DevOps Setup & Quick Commands

Handy guide to install & set up Jenkins, Maven, and a custom Docker Ubuntu image.

Jenkins – Automation Server

Install Jenkins on Ubuntu (requires Java 17+):

1. Update packages:

```
sudo apt update
```

2. Install Java:

```
sudo apt install openjdk-17-jdk -y
```

3. Add Jenkins GPG key & repository:

```
curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo
tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null

echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]
  \ https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list  >  /dev/null
```

4. Update again to fetch Jenkins:

```
sudo apt update
```

5. Install Jenkins:

```
sudo apt install jenkins -y
```

6. Start Jenkins:

```
sudo systemctl start jenkins
```

7. Enable Jenkins on boot:

```
sudo systemctl enable jenkins
```

Maven (mvnd) – Build Automation Tool

Install Apache Maven Daemon (mvnd):

1. Download mvnd tar.gz to /opt:

```
sudo wget -P /opt https://dlcdn.apache.org/maven/mvnd/2.0.0-rc-
3/maven- mvnd-2.0.0-rc-3-linux-amd64.tar.gz
```

2. Navigate to /opt:

```
cd /opt
```

3. Extract the archive:

```
sudo tar -xzf maven-mvnd-2.0.0-rc-3-linux-amd64.tar.gz
```

4. (Optional) Rename folder:

```
sudo mv maven-mvnd-2.0.0-rc-3-linux-amd64 mvnd
```

5. Add mvnd to PATH temporarily:

```
export PATH=$PATH:/opt/mvnd/bin
```

6. To make it permanent, add to ~/.bashrc:

```
echo 'export PATH=$PATH:/opt/mvnd/bin' >> ~/.bashrc
source ~/.bashrc
```

7. Test mvnd:

```
mvnd --version
```

Docker – Custom Ubuntu Image

Build & run your own Ubuntu Docker container:

1. Pull the latest Ubuntu image:

```
docker pull ubuntu:latest
```

2. Create project folder and go into it:

```
mkdir myubuntu && cd myubuntu
```

3. Create Dockerfile with tools:

```
echo -e "FROM ubuntu:latest\nRUN apt update && apt install -y curl
vim\nCMD [\"/bin/bash\"]" > Dockerfile
```

4. Build Docker image named 'myubuntuimage':

```
docker build -t myubuntuimage .
```

5. Run container interactively, named 'myubuntucontainer':

```
docker run -it --name myubuntucontainer myubuntuimage
```

6. (Inside container): do work, then type 'exit' to stop.

7. Restart the stopped container:

```
docker start myubuntucontainer
```

8. Attach to container or open a new shell:

```
docker exec -it myubuntucontainer /bin/bash
```

9. List all images & containers:

```
docker images && docker ps -a
```

10. Cleanup: stop & remove container & image:

```
docker stop myubuntucontainer && docker rm myubuntucontainer && docker rmi myubuntuimage
```

PRESENTED BY

NAVEEN PRASANTH P