==========

Step-by-Step Linux Practical Assignment Lab .

This covers **User Management, File Permissions, Apache/Nginx/Tomcat setup, Python, Cron, LAMP, SED, Find, Grep, VI, Log Management, Disk Usage, IP checks, Environment Variables**, etc.

🐧 Linux Admin Beginner Assignment – Full Step-by-Step Lab

✅ 1. User Management: Create 5 Users and Grant Sudo to One

bash

```
# Create 5 users
sudo useradd user1
sudo useradd user2
sudo useradd user3
sudo useradd user4
sudo useradd user5

# Set password for users
echo "password123" | sudo passwd --stdin user1
echo "password123" | sudo passwd --stdin user2
echo "password123" | sudo passwd --stdin user3
echo "password123" | sudo passwd --stdin user4
echo "password123" | sudo passwd --stdin user5

# Grant sudo to user1
```

==========

```
sudo usermod -aG wheel user1
```

✅ 2. Create WordPress Directory Tree and Set Permissions (Read/Write for Others)

bash

```
# Create directory structure
mkdir -p /var/www/html/wordpress

# Set permissions: Read/Write for Others
chmod -R o+rw /var/www/html/wordpress

# Verify
ls -ld /var/www/html/wordpress
```

✅ 3. Create Python Scripts Directory and Give Execute Permission

bash

```
mkdir /home/user1/python_scripts

# Give execute permission
chmod -R +x /home/user1/python_scripts
```

✅ 4. SED Command Use Case: Replace Text in a File

bash

==========

```
echo "Welcome to Linux Class" > demo.txt

# Replace 'Linux' with 'Cloud'
sed -i 's/Linux/Cloud/g' demo.txt

# View file
cat demo.txt
```

✅ 5. FIND Command: Find Files Modified in Last 1 Day

bash

```
find /home -type f -mtime -1
```

✅ 6. GREP Use Case: pecific Keyword in Logs

bash

```
grep "error" /var/log/messages
```

✅ 7. VI Editor Use Cases:

- Open file:
  `vi demo.txt`

- Insert text
  Press `i`, type something.

==========

- Save and exit
  Press `Esc`, then type `:wq`

- Search word in vi:
  Press `/keyword` then Enter.

✅ 8. LAMP Setup (Linux, Apache, MySQL, PHP):

bash

```
# Apache
sudo yum install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd

# MySQL (MariaDB)
sudo yum install mariadb-server -y
sudo systemctl start mariadb
sudo systemctl enable mariadb

# PHP
sudo yum install php php-mysql -y
sudo systemctl restart httpd
```

✅ 9. NGINX Setup: as web server

bash

```
sudo yum install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
```

==========

✅ 10. Download Sample File from URL and Unzip:

bash

```
wget https://wordpress.org/latest.zip
unzip latest.zip
```

✅ 11. Download tar.gz and Untar It:

bash

```
wget http://example.com/sample.tar.gz
tar -xvzf sample.tar.gz
```

✅ 12. Start and Check Status of Tomcat and NGINX:

bash

```
# Tomcat
sudo systemctl start tomcat
sudo systemctl status tomcat

# NGINX
sudo systemctl start nginx
sudo systemctl status nginx
```

✅ 13. Check Disk Usage and Memory Usage:

==========

bash

```
df -h          # Disk
free -m        # Memory
```

✅ 14. Set Hostname:

bash

```
sudo hostnamectl set-hostname webserver
hostname
```

✅ 15. Check Private and Public IP Address:

bash

```
# Private IP
ip a

# Public IP
curl ifconfig.me
```

✅ 16. Find and Kill Specific Service (Example: Apache):

bash

```
ps -ef | grep httpd
sudo kill <PID>
```

==========

✅ 17. Mount File System Example:

bash

```
# Check available disks
lsblk

# Format and Mount Example Disk (e.g., /dev/xvdf)
sudo mkfs.ext4 /dev/xvdf
sudo mkdir /mnt/data
sudo mount /dev/xvdf /mnt/data

# Check mount
df -h
```

✅ 18. Crontab Use Case: Run Python Script Every Day at 12 AM:

bash

```
crontab -e

# Add below line
0 0 * * * /usr/bin/python3 /home/user1/python_scripts/myscript.py >>
/home/user1/python_scripts/log.txt 2>&1
```

✅ 19. Read System, NGINX, and Tomcat Logs:

bash

```
# System Logs
```

==========

```
tail -f /var/log/messages

# NGINX Logs
tail -f /var/log/nginx/access.log

# Tomcat Logs
tail -f /opt/tomcat/logs/catalina.out
```

✅ 20. Log Rotation Automatic (Example for NGINX):

bash

```
sudo logrotate /etc/logrotate.conf
```

Configuration file: `/etc/logrotate.d/nginx`

✅ 21. Increase /tmp Space and Swap (Example):

- **Increase Swap:**

bash

```
sudo fallocate -l 1G /swapfile
sudo chmod 600 /swapfile
sudo mkswap /swapfile
sudo swapon /swapfile
```

==========

- **/tmp Space:**
  Extend disk or mount a larger disk on `/tmp` as per requirement.

✅ 22. Environment Variables Example:

bash

```
export APP_ENV=production
echo $APP_ENV
```

🐧 Full Web Hosting Practical Lab – AWS EC2 (Amazon Linux 2) – With Cloud Infotech Website Content

---

✅ Initial Setup for All Labs:

Step 1: Launch AWS EC2 Instance (Amazon Linux 2)

- Instance Type: **t2.micro (Free Tier) or higher**
- Open **Security Group Ports:**
  - **22 (SSH)**
  - **80 (HTTP)**

==========

- **8080 (Tomcat & Jenkins)**
- **5000 (Flask App)**

✅ 1. Static Website on NGINX (Cloud Infotech Website)

Step 1: Install NGINX:

bash

```
sudo amazon-linux-extras enable nginx1
sudo yum install nginx -y
sudo systemctl start nginx
sudo systemctl enable nginx
```

Step 2: Deploy Cloud Infotech Website Homepage:

bash

```
sudo vi /usr/share/nginx/html/index.html
```

Paste this content:

html

```
<!DOCTYPE html>
<html>
<head>
    <title>Cloud Infotech Trainings - Smart Degree Revolution</title>
    <style>
        body { font-family: Arial; background-color: #f2f2f2;
text-align: center; margin-top: 50px; }
```

==========

```
        h1 { color: #007BFF; }
        p { font-size: 18px; color: #333; }
        .highlight { color: #FF5722; font-weight: bold; }
    </style>
</head>
<body>
    <h1>Welcome to Cloud Infotech Trainings</h1>
    <p>Smart Degree + IT Job Ready Skills + MMA Sports</p>
    <p>UGC Approved Universities | 100% Placement Support</p>
    <p>Contact: <strong>8688253560</strong> |
info@cloudinfotechsolutions.com</p>
</body>
</html>
```

Access:

```
http://<EC2 Public-IP>/
```

✅ 2. Apache PHP (LAMP Stack) with Cloud Infotech Info

Step 1: Install Apache and PHP:

bash

```
sudo yum install httpd -y
sudo amazon-linux-extras enable php7.4
sudo yum install php php-mysqlnd -y
sudo systemctl start httpd
sudo systemctl enable httpd
```

Step 2: Deploy Sample PHP Cloud Info Page:

bash

==========

```
sudo vi /var/www/html/index.php
```

Paste:

php

```
<?php
echo "<h1>Cloud Infotech Trainings - PHP LAMP Site</h1>";
echo "<p>Smart Degree + IT + Placement | Contact: 8688253560</p>";
?>
```

Access:

```
http://<EC2 Public-IP>/index.php
```

✅ 3. WordPress Hosting (with MySQL)

Step 1: Install MySQL (MariaDB):

bash

```
sudo yum install mariadb-server -y
sudo systemctl start mariadb
sudo systemctl enable mariadb
mysql_secure_installation
```

Step 2: Configure MySQL Database:

bash

```
mysql -u root -p

CREATE DATABASE cloudwordpress;
```

==========

```
CREATE USER 'wpuser'@'localhost' IDENTIFIED BY 'password123';
GRANT ALL PRIVILEGES ON cloudwordpress.* TO 'wpuser'@'localhost';
FLUSH PRIVILEGES;
exit;
```

Step 3: Install WordPress:

bash

```
cd /tmp
wget https://wordpress.org/latest.tar.gz
tar -xvzf latest.tar.gz
sudo cp -r wordpress /var/www/html/
sudo chown -R apache:apache /var/www/html/wordpress
sudo systemctl restart httpd
```

Step 4: Configure WordPress:

Edit `/var/www/html/wordpress/wp-config.php` with database details:

php

```
DB_NAME: cloudwordpress
DB_USER: wpuser
DB_PASSWORD: password123
```

Access:
`http://<EC2 Public-IP>/wordpress/`

✅ 4. Tomcat9 Java Hosting (Cloud Infotech Sample Java App)

==========

Step 1: Java 11:

bash

```
sudo amazon-linux-extras enable corretto11
sudo yum install java-11-amazon-corretto -y
```

Step 2: Tomcat9 Manual Installation:

bash

```
cd /opt
sudo wget
https://downloads.apache.org/tomcat/tomcat-9/v9.0.85/bin/apache-tomcat
-9.0.85.tar.gz
sudo tar xvf apache-tomcat-9.0.85.tar.gz
sudo mv apache-tomcat-9.0.85 tomcat9
```

Step 3: Deploy Sample WAR App:

bash

```
cd /opt/tomcat9/webapps
sudo wget
https://tomcat.apache.org/tomcat-7.0-doc/appdev/sample/sample.war
```

Start Tomcat:

bash

```
cd /opt/tomcat9/bin
sudo ./startup.sh
```

Access:

```
http://<EC2 Public-IP>:8080/sample/
```

==========

✅ 5. Python Flask Hosting (Cloud Infotech Flask Welcome Page)

Step 1: Install Python and Flask:

bash

```
sudo yum install python3 -y
pip3 install flask
```

Step 2: Create Flask App:

bash

```
mkdir ~/flaskapp
cd ~/flaskapp

cat <<EOF > app.py
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello():
    return "<h1>Welcome to Cloud Infotech Flask Web
App</h1><p>Contact: 8688253560</p>"

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
EOF
```

Run Flask:

==========

bash

```
python3 app.py
```

Access:
`http://<EC2 Public-IP>:5000/`

✅ 6. Jenkins Setup (For Future CI/CD Labs)

Step 1: Java:

bash

```
sudo yum install java-11-amazon-corretto -y
```

Step 2: Jenkins Install:

bash

```
sudo wget -O /etc/yum.repos.d/jenkins.repo
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import
https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo yum install jenkins -y
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

Unlock Jenkins:

bash

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

==========

Access Jenkins:
`http://<EC2 Public-IP>:8080`

✅ ✅ Final Summary:

Platform	URL
NGINX Static (Cloud Infotech)	`http://<IP>/`
Apache PHP	`http://<IP>/index.php`
WordPress	`http://<IP>/wordpress/`
Tomcat Java	`http://<IP>:8080/sample/`
Flask Python	`http://<IP>:5000/`
Jenkins	`http://<IP>:8080/`