# Week 4: Initial System Configuration and Security Implementation

### 1. Aim of the Week

The aim of Week 4 was to deploy the Ubuntu server and implement foundational security controls. This week focused on securing remote access, managing users and privileges, and restricting network access using a firewall. All configuration tasks were performed remotely via SSH from the workstation, following professional system administration practices.

### 2. SSH Service Installation and Verification

To enable secure remote administration, the OpenSSH server was installed on the Ubuntu server. This allows the system to accept SSH connections from the workstation.

The SSH service was installed and verified using the package manager. Once installed, the service was confirmed to be active and ready to accept remote connections.

This step ensures that all further administration can be carried out remotely, which is a core requirement of this coursework.

### 3. Network Configuration and Server IP Identification

The server's network configuration was examined to identify its IP address. This is required to establish SSH connections and to configure firewall rules correctly.

Using IPv4 address inspection commands, the server was confirmed to be operating on the 192.168.56.0/24 network. The active network interface was identified, and routing information confirmed that the server was reachable from the workstation.

This IP address was later used for SSH access and firewall configuration.

### 4. User Management and Privilege Control

To follow the principle of least privilege, system administration was not performed using the root account. Instead, a non-root administrative user was created and assigned sudo privileges.

This approach limits the potential impact of account compromise and aligns with best security practices. Administrative tasks were carried out using this account with elevated privileges only when required.

This ensures that routine access does not have unrestricted control over the system.

---

**5. SSH Key-Based Authentication**

SSH key-based authentication was configured to improve security. This method replaces password-based authentication with cryptographic keys, reducing the risk of brute-force attacks.

An SSH key pair was generated on the workstation and securely copied to the server for the administrative user. Successful login using the SSH key confirmed that key-based authentication was functioning correctly.

Password-based SSH access was then disabled to ensure only authorized key holders could access the server.

**6. SSH Configuration Before and After Changes**

Before modifying the SSH configuration, a backup of the configuration file was created. This allows recovery if configuration errors occur.

The SSH configuration file was then updated to:

- Enable public key authentication
- Disable password authentication
- Disable direct root login

These changes significantly improve the security of remote access to the server.

The configuration was validated and the SSH service restarted to apply the changes.

## 7. Firewall Configuration

A firewall was configured using UFW to restrict access to the server. The firewall was set to deny all incoming traffic by default while allowing outgoing connections.

SSH access was explicitly allowed **only from the workstation's IP address**, ensuring that remote access is restricted to a trusted source.

After enabling the firewall, the full ruleset was reviewed to confirm that only the intended traffic was permitted.

## 8. Remote Administration Evidence

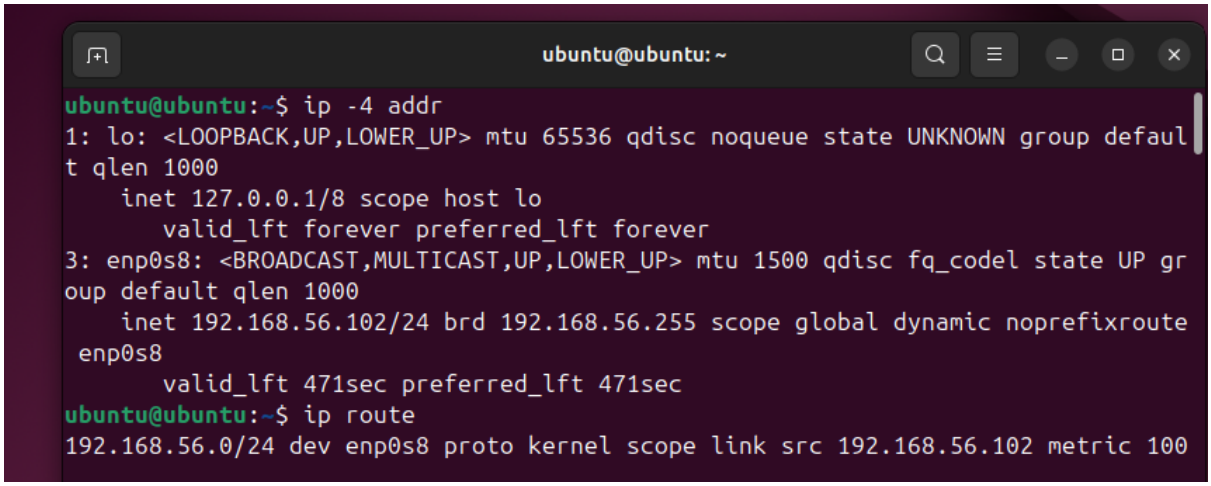
All administrative tasks during this week were performed remotely via SSH from the workstation. Commands such as network inspection, service management, and privilege escalation were executed within the SSH session.

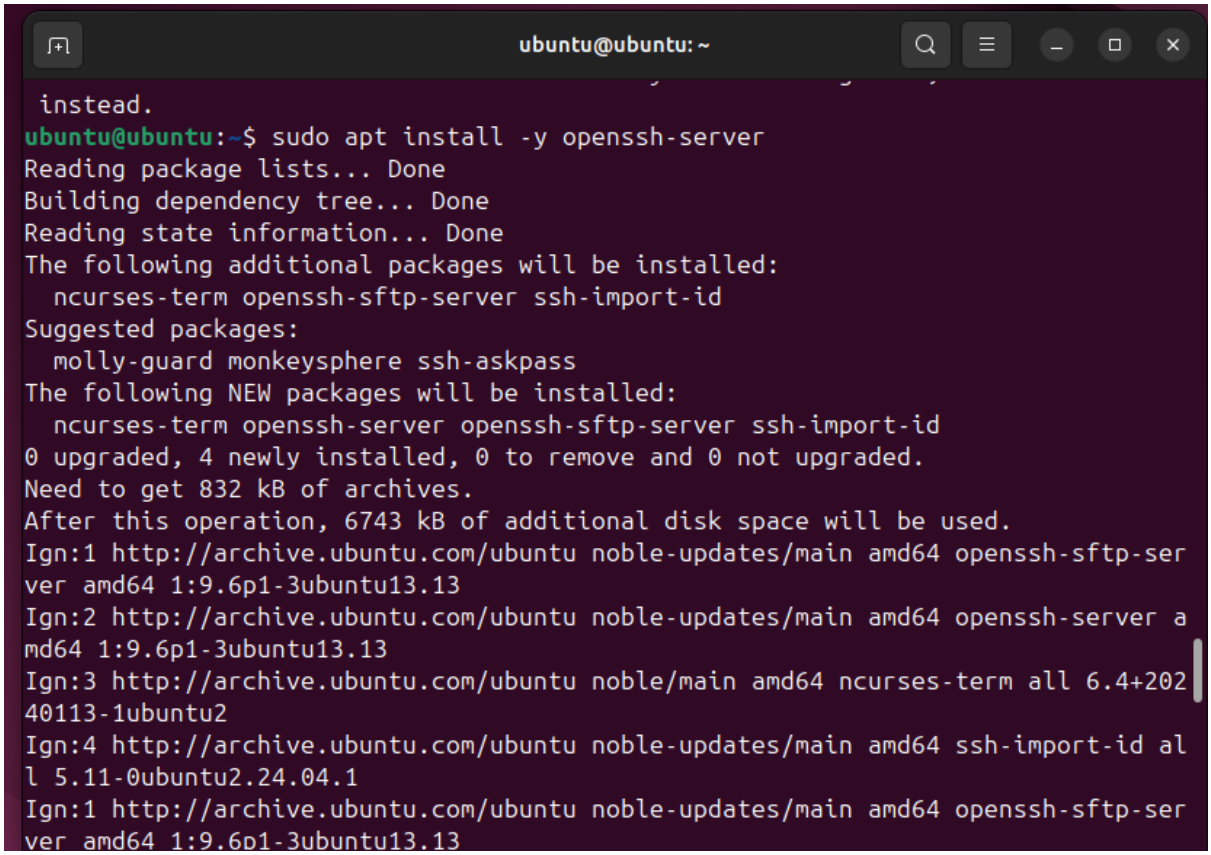
This demonstrates compliance with the administrative constraint and reflects real-world server management practices.

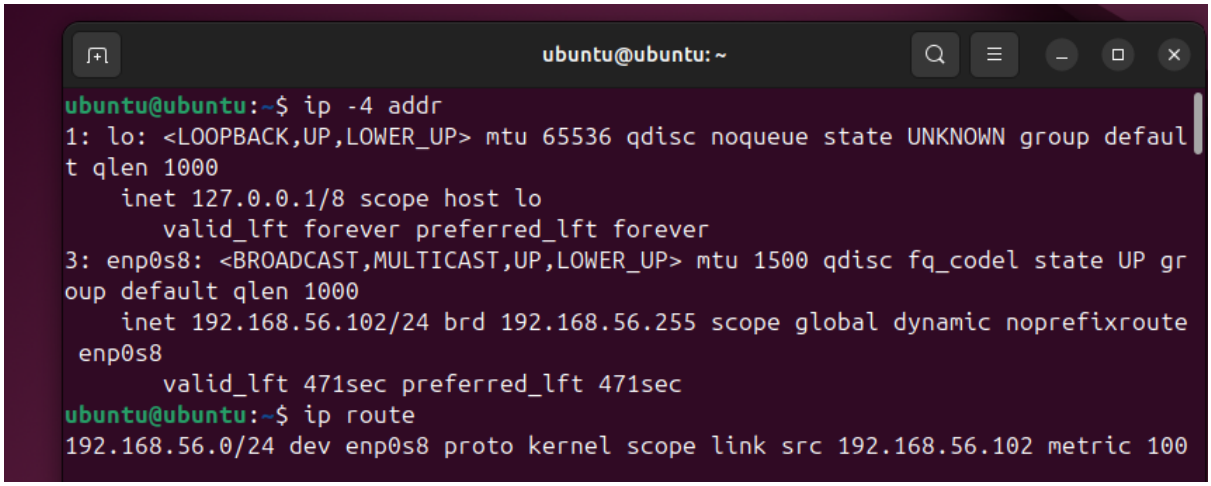
## 9. Reflection

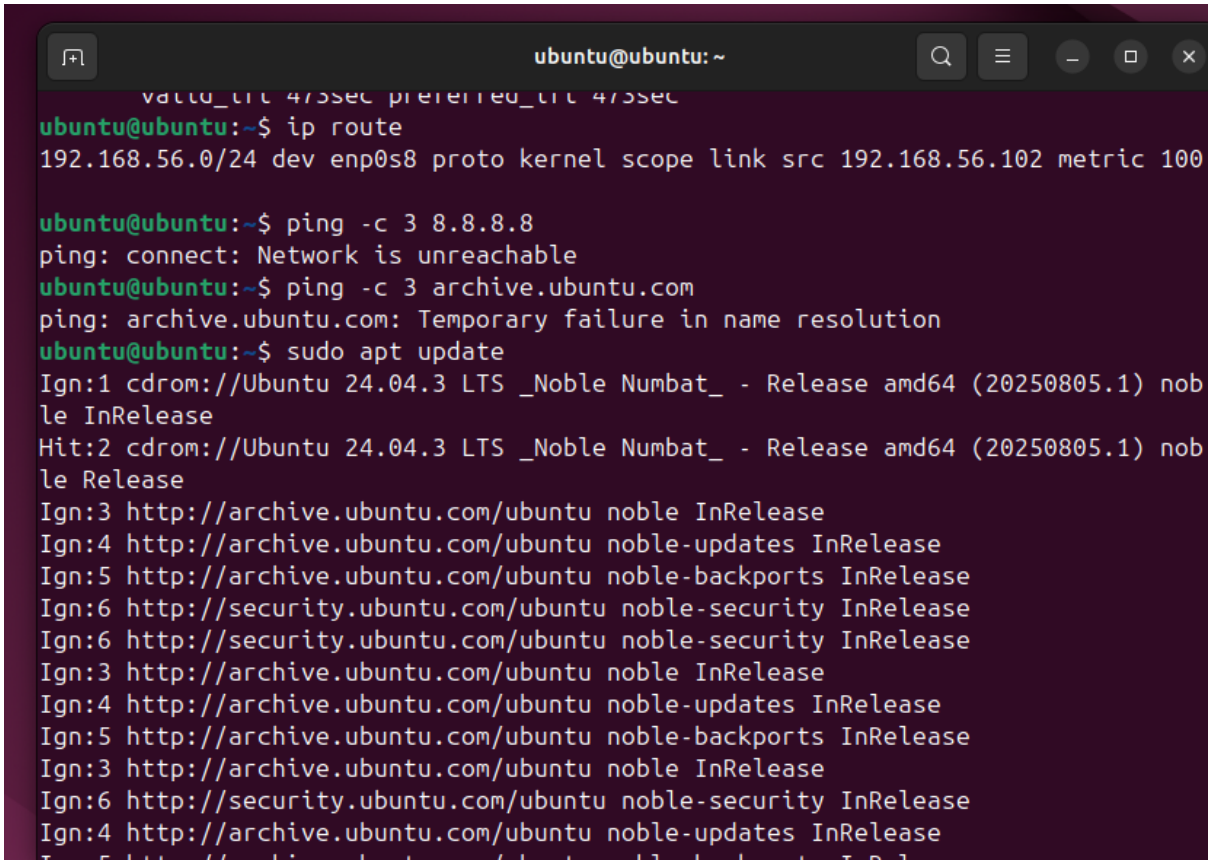
Week 4 was a significant step in securing the server. Implementing SSH key-based authentication and firewall rules improved my understanding of how access control and network security work together. Managing users and privileges highlighted the importance of limiting access to reduce security risks.

Completing all configuration remotely reinforced professional system administration techniques and prepared the server for advanced security monitoring and performance evaluation in later weeks.


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ip -4 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute
enp0s8
valid_lft 471sec preferred_lft 471sec
ubuntu@ubuntu:~$ ip route
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.102 metric 100


ubuntu@ubuntu: ~
instead.
ubuntu@ubuntu:~$ sudo apt install -y openssh-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
ncurses-term openssh-sftp-server ssh-import-id
Suggested packages:
molly-guard monkeysphere ssh-askpass
The following NEW packages will be installed:
ncurses-term openssh-server openssh-sftp-server ssh-import-id
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 832 kB of archives.
After this operation, 6743 kB of additional disk space will be used.
Ign:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-sftp-ser
ver amd64 1:9.6p1-3ubuntu13.13
Ign:2 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-server a
md64 1:9.6p1-3ubuntu13.13
Ign:3 http://archive.ubuntu.com/ubuntu noble/main amd64 ncurses-term all 6.4+202
40113-1ubuntu2
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 ssh-import-id al
l 5.11-0ubuntu2.24.04.1
Ign:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 openssh-sftp-ser
ver amd64 1:9.6p1-3ubuntu13.13


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ip -4 addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
inet 192.168.56.102/24 brd 192.168.56.255 scope global dynamic noprefixroute
enp0s8
valid_lft 471sec preferred_lft 471sec
ubuntu@ubuntu:~$ ip route
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.102 metric 100


ubuntu@ubuntu: ~
ubuntu@ubuntu:~$ ip route
192.168.56.0/24 dev enp0s8 proto kernel scope link src 192.168.56.102 metric 100
ubuntu@ubuntu:~$ ping -c 3 8.8.8.8
ping: connect: Network is unreachable
ubuntu@ubuntu:~$ ping -c 3 archive.ubuntu.com
ping: archive.ubuntu.com: Temporary failure in name resolution
ubuntu@ubuntu:~$ sudo apt update
Ign:1 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) nob
le InRelease
Hit:2 cdrom://Ubuntu 24.04.3 LTS _Noble Numbat_ - Release amd64 (20250805.1) nob
le Release
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Ign:5 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Ign:3 http://archive.ubuntu.com/ubuntu noble InRelease
Ign:6 http://security.ubuntu.com/ubuntu noble-security InRelease
Ign:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease