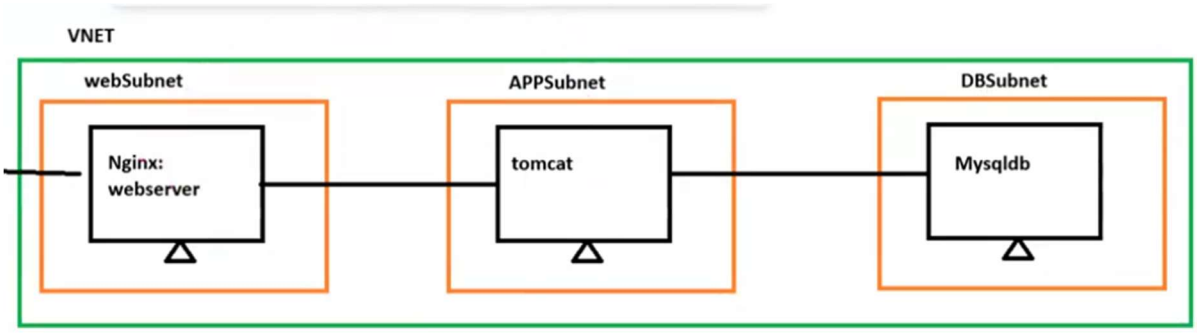
# Three Tier Architecture

→In the three architecture here we can connect web server to app server and app server to db server.

→ It is mainly used for connecting the real world application.

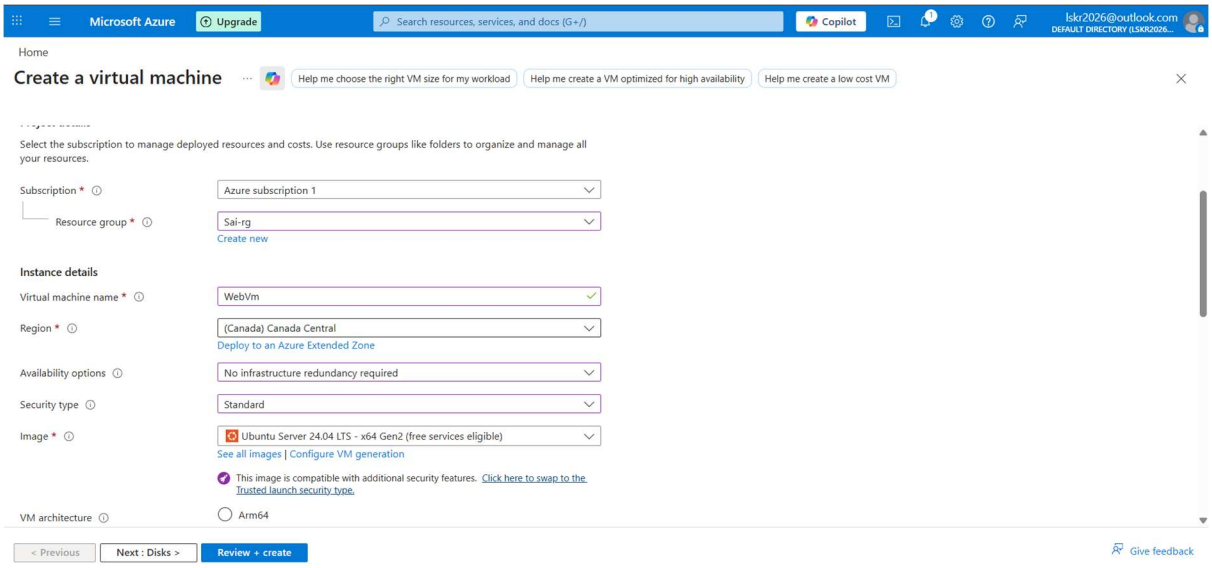
→Here we can connect three machines.

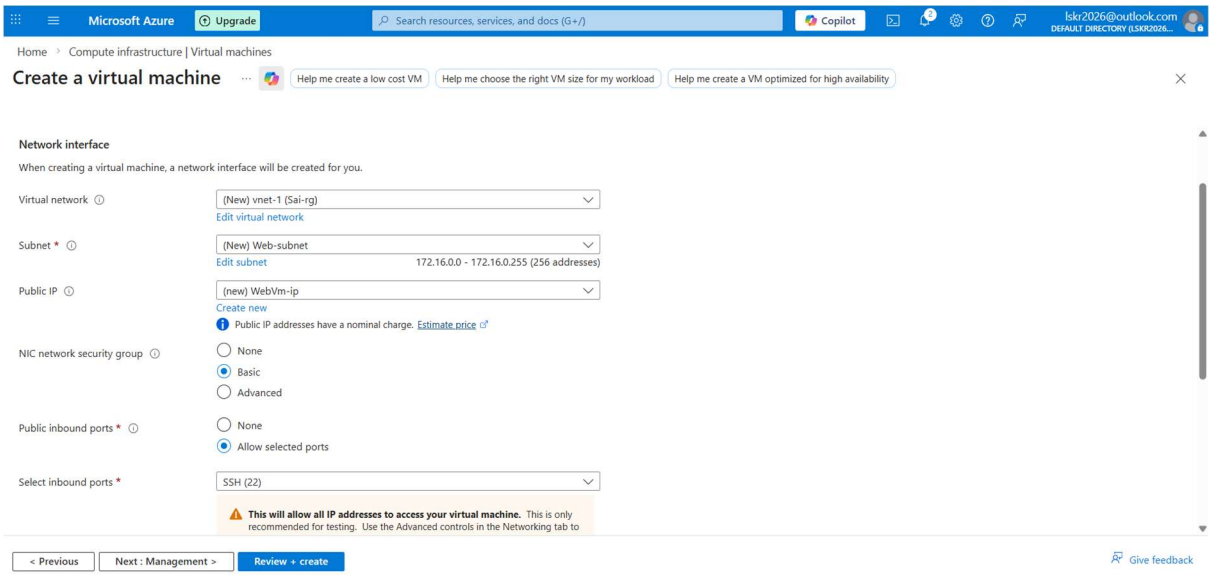



→First we want to create a Resource Group.

→Then create a vnet in the vnet we can create two virtual machines.

→Creating a Virtual network.



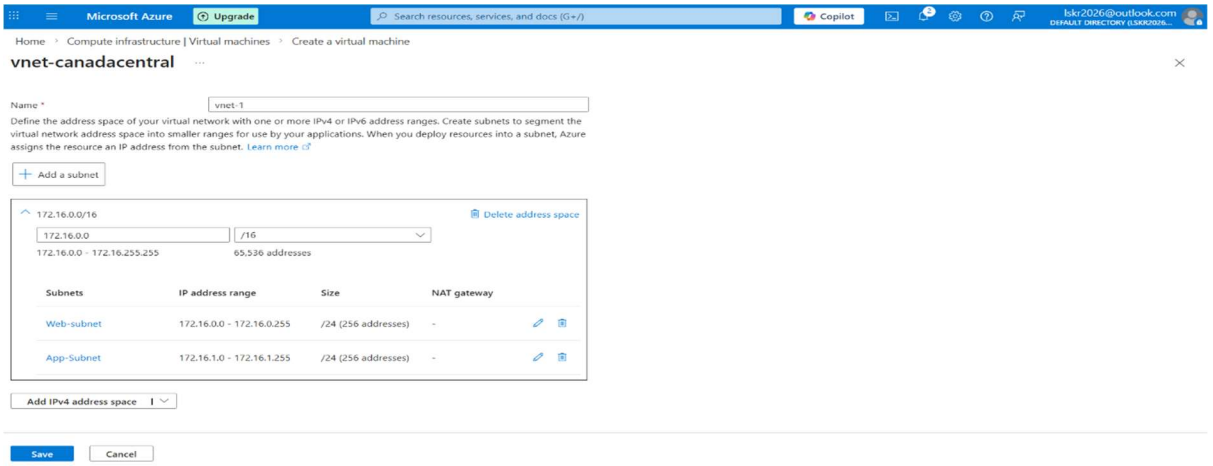


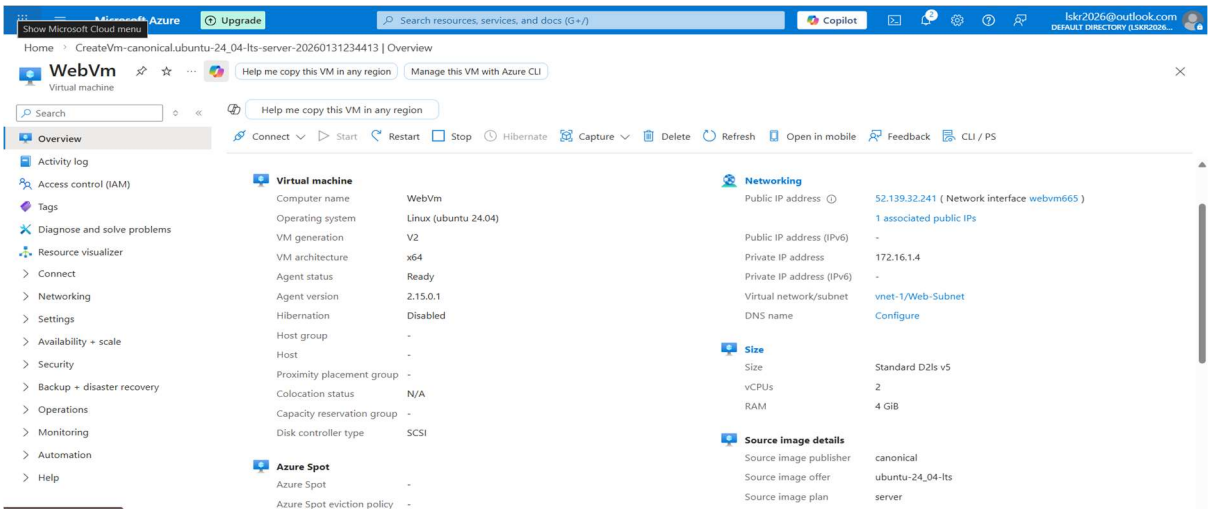

→Adding Virtual network name and ip address.

→Creating two Subnets

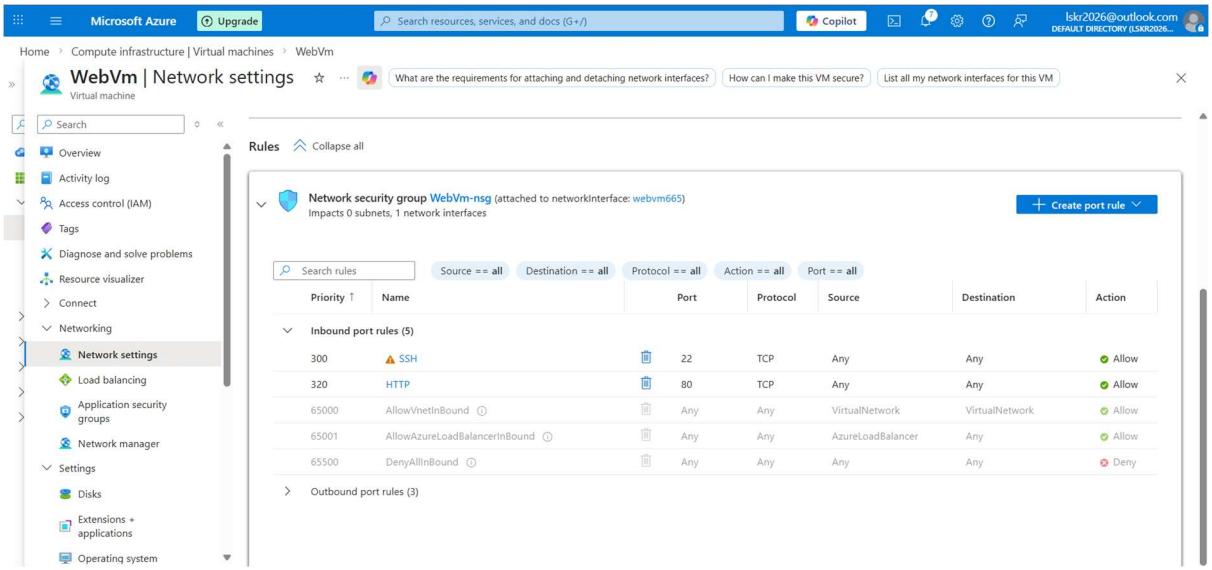
→One is Web Subnet

→And is another Subnet is App Subnet

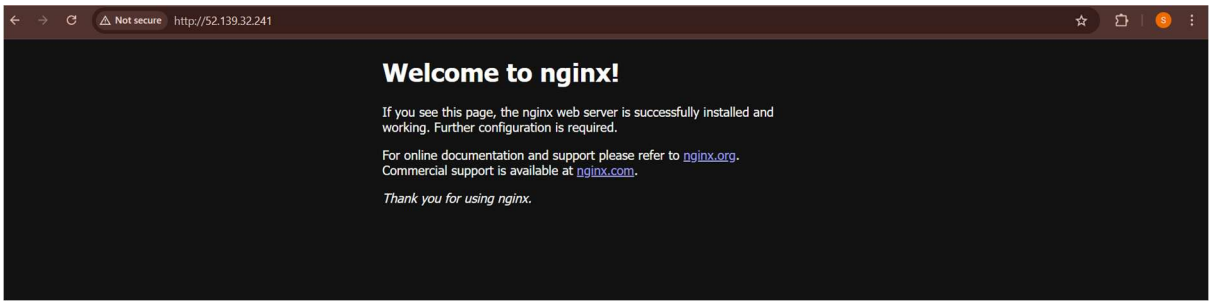


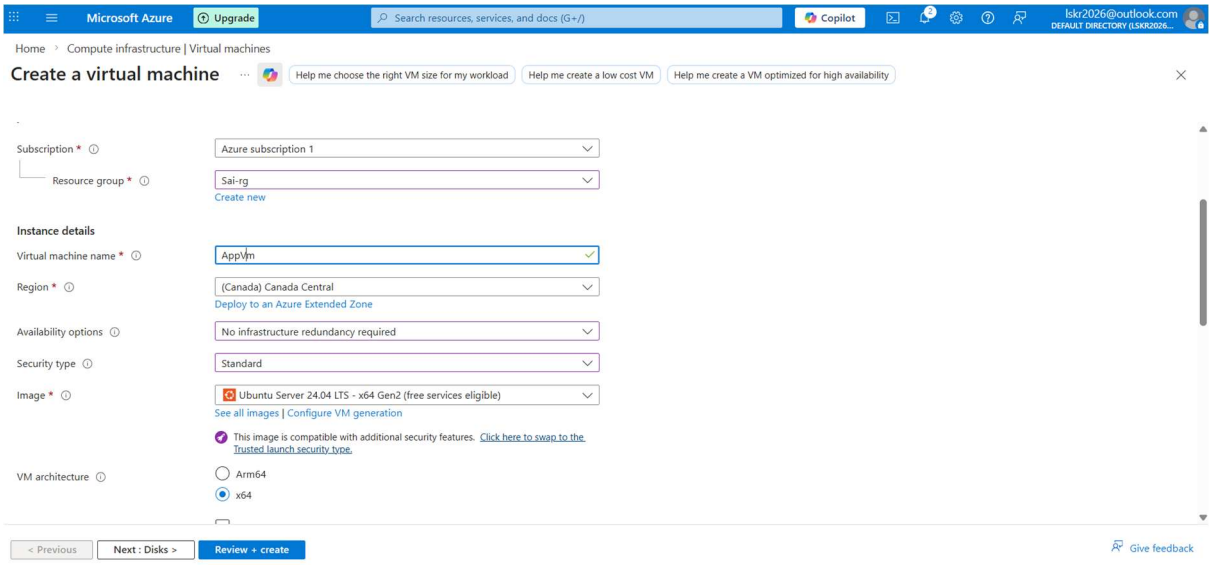
→Network Security Group for WebVm



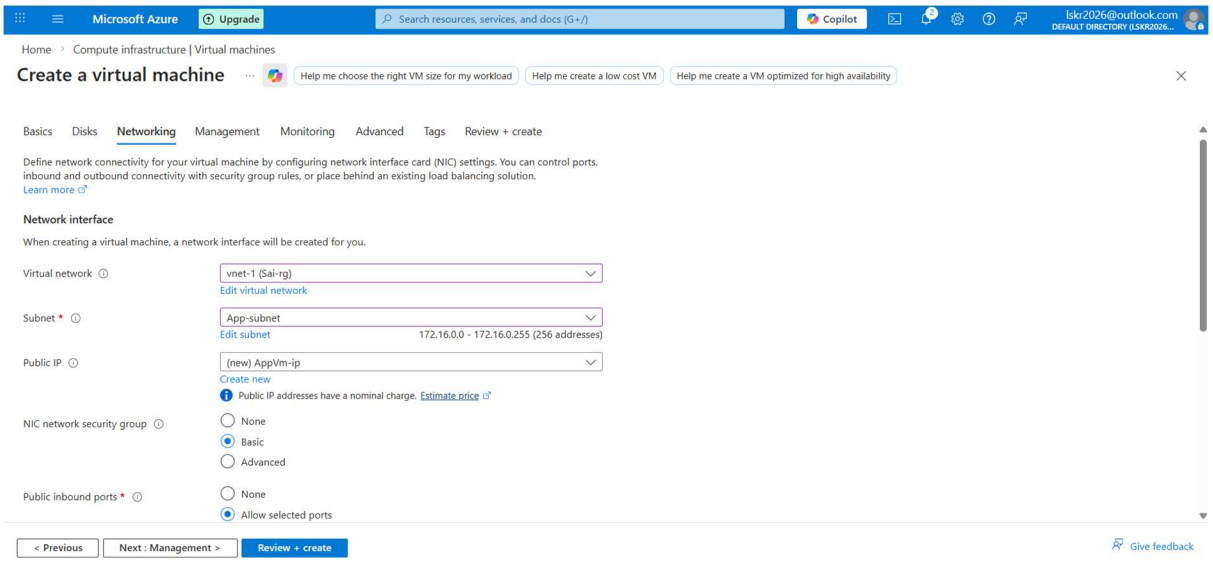

→Installed nginx and browsing nginx on the browser



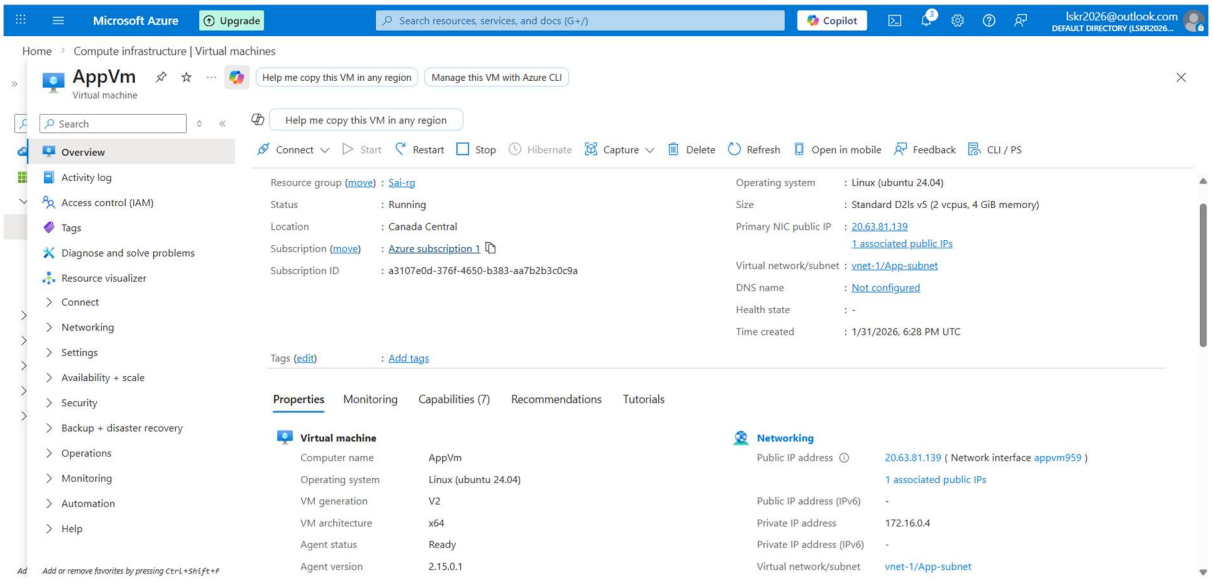

→Creating an App Virtual machine.



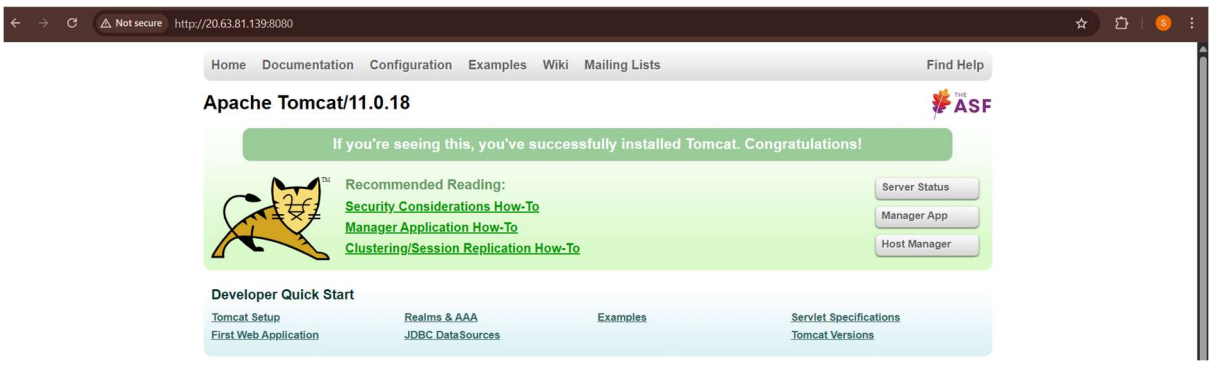

→Selecting the virtual network and subnet already we have created for App VM



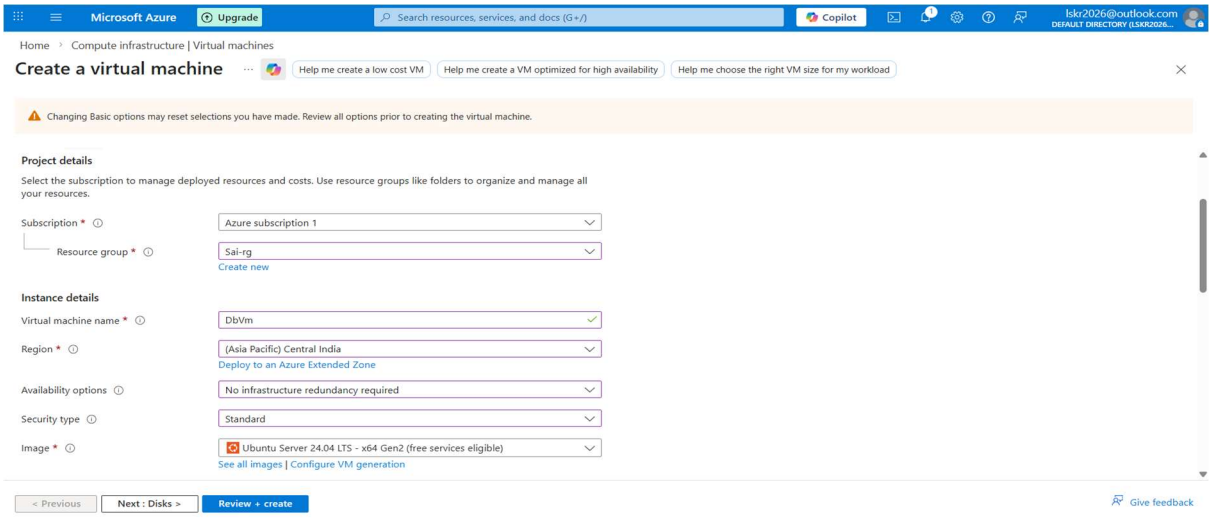

→Created APP Virtual  Machine.




→Installed Tomcat on the APP VM and browsing the public ip of AppVm.

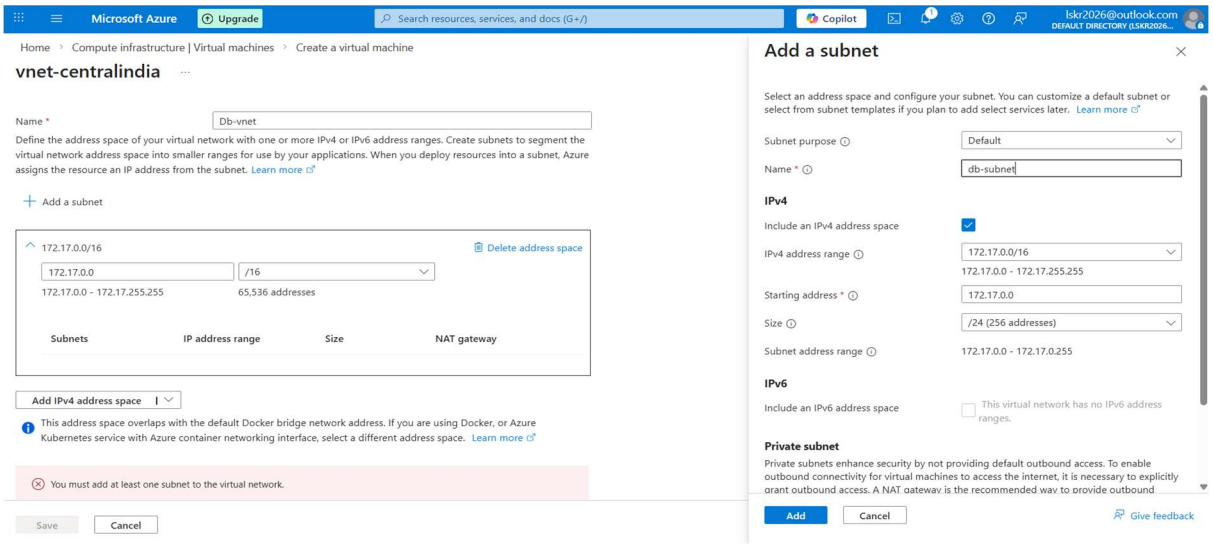


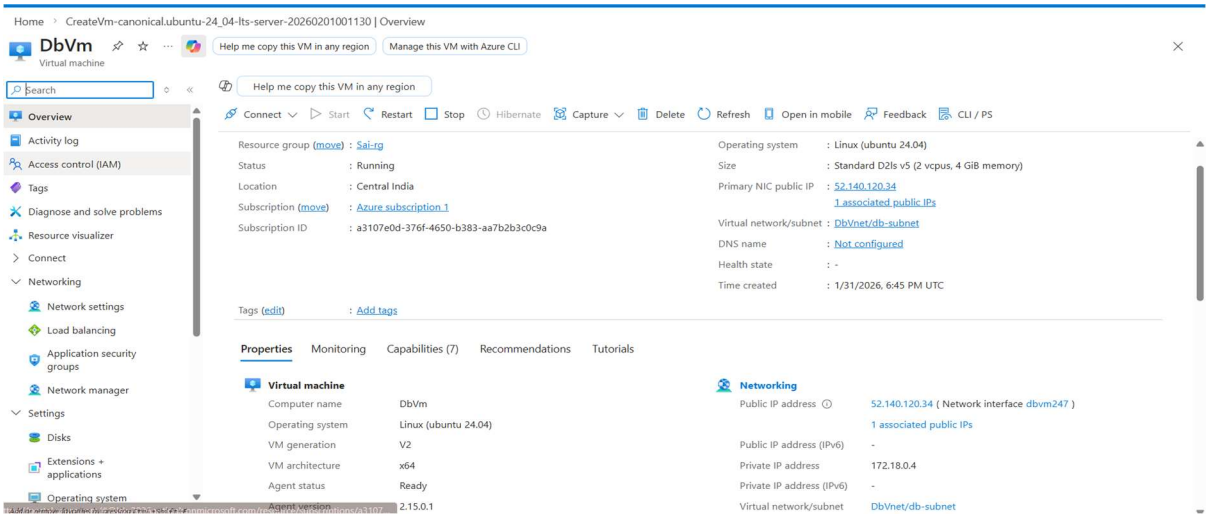
→Creating a Db Virtual machine.




→Creating a Vnet for Db.

→For free tier we can only create 4cpu's in single region that's why we create two Vm's in single region. Third Vm created in another region.



→Installed mysql on DbVm.

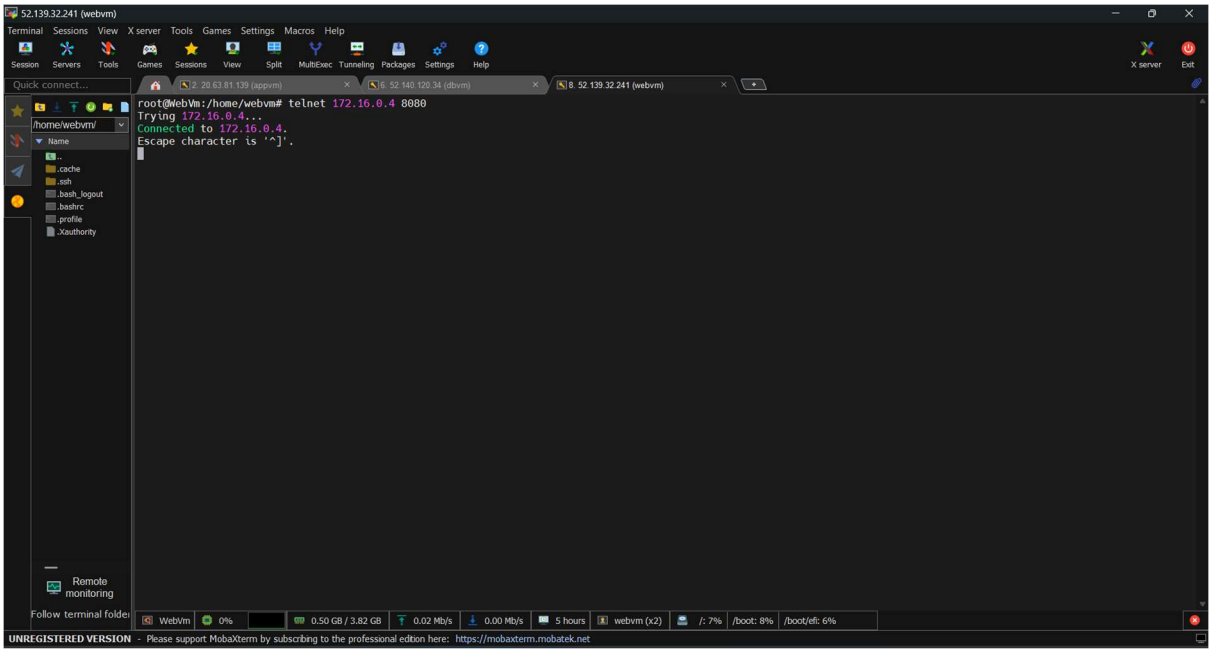
→Changed the bind address to 0.0.0.0

```
root@DbVm:/home/dbvm# systemctl status mysql
● mysql.service - MySQL Community Server
     Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
     Active: active (running) since Sat 2026-01-31 18:49:49 UTC; 2min 22s ago
    Process: 3238 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
   Main PID: 3247 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 4662)
     Memory: 368.2M (peak: 380.9M)
        CPU: 800ms
     CGroup: /system.slice/mysql.service
             └─3247 /usr/sbin/mysqld

Jan 31 18:49:48 DbVm systemd[1]: Starting mysql.service - MySQL Community Server...
Jan 31 18:49:49 DbVm systemd[1]: Started mysql.service - MySQL Community Server.
root@DbVm:/home/dbvm#
```

→Connection from Web server to App Server.

→telnet <appvtip> 8080

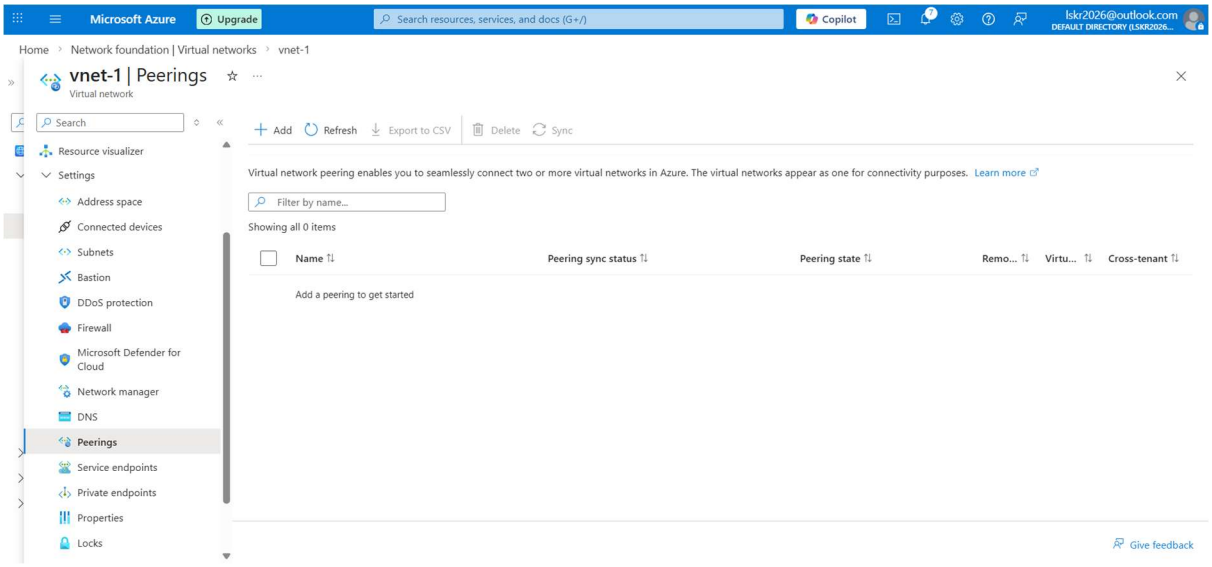



→Web server and app server both are in the same vnet that's why the can communicate.

→But making the connection from app server to Db server both are in different regions and in different Virtual networks.

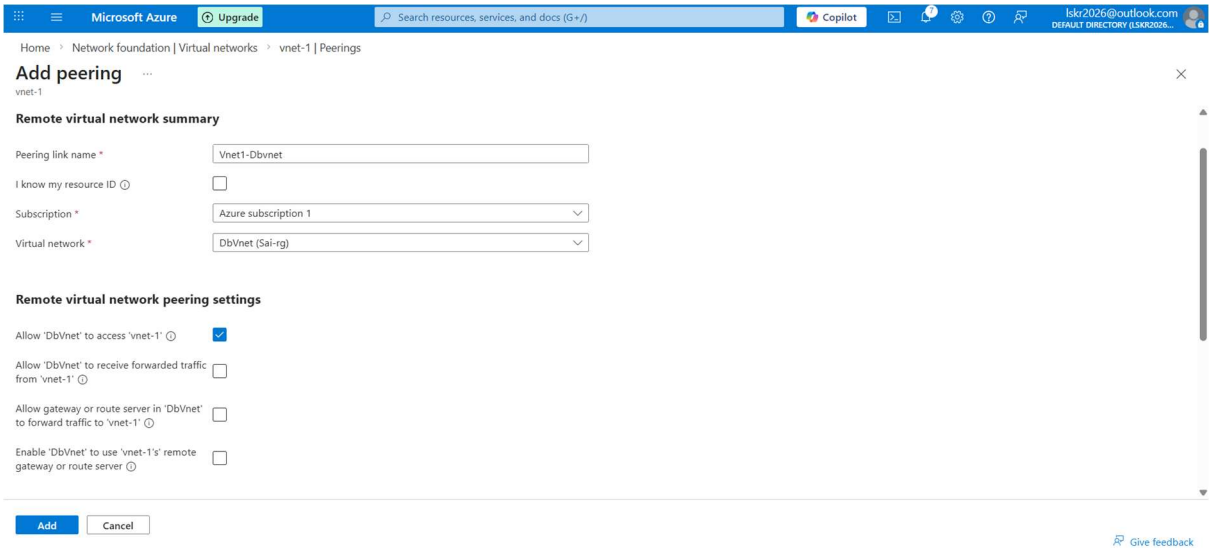
→We want to make communication between two Vnets. First we did Vnet Peering then only we check the communication between app and Db.

→Go to the Vnet and select peerings and click on add.

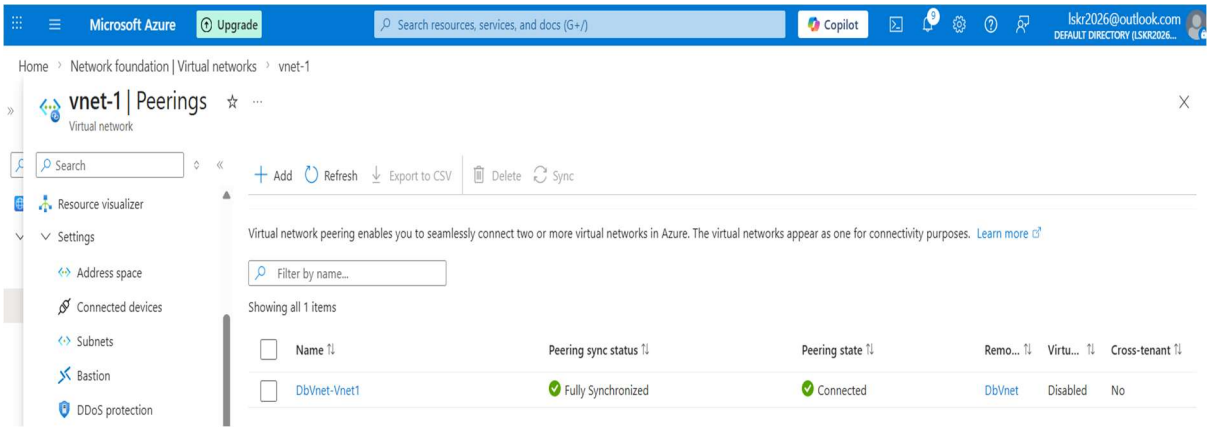



→Here I can open from vnet1 and make communication to DbVnet

→Selecting Virtual network as DbVnet and finally click on add.




→The peering can be done.



→Here we check the communication between the AppVnet and DbVnet.

→Before that we check ping <dbpvt ip>.

→If the peering has done the communication will work.