

Project Report

V.SWETHA

1. Project Overview

Project Title: 3-Tier Azure Deployment

Project Aim:

The primary objective of this project is to design and implement a 3-tier architecture infrastructure on Microsoft Azure. The architecture separates the presentation layer (web), application logic layer, and data storage layer, enhancing scalability, security, and ease of management.

Objectives:

- **Create three subnets in a Virtual Network (VNet): WebSubnet, AppSubnet, and DbSubnet**

Home > Network foundation | Virtual networks > vnetchitti

The screenshot shows the 'Subnets' page for a virtual network named 'vnetchitti'. The left sidebar contains navigation links: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Resource visualizer, Settings (Address space, Connected devices, Subnets, Bastion, DDoS protection), and Subnets (selected). The main content area has a search bar and a table of subnets. The table has columns: Name, IPv4, IPv6, Available IPs, Delegated to, Security group, and Route table. There are three subnets listed: 'webserver', 'appserver', and 'dbserver', each with an IPv4 address of /24 and 250 available IPs.

Name	IPv4	IPv6	Available IPs	Delegated to	Security group	Route table
webserver	10.0.0.0/24	-	250	-	-	-
appserver	10.0.1.0/24	-	250	-	-	-
dbserver	10.0.2.0/24	-	250	-	-	-

- **Deploy virtual machines in each subnet for respective roles.**

Home > Compute infrastructure

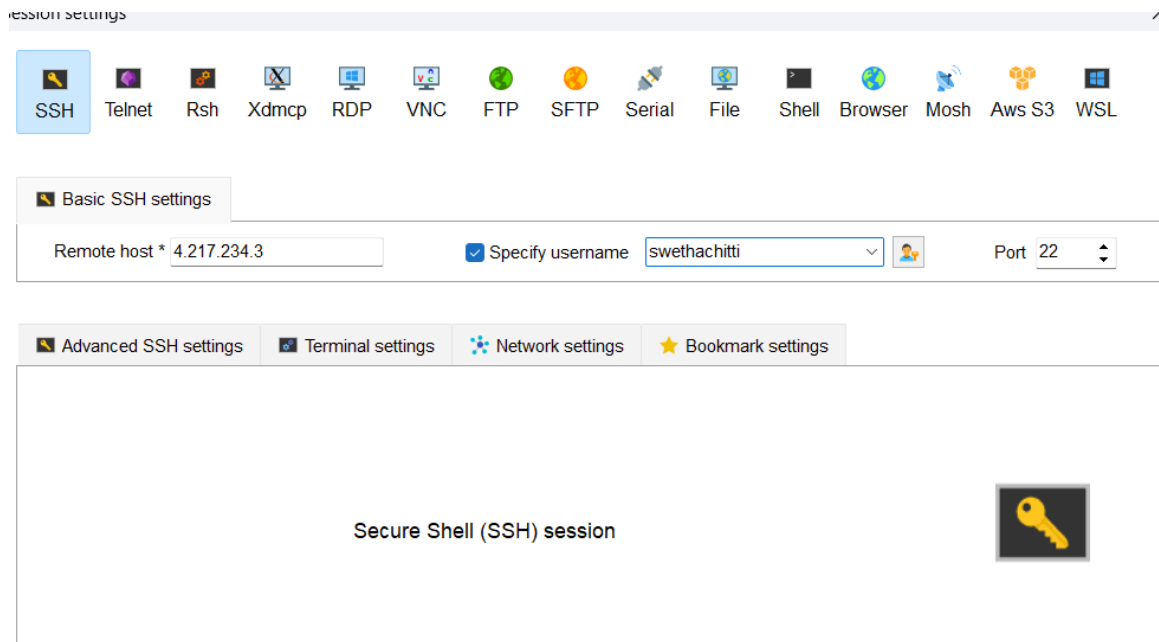
The screenshot shows the 'Virtual machines' page for 'Compute infrastructure'. The left sidebar contains navigation links: Overview, All resources, Infrastructure (Virtual machines, Virtual Machine Scale Set (VMSS), Compute Fleet, Disks + images, Custom images, Disks, Snapshots, Disk encryption sets), and Virtual machines (selected). The main content area has a search bar and a table of virtual machines. The table has columns: Name, Subscription, Resource Group, Location, Status, Operating syst..., Size, Public IP addre..., and Disks. There are five virtual machines listed: 'appasg', 'appsubnet', 'chittiappvm02', 'chittidbvm03', and 'chittivm01'. The last three are running Linux on Standard_B1s disks.

Name	Subscription	Resource Group	Location	Status	Operating syst...	Size	Public IP addre...	Disks
appasg	Azure subscript...	Nani	Spain Central	Running	Linux	Standard_B1s	158.158.32.105	1
appsubnet	Azure subscript...	REVATHI123	West Europe	Running	Linux	Standard_B1s	9.163.0.52	1
chittiappvm02	Azure subscript...	chitti	Korea Central	Running	Linux	Standard_B1s	4.230.26.195	1
chittidbvm03	Azure subscript...	CHITTI	Korea Central	Running	Linux	Standard_B1s	4.217.193.61	1
chittivm01	Azure subscript...	chitti	Korea Central	Running	Linux	Standard_B1s	4.217.234.3	1

2. Web Tier Setup – Installing NGINX

Step-by-step installation on WEBVM01:

1. **Connect using MobaXterm via SSH.**



2. Switch to root user:

- `sudo su`

3. Update packages:

- `apt update`

4. Install NGINX:

- `apt install nginx -y`

5. Enable and start NGINX: • `systemctl start nginx`

- `systemctl status nginx`

```
1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Sat Jul  5 12:14:48 2025 from 27.6.163.102
swethachitti@chittivm01:~$ sudo su
root@chittivm01:/home/swethachitti# apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [162 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:9 http://azure.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7084 B]
Get:10 http://azure.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.3 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:12 http://azure.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:13 http://azure.archive.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:16 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 4017 kB in 1s (1121 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
root@chittivm01:/home/swethachitti# apt update install nginx -y
E: The update command takes no arguments
root@chittivm01:/home/swethachitti# apt install nginx -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
nginx is already the newest version (1.24.0-2ubuntu7.4).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
root@chittivm01:/home/swethachitti#
```

6. Allow **port 80** in NSG settings.

chittivm01 | Network settings

Rules Collapse all

Network security group **chittivm01-nsg** (attached to networkInterface: chittivm01428)
Impacts 0 subnets, 1 network interfaces

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

Priority	Name	Port	Protocol	Source	Destination	Action
Inbound port rules (5)						
300	SSH	22	TCP	Any	Any	Allow
310	AllowAnyHTTPInbound	80	TCP	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny
Outbound port rules (3)						

7. Access via browser: **http://<public-ip>:80**

- http://172.172.140.88:80**

WhatsApp chittivm01 - Microsoft Azure Welcome to nginx!

Not secure 4.217.234.3

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.

3. Application Tier Setup – Apache Tomcat

Steps for **chittiappvm02**:

1. Connect via SSH and switch to root:

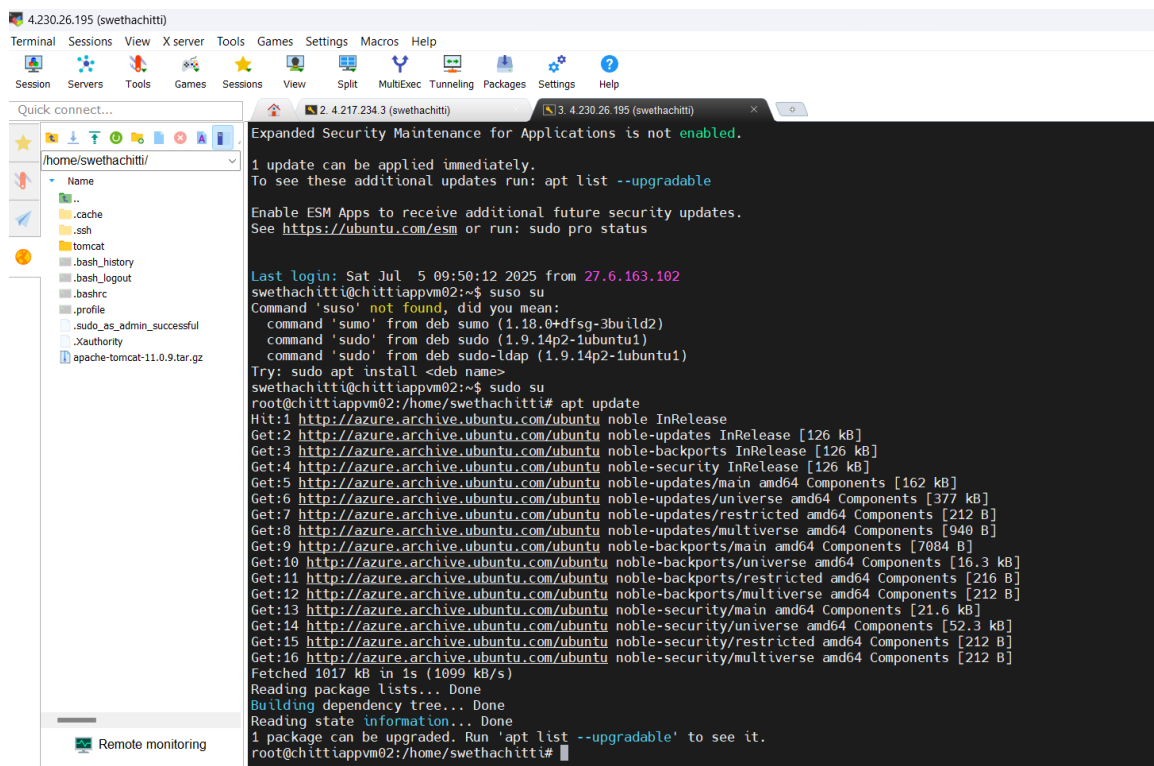
- **sudo su**

2. Install Java:

- **apt update**
- **apt install default-jdk -y**

3. Verify Java installation:

- **java -version**



The screenshot shows a terminal window with a file manager on the left. The terminal output is as follows:

```
4.230.26.195 (swethachitti)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/swethachitti/
Name
..
.cache
.ssh
tomcat
.bash_history
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.xauthority
apache-tomcat-11.0.9.tar.gz

Expanded Security Maintenance for Applications is not enabled.
1 update can be applied immediately.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

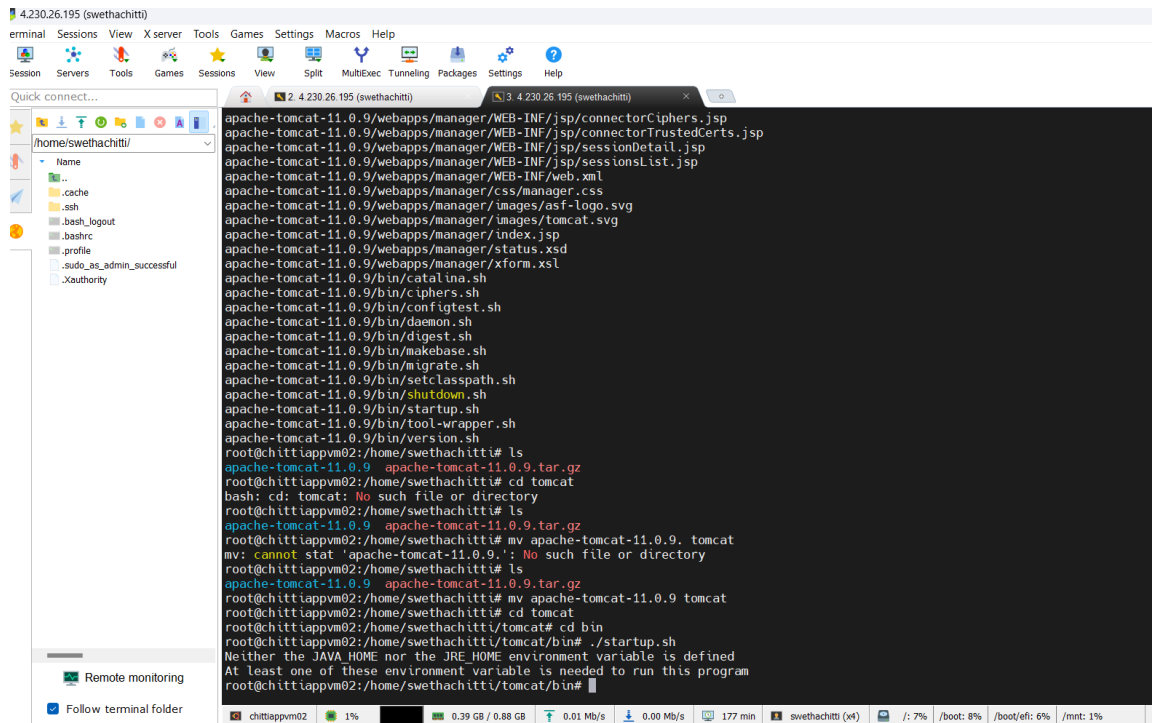
Last login: Sat Jul 5 09:50:12 2025 from 27.6.163.102
swethachitti@chittiappvm02:~$ sudo su
Command 'sudo' not found, did you mean:
  command 'sumo' from deb sumo (1.18.0+dfsg-3build2)
  command 'sudo' from deb sudo (1.9.14p2-1ubuntu1)
  command 'sudo' from deb sudo-ldap (1.9.14p2-1ubuntu1)
Try: sudo apt install <deb name>
swethachitti@chittiappvm02:~$ sudo su
root@chittiappvm02:/home/swethachitti# apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [162 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [377 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:9 http://azure.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7084 B]
Get:10 http://azure.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.3 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:12 http://azure.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:13 http://azure.archive.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:16 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 1017 kB in 1s (1099 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.
root@chittiappvm02:/home/swethachitti#
```

4. Download and extract Tomcat in /opt/:

- **wget [Tomcat URL]**
- **tar -xvzf apache-tomcat.tar.gz**
- **mv folder to 'tomcat'**

5. Start Tomcat: **./startup.sh**

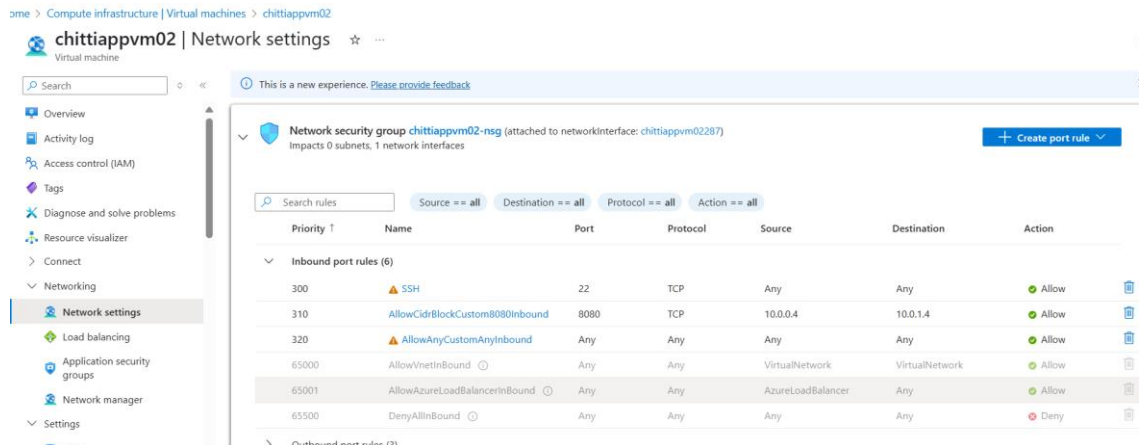
6. Start Tomcat: `./shutdown.sh`



```
4.230.26.195 (swethachitti)
terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/swethachitti/
Name
..
.cocher
.sh
.bash_logout
.bashrc
.profile
.sudo_as_admin_successful
.xauthority
Remote monitoring
Follow terminal folder
chittiappvm02 1% 0.39 GB / 0.88 GB 0.01 Mb/s 0.00 Mb/s 177 min swethachitti (x4) /: 7% /boot: 8% /boot/efi: 6% /mnt: 1%
```

REGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

8. Allow port 8080 in NSG.



chittiappvm02 | Network settings

Network security group **chittiappvm02-nsg** (attached to networkInterface: chittiappvm02287)

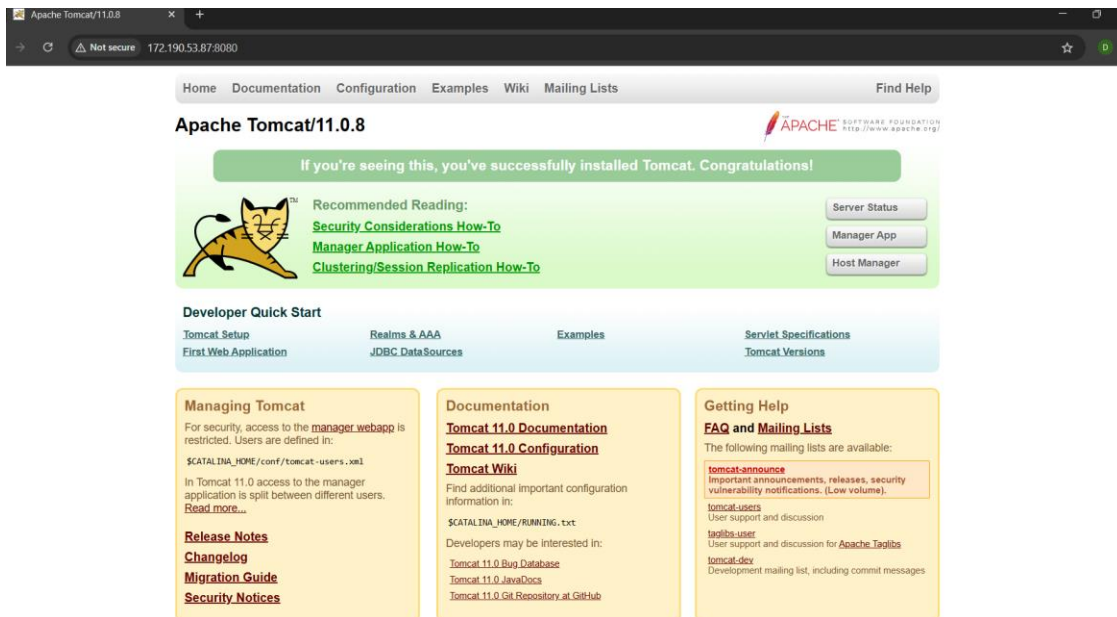
Impacts 0 subnets, 1 network interfaces

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

Priority ↑	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	AllowCidrBlockCustom8080Inbound	8080	TCP	10.0.0.4	10.0.1.4	Allow
320	AllowAnyCustomAnyInbound	Any	Any	Any	Any	Allow
65000	AllowVnetInbound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInbound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInbound	Any	Any	Any	Any	Deny

Outbound port rules (3)

9. Access Tomcat using: `http://<public-ip>:8080`



4. Database Tier Setup – MySQL

Steps for **chittidbvm03**:

1. Connect via SSH and switch to root:

- **sudo su**

2. Update packages:

- **apt update**

3. Install MySQL:

- **apt install mysql-server -y**

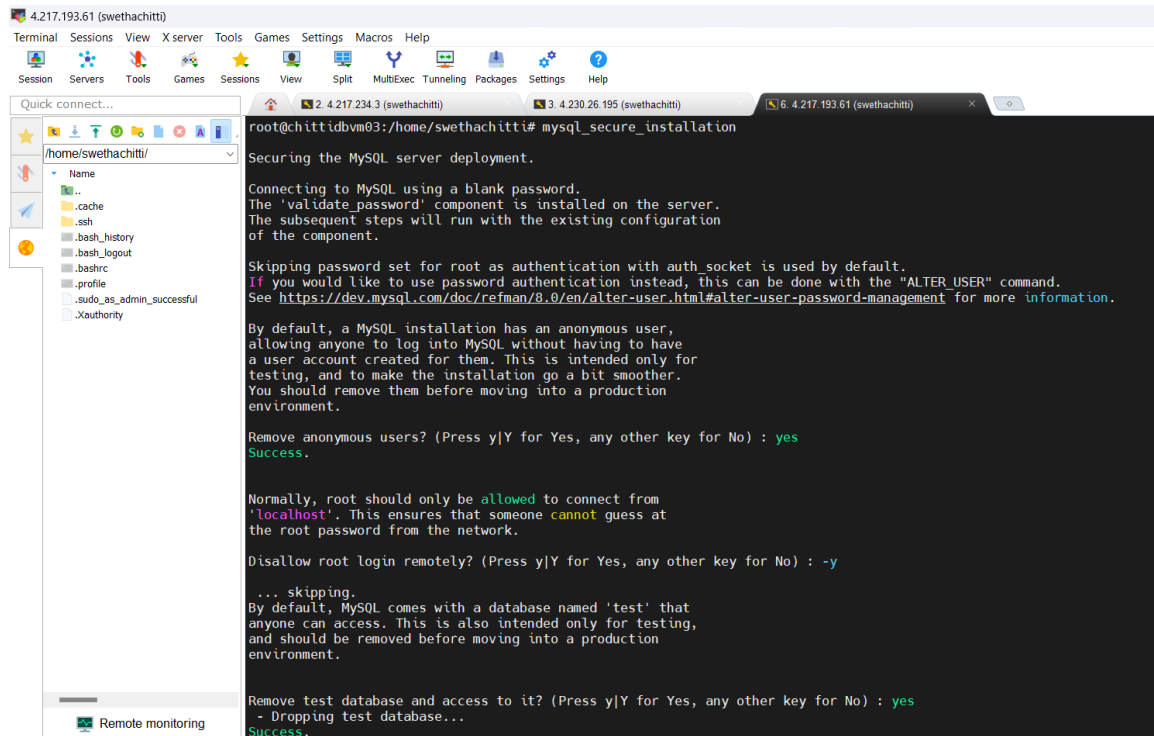
```

172.172.140.88 (anubadmin)
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split Multitac Tunneling Packages Settings Help
Quick connect... 172.172.140.88 (anubadmin) 172.172.140.88 (anubadmin)
anubadmin@ANUBVM:~$ sudo su
root@ANUBVM:/home/anubadmin# apt update
Hit:1 http://azure.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://azure.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Get:3 http://azure.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]
Get:4 http://azure.archive.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://azure.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [162 kB]
Get:6 http://azure.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [277 kB]
Get:7 http://azure.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]
Get:8 http://azure.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]
Get:9 http://azure.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7684 B]
Get:10 http://azure.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [16.4 kB]
Get:11 http://azure.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [216 B]
Get:12 http://azure.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]
Get:13 http://azure.archive.ubuntu.com/ubuntu noble-security/main amd64 Components [21.6 kB]
Get:14 http://azure.archive.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.3 kB]
Get:15 http://azure.archive.ubuntu.com/ubuntu noble-security/restricted amd64 Components [208 B]
Get:16 http://azure.archive.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Fetched 1017 kB in 1s (884 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
1 package can be upgraded. Run 'apt list --upgradable' to see it.

```

4. Secure MySQL:

- **mysql_secure_installation**



The screenshot shows a terminal window with the command `mysql_secure_installation` executed. The output is as follows:

```
root@chittidbvm03:/home/swethachitti# mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.
The 'validate_password' component is installed on the server.
The subsequent steps will run with the existing configuration
of the component.

Skipping password set for root as authentication with auth_socket is used by default.
If you would like to use password authentication instead, this can be done with the "ALTER USER" command.
See https://dev.mysql.com/doc/refman/8.0/en/alter-user.html#alter-user-password-management for more information.

By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : yes
Success.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : -y

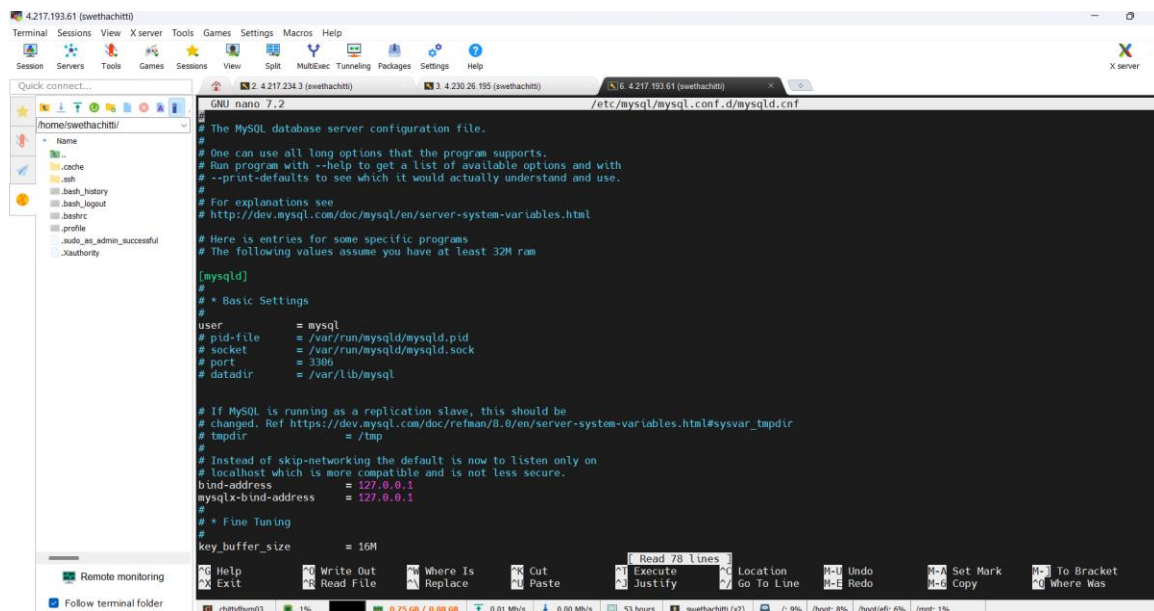
... skipping.
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) : yes
- Dropping test database...
Success.
```

5. Edit config file:

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

6. Modify bind-address to 0.0.0.0 or specific internal IP.



The screenshot shows a terminal window with the command `nano /etc/mysql/mysql.conf.d/mysqld.cnf` executed. The output is as follows:

```
GNU nano 2.2 /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        = 127.0.0.1
mysqlx-bind-address = 127.0.0.1
#
# * Fine Tuning
#
key_buffer_size     = 16M
```


7. Restart MySQL:

- `systemctl restart mysql`

- `systemctl restart mysql`

8. Check MySQL status:

- `systemctl status mysql`

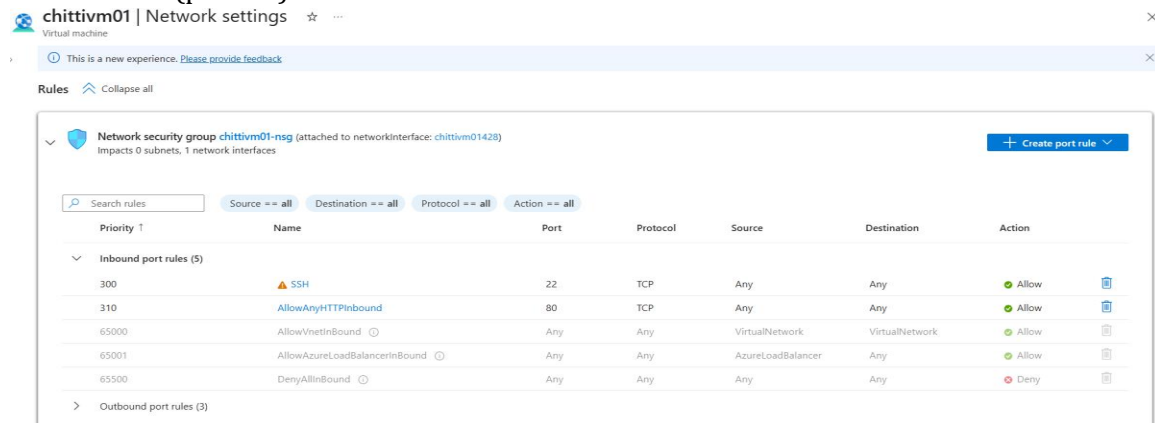
- `systemctl status mysql`

A screenshot of a terminal window titled "4.217.193.61 (swethachitti)". The terminal shows the following sequence of events:
1. A file manager window is open in the background, showing the directory structure of "/home/swethachitti/".
2. The terminal displays "... skipping." followed by a message: "By default, MySQL comes with a database named 'test' that anyone can access. This is also intended only for testing, and should be removed before moving into a production environment."
3. A prompt "Remove test database and access to it? (Press y/Y for Yes, any other key for No) : " is shown, followed by the user input "yes".
4. The terminal shows "- Dropping test database..." and "Success."
5. A prompt "- Removing privileges on test database..." is shown, followed by "Success."
6. A message "Reloading the privilege tables will ensure that all changes made so far will take effect immediately."
7. A prompt "Reload privilege tables now? (Press y/Y for Yes, any other key for No) : " is shown, followed by the user input "yes".
8. The terminal shows "Success."
9. A message "All done!" is displayed.
10. The user runs the command "root@chittidbvm03:/home/swethachitti# nano /etc/mysql/mysql.conf.d/mysqld.cnf".
11. The user runs "root@chittidbvm03:/home/swethachitti# systemctl restart mysql".
12. The user runs "root@chittidbvm03:/home/swethachitti# systemctl status mysql".
13. The terminal displays the status of the MySQL service:
● mysql.service - MySQL Community Server
Loaded: loaded (/usr/lib/systemd/system/mysql.service; enabled; preset: enabled)
Active: active (running) since Mon 2025-07-07 13:19:53 UTC; 42s ago
Process: 88785 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
Main PID: 88794 (mysqld)
Status: "Server is operational"
Tasks: 38 (limit: 1056)
Memory: 389.1M (peak: 407.1M)
CPU: 1.085s
CGroup: /system.slice/mysql.service
└─88794 /usr/sbin/mysqld
14. At the bottom, there are two lines of system logs:
Jul 07 13:19:51 chittidbvm03 systemd[1]: Starting mysql.service - MySQL Community Server...
Jul 07 13:19:53 chittidbvm03 systemd[1]: Started mysql.service - MySQL Community Server.
15. The terminal prompt is now "root@chittidbvm03:/home/swethachitti#".

5. Network Security Groups (NSG) Configuration

Each tier is protected by NSG rules to control traffic securely between layers.

- **swethaWEBVM:**
 - Allow HTTP/HTTPS (ports 80, 443)
 - Allow SSH(port 22)



• swethaVMAPP(NSG):

- Allow from chittiAPPvm02 on port 8080

chittiappvm02 | Network settings

Network security group **chittiappvm02-nsg** (attached to networkInterface: chittiappvm02287)
Impacts 0 subnets, 1 network interfaces

Search rules

Priority ↑	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	AllowCidrBlockCustom8080Inbound	8080	TCP	10.0.0.4	10.0.1.4	Allow
320	AllowAnyCustomAnyInbound	Any	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Outbound port rules (3)

• chittidbvm03:

- Allow only from chittidbvm03 on port 3306

- Deny all other inbound connections

chittidbvm03 | Network settings

Network security group **chittidbvm03-nsg** (attached to networkInterface: chittidbvm03220)
Impacts 0 subnets, 1 network interfaces

Search rules

Priority ↑	Name	Port	Protocol	Source	Destination	Action
300	SSH	22	TCP	Any	Any	Allow
310	AllowCidrBlockCustom3306Inbound	3306	Any	10.0.1.4	10.0.2.4	Allow
320	DenyCidrBlockCustomAnyInbound	Any	Any	10.0.0.4	10.0.2.4	Deny
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalancer	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Outbound port rules (3)

This ensures layered security and isolation.

Connection status:

- Checking Connection status from **chittiappvm02** to **chittidbvm03**
(**telnet<ipaddress>portnumber**)

```
10.0.2.4: command not found
root@chittidbvm03:/home/swethachitti# telnet 10.0.1.4
Trying 10.0.1.4...
telnet: Unable to connect to remote host: Connection refused
root@chittidbvm03:/home/swethachitti#
```

chittidbvm03 0% 0.40 GB / 0.88 GB 0.01 Mb/s 0.00 Mb/s 257 min swethachitti (x) /: 7% /boot: 8% /boot/efi: 6% /mnt: 1%

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>

- Checking Connection status from **chittiVMWEB TO chittiVMDB**
03(telnet<ipaddress>portnumber)

```
root@chittidbvm03:/home/swethachitti# telnet 10.0.0.4 3306
Trying 10.0.0.4...
```

Conclusion:

The implementation of a 3-tier architecture in Azure improves security, scalability, and maintainability. By isolating each layer with dedicated VMs and NSG rules, the infrastructure supports modern application needs securely. Using MobaXterm, Linux commands, and service configuration, each component was successfully deployed and tested.