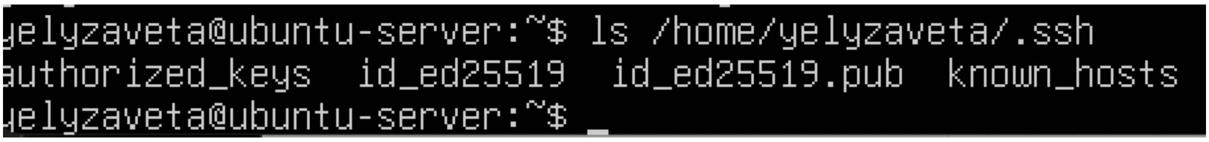
Week 8 – Assessment week 4

Task 1

SSH access was secured by configuring key-based authentication. This approach is more secure than password-only login because the server accepts connections only from clients that possess the correct private key.

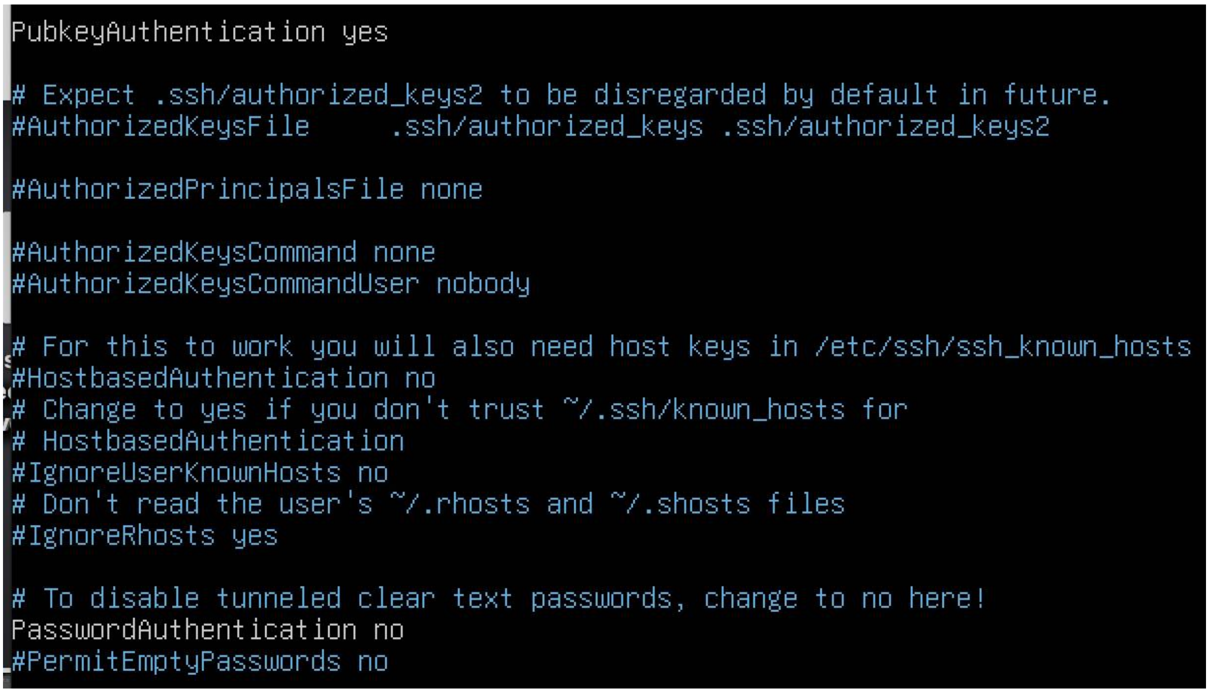
An SSH key pair was generated on the workstation using ssh-keygen. The public key was then copied to the server user account using ssh-copy-id, which adds the key to the server's ~/.ssh/authorized_keys file. After this, the connection was tested to confirm the server allows authentication using the SSH key.

Evidence of SSH Key Condiguration



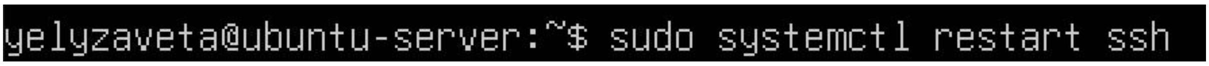
```
yelyzaveta@ubuntu-server:~$ ls /home/yelyzaveta/.ssh
authorized_keys  id_ed25519  id_ed25519.pub  known_hosts
yelyzaveta@ubuntu-server:~$ _
```

Output of the command showing SSH key files present in the user's .ssh directory.



```
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 no
#PermitEmptyPasswords no
```

SSH configuration file showing that public key authentication is enabled and password authentication is disabled.



```
yelyzaveta@ubuntu-server:~$ sudo systemctl restart ssh
```

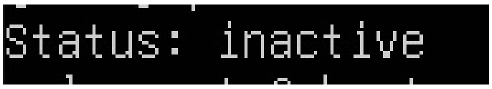
Restart of the SSH server to apply configuration changes.

Task 2

Firewall Condiguration Overview

A firewall was configured on the Ubuntu Server using UFW (Uncomplicated Firewall) to restrict SSH access to a single trusted workstation.
All incoming connections are denied by default, while outgoing connections are allowed.
SSH access (port 22) is explicitly permitted only from the administrator's workstation IP address, providing an additional layer of security against unauthorised access.

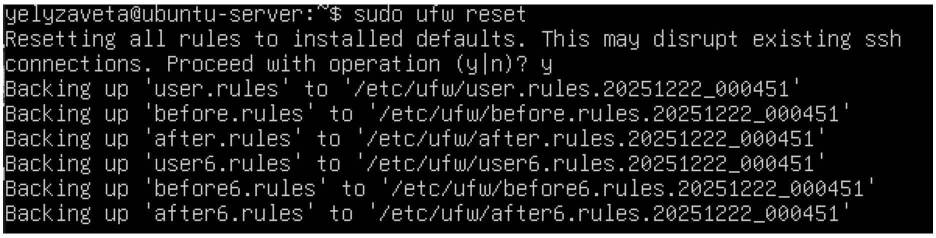
Step 1



```
Status: inactive
```

The firewall was initially inactive, allowing configuration from a clear state.

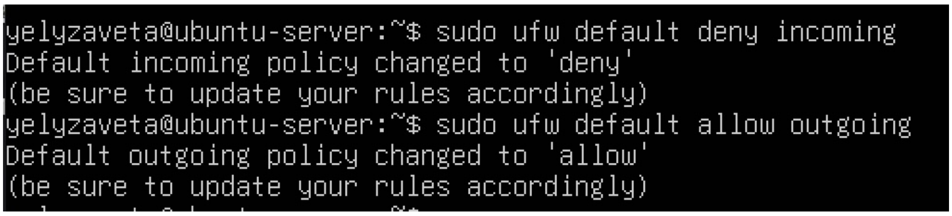
Step 2



```
yelyzaveta@ubuntu-server:~$ sudo ufw reset
Resetting all rules to installed defaults. This may disrupt existing ssh
connections. Proceed with operation (y|n)? y
Backing up 'user.rules' to '/etc/ufw/user.rules.20251222_000451'
Backing up 'before.rules' to '/etc/ufw/before.rules.20251222_000451'
Backing up 'after.rules' to '/etc/ufw/after.rules.20251222_000451'
Backing up 'user6.rules' to '/etc/ufw/user6.rules.20251222_000451'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20251222_000451'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20251222_000451'
```

Existing firewall rules were cleared to ensure a clean and controlled configuration.

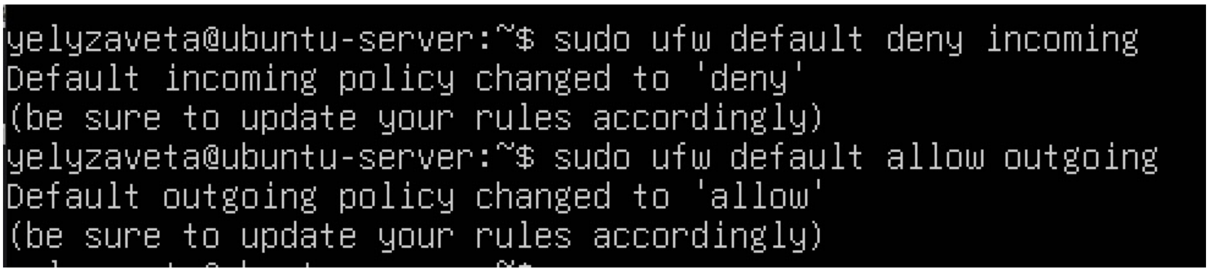
Step 3



```
yelyzaveta@ubuntu-server:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
yelyzaveta@ubuntu-server:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Default firewall policies were configured to deny all incoming traffic and allow all outgoing traffic.

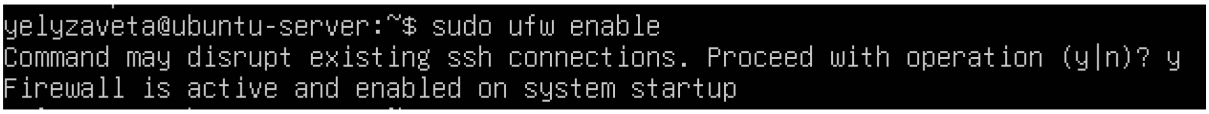
Step 4



```
yelyzaveta@ubuntu-server:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
yelyzaveta@ubuntu-server:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

SSH access was restricted to a single trusted workstation by allowing port 22 traffic only from the administrator's IP address.

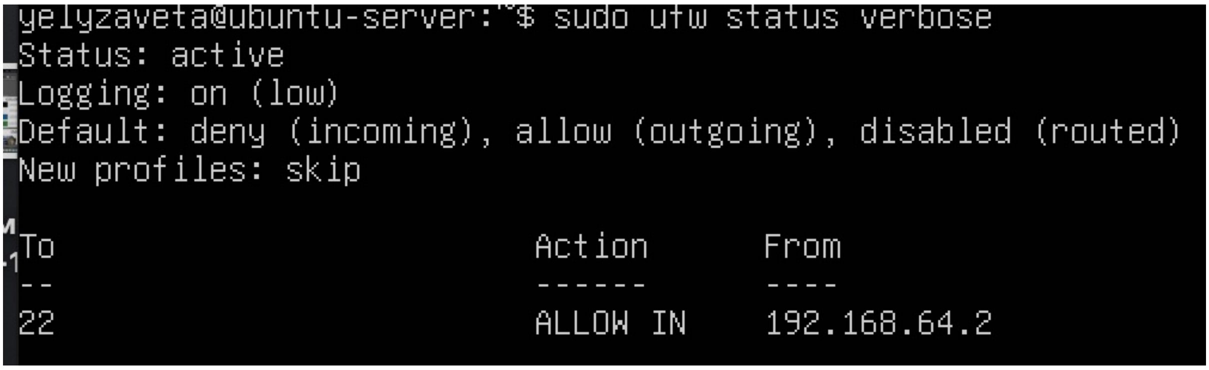
Step 5



```
yelyzaveta@ubuntu-server:~$ sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
```

The firewall was enabled to enforce all configured rules.

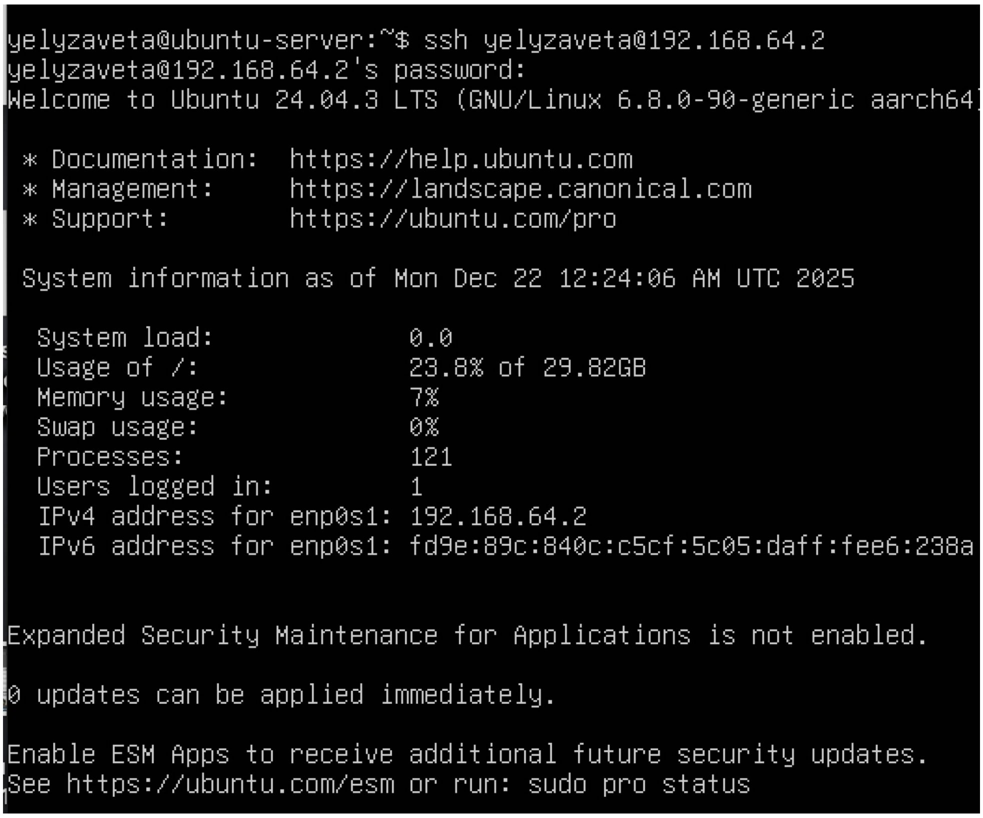
Step 6



```
yelyzaveta@ubuntu-server:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22                         ALLOW IN    192.168.64.2
```

The complete firewall ruleset was displayed in verbose mode to confirm correct configuration.

Step 7



```
yelyzaveta@ubuntu-server:~$ ssh yelyzaveta@192.168.64.2
yelyzaveta@192.168.64.2's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic aarch64)

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

 System information as of Mon Dec 22 12:24:06 AM UTC 2025

  System load:             0.0
  Usage of /:              23.8% of 29.82GB
  Memory usage:            7%
  Swap usage:              0%
  Processes:               121
  Users logged in:         1
  IPv4 address for enp0s1: 192.168.64.2
  IPv6 address for enp0s1: fd9e:89c:840c:c5cf:5c05:daff:fee6:238a


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
```

An SSH connection was tested from the authorised workstation to confirm that access is permitted through the firewall.

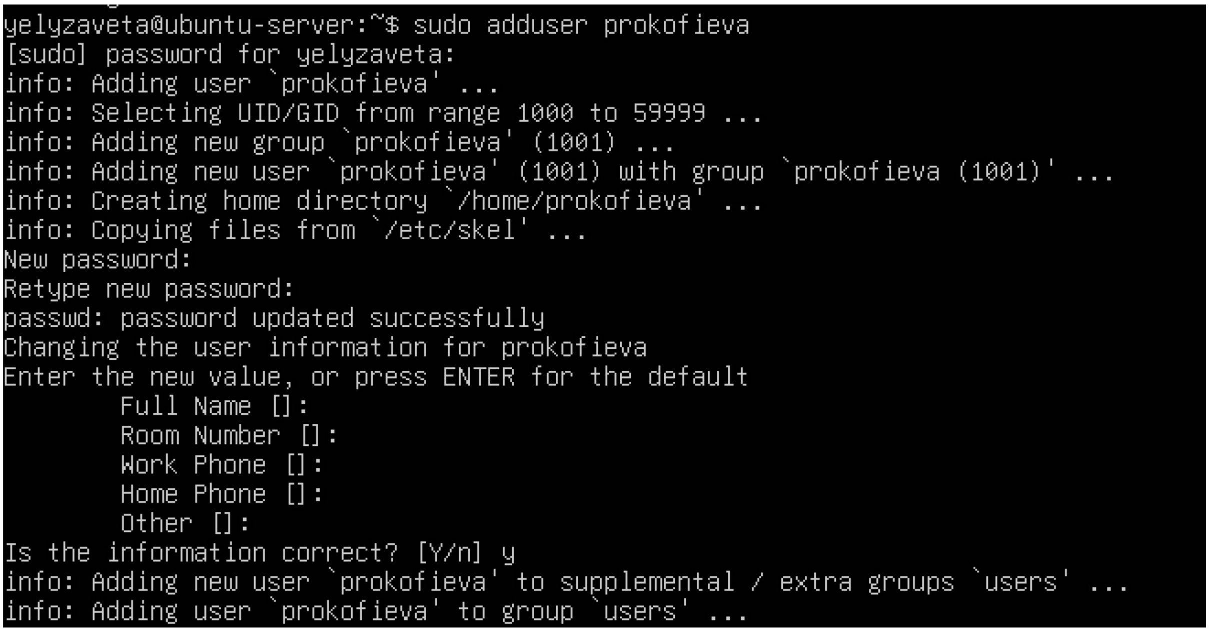
Outcome

The firewall successfully restricts SSH access to a single trusted workstation while blocking all other incoming connections. This configuration significantly reduces the attack surface of the server and ensures secure remote administration.

Task 3

To improve system security, direct administrative access using the root account was avoided. A dedicated non-root administrative user was created and granted controlled sudo privileges. This approach follows the principle of least privilege and reduces the risk associated with unrestricted root access.

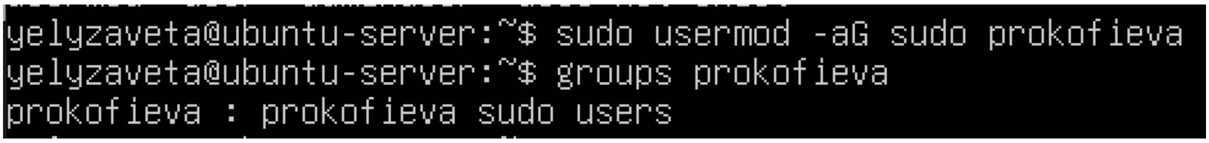
Step 1



```
yelyzaveta@ubuntu-server:~$ sudo adduser prokofieva
[sudo] password for yelyzaveta:
info: Adding user `prokofieva' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `prokofieva' (1001) ...
info: Adding new user `prokofieva' (1001) with group `prokofieva (1001)' ...
info: Creating home directory `/home/prokofieva' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for prokofieva
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
info: Adding new user `prokofieva' to supplemental / extra groups `users' ...
info: Adding user `prokofieva' to group `users' ...
```

A new user account was created for administrative purposes.

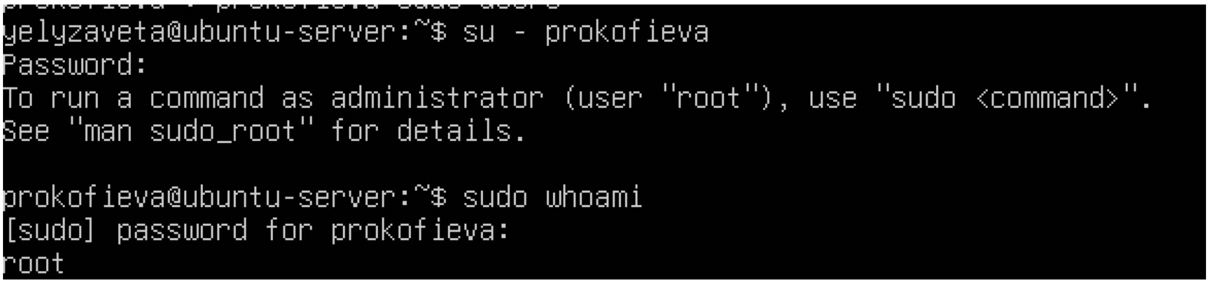
Step 2



```
yelyzaveta@ubuntu-server:~$ sudo usermod -aG sudo prokofieva
yelyzaveta@ubuntu-server:~$ groups prokofieva
prokofieva : prokofieva sudo users
```

The newly created user was added to the sudo group to allow administrative actions when required.

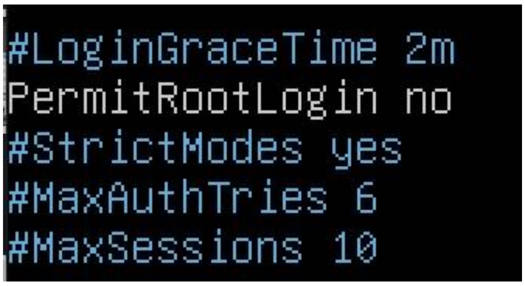
Step 3



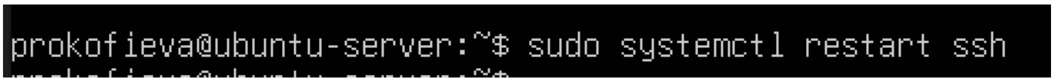
```
yelyzaveta@ubuntu-server:~$ su - prokofieva
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

prokofieva@ubuntu-server:~$ sudo whoami
[sudo] password for prokofieva:
root
```

The new user account was tested to ensure it can perform administrative tasks using sudo.

Step 4



```
#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
#MaxAuthTries 6
#MaxSessions 10
```



```
prokofieva@ubuntu-server:~$ sudo systemctl restart ssh
```

Direct SSH access for the root account was disabled to prevent unauthorised administrative access.

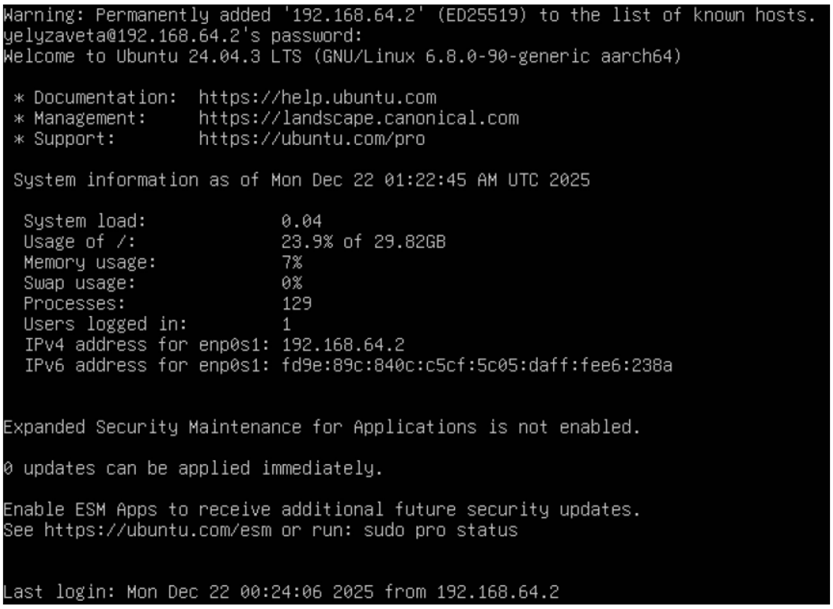
Outcome

A non-root administrative user was successfully created and configured with controlled sudo access. Root SSH login was disabled, ensuring that all administrative actions are traceable to individual user accounts. This configuration improves accountability, reduces security risks, and aligns with best practices for secure server administration.

Task 4

This task demonstrates successful remote administration of the server using SSH from an authorised workstation. The evidence confirms that SSH access is operational and restricted according to the security configuration implemented in previous tasks.

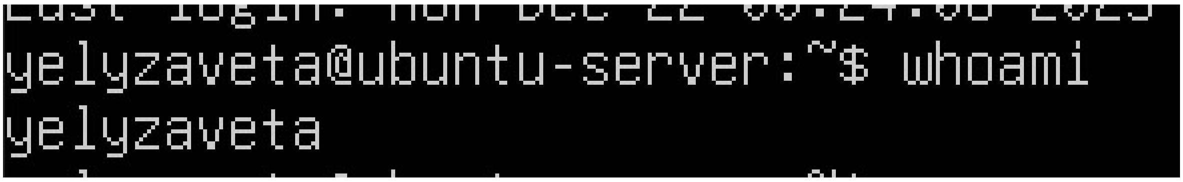
Step 1



```
Warning: Permanently added '192.168.64.2' (ED25519) to the list of known hosts.
yelyzaveta@192.168.64.2's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic aarch64)

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

 System information as of Mon Dec 22 01:22:45 AM UTC 2025

  System load:            0.04
  Usage of /:             23.9% of 29.82GB
  Memory usage:           7%
  Swap usage:             0%
  Processes:              129
  Users logged in:        1
  IPv4 address for enp0s1: 192.168.64.2
  IPv6 address for enp0s1: fd9e:89c:840c:c5cf:5c05:daff:fee6:238a


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 00:24:06 2025 from 192.168.64.2
```

An SSH connection was initiated from the authorised client machine to the Ubuntu server using the configured user account.

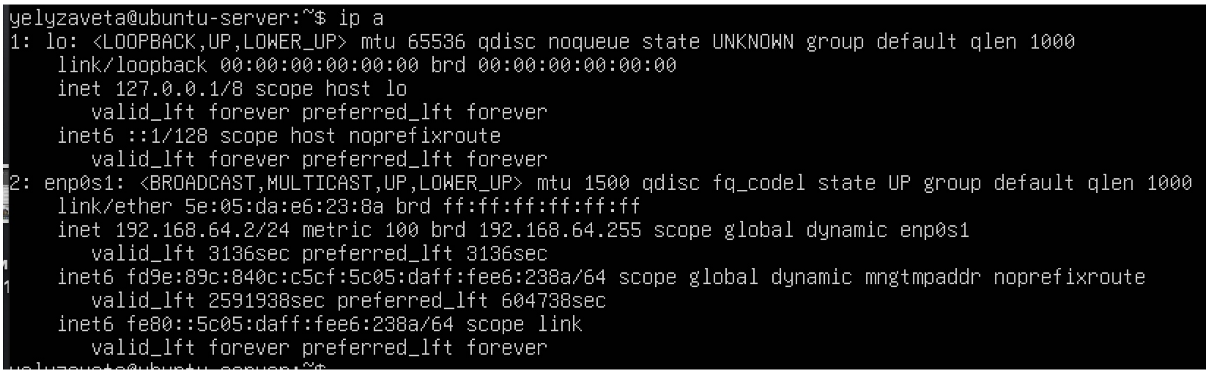
Step 2



```
Last login: Mon Dec 22 00:24:08 2025
yelyzaveta@ubuntu-server:~$ whoami
yelyzaveta
```

Once connected, commands were executed remotely to confirm active SSH access.

Step 3



```
yelyzaveta@ubuntu-server:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 5e:05:da:e6:23:8a brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.2/24 metric 100 brd 192.168.64.255 scope global dynamic enp0s1
       valid_lft 3136sec preferred_lft 3136sec
    inet6 fd9e:89c:840c:c5cf:5c05:daff:fee6:238a/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 2591938sec preferred_lft 604738sec
    inet6 fe80::5c05:daff:fee6:238a/64 scope link
       valid_lft forever preferred_lft forever
yelyzaveta@ubuntu-server:~$
```

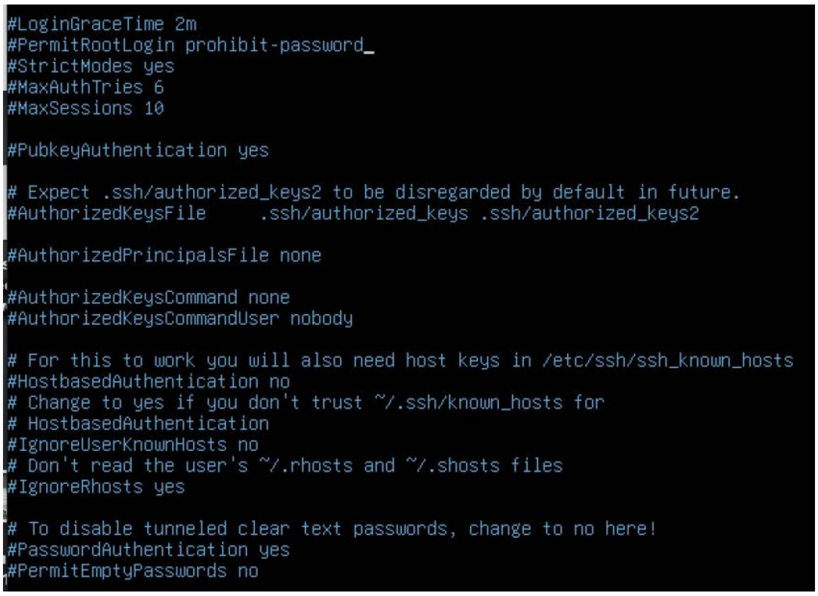
The server IP address was checked to confirm that the session is running on the remote server and not locally.

Outcome

The screenshots confirm a successful SSH connection from an authorised workstation to the server. All commands were executed remotely, demonstrating effective and secure remote administration. This evidence validates the SSH configuration implemented in Task 1 and supports the firewall restrictions applied in Task 2.

Task 5

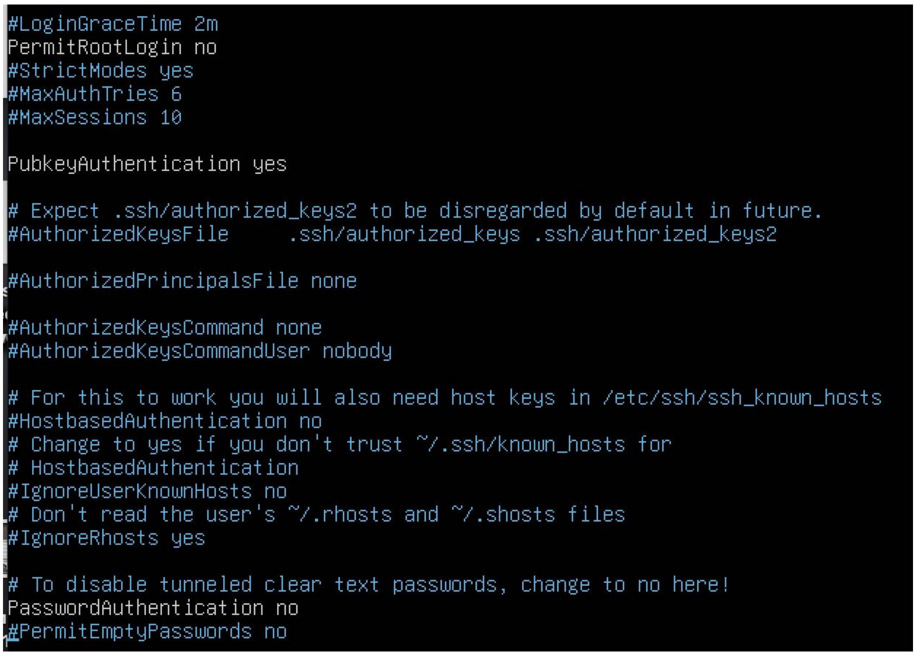
Step 1



```
#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
```

Before security hardening, the SSH daemon configuration allowed password-based authentication and did not explicitly enforce key-based access.

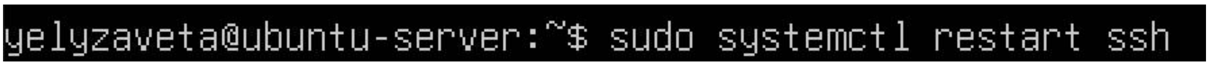
Step 2



```
#LoginGraceTime 2m
PermitRootLogin no
#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 no
#PermitEmptyPasswords no
```

The SSH configuration was updated to enforce key-based authentication and disable password-based login.

Step 3



```
yelyzaveta@ubuntu-server:~$ sudo systemctl restart ssh
```

The SSH service was restarted to apply the updated configuration.

Comparison Summary

Setting	Before Configuration	After Configuration
Public key authentication	Not enforced (commented)	Enabled
Password authentication	Enabled	Disabled
Root SSH login	Allowed	Disabled

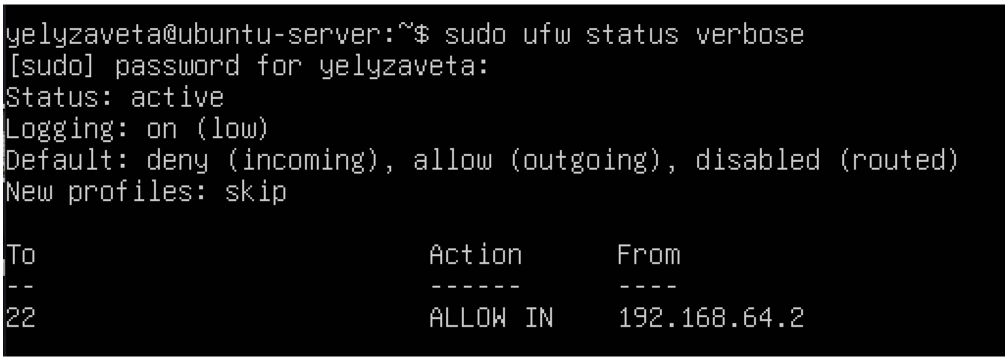
Outcome

The before-and-after comparison confirms that SSH security was successfully strengthened. Password-based authentication and root login were disabled, while key-based authentication was enforced. These changes significantly reduce the risk of unauthorised access and align with best practices for secure server administration.

Task 6

This task documents the complete firewall ruleset configured on the server using UFW. The ruleset confirms that the firewall is active, restrictive by default, and permits SSH access only from a trusted workstation.

Step 1



```
yelyzaveta@ubuntu-server:~$ sudo ufw status verbose
[sudo] password for yelyzaveta:
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22                         ALLOW IN    192.168.64.2
```

The full firewall configuration was displayed in verbose mode to verify all active rules.

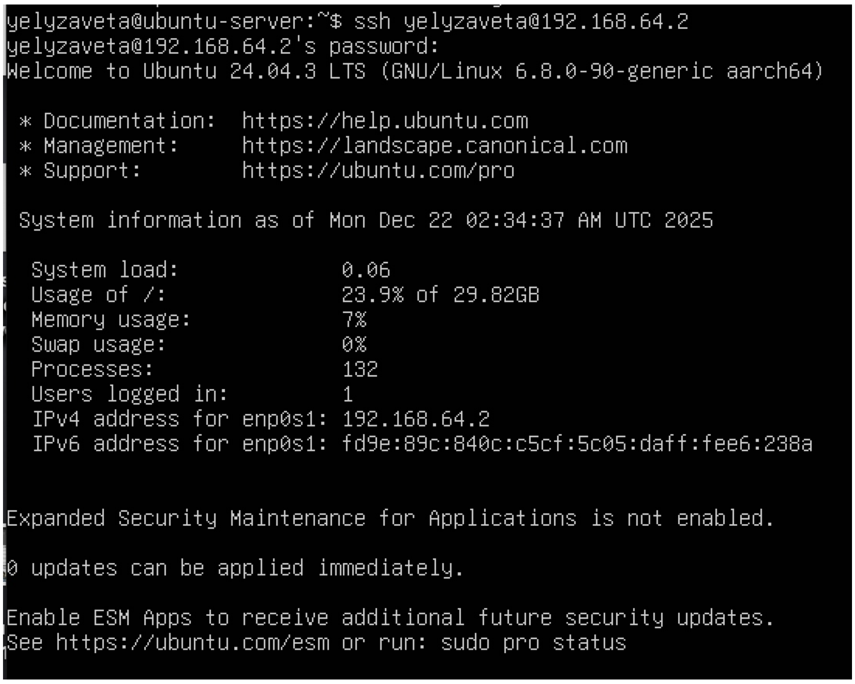
The SSH rule confirms that port 22 is only accessible from the authorised workstation.

Outcome

The firewall ruleset confirms that the server is protected by restrictive default policies and controlled access rules. SSH access is limited to a single trusted workstation, reducing exposure to unauthorised access attempts and supporting secure remote administration.

Task 7

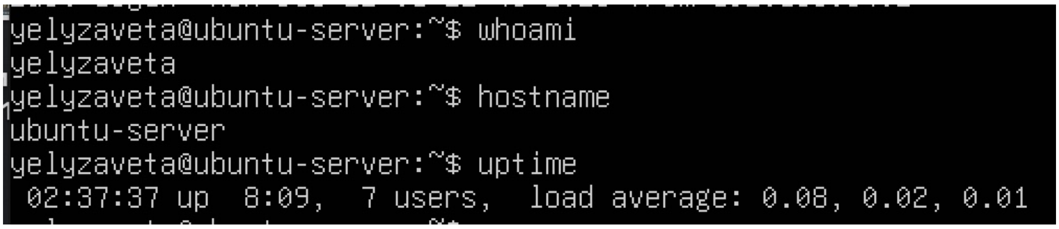
Step 1



```
yelyzaveta@ubuntu-server:~$ ssh yelyzaveta@192.168.64.2
yelyzaveta@192.168.64.2's password:
Welcome to Ubuntu 24.04.3 LTS (GNU/Linux 6.8.0-90-generic aarch64)

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

 System information as of Mon Dec 22 02:34:37 AM UTC 2025

  System load:             0.06
  Usage of /:              23.9% of 29.82GB
  Memory usage:            7%
  Swap usage:              0%
  Processes:               132
  Users logged in:         1
  IPv4 address for enp0s1: 192.168.64.2
  IPv6 address for enp0s1: fd9e:89c:840c:c5cf:5c05:daff:fee6:238a


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
```

An SSH connection was established from the authorised workstation to the server.

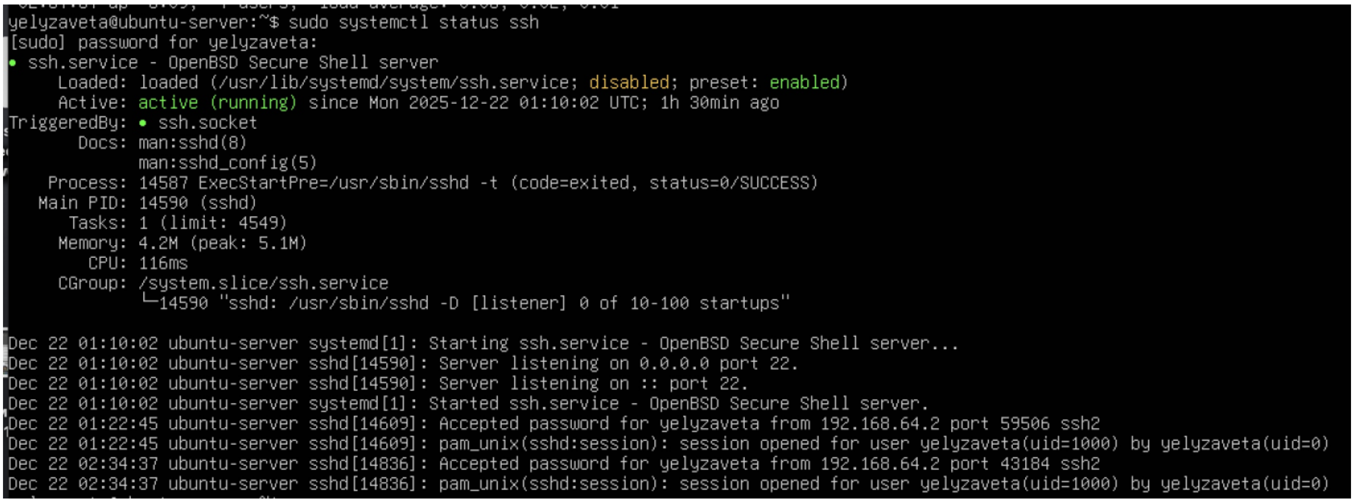
Step 2



```
yelyzaveta@ubuntu-server:~$ whoami
yelyzaveta
yelyzaveta@ubuntu-server:~$ hostname
ubuntu-server
yelyzaveta@ubuntu-server:~$ uptime
 02:37:37 up  8:09,  7 users,  load average: 0.08, 0.02, 0.01
```

Administrative commands were executed over the SSH connection to demonstrate remote system management.

Step 3



```
yelyzaveta@ubuntu-server:~$ sudo systemctl status ssh
[sudo] password for yelyzaveta:
● ssh.service - OpenBSD Secure Shell server
     Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
     Active: active (running) since Mon 2025-12-22 01:10:02 UTC; 1h 30min ago
TriggeredBy: ● ssh.socket
       Docs: man:sshd(8)
             man:sshd_config(5)
    Process: 14587 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 14590 (sshd)
      Tasks: 1 (limit: 4549)
     Memory: 4.2M (peak: 5.1M)
        CPU: 116ms
     CGroup: /system.slice/ssh.service
             └─14590 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Dec 22 01:10:02 ubuntu-server systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
Dec 22 01:10:02 ubuntu-server sshd[14590]: Server listening on 0.0.0.0 port 22.
Dec 22 01:10:02 ubuntu-server sshd[14590]: Server listening on :: port 22.
Dec 22 01:10:02 ubuntu-server systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
Dec 22 01:22:45 ubuntu-server sshd[14609]: Accepted password for yelyzaveta from 192.168.64.2 port 59506 ssh2
Dec 22 01:22:45 ubuntu-server sshd[14609]: pam_unix(sshd:session): session opened for user yelyzaveta(uid=1000) by yelyzaveta(uid=0)
Dec 22 02:34:37 ubuntu-server sshd[14836]: Accepted password for yelyzaveta from 192.168.64.2 port 43184 ssh2
Dec 22 02:34:37 ubuntu-server sshd[14836]: pam_unix(sshd:session): session opened for user yelyzaveta(uid=1000) by yelyzaveta(uid=0)
```

A privileged administrative command was executed using sudo to confirm administrative access.

Outcome

The evidence confirms that the server is fully managed remotely via SSH. Both standard and administrative commands were successfully executed from the authorised workstation, demonstrating secure and effective remote administration.