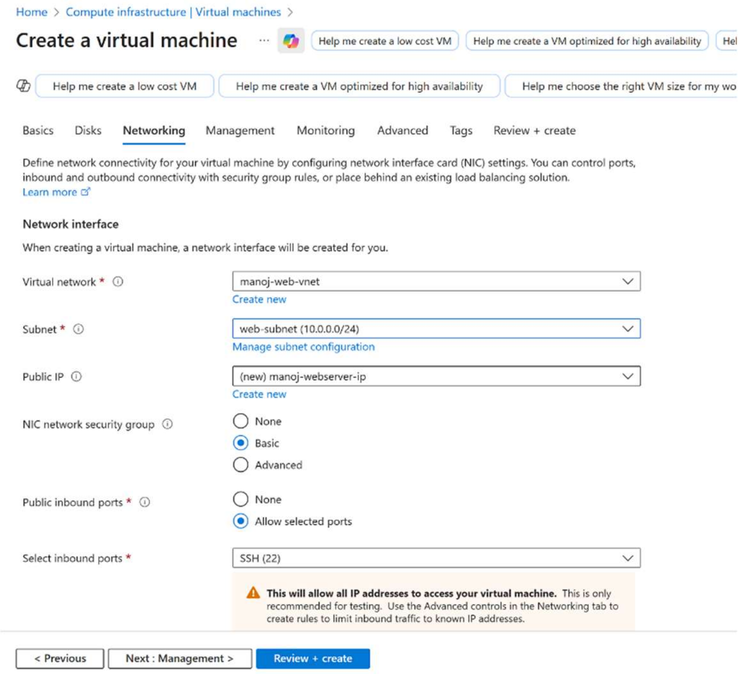
Enable Ultra Disk compatibility : □
Ultra disk is supported in Availability Zone(s) 1,2,3 for the selected VM size Standard_D2s_v3.

Data disks for manoj-webserver

You can add and configure additional data disks for your virtual machine or attach existing disks. This VM also comes with a

< Previous | Next : Networking > | Review + create

Select the disk type, here i selected HDD, and click on next.


Home > Compute infrastructure | Virtual machines >
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my wo
Basics
Disks
Networking
Management
Monitoring
Advanced
Tags
Review + create
< Previous
Next : Management >
Review + create

Create a virtual machine

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. Learn more

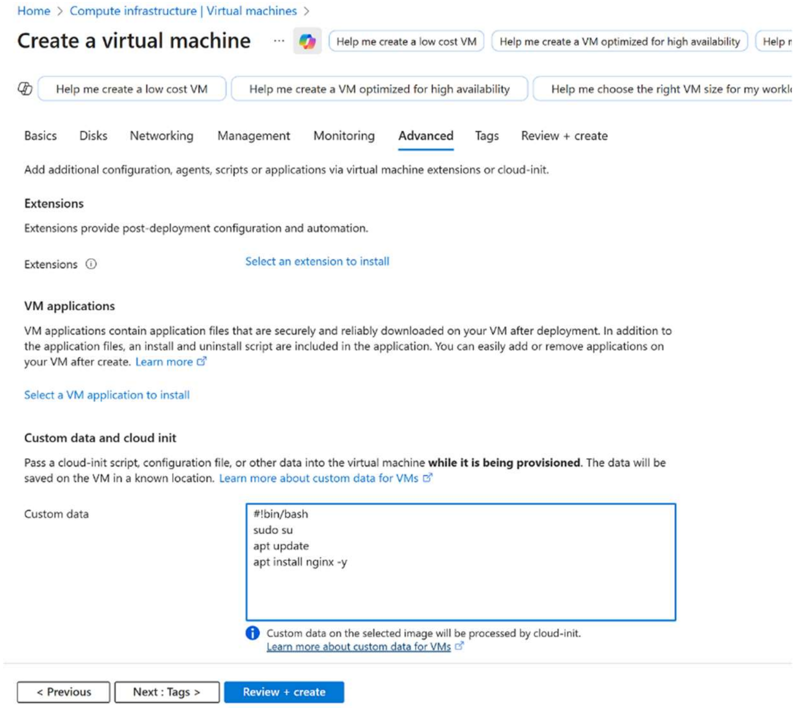
Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network *	manoj-web-vnet Create new
Subnet *	web-subnet (10.0.0.0/24) Manage subnet configuration
Public IP	(new) manoj-webserver-ip Create new
NIC network security group	○ None ● Basic ○ Advanced
Public inbound ports *	○ None ● Allow selected ports
Select inbound ports *	SSH (22)

This will allow all IP addresses to access your virtual machine. This is only recommended for testing. Use the Advanced controls in the Networking tab to create rules to limit inbound traffic to known IP addresses.

Here select the VNet (manoj-web-vnet), which is created for web server and select the subnet range and click on next and move to Advanced settings page.


Home > Compute infrastructure | Virtual machines >
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my workl
Basics
Disks
Networking
Management
Monitoring
Advanced
Tags
Review + create
< Previous
Next : Tags >
Review + create

Create a virtual machine

Add additional configuration, agents, scripts or applications via virtual machine extensions or cloud-init.

Extensions

Extensions provide post-deployment configuration and automation.

Extensions: Select an extension to install

VM applications

VM applications contain application files that are securely and reliably downloaded on your VM after deployment. In addition to the application files, an install and uninstall script are included in the application. You can easily add or remove applications on your VM after create. Learn more

Select a VM application to install

Custom data and cloud init

Pass a cloud-init script, configuration file, or other data into the virtual machine **while it is being provisioned**. The data will be saved on the VM in a known location. Learn more about custom data for VMs

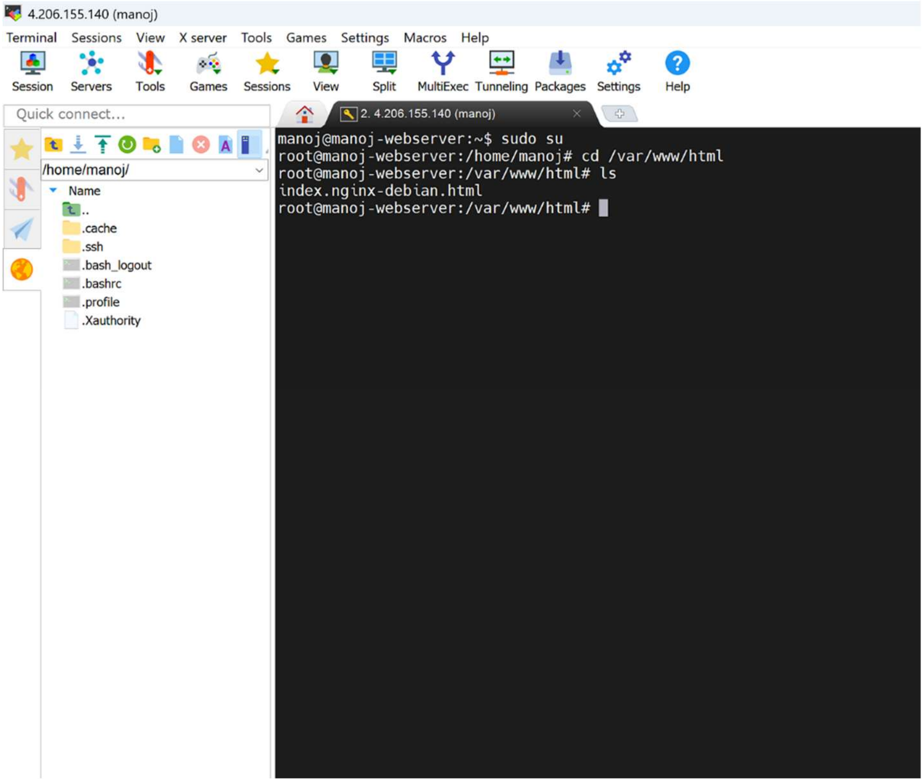
Custom data

```
#!bin/bash
sudo su
apt update
apt install nginx -y
```

Custom data on the selected image will be processed by cloud-init.
Learn more about custom data for VMs

In this server we need to install **Nginx server**, that is used and wed server (frontend server), so write the commands as shown in custom data, this will automatically install nginx server while creating the VM, and then click on review + create.

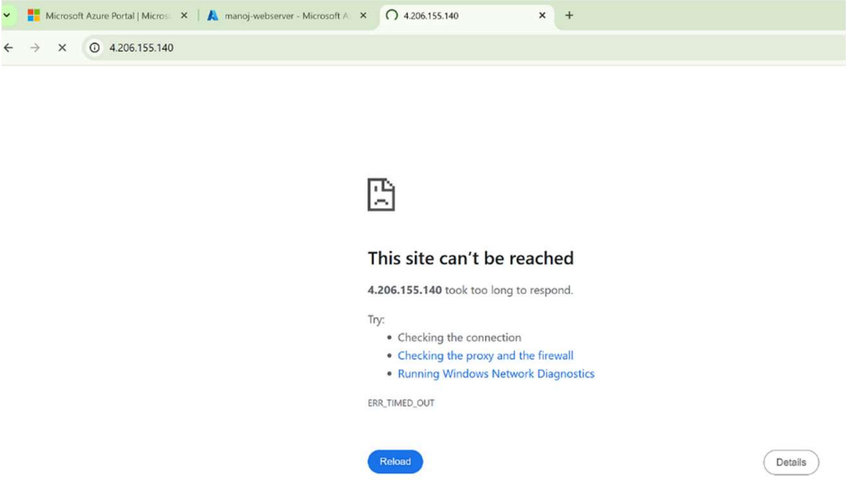
Now lets connect to the server and check whether the nginx server is installed or not. Here i used **Mobaxtream,** to connect to the Vm's using SSH and provide the **public ip** of the machine and enter password.


4.206.155.140 (manoj)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
2. 4.206.155.140 (manoj)
/home/manoj/
Name
..
.cache
.ssh
.bash_logout
.bashrc
.profile
.Xauthority

```
manoj@manoj-webserver:~$ sudo su
root@manoj-webserver:/home/manoj# cd /var/www/html
root@manoj-webserver:/var/www/html# ls
index.nginx-debian.html
root@manoj-webserver:/var/www/html#
```

Execute the **"cd /var/www/html"** command to move to the directory and execute ls command to check the nginx is installed or not. In the screenshot you can check the nginx server is showed.

Now lets try to access the server by typing our server public ip address in chrome.


Microsoft Azure Portal | Micros
manoj-webserver - Microsoft A
4.206.155.140
4.206.155.140
Reload
Details

This site can't be reached

4.206.155.140 took too long to respond.

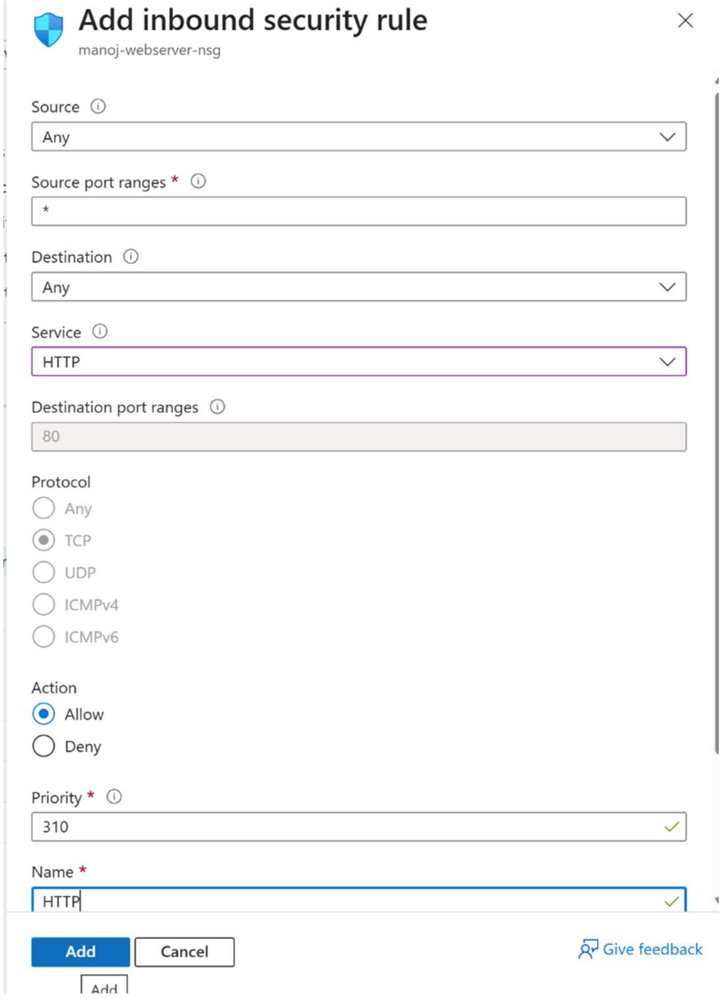
Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_TIMED_OUT

So we can see the server is not been acessed, so here we need to set **inbound rules** to allow http port no.80.

Go to web server and go to network setting and scroll down, there you can find inboud and outbound rules and set them as follow and click add.



Add inbound security rule

manoj-webserver-nsg

Source: Any

Source port ranges *: *

Destination: Any

Service: HTTP

Destination port ranges: 80

Protocol

○ Any
● TCP
○ UDP
○ ICMPv4
○ ICMPv6

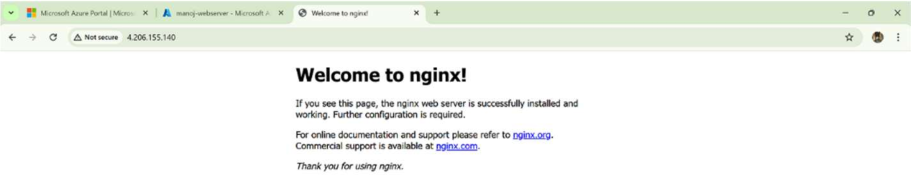
Action

● Allow
○ Deny

Priority *: 310

Name *: HTTP

Now anyone can access the nginx server, now go to chrome and search with our public ip.



Microsoft Azure Portal | Micros
manoj-webserver - Microsoft A
Welcome to nginx!
Not secure

4.206.155.140

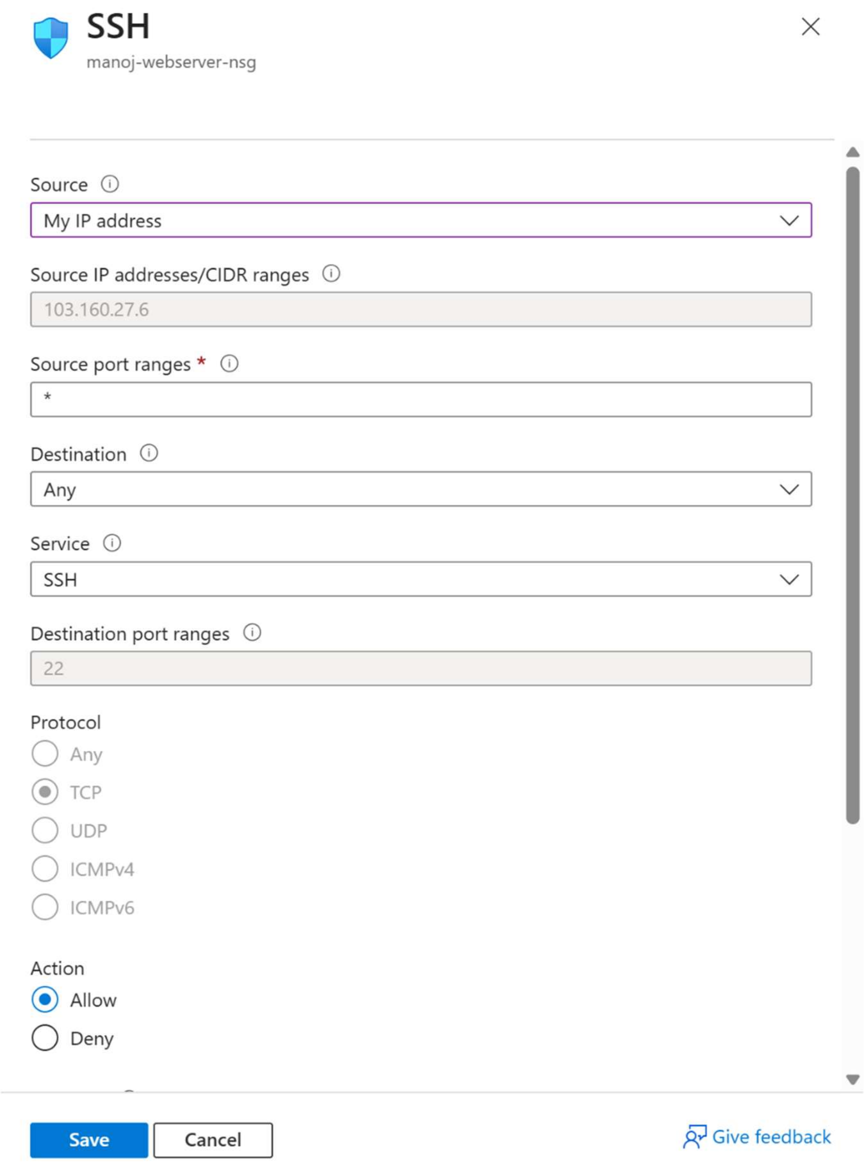
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

So the nginx server is being able to access, and we also need to change the ssh inbound rule, because only the host should have the access to the main host server. Now go to network settings and click on ssh edit and give the ip address of the host machine to access the main web server.



SSH
manoj-webserver-nsg

Source: My IP address

Source IP addresses/CIDR ranges: 103.160.27.6

Source port ranges *: *

Destination: Any

Service: SSH

Destination port ranges: 22

Protocol
○ Any
● TCP
○ UDP
○ ICMPv4
○ ICMPv6

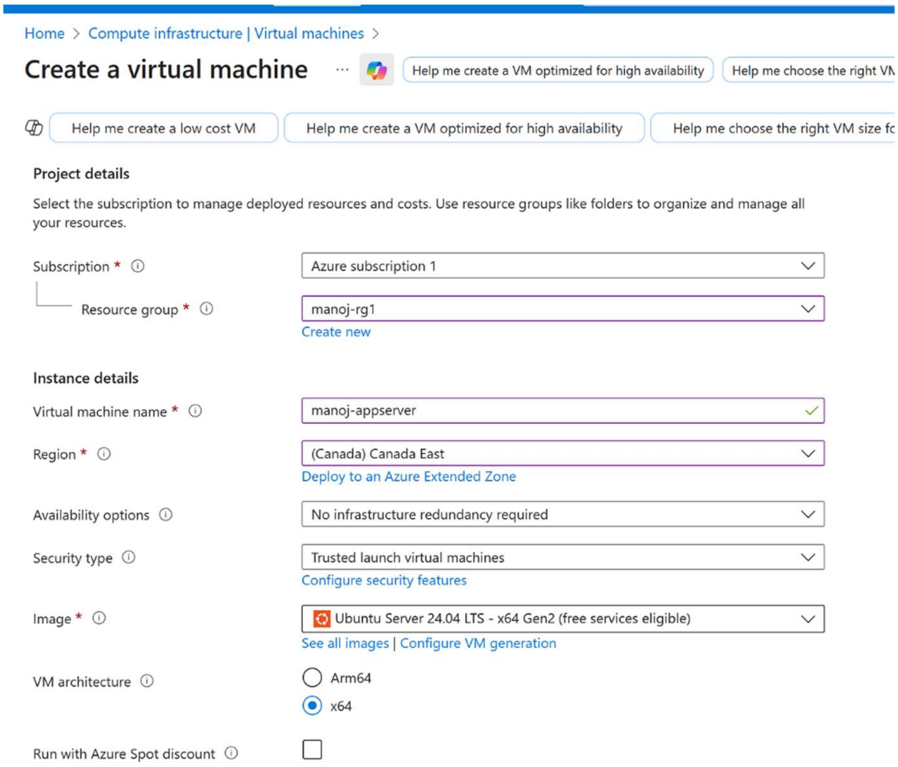
Action
● Allow
○ Deny

App Server:

Now its time to create App Server, Search for Virtual Machines and click on create and select first option (virtual machine).

Follow the same steps while of WebServer while creating App Server VM as well, but there will be some changes.

The region will be different. The region should be as same as the 2nd VNet i.e; app-db-vnet. Here the both App server and DB Server will be in same VNet with different subnets.


Home > Compute infrastructure | Virtual machines >
Help me create a VM optimized for high availability
Help me choose the right VN
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size fc

Create a virtual machine

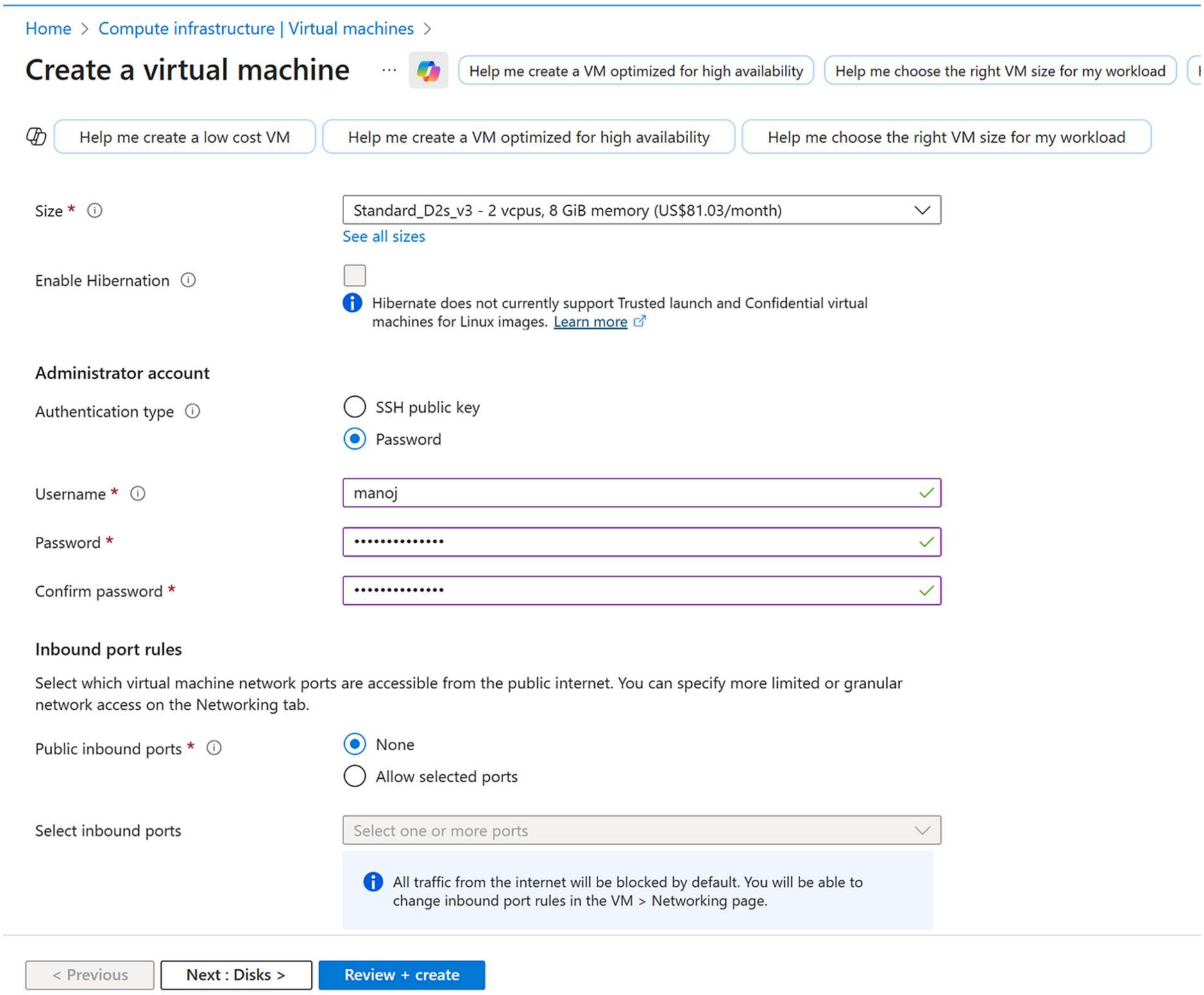
Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * ⓘ	Azure subscription 1
Resource group * ⓘ	manoj-rg1 Create new

Instance details

Virtual machine name * ⓘ	manoj-appserver ✓
Region * ⓘ	(Canada) Canada East Deploy to an Azure Extended Zone
Availability options ⓘ	No infrastructure redundancy required
Security type ⓘ	Trusted launch virtual machines Configure security features
Image * ⓘ	Ubuntu Server 24.04 LTS - x64 Gen2 (free services eligible) See all images | Configure VM generation
VM architecture ⓘ	○ Arm64 ● x64
Run with Azure Spot discount ⓘ	□


Home > Compute infrastructure | Virtual machines >
Help me create a VM optimized for high availability
Help me choose the right VM size for my workload
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my workload
< Previous
Next : Disks >
Review + create

Create a virtual machine

Size * ⓘ	Standard_D2s_v3 - 2 vcpus, 8 GiB memory (US$81.03/month) See all sizes
Enable Hibernation ⓘ	□ Hibernate does not currently support Trusted launch and Confidential virtual machines for Linux images. Learn more

Administrator account

Authentication type ⓘ	○ SSH public key ● Password
Username * ⓘ	manoj ✓
Password *	•••••••••••••• ✓
Confirm password *	•••••••••••••• ✓

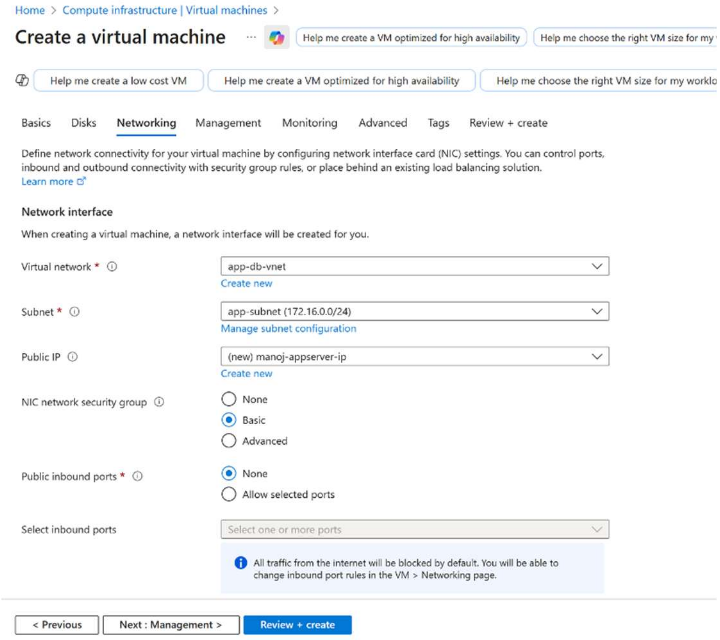
Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports * ⓘ	● None ○ Allow selected ports
Select inbound ports	Select one or more ports

All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Here deny the ports, we will add inbound rules later.



Home > Compute infrastructure | Virtual machines >
Help me create a VM optimized for high availability
Help me choose the right VM size for my
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my worklo
Basics
Disks
Networking
Management
Monitoring
Advanced
Tags
Review + create
< Previous
Next : Management >
Review + create

Create a virtual machine

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution.
Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * : app-db-vnet
Create new

Subnet * : app-subnet (172.16.0.0/24)
Manage subnet configuration

Public IP : (new) manoj-appserver-ip
Create new

NIC network security group: ○ None ● Basic ○ Advanced

Public inbound ports * : ● None ○ Allow selected ports

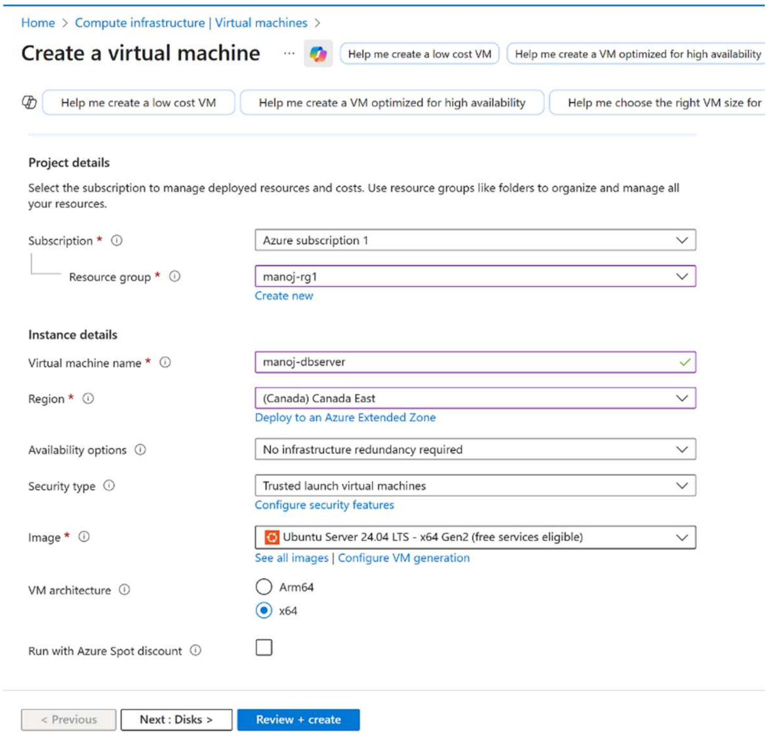
Select inbound ports: Select one or more ports

All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Select app-db-vnet and select 172.16.0.0/24 subnet for this server and click on review + create.

DB Server:

Now create another VM — DB Server as same as APP server by selecting different subnet 172.16.1.0/24.



Home > Compute infrastructure | Virtual machines >
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for
< Previous
Next : Disks >
Review + create

Create a virtual machine

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription * : Azure subscription 1

Resource group * : manoj-rg1
Create new

Instance details

Virtual machine name * : manoj-dbserver

Region * : (Canada) Canada East
Deploy to an Azure Extended Zone

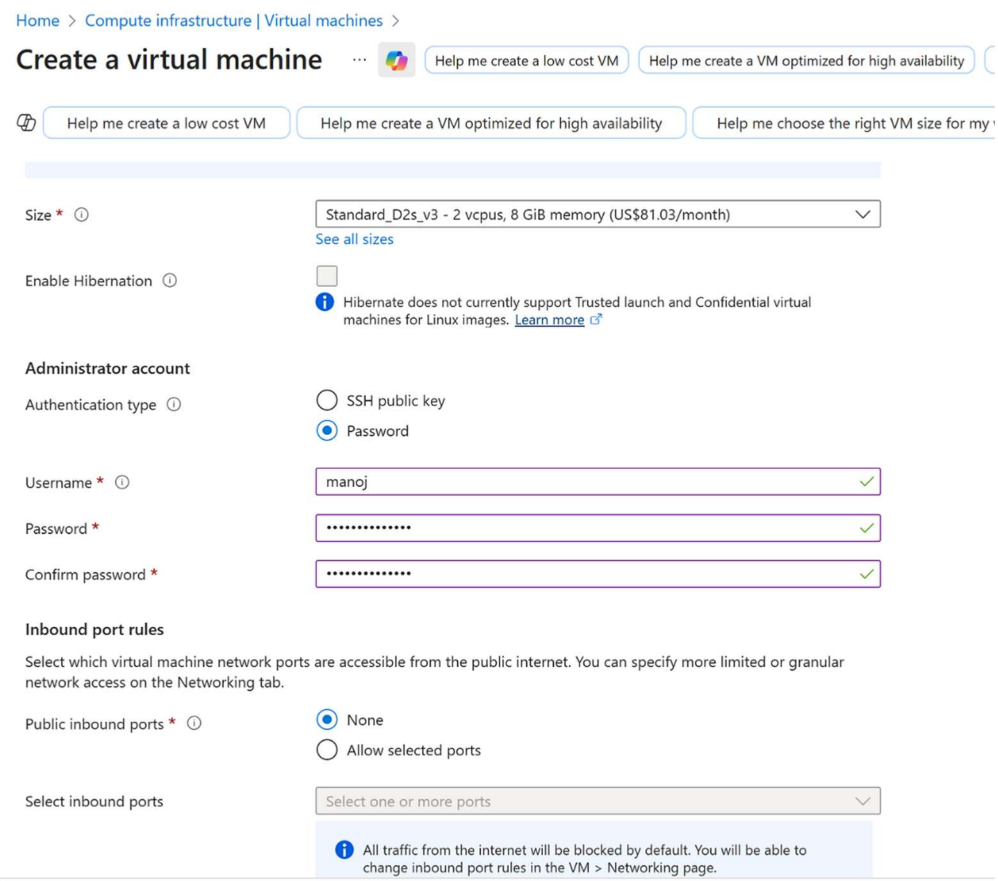
Availability options: No infrastructure redundancy required

Security type: Trusted launch virtual machines
Configure security features

Image * : Ubuntu Server 24.04 LTS - x64 Gen2 (free services eligible)
See all images | Configure VM generation

VM architecture: ○ Arm64 ● x64

Run with Azure Spot discount: □



Home > Compute infrastructure | Virtual machines >
Create a virtual machine ...
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my v

Size * : Standard_D2s_v3 - 2 vcpus, 8 GiB memory (US$81.03/month)

See all sizes

Enable Hibernation □

Hibernate does not currently support Trusted launch and Confidential virtual machines for Linux images. Learn more

Administrator account

Authentication type
○ SSH public key
● Password

Username * : manoj ✓

Password * : •••••••••••••• ✓

Confirm password * : •••••••••••••• ✓

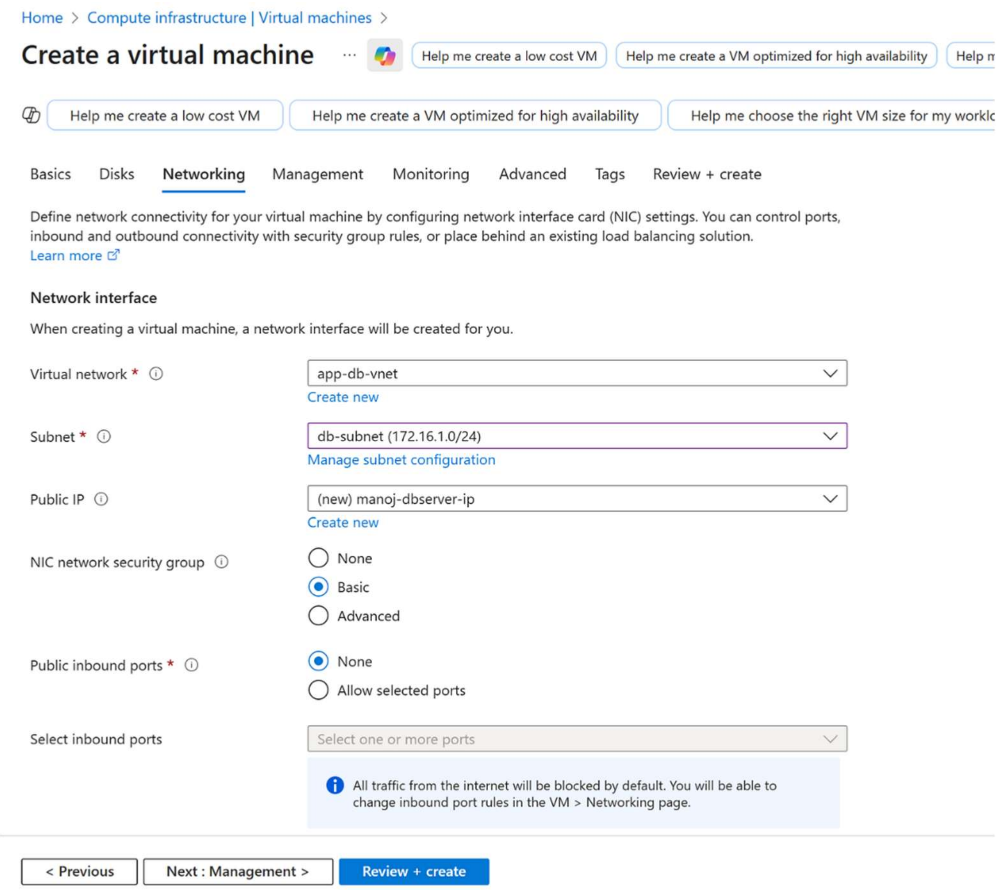
Inbound port rules

Select which virtual machine network ports are accessible from the public internet. You can specify more limited or granular network access on the Networking tab.

Public inbound ports *
● None
○ Allow selected ports

Select inbound ports: Select one or more ports

All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.



Home > Compute infrastructure | Virtual machines >
Create a virtual machine ...
Help me create a low cost VM
Help me create a VM optimized for high availability
Help m
Help me create a low cost VM
Help me create a VM optimized for high availability
Help me choose the right VM size for my worklo
Basics
Disks
Networking
Management
Monitoring
Advanced
Tags
Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. Learn more

Network interface

When creating a virtual machine, a network interface will be created for you.

Virtual network * : app-db-vnet

Create new

Subnet * : db-subnet (172.16.1.0/24)

Manage subnet configuration

Public IP : (new) manoj-dbserver-ip

Create new

NIC network security group
○ None
● Basic
○ Advanced

Public inbound ports *
● None
○ Allow selected ports

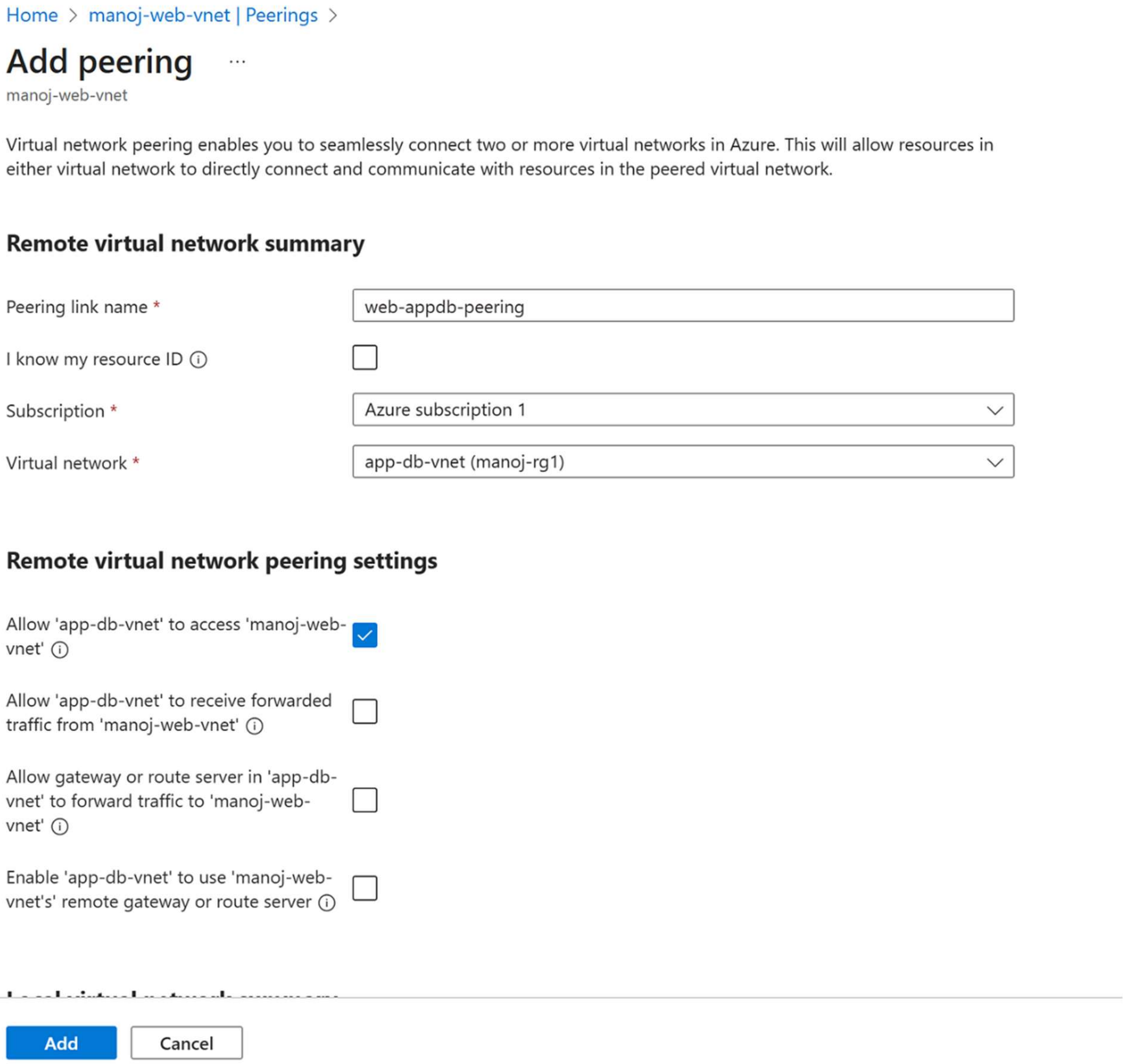
Select inbound ports: Select one or more ports

All traffic from the internet will be blocked by default. You will be able to change inbound port rules in the VM > Networking page.

Step 4: Setting Up Peering:

We need to do Network Peering because we used two different VNet's to deploy our servers and it is necessary that the three servers should communicate with each other, so we use peering.

Peering: Acts as bridge between two virtual networks.



Home > manoj-web-vnet | Peerings >

Add peering

manoj-web-vnet

Virtual network peering enables you to seamlessly connect two or more virtual networks in Azure. This will allow resources in either virtual network to directly connect and communicate with resources in the peered virtual network.

Remote virtual network summary

Peering link name *	web-appdb-peering
I know my resource ID ⓘ	□
Subscription *	Azure subscription 1
Virtual network *	app-db-vnet (manoj-rg1)

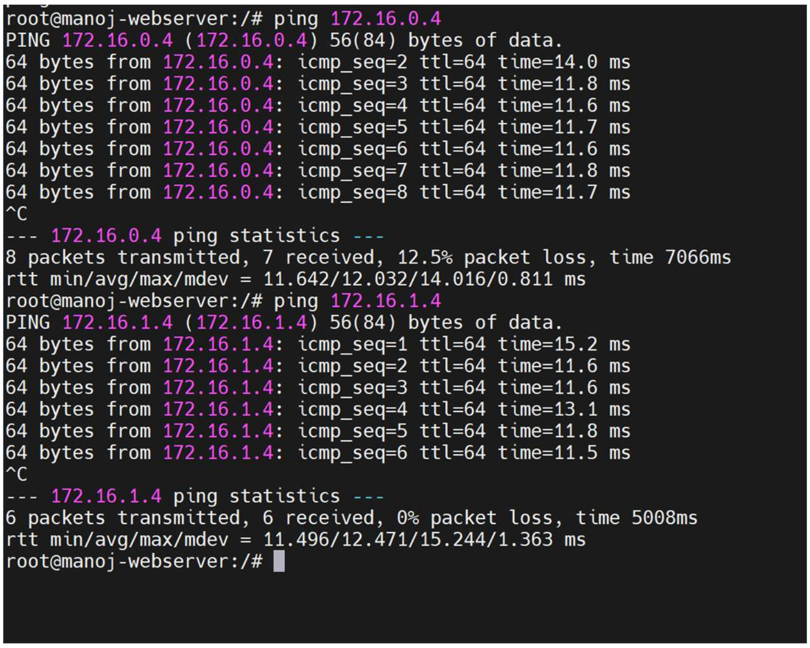
Remote virtual network peering settings

Allow 'app-db-vnet' to access 'manoj-web-vnet' ⓘ	☑
Allow 'app-db-vnet' to receive forwarded traffic from 'manoj-web-vnet' ⓘ	□
Allow gateway or route server in 'app-db-vnet' to forward traffic to 'manoj-web-vnet' ⓘ	□
Enable 'app-db-vnet' to use 'manoj-web-vnet's' remote gateway or route server ⓘ	□

Local virtual network summary

Go to any one of the two VNet's, ex:manoj-web-vnet and click on peering option from the left side window and give a name to the peering and select the virtual network (destination vnet) i.e; app-db-vnet and click on add.

This will add the peering and we can verify whether the server are communicating or not by using "ping" command.



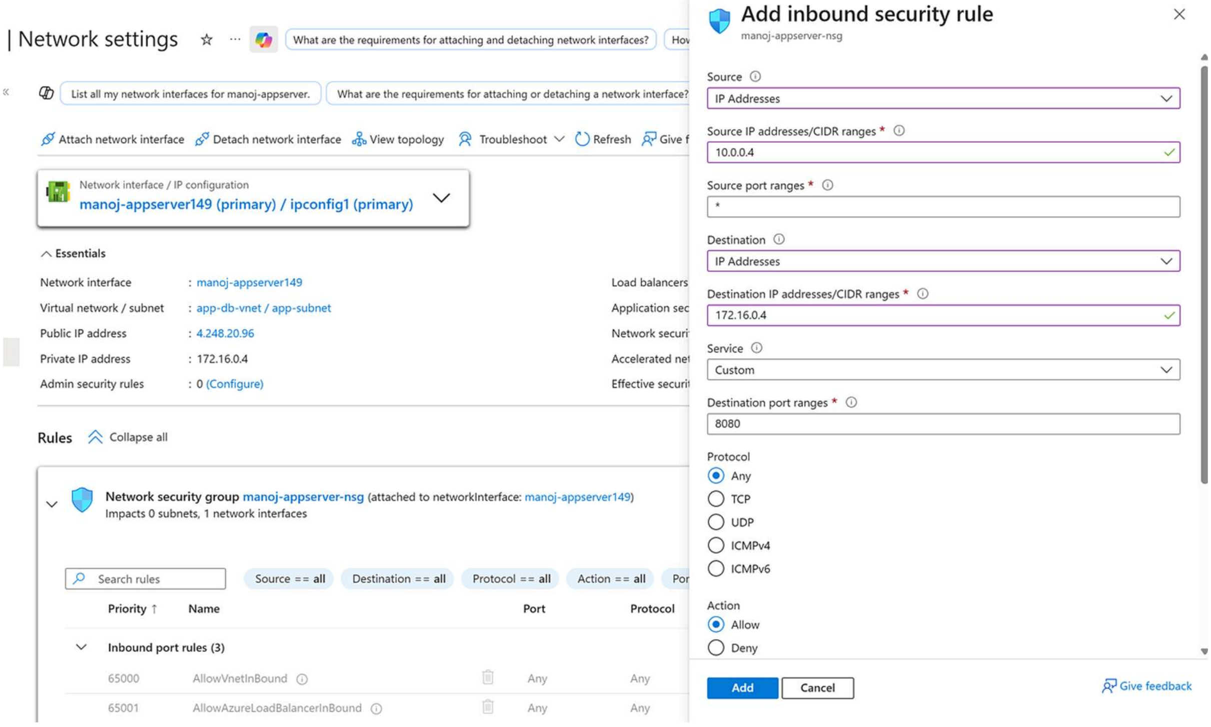
```
root@manoj-webserver:/# ping 172.16.0.4
PING 172.16.0.4 (172.16.0.4) 56(84) bytes of data.
64 bytes from 172.16.0.4: icmp_seq=2 ttl=64 time=14.0 ms
64 bytes from 172.16.0.4: icmp_seq=3 ttl=64 time=11.8 ms
64 bytes from 172.16.0.4: icmp_seq=4 ttl=64 time=11.6 ms
64 bytes from 172.16.0.4: icmp_seq=5 ttl=64 time=11.7 ms
64 bytes from 172.16.0.4: icmp_seq=6 ttl=64 time=11.6 ms
64 bytes from 172.16.0.4: icmp_seq=7 ttl=64 time=11.8 ms
64 bytes from 172.16.0.4: icmp_seq=8 ttl=64 time=11.7 ms
^C
--- 172.16.0.4 ping statistics ---
8 packets transmitted, 7 received, 12.5% packet loss, time 7066ms
rtt min/avg/max/mdev = 11.642/12.032/14.016/0.811 ms
root@manoj-webserver:/# ping 172.16.1.4
PING 172.16.1.4 (172.16.1.4) 56(84) bytes of data.
64 bytes from 172.16.1.4: icmp_seq=1 ttl=64 time=15.2 ms
64 bytes from 172.16.1.4: icmp_seq=2 ttl=64 time=11.6 ms
64 bytes from 172.16.1.4: icmp_seq=3 ttl=64 time=11.6 ms
64 bytes from 172.16.1.4: icmp_seq=4 ttl=64 time=13.1 ms
64 bytes from 172.16.1.4: icmp_seq=5 ttl=64 time=11.8 ms
64 bytes from 172.16.1.4: icmp_seq=6 ttl=64 time=11.5 ms
^C
--- 172.16.1.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 11.496/12.471/15.244/1.363 ms
root@manoj-webserver:/# 
```

Step 5: Adding Inbound and Outbound Rules:

Web Server (Frontend): for web server we already added the inbound rules above and it should be accessed by public as it is the frontend page.

App Server (Backend): App Server should be private and should not be access by public.

- App server should only communicate with web server, so set the rules as follow.



Network interface / IP configuration
manoj-appserver149 (primary) / ipconfig1 (primary)

Network interface : manoj-appserver149
Virtual network / subnet : app-db-vnet / app-subnet
Public IP address : 4.248.20.96
Private IP address : 172.16.0.4
Admin security rules : 0 (Configure)

Network security group manoj-appserver-nsg (attached to networkInterface: manoj-appserver149)
Impacts 0 subnets, 1 network interfaces

Priority ↑	Name	Port	Protocol
Inbound port rules (3)			
65000	AllowVnetInBound	Any	Any
65001	AllowAzureLoadBalancerInBound	Any	Any

Add inbound security rule
manoj-appserver-nsg

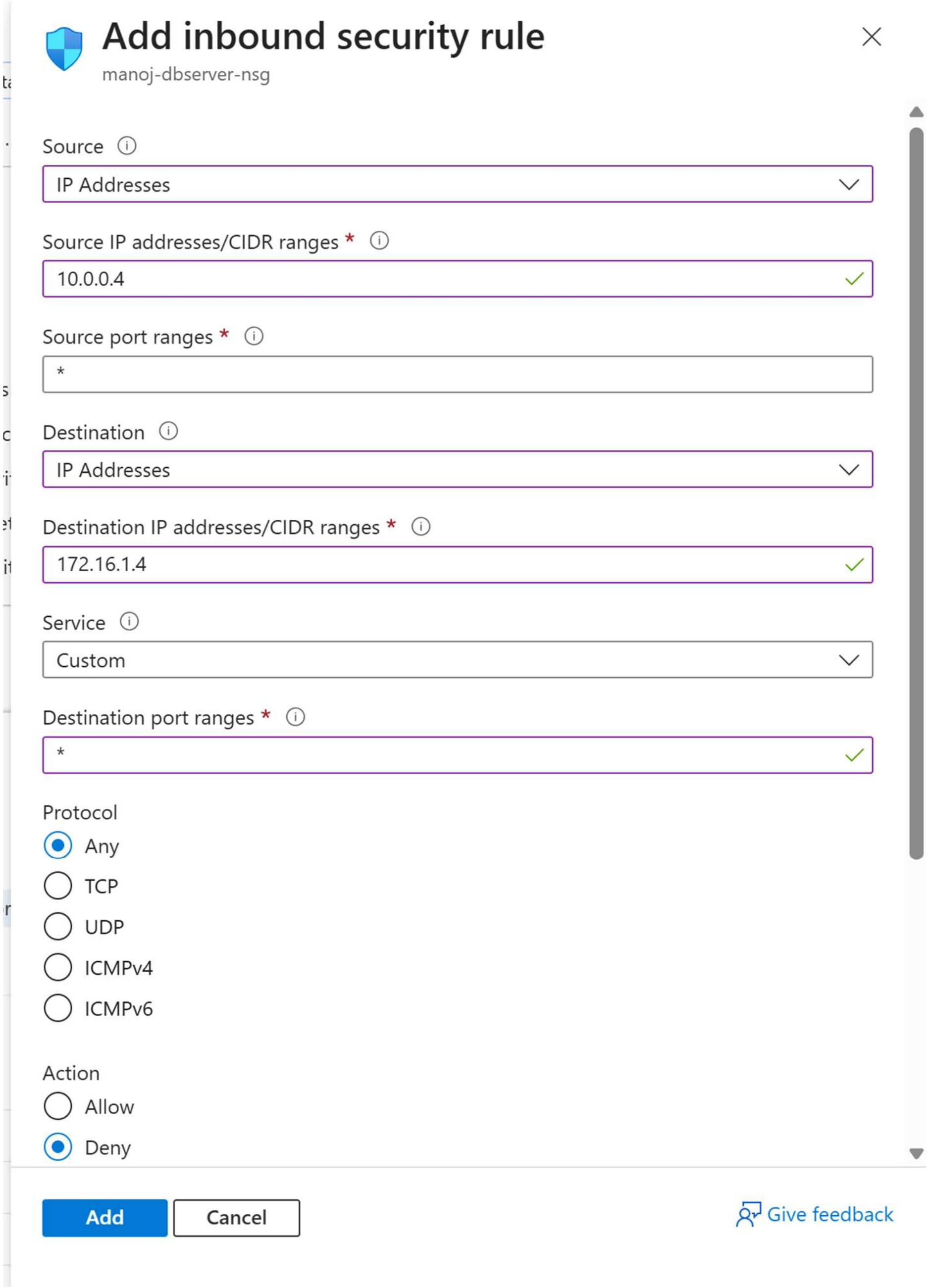
Source: IP Addresses
Source IP addresses/CIDR ranges *: 10.0.0.4
Source port ranges *: *
Destination: IP Addresses
Destination IP addresses/CIDR ranges *: 172.16.0.4
Service: Custom
Destination port ranges *: 8080

Protocol
● Any
○ TCP
○ UDP
○ ICMPv4
○ ICMPv6

Action
● Allow
○ Deny

Here the source should be the private ip of webserver (10.0.0.4) and destination should be app server private ip (172.16.0.4).

DB Server (Database): The DB Server should not have connection with the web server (frontend), so we deny all ports or requests from webserver, here we give source webserver (10.0.0.4) and destination db server (172.16.1.4) private anddress and select any ports and select deny.



Add inbound security rule

manoj-dbserver-nsg

Source: IP Addresses

Source IP addresses/CIDR ranges *: 10.0.0.4

Source port ranges *: *

Destination: IP Addresses

Destination IP addresses/CIDR ranges *: 172.16.1.4

Service: Custom

Destination port ranges *: *

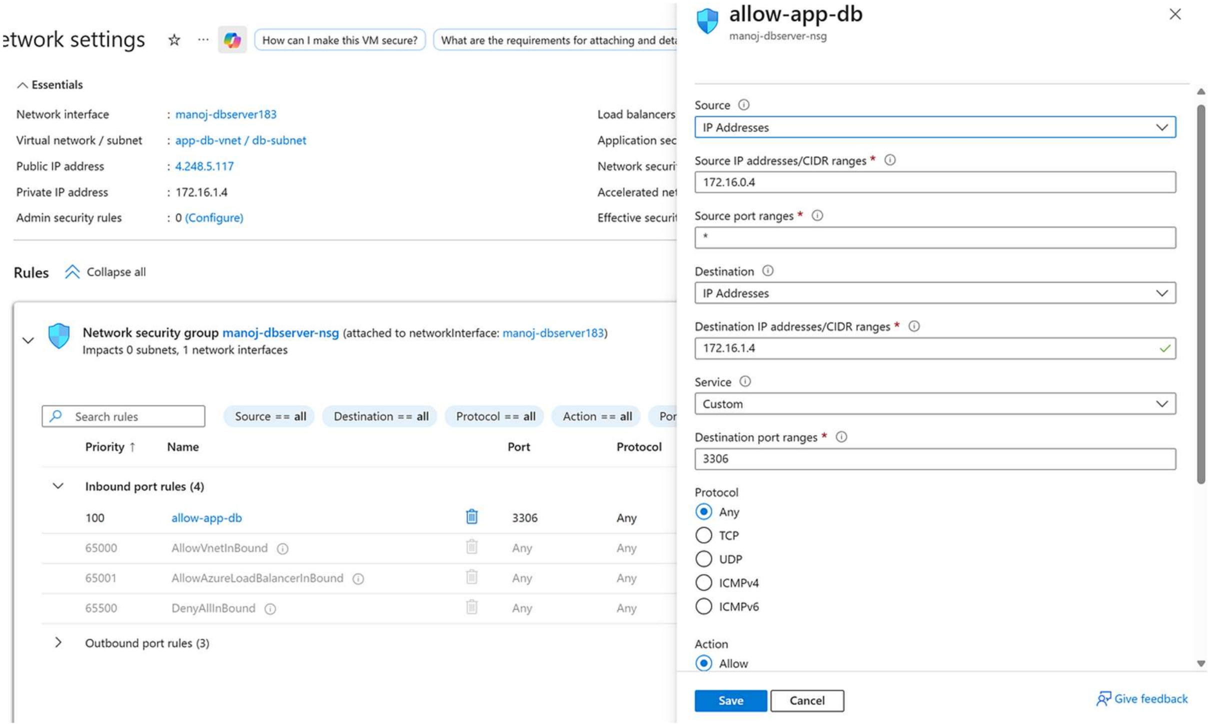
Protocol

● Any
○ TCP
○ UDP
○ ICMPv4
○ ICMPv6

Action

○ Allow
● Deny

But database make requests and response from the app server so we give 3306 — mysql port access by giving appserver private ip as source and db server private ip as destinationand allow 3306 port.



etwork settings ☆ ···
How can I make this VM secure?
What are the requirements for attaching and deta

Essentials

Network interface : manoj-dbserver183

Virtual network / subnet : app-db-vnet / db-subnet

Public IP address : 4.248.5.117

Private IP address : 172.16.1.4

Admin security rules : 0 (Configure)

Load balancers

Application sec

Network securi

Accelerated ne

Effective securi

Rules Collapse all

Network security group manoj-dbserver-nsg (attached to networkInterface: manoj-dbserver183)
Impacts 0 subnets, 1 network interfaces

Search rules | Source == all | Destination == all | Protocol == all | Action == all | Por

Priority ↑	Name	Port	Protocol
Inbound port rules (4)			
100	allow-app-db	3306	Any
65000	AllowVnetInBound	Any	Any
65001	AllowAzureLoadBalancerInBound	Any	Any
65500	DenyAllInBound	Any	Any
Outbound port rules (3)			

allow-app-db
manoj-dbserver-nsg

Source: IP Addresses

Source IP addresses/CIDR ranges *: 172.16.0.4

Source port ranges *: *

Destination: IP Addresses

Destination IP addresses/CIDR ranges *: 172.16.1.4

Service: Custom

Destination port ranges *: 3306

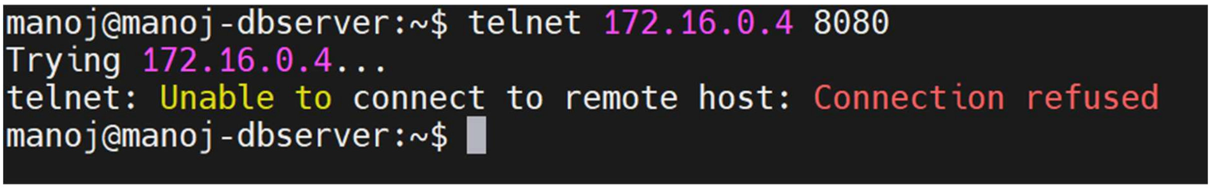
Protocol
● Any
○ TCP
○ UDP
○ ICMPv4
○ ICMPv6

Action
● Allow

Step 6: Installing and Configuring Tomcat in AppServer:

- So the setting up of all servers are done and now lets see the connectivity between the,

webserver and app server - telnet 172.16.0.4 8080 , command. Here we will get error because we didnot install tomcat in appserver.

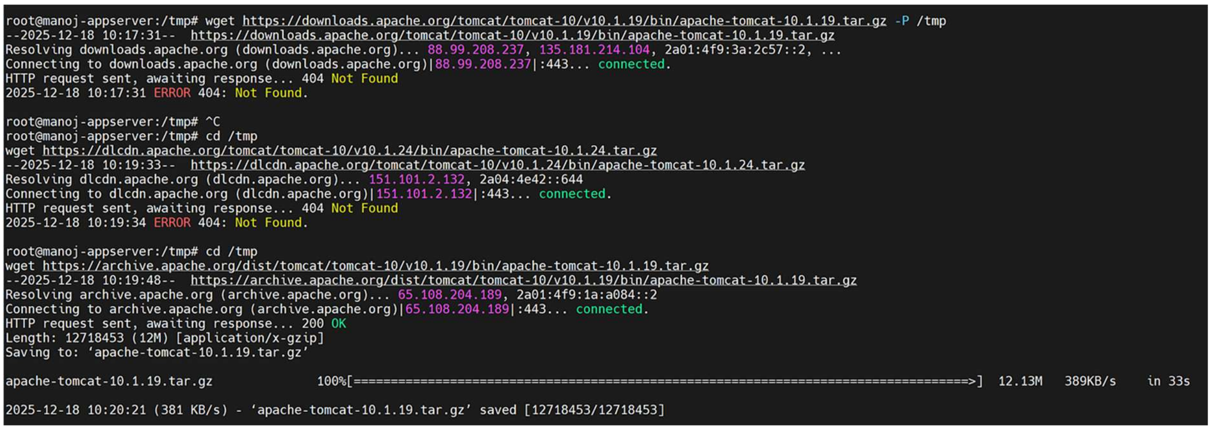


```
manoj@manoj-dbserver:~$ telnet 172.16.0.4 8080
Trying 172.16.0.4...
telnet: Unable to connect to remote host: Connection refused
manoj@manoj-dbserver:~$ 
```

So follow the below commands to install and run tomcat in appserver. Here we can directly run appserver from webserver by the using SSH command: **ssh manoj@172.16.0.4,** we can directly move to appserver and execute the below commands from root user.

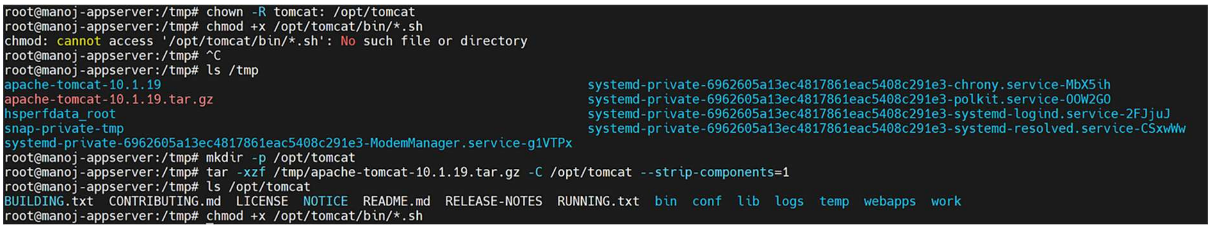
- apt update
- apt install openjdk-11-jdk -y
- useradd -m -U -d /opt/tomcat -s /bin/false tomcat - to create tomcat user
- cd /tmp
  wget https://archive.apache.org/dist/tomcat/tomcat10/v10.1.19/bin/apache-tomcat-10.1.19.tar.gz - to install tomcat from internet

- mkdir -p /opt/tomcat - create a directory
- tar -xzf /tmp/apache-tomcat-10.1.19.tar.gz -C /opt/tomcat — strip-components=1 - extract after download in the created directory



```
root@manoj-appserver:/tmp# wget https://downloads.apache.org/tomcat/tomcat-10/v10.1.19/bin/apache-tomcat-10.1.19.tar.gz -P /tmp
--2025-12-18 10:17:31--  https://downloads.apache.org/tomcat/tomcat-10/v10.1.19/bin/apache-tomcat-10.1.19.tar.gz
Resolving downloads.apache.org (downloads.apache.org)... 88.99.208.237, 135.181.214.104, 2a01:4f9:3a:2c57::2, ...
Connecting to downloads.apache.org (downloads.apache.org)|88.99.208.237|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2025-12-18 10:17:31 ERROR 404: Not Found.

root@manoj-appserver:/tmp# ^C
root@manoj-appserver:/tmp# cd /tmp
wget https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.24/bin/apache-tomcat-10.1.24.tar.gz
--2025-12-18 10:19:33--  https://dlcdn.apache.org/tomcat/tomcat-10/v10.1.24/bin/apache-tomcat-10.1.24.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2025-12-18 10:19:34 ERROR 404: Not Found.

root@manoj-appserver:/tmp# cd /tmp
wget https://archive.apache.org/dist/tomcat/tomcat-10/v10.1.19/bin/apache-tomcat-10.1.19.tar.gz
--2025-12-18 10:19:48--  https://archive.apache.org/dist/tomcat/tomcat-10/v10.1.19/bin/apache-tomcat-10.1.19.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 12718453 (12M) [application/x-gzip]
Saving to: ‘apache-tomcat-10.1.19.tar.gz’

apache-tomcat-10.1.19.tar.gz           100%[===============================================================================>]  12.13M   389KB/s    in 33s

2025-12-18 10:20:21 (381 KB/s) - ‘apache-tomcat-10.1.19.tar.gz’ saved [12718453/12718453]
```

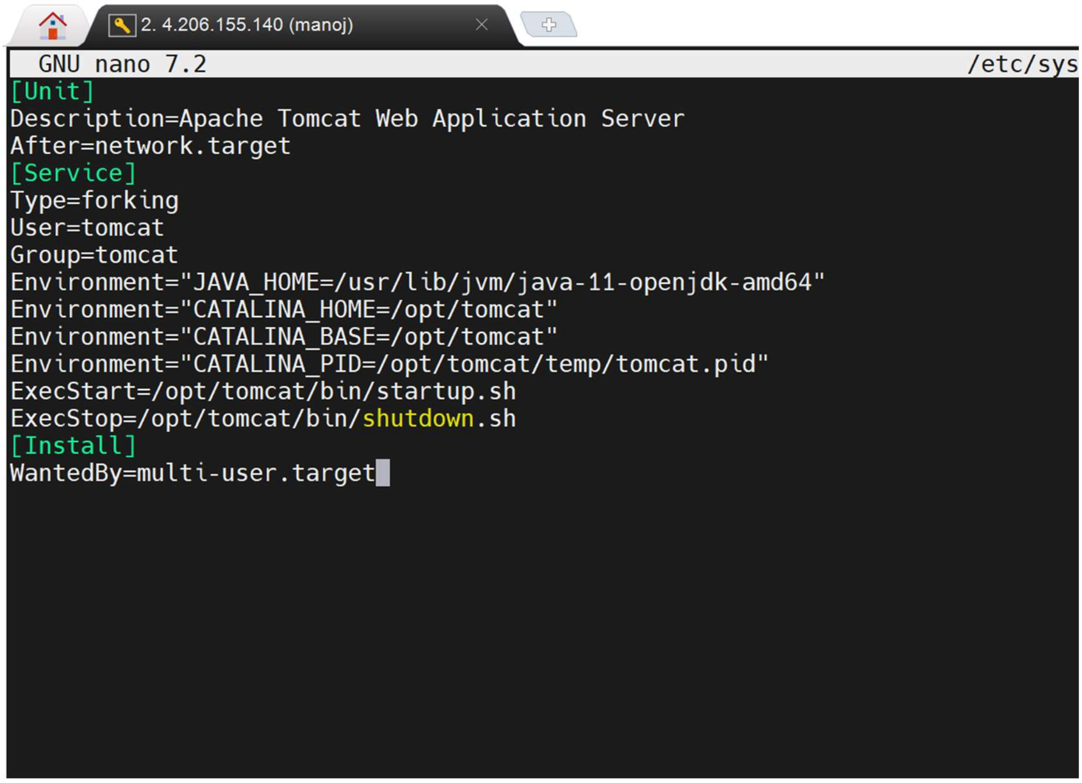
- ls /opt/tomcat - must see (bin conf lib logs webapps work temp)



```
root@manoj-appserver:/tmp# chown -R tomcat: /opt/tomcat
root@manoj-appserver:/tmp# chmod +x /opt/tomcat/bin/*.sh
chmod: cannot access '/opt/tomcat/bin/*.sh': No such file or directory
root@manoj-appserver:/tmp# ^C
root@manoj-appserver:/tmp# ls /tmp
apache-tomcat-10.1.19                                                        systemd-private-6962605a13ec4817861eac5408c291e3-chrony.service-MbX5ih
apache-tomcat-10.1.19.tar.gz                                                 systemd-private-6962605a13ec4817861eac5408c291e3-polkit.service-0OW2GO
hsperfdata_root                                                              systemd-private-6962605a13ec4817861eac5408c291e3-systemd-logind.service-2FJjuJ
snap-private-tmp                                                             systemd-private-6962605a13ec4817861eac5408c291e3-systemd-resolved.service-CSxwWw
systemd-private-6962605a13ec4817861eac5408c291e3-ModemManager.service-g1VTPx
root@manoj-appserver:/tmp# mkdir -p /opt/tomcat
root@manoj-appserver:/tmp# tar -xzf /tmp/apache-tomcat-10.1.19.tar.gz -C /opt/tomcat --strip-components=1
root@manoj-appserver:/tmp# ls /opt/tomcat
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  README.md  RELEASE-NOTES  RUNNING.txt  bin  conf  lib  logs  temp  webapps  work
root@manoj-appserver:/tmp# chmod +x /opt/tomcat/bin/*.sh
```

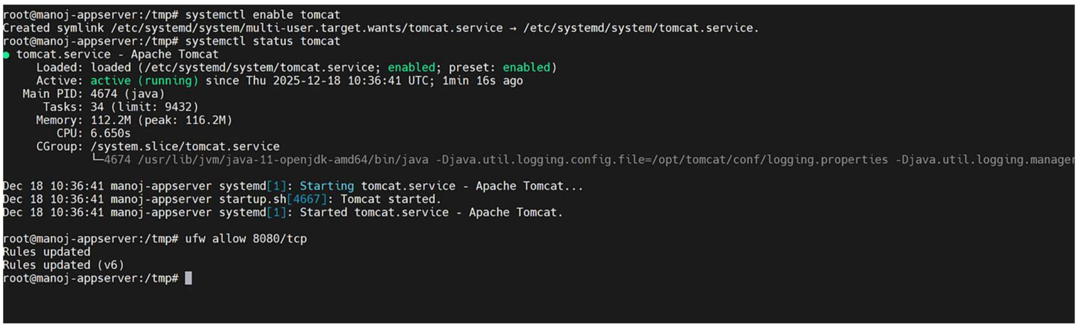
- chown -R tomcat:tomcat /opt/tomcat - change ownership
- chmod +x /opt/tomcat/bin/*.sh - change permission
- nano /etc/systemd/system/tomcat.service - to create systemed file

```
[Unit]
Description=Apache Tomcat
After=network.target

[Service]
Type=forking
User=tomcat
Group=tomcat

Environment=JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
Environment=CATALINA_HOME=/opt/tomcat
Environment=CATALINA_BASE=/opt/tomcat

ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh

[Install]
WantedBy=multi-user.target
```



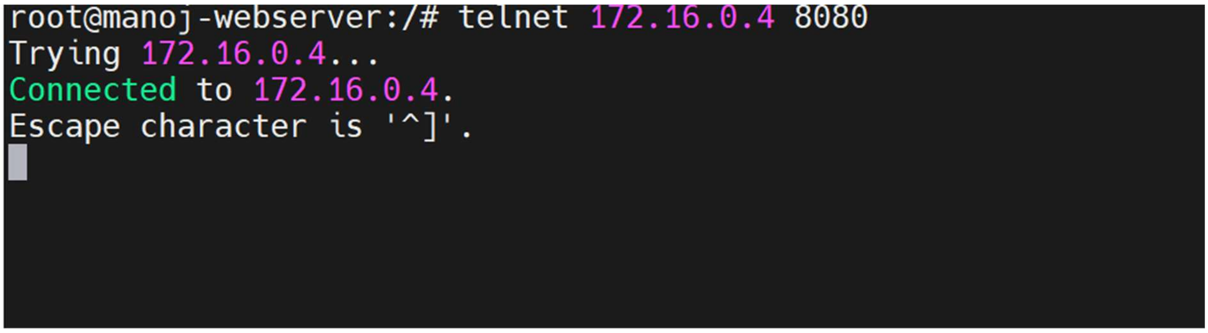
```
  GNU nano 7.2                                                   /etc/sys
[Unit]
Description=Apache Tomcat Web Application Server
After=network.target
[Service]
Type=forking
User=tomcat
Group=tomcat
Environment="JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64"
Environment="CATALINA_HOME=/opt/tomcat"
Environment="CATALINA_BASE=/opt/tomcat"
Environment="CATALINA_PID=/opt/tomcat/temp/tomcat.pid"
ExecStart=/opt/tomcat/bin/startup.sh
ExecStop=/opt/tomcat/bin/shutdown.sh
[Install]
WantedBy=multi-user.target
```

- systemctl daemon-reload
- systemctl start tomcat
- systemctl enable tomcat



```
root@manoj-appserver:/tmp# systemctl enable tomcat
Created symlink /etc/systemd/system/multi-user.target.wants/tomcat.service → /etc/systemd/system/tomcat.service.
root@manoj-appserver:/tmp# systemctl status tomcat
● tomcat.service - Apache Tomcat
     Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-12-18 10:36:41 UTC; 1min 16s ago
   Main PID: 4674 (java)
      Tasks: 34 (limit: 9432)
     Memory: 112.2M (peak: 116.2M)
        CPU: 6.650s
     CGroup: /system.slice/tomcat.service
             └─4674 /usr/lib/jvm/java-11-openjdk-amd64/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager

Dec 18 10:36:41 manoj-appserver systemd[1]: Starting tomcat.service - Apache Tomcat...
Dec 18 10:36:41 manoj-appserver startup.sh[4667]: Tomcat started.
Dec 18 10:36:41 manoj-appserver systemd[1]: Started tomcat.service - Apache Tomcat.

root@manoj-appserver:/tmp# ufw allow 8080/tcp
Rules updated
Rules updated (v6)
root@manoj-appserver:/tmp# 
```

- ss -tulnp | grep 8080 — to check 8080 port is open or not
- Now go to webserver root address and retype the telnet command: telnet 172.16.0.4 8080



```
root@manoj-webserver:/# telnet 172.16.0.4 8080
Trying 172.16.0.4...
Connected to 172.16.0.4.
Escape character is '^]'.
```

Step 7: Installing and Configuring MySql database in DB Server:

Now we need to install MySql database in DB Server and try to connect from appserver to dbserver using: telnet 172.16.1.4

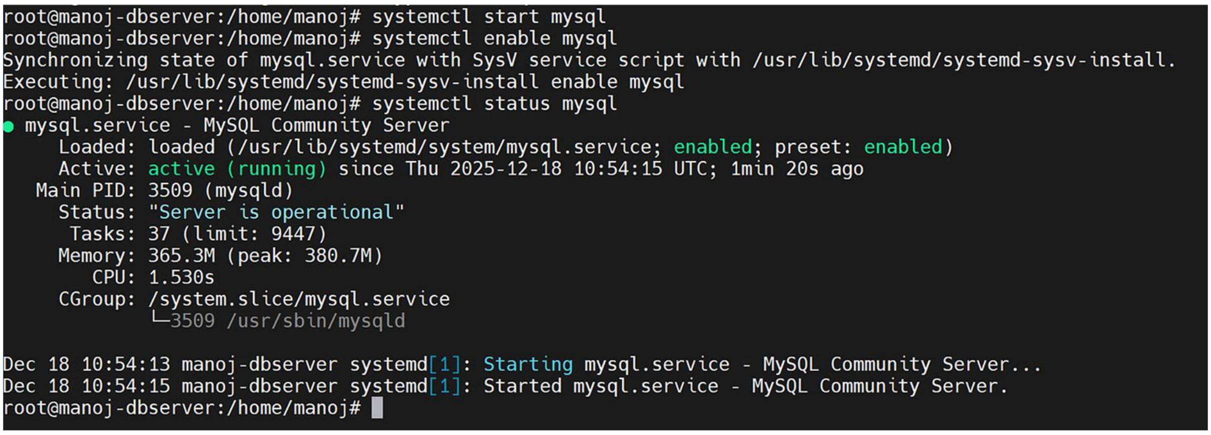
Now lets connect to app server from web server usinf SSH command and then from app server we connect to DB server using SSH command :

ssh manoj@172.16.1.4

We cannot directly connect to dbserver from webserver because we denied all the ports communication of dbserver with webserver.

Execute the following commands to install and configure MySql database:

- apt update
- apt install mysql-server -y
- systemctl start mysql
- systemctl enable mysql
- systemctl status mysql



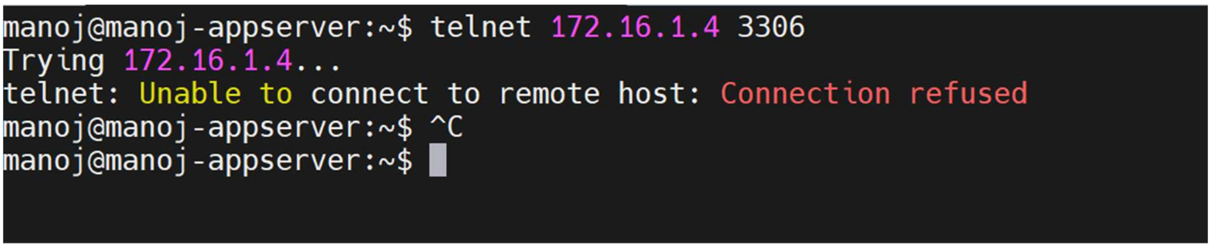
```
root@manoj-dbserver:/home/manoj# systemctl start mysql
root@manoj-dbserver:/home/manoj# systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mysql
root@manoj-dbserver:/home/manoj# systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-12-18 10:54:15 UTC; 1min 20s ago
   Main PID: 3509 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 9447)
     Memory: 365.3M (peak: 380.7M)
        CPU: 1.530s
     CGroup: /system.slice/mysql.service
             └─3509 /usr/sbin/mysqld

Dec 18 10:54:13 manoj-dbserver systemd[1]: Starting mysql.service - MySQL Community Server...
Dec 18 10:54:15 manoj-dbserver systemd[1]: Started mysql.service - MySQL Community Server.
root@manoj-dbserver:/home/manoj# 
```

- mysql_secure_installation - and press yes for all
- ss -tulnp | grep 3306

tcp LISTEN 0 70 127.0.0.1:33060 0.0.0.0:* users:((“mysqld”,pid=3509,fd=21)) tcp LISTEN 0 151 127.0.0.1:3306 0.0.0.0:* users:((“mysqld”,pid=3509,fd=23))

You will get this msg after executing the above command, this indicates the port number is running and listening correctly.

Now try to connect db server from app server, but the connection will be refused.

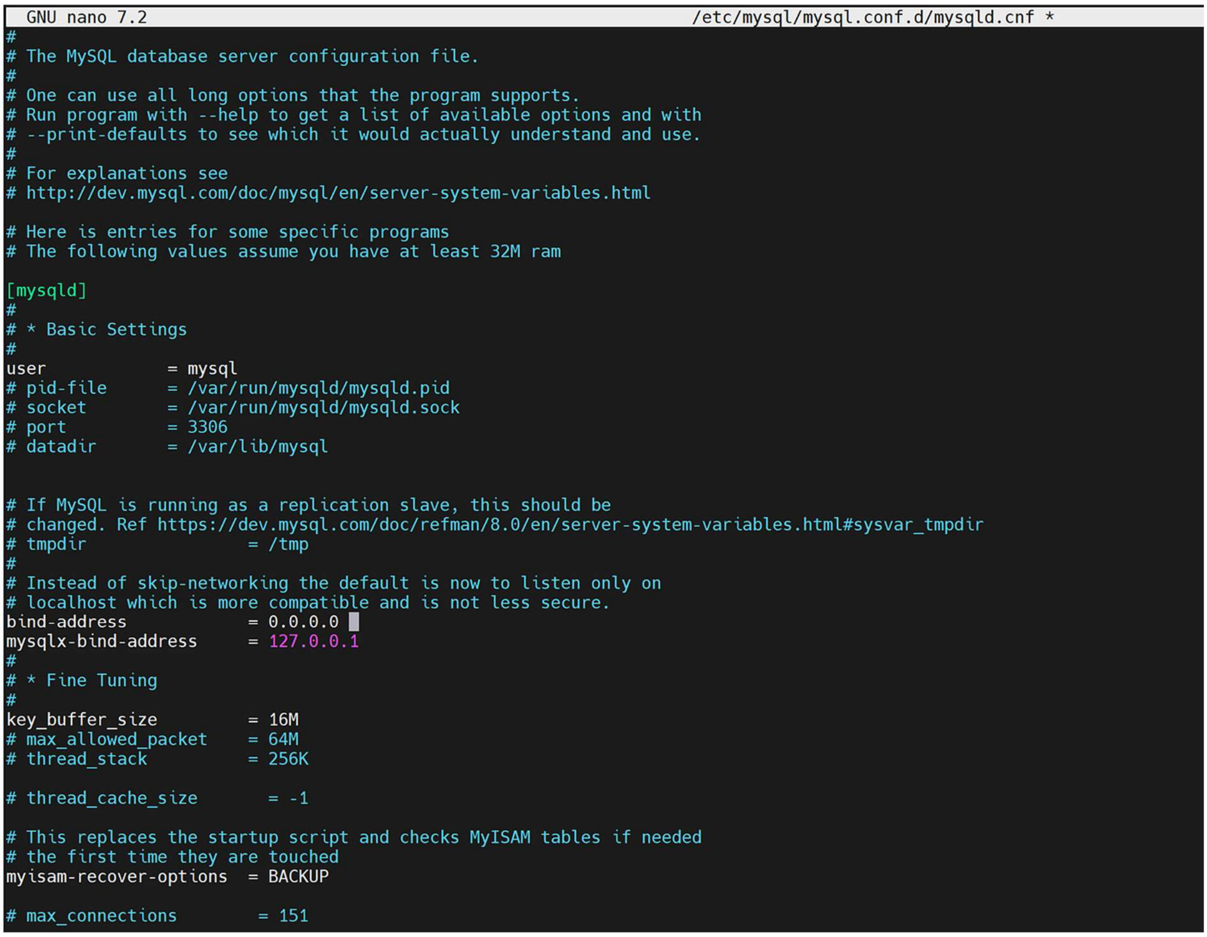


```
manoj@manoj-appserver:~$ telnet 172.16.1.4 3306
Trying 172.16.1.4...
telnet: Unable to connect to remote host: Connection refused
manoj@manoj-appserver:~$ ^C
manoj@manoj-appserver:~$ 
```

This is because in above grep command the ip is 127.0.0.1:3306, so it will connect to only that ip, so we need to chane it to 0.0.0.0:3306 to connect using any ip address. Go to db server again and execute

- nano /etc/mysql/mysql.conf.d/mysqld.cnf

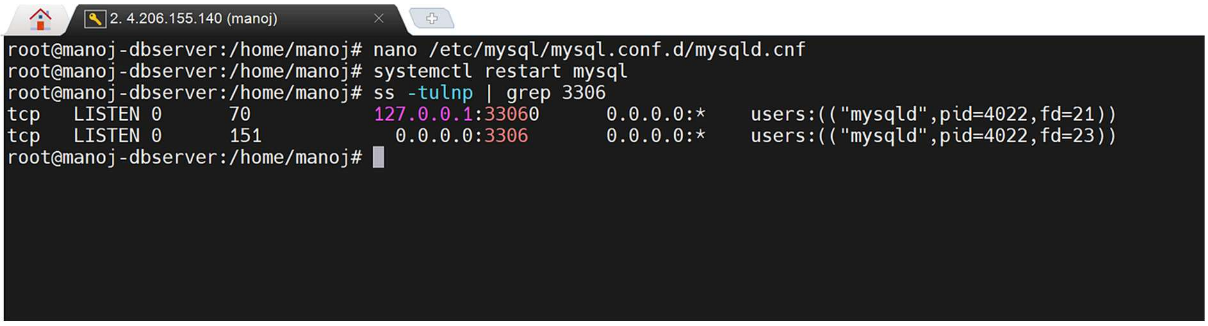
execute this cmd you will get as below image



```
  GNU nano 7.2                                             /etc/mysql/mysql.conf.d/mysqld.cnf *
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0 
mysqlx-bind-address     = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K

# thread_cache_size       = -1

# This replaces the startup script and checks MyISAM tables if needed
# the first time they are touched
myisam-recover-options  = BACKUP

# max_connections        = 151
```

Here change the bind-address from 127.0.0.1 to 0.0.0.0

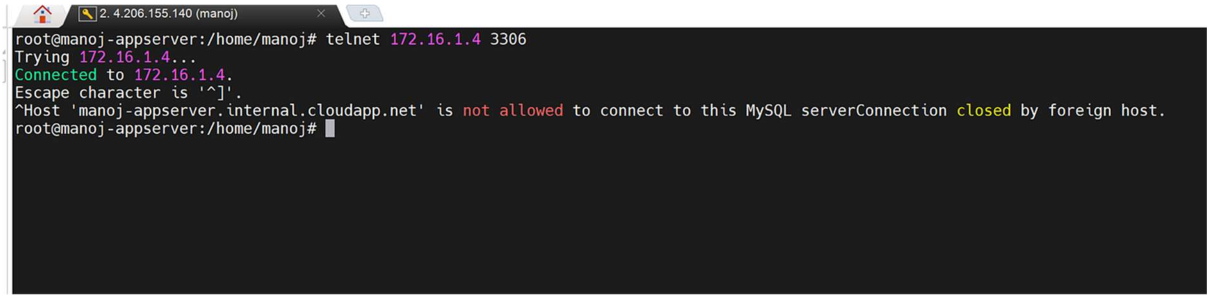
- again execute: ss -tulnp | grep 3306



```
root@manoj-dbserver:/home/manoj# nano /etc/mysql/mysql.conf.d/mysqld.cnf
root@manoj-dbserver:/home/manoj# systemctl restart mysql
root@manoj-dbserver:/home/manoj# ss -tulnp | grep 3306
tcp   LISTEN 0      70           127.0.0.1:33060      0.0.0.0:*    users:(("mysqld",pid=4022,fd=21))
tcp   LISTEN 0      151            0.0.0.0:3306       0.0.0.0:*    users:(("mysqld",pid=4022,fd=23))
root@manoj-dbserver:/home/manoj# 
```

127.0.0.0:3306 is changed to 0.0.0.0:3306

Now try to connect to database server from appserver,



```
root@manoj-appserver:/home/manoj# telnet 172.16.1.4 3306
Trying 172.16.1.4...
Connected to 172.16.1.4.
Escape character is '^]'.
^Host 'manoj-appserver.internal.cloudapp.net' is not allowed to connect to this MySQL serverConnection closed by foreign host.
root@manoj-appserver:/home/manoj# 
```

So the connection is established between appserver and dbserver.

Conclusion:

This project successfully demonstrates the implementation of a **3-tier architecture on Microsoft Azure**, following industry best practices for scalability, security, and maintainability. By separating the Web Server, Application Server, and Database Server into individual tiers, the system achieves better isolation, easier management, and improved performance.

The use of **Azure Virtual Networks, subnets, and Network Security Groups** ensures secure communication between tiers while restricting unauthorized access. The Web Server acts as the public entry point, the Application Server handles business logic using Apache Tomcat, and the Database Server securely stores data using MySQL in a private subnet.

Overall, this architecture reflects a real-world enterprise deployment model and provides a strong foundation for building scalable cloud applications. The project highlights how Azure services can be effectively used to design and deploy a secure multi-tier application environment.