Phase 4: Initial System Configuration & Core Security Hardening (Week 4)

4.1 Phase Objective

The objective of Phase 4 was to deploy the server in a secure, production-ready state by implementing **core security controls**. This phase focused on securing remote access, enforcing least-privilege principles, and reducing the system's exposed attack surface through SSH hardening and firewall configuration.

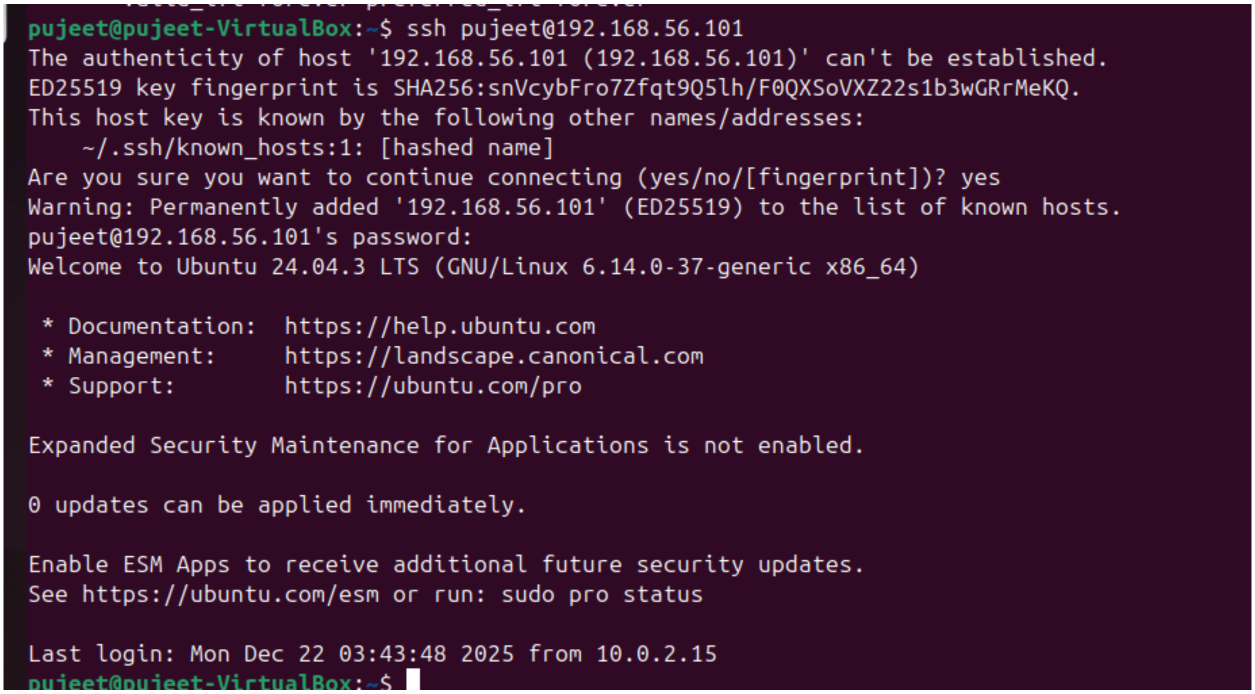
All configuration and verification tasks were performed **remotely via SSH**, reflecting real-world Linux server administration practices and complying with the coursework requirement for headless server management.

4.2 Initial Remote Access Verification

An initial SSH connection was established to confirm baseline remote accessibility and validate network connectivity between the workstation and the server.

Command executed:

```
ssh pujeet@192.168.56.101
```



```
       valid_lft forever preferred_lft forever
pujeet@pujeet-VirtualBox:~$ ssh pujeet@192.168.56.101
The authenticity of host '192.168.56.101 (192.168.56.101)' can't be established.
ED25519 key fingerprint is SHA256:snVcybFro7Zfqt9Q5lh/F0QXSoVXZ22s1b3wGRrMeKQ.
This host key is known by the following other names/addresses:
    ~/.ssh/known_hosts:1: [hashed name]
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.56.101' (ED25519) to the list of known hosts.
pujeet@192.168.56.101's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

Last login: Mon Dec 22 03:43:48 2025 from 10.0.2.15
pujeet@pujeet-VirtualBox:~$
```

The successful login confirmed that:

- The SSH service was reachable
- Network configuration was correct
- Remote administration was possible

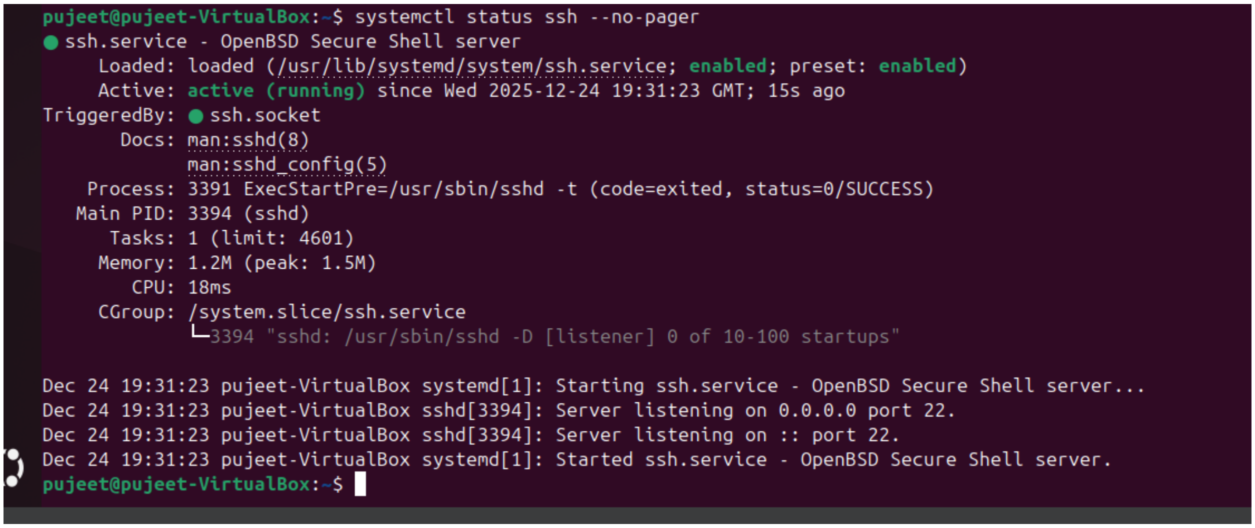
Initial SSH connection to the server
(Screenshot showing first SSH login prompt and successful connection)

4.3 SSH Service Verification

Before applying any hardening changes, the SSH service status was verified to ensure it was active and managed by systemd.

Command executed:

```
systemctl status ssh --no-pager
```



```
pujeet@pujeet-VirtualBox:~$ systemctl status ssh --no-pager
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-12-24 19:31:23 GMT; 15s ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 3391 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3394 (sshd)
      Tasks: 1 (limit: 4601)
     Memory: 1.2M (peak: 1.5M)
        CPU: 18ms
     CGroup: /system.slice/ssh.service
             └─3394 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 24 19:31:23 pujeet-VirtualBox systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Dec 24 19:31:23 pujeet-VirtualBox sshd[3394]: Server listening on 0.0.0.0 port 22.
Dec 24 19:31:23 pujeet-VirtualBox sshd[3394]: Server listening on :: port 22.
Dec 24 19:31:23 pujeet-VirtualBox systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
pujeet@pujeet-VirtualBox:~$ █
```

The output confirmed that:

- SSH daemon was active (running)
- Service was enabled at boot

SSH service active and running
(Screenshot showing ssh.service status)

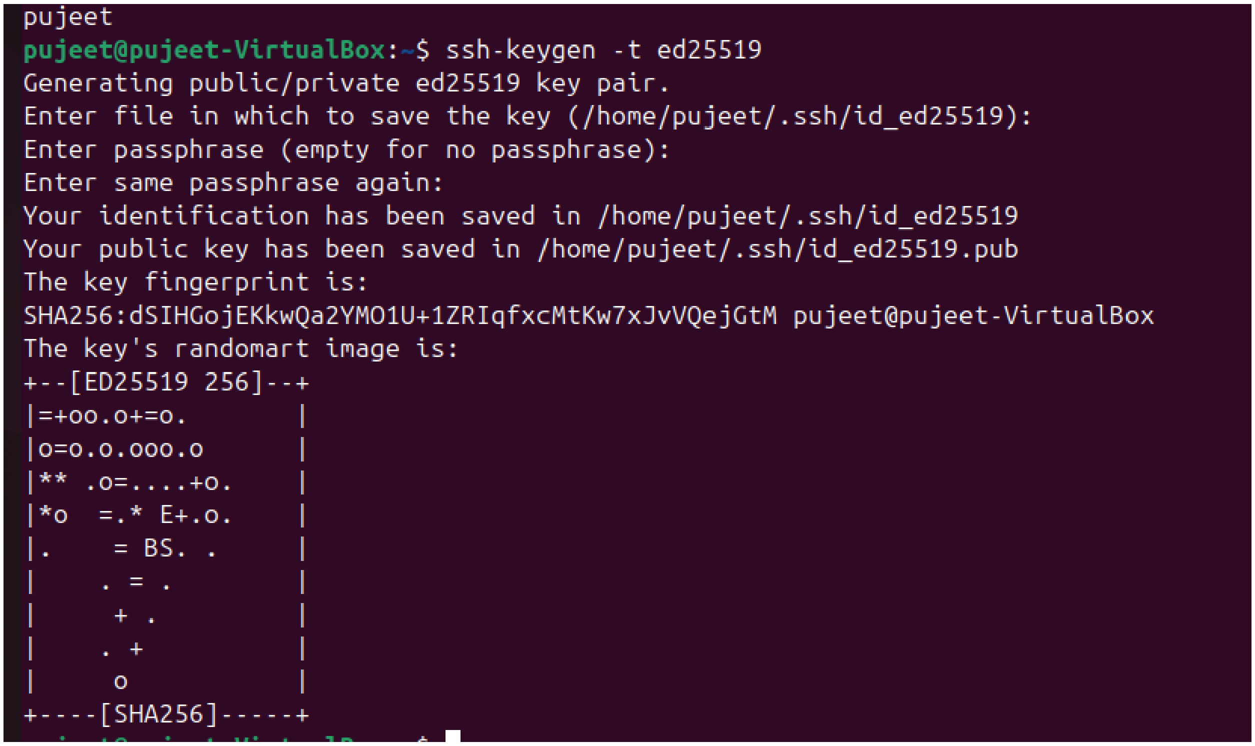
4.4 SSH Key-Based Authentication Setup

To improve authentication security, **SSH key-based authentication** was configured using modern cryptographic standards.

Key Generation (Workstation)

An Ed25519 key pair was generated on the workstation:

```
ssh-keygen -t ed25519
```

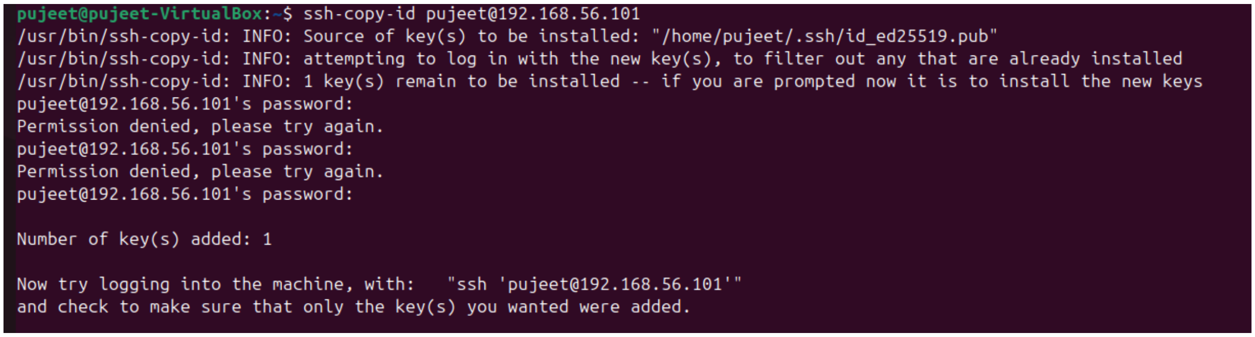


```
pujeet
pujeet@pujeet-VirtualBox:~$ ssh-keygen -t ed25519
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/pujeet/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pujeet/.ssh/id_ed25519
Your public key has been saved in /home/pujeet/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:dSIHGojEKkwQa2YMO1U+1ZRIqfxcMtKw7xJvVQejGtM pujeet@pujeet-VirtualBox
The key's randomart image is:
+--[ED25519 256]--+
|=+oo.o+=o.       |
|o=o.o.ooo.o      |
|** .o=....+o.    |
|*o  =.* E+.o.    |
|.    = BS. .     |
|    . = .        |
|     + .         |
|    . +          |
|     o           |
+----[SHA256]-----+
```

Key Deployment

The public key was securely transferred to the server:

```
ssh-copy-id pujeet@192.168.56.101
```



```
pujeet@pujeet-VirtualBox:~$ ssh-copy-id pujeet@192.168.56.101
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/pujeet/.ssh/id_ed25519.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
pujeet@192.168.56.101's password:
Permission denied, please try again.
pujeet@192.168.56.101's password:
Permission denied, please try again.
pujeet@192.168.56.101's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'pujeet@192.168.56.101'"
and check to make sure that only the key(s) you wanted were added.
```

This configuration eliminates reliance on passwords and significantly reduces vulnerability to brute-force attacks.

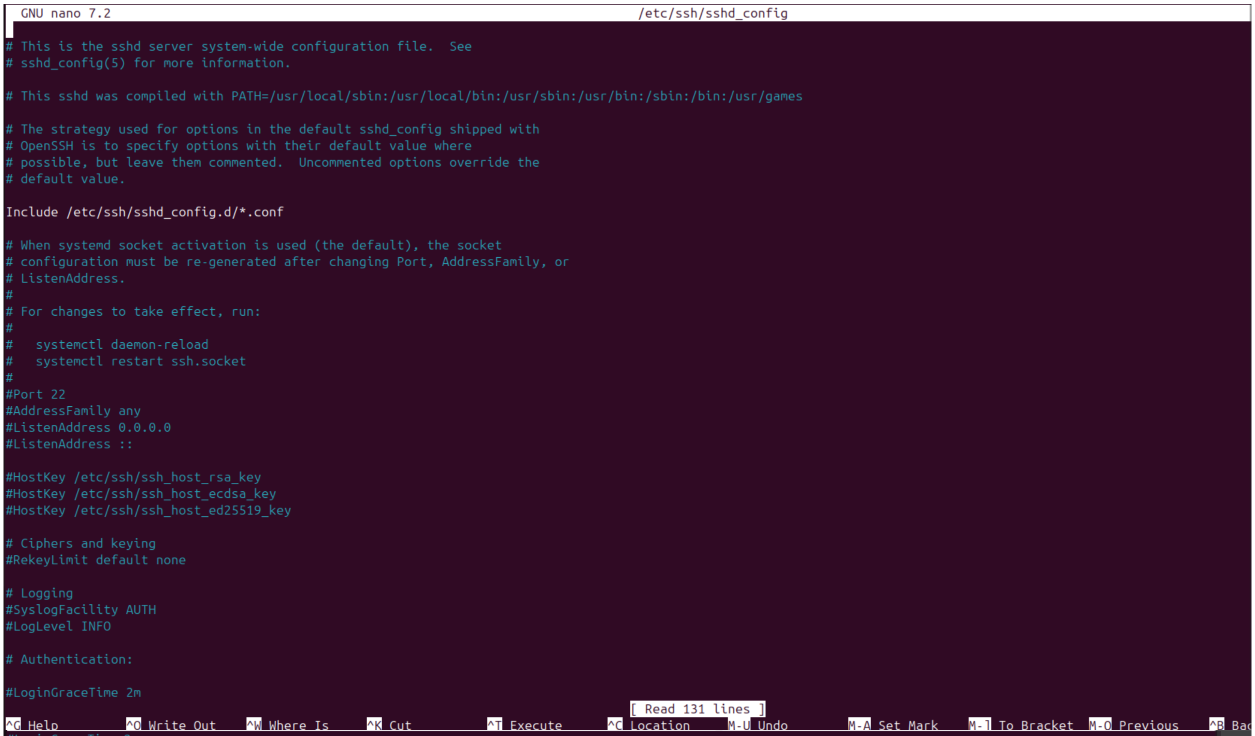
SSH key generation (Ed25519)
Public key successfully copied to server

4.5 SSH Daemon Hardening Configuration

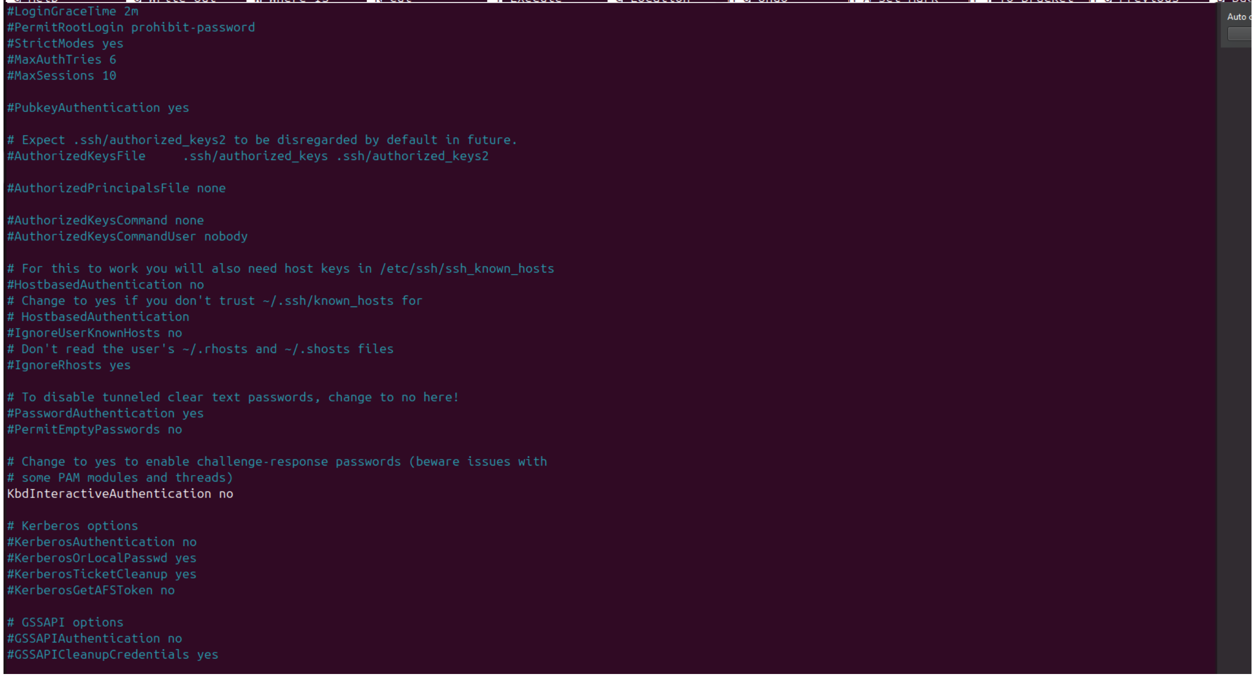
The SSH daemon configuration file was modified to enforce strict access controls.

File edited:

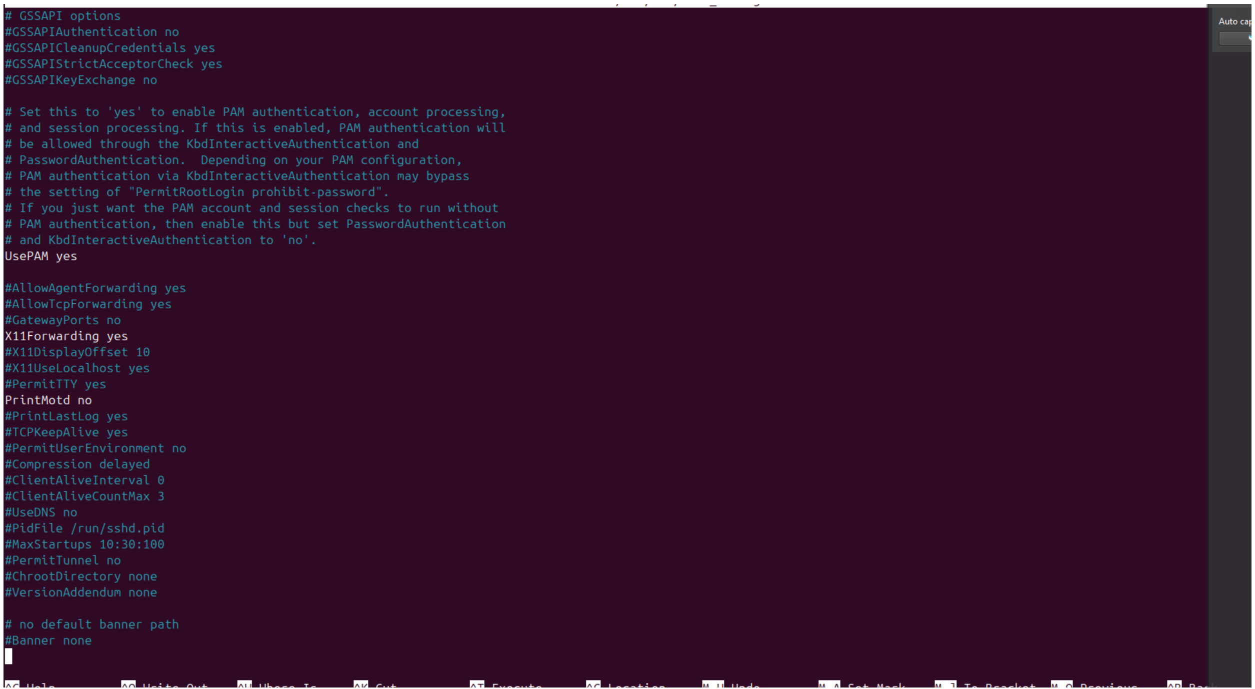
```
sudo nano /etc/ssh/sshd_config
```



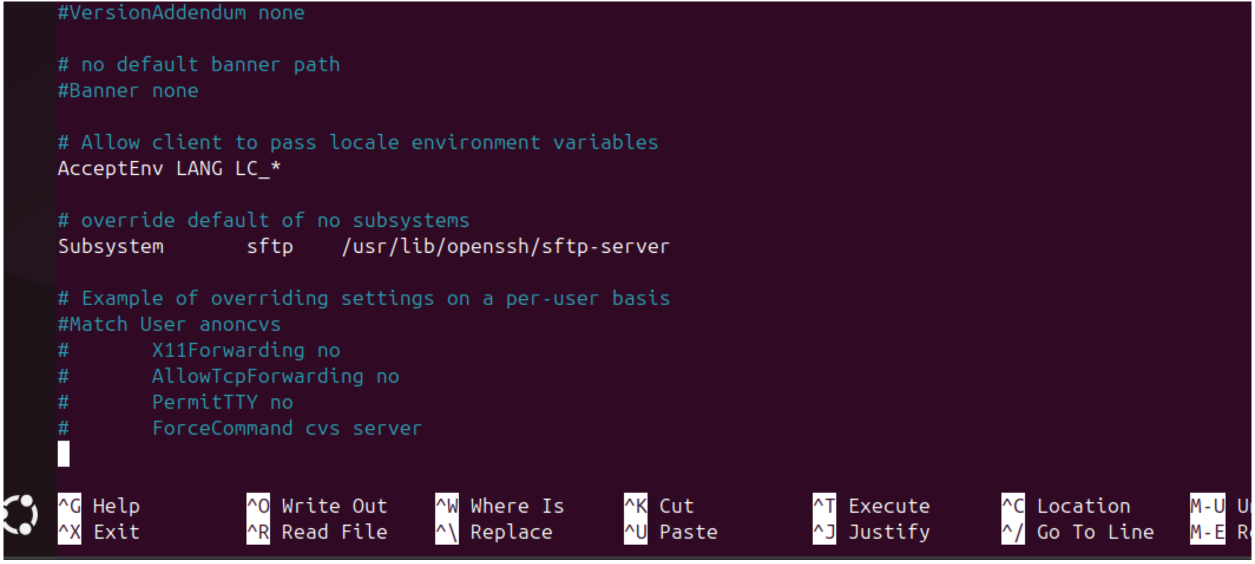
```
  GNU nano 7.2                                                                         /etc/ssh/sshd_config

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

# When systemd socket activation is used (the default), the socket
# configuration must be re-generated after changing Port, AddressFamily, or
# ListenAddress.
#
# For changes to take effect, run:
#
#   systemctl daemon-reload
#   systemctl restart ssh.socket
#
#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
                                                                                   [ Read 131 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark    M-] To Bracket  M-O Previous    ^B Bac
```



```
#LoginGraceTime 2m
#PermitRootLogin prohibit-password
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10

#PubkeyAuthentication yes

# Expect .ssh/authorized_keys2 to be disregarded by default in future.
#AuthorizedKeysFile     .ssh/authorized_keys .ssh/authorized_keys2

#AuthorizedPrincipalsFile none

#AuthorizedKeysCommand none
#AuthorizedKeysCommandUser nobody

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
KbdInteractiveAuthentication no

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
```



```
# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes
#GSSAPIStrictAcceptorCheck yes
#GSSAPIKeyExchange no

# Set this to 'yes' to enable PAM authentication, account processing,
# and session processing. If this is enabled, PAM authentication will
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication.  Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin prohibit-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/sshd.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none
```



```
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

# override default of no subsystems
Subsystem       sftp    /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#       X11Forwarding no
#       AllowTcpForwarding no
#       PermitTTY no
#       ForceCommand cvs server
```

Security Controls Implemented

The following settings were applied:

```
PermitRootLogin no
PasswordAuthentication no
PubkeyAuthentication yes
MaxAuthTries 3
PermitEmptyPasswords no
KbdInteractiveAuthentication no
```

These changes ensure:

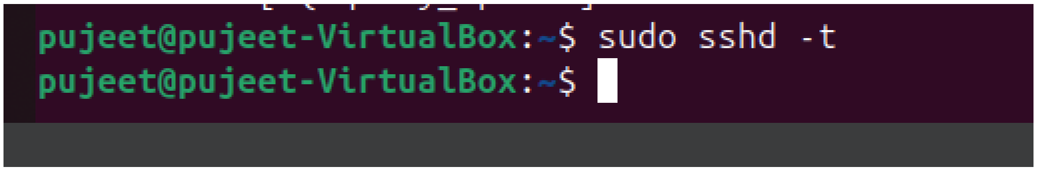
- Root login is completely disabled
- Password-based authentication is prohibited
- Only key-based authentication is allowed
- Authentication attempts are rate-limited

4.6 SSH Configuration Validation and Service Reload

Before applying the new SSH configuration, the syntax was validated to prevent misconfiguration that could result in loss of remote access.

Command executed:

```
sudo sshd -t
```



```
pujeet@pujeet-VirtualBox:~$ sudo sshd -t
pujeet@pujeet-VirtualBox:~$
```

The command returned no output, confirming that the SSH configuration was valid.

The SSH service was then reloaded to apply the new security settings. Successful application of the configuration was verified through a subsequent SSH login using key-based authentication.

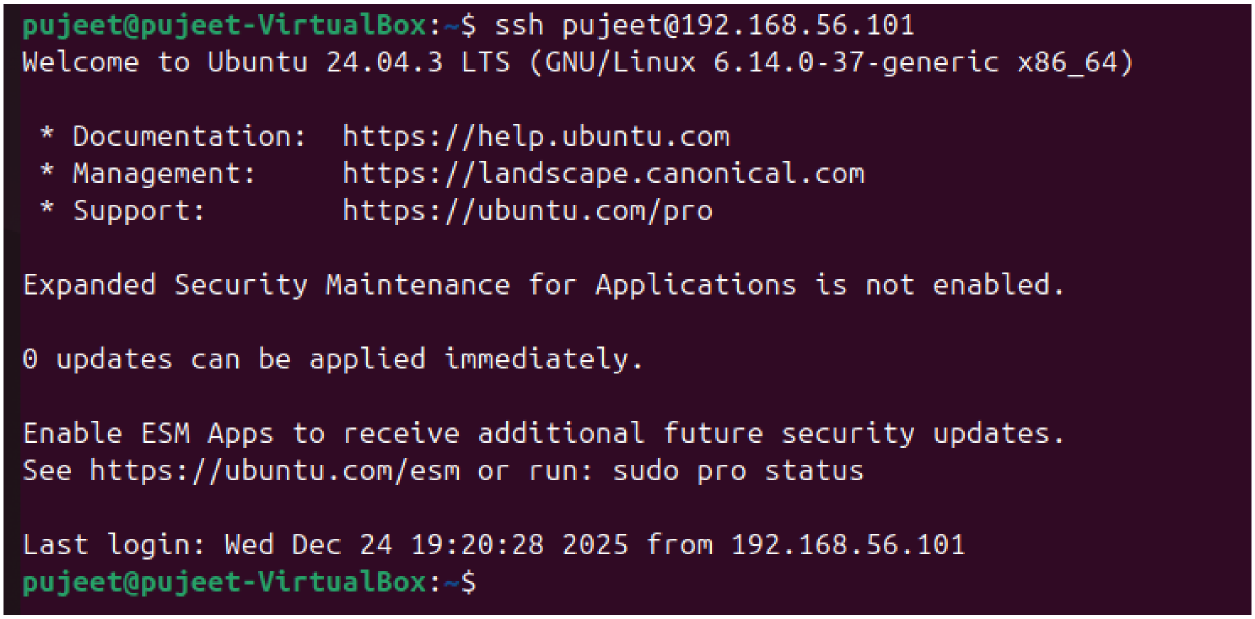
SSH configuration validation using `sshd -t`

4.7 Verification of Secure SSH Access

A new SSH session was initiated to confirm that the hardened configuration was functioning as intended.

Command executed:

```
ssh pujeet@192.168.56.101
```



```
pujeet@pujeet-VirtualBox:~$ ssh pujeet@192.168.56.101
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.14.0-37-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

Last login: Wed Dec 24 19:20:28 2025 from 192.168.56.101
pujeet@pujeet-VirtualBox:~$
```

The login completed successfully **without a password prompt**, confirming that:

- Password authentication was disabled
- SSH key-based authentication was enforced
- Remote access remained functional after hardening

User identity was verified to ensure access was granted to a non-root account:

```
whoami
```

Successful key-based SSH login and user identity verification

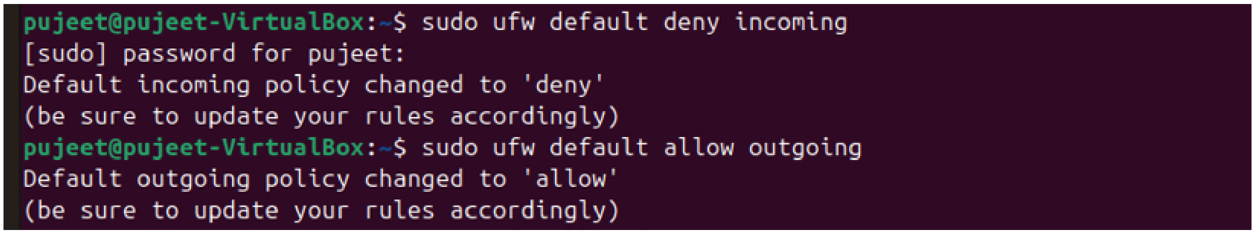
---

4.8 Firewall Configuration Using UFW

The Uncomplicated Firewall (UFW) was configured to restrict inbound traffic and reduce network exposure.

Default Firewall Policies

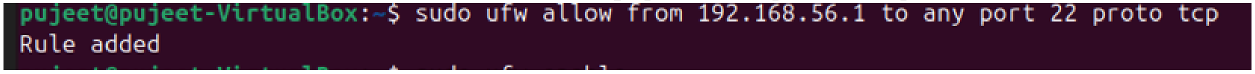
```
sudo ufw default deny incoming
sudo ufw default allow outgoing
```



```
pujeet@pujeet-VirtualBox:~$ sudo ufw default deny incoming
[sudo] password for pujeet:
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
pujeet@pujeet-VirtualBox:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Restrict SSH Access to Trusted Workstation

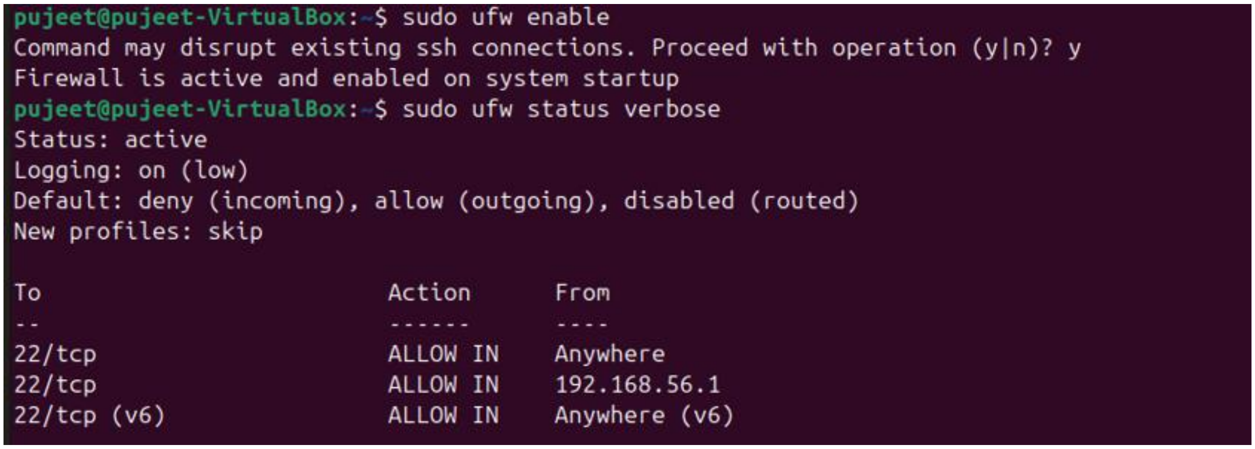
```
sudo ufw allow from 192.168.56.1 to any port 22 proto tcp
```



```
pujeet@pujeet-VirtualBox:~$ sudo ufw allow from 192.168.56.1 to any port 22 proto tcp
Rule added
```

The firewall was enabled and verified:

```
sudo ufw enable
sudo ufw status verbose
```



```
pujeet@pujeet-VirtualBox:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
pujeet@pujeet-VirtualBox:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
22/tcp                     ALLOW IN    192.168.56.1
22/tcp (v6)                ALLOW IN    Anywhere (v6)
```

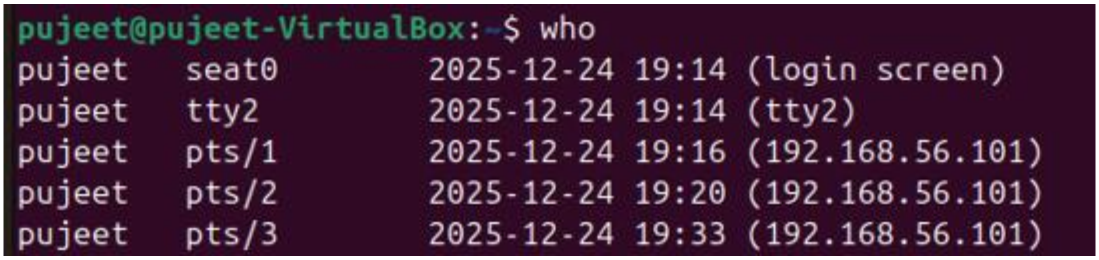
UFW firewall enabled
UFW rules restricting SSH access to workstation IP

4.9 Active Session Verification

Active SSH sessions were verified to confirm controlled remote access.

Command executed:

```
Who
```



```
pujeet@pujeet-VirtualBox:~$ who
pujeet   seat0        2025-12-24 19:14 (login screen)
pujeet   tty2         2025-12-24 19:14 (tty2)
pujeet   pts/1        2025-12-24 19:16 (192.168.56.101)
pujeet   pts/2        2025-12-24 19:20 (192.168.56.101)
pujeet   pts/3        2025-12-24 19:33 (192.168.56.101)
```

The output confirmed active SSH sessions originating from the trusted workstation IP.

Active SSH session verification

4.10 Reflection: Security Impact

Disabling password authentication and root login significantly reduces the risk of brute-force and credential-based attacks. Enforcing SSH key authentication ensures stronger cryptographic security, while firewall restrictions limit exposure to trusted sources only.

This phase demonstrates the effective application of **defence-in-depth**, combining authentication hardening and network-level controls. The resulting configuration aligns with industry-standard Linux server security practices used in enterprise and cloud environments.