
Initial Setup (Fresh System)

1. Update the system

sudo apt update -y sudo apt upgrade -y

2. Install Git

sudo apt install git -y git --version

3. Install Java (required for Jenkins and Maven)

sudo apt install openjdk-11-jdk -y java -version

4. Install Maven

sudo apt install maven -y mvn -version

5. Install Ansible

sudo apt install ansible -y ansible --version

6. Install Docker

sudo apt install apt-transport-https ca-certificates curl software-properties-common -y curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg echo "deb [arch=\$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null sudo apt update sudo apt install docker-ce docker-ce-cli containerd.io -y sudo systemctl enable docker sudo systemctl start docker docker --version

7. Install Jenkins

wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add - sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list' sudo apt update sudo apt install jenkins -y sudo systemctl enable jenkins sudo systemctl start jenkins sudo systemctl status jenkins Access Jenkins at: http\://:8080 Unlock using /var/lib/jenkins/secrets/initialAdminPassword

---

Set 4 Solutions

Q1: Ansible Playbook – Install Nginx

```
---
- name: Install and start Nginx
  hosts: all
  become: yes
  tasks:
    - name: Install Nginx
      apt:
        name: nginx
        state: present
        update_cache: yes

    - name: Ensure Nginx is running
      service:
        name: nginx
        state: started
        enabled: yes
```

Run with: ansible-playbook -i , nginx.yml Check: systemctl status nginx

Q2: Docker – Run container and show logs

```
docker run -d --name myapp_container <image_name>
docker logs -f myapp_container
```

Q3: Manual Maven build in Jenkins

1. Jenkins → New Item → Freestyle Project
2. Configure Git repository
3. Build → Invoke top-level Maven targets, Goals: clean package
4. Build Now → Console Output shows: [INFO] BUILD SUCCESS

Set 5 Solutions

Q1: Ansible Playbook – Install Apache2

```
---
- name: Install and start Apache2
  hosts: all
  become: yes
  tasks:
    - name: Install Apache2
      apt:
        name: apache2
        state: present
        update_cache: yes

    - name: Ensure Apache2 is running
      service:
        name: apache2
        state: started
        enabled: yes
```

Accessible: http\://

Q2: Dockerfile, Build, and Push

```
FROM python:3.11-slim
WORKDIR /app
COPY . /app
RUN pip install flask
CMD ["python", "app.py"]
```

```
docker build -t <dockerhub-username>/myapp:latest .
docker login
docker push <dockerhub-username>/myapp:latest
```

Q3: Git Version Control

```
git init
echo "Hello World" > file.txt
git add file.txt
git commit -m "Initial commit"
```

```
echo "Version 2" >> file.txt
git add file.txt
git commit -m "Second commit"
git remote add origin https://github.com/<username>/<repo>.git
git push -u origin main
```

Set 6 Solutions

Q1: Ansible Playbook – Install HAProxy

```
---
- name: Install and start HAProxy
  hosts: all
  become: yes
  tasks:
    - name: Update package repository
      apt:
        update_cache: yes

    - name: Install HAProxy
      apt:
        name: haproxy
        state: present

    - name: Ensure HAProxy is running
      service:
        name: haproxy
        state: started
        enabled: yes
```

Q2: Docker – Python App

Dockerfile:

```
FROM python:3.11-slim
WORKDIR /app
COPY . /app
CMD ["python", "app.py"]
```

app.py:

```
print("Hello from Docker and Python!")
```

```
docker build -t python-hello .
docker run --rm python-hello
```

Output: Hello from Docker and Python!

Q3: Git – Push index.html

```
git init
echo "<h1>Welcome</h1>" > index.html
git add index.html
git commit -m "Initial commit"
git remote add origin https://github.com/<username>/<repo>.git
git push -u origin main
```

GitHub shows the file and commit message.