Case Study 1: Setting Up a Personal Linux Server

Objective:

Learn how to set up and manage a basic Linux server for hosting files, websites, and services.

Scenario:

A small business wants to set up a local Linux server for file sharing and hosting their website.

Steps:

1. **Install Linux:**
    - Choose a lightweight Linux distribution (e.g., Ubuntu Server or CentOS).
    - Install and configure the OS on a virtual machine or physical hardware.
2. **Set Up SSH Access:**
    - Configure sshd to allow secure remote access.
    - Restrict root login and enable key-based authentication.
3. **File Sharing:**
    - Set up file sharing using Samba or NFS for local network access.
4. **Web Hosting:**
    - Install and configure a web server (Apache or Nginx).
    - Host a simple static HTML site.
5. **Backup Solution:**

    - Automate backups using rsync or tar and schedule them with cron.

Key Skills Practiced:

- Server setup and hardening
- File sharing and permissions
- Web server configuration

Backup and scheduling

SOLUTION

Here is a step-by-step guide to setting up and managing a basic Linux server for hosting files, websites, and services:

1. Choose and Install a Linux Distribution

Steps:

1. **Select a Distribution**:
   - Popular choices: Ubuntu Server, CentOS, Debian, or Rocky Linux.
2. **Download ISO**:
   - Visit the official website of the chosen Linux distribution and download the ISO file.
3. **Install Linux**:
   - Create a bootable USB drive using tools like Rufus or Balena Etcher.
   - Boot from the USB and follow the installation prompts.
   - Set up a hostname and choose partitioning options (LVM is recommended for flexibility).

2. Perform Initial Configuration

Steps:

1. **Update the System**:

   ```
   bash
   CopyEdit
   sudo apt update && sudo apt upgrade -y   # For
   Ubuntu/Debian
   sudo yum update -y                       # For
   CentOS/RHEL
   ```

2. **Set a Static IP Address**:
   - Edit the network configuration file (e.g., `/etc/netplan/` for Ubuntu or `/etc/sysconfig/network-scripts/` for CentOS).
   - Example for Ubuntu (Netplan):

```
yaml
CopyEdit
network:
  version: 2
  ethernets:
    eth0:
      addresses:
        - 192.168.1.100/24
      gateway4: 192.168.1.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

- Apply changes:

```
bash
CopyEdit
sudo netplan apply
```

3. **Set Hostname**:

```
bash
CopyEdit
sudo hostnamectl set-hostname server-name
```

3. Set Up Secure SSH Access

Steps:

1. **Install OpenSSH Server**:

```
bash
CopyEdit
sudo apt install openssh-server -y  # For
Ubuntu/Debian
sudo yum install openssh-server -y  # For
CentOS/RHEL
```

2. **Enable and Start SSH**:

```
bash
CopyEdit
sudo systemctl enable ssh
sudo systemctl start ssh
```

3. **Harden SSH**:
   - Edit the SSH configuration file:

     ```
     bash
     CopyEdit
     sudo nano /etc/ssh/sshd_config
     ```

   - Recommended changes:
     - Change the default port (e.g., `Port 2222`).
     - Disable root login: `PermitRootLogin no`.
     - Enable key-based authentication: `PasswordAuthentication no`.
   - Restart SSH:

     ```
     bash
     CopyEdit
     sudo systemctl restart ssh
     ```

4. **Set Up SSH Keys**:
   - Generate keys on your local machine:

     ```
     bash
     CopyEdit
     ssh-keygen -t rsa
     ```

   - Copy the public key to the server:

     ```
     bash
     CopyEdit
     ssh-copy-id user@server-ip
     ```

4. Set Up a Web Server

Steps:

1. **Install Apache or Nginx**:
   - For Apache:

     ```
     bash
     CopyEdit
     ```

```
sudo apt install apache2 -y   # For
Ubuntu/Debian
sudo yum install httpd -y     # For
CentOS/RHEL
```

- For Nginx:

  ```
  bash
  CopyEdit
  sudo apt install nginx -y     # For
  Ubuntu/Debian
  sudo yum install nginx -y     # For
  CentOS/RHEL
  ```

2. **Start and Enable the Web Server**:

   ```
   bash
   CopyEdit
   sudo systemctl start apache2   # For Apache on
   Ubuntu
   sudo systemctl start httpd     # For Apache on
   CentOS
   sudo systemctl start nginx     # For Nginx
   sudo systemctl enable apache2/httpd/nginx
   ```

3. **Test the Web Server**:
   - Open a browser and visit your server's IP address (e.g., `http://192.168.1.100`).
   - You should see the default web server page.
4. **Host a Website**:
   - Place your HTML files in the web server's root directory:
     - Apache: `/var/www/html/`
     - Nginx: `/usr/share/nginx/html/`

---

5. Set Up File Sharing

Steps:

1. **Install Samba (for Windows/Linux clients)**:

   ```
   bash
   CopyEdit
   ```

```
sudo apt install samba -y  # For Ubuntu/Debian
sudo yum install samba -y  # For CentOS/RHEL
```

2. **Configure Samba**:
    - Edit the Samba configuration file:

      ```
      bash
      CopyEdit
      sudo nano /etc/samba/smb.conf
      ```

    - Add a share:

      ```
      ini
      CopyEdit
      [SharedFolder]
      path = /srv/shared
      browseable = yes
      read only = no
      guest ok = yes
      ```

    - Create the shared directory:

      ```
      bash
      CopyEdit
      sudo mkdir -p /srv/shared
      sudo chmod 777 /srv/shared
      ```

3. **Start Samba**:

   ```
   bash
   CopyEdit
   sudo systemctl start smbd
   sudo systemctl enable smbd
   ```

4. **Access the Share**:
    - Access the share from a Windows/Linux client using the server IP.

---

6. Automate Backups

Steps:

1. **Install Rsync**:

```
bash
CopyEdit
sudo apt install rsync -y   # For Ubuntu/Debian
sudo yum install rsync -y   # For CentOS/RHEL
```

2. **Create a Backup Script**:
   - Example script (`backup.sh`):

     ```
     bash
     CopyEdit
     #!/bin/bash
     rsync -av --delete /var/www/html/
     /backup/html/
     ```

   - Make it executable:

     ```
     bash
     CopyEdit
     chmod +x backup.sh
     ```

3. **Schedule Backups**:
   - Edit the `cron` jobs:

     ```
     bash
     CopyEdit
     crontab -e
     ```

   - Add a backup schedule (e.g., daily at midnight):

     ```
     bash
     CopyEdit
     0 0 * * * /path/to/backup.sh
     ```

---

7. Monitor and Maintain the Server

Steps:

1. **Install Monitoring Tools**:
   - System performance: `htop`, `glances`
   - Logs: `journalctl`, `logwatch`

```
bash
CopyEdit
```

```
sudo apt install htop glances -y
```

2. **Enable Automatic Updates**:

```
bash
CopyEdit
sudo apt install unattended-upgrades -y
sudo dpkg-reconfigure --priority=low unattended-
upgrades
```

3. **Check Disk Usage**:

```
bash
CopyEdit
df -h
```

4. **Clean Up Unused Files**:

```
bash
CopyEdit
sudo apt autoremove -y
sudo apt clean
```