

DEVOPS LABS – STEP-BY-STEP (EASY VERSION)

LAB 1 – Git Branching, Merging, Pull Request

1. Open browser → go to github.com

2. Login → open your repository

3. Click “Code” → copy URL

4. Open Terminal → type:

```
git clone
```

```
cd
```

5. Create branch:

```
git checkout -b new-branch
```

6. Edit file → save

7. Run:

```
git add .
```

```
git commit -m "update"
```

```
git push
```

8. Go to GitHub → click “Compare & pull request”

9. Click “Create pull request” → “Merge”

LAB 2 – Jenkins CI/CD

1. Open browser → type: <http://localhost:8080>

2. Click “New Item”

3. Enter name → select “Pipeline”

4. Scroll to Pipeline section

5. Select “Pipeline script”

6. Paste simple pipeline

7. Click Save

8. Click “Build Now”

9. View “Console Output”

LAB 3 – GitHub Actions

1. Go to GitHub repo
2. Click “Actions”
3. Click “New workflow”
4. Select “Simple CI”
5. Click “Set up workflow”
6. Commit → GitHub Actions runs

LAB 4 – Ansible Playbook

1. Create folder → open terminal
2. Create playbook.yml file
3. Add simple Nginx install playbook
4. Create hosts file
5. Run:

```
ansible-playbook -i hosts playbook.yml
```

LAB 5 – Terraform EC2

1. Create folder → make main.tf
2. Add AWS provider + EC2 instance code
3. Run:
`terraform init`
`terraform apply`
4. Check EC2 in AWS Console

LAB 6 – Docker + Compose

1. Create Dockerfile → add FROM nginx
2. Build:
`docker build -t myapp .`
3. Run:
`docker run -p 8080:80 myapp`
4. Create docker-compose.yml

5. Run:

```
docker compose up
```

LAB 7 – Kubernetes (Minikube)

1. Start cluster:

```
minikube start
```

2. Deploy app:

```
kubectl create deployment web --image=nginx
```

3. Expose:

```
kubectl expose deployment web --type=NodePort --port=80
```

4. Open service:

```
minikube service web
```

LAB 8 – Helm

1. Add repo:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

2. Install:

```
helm install myapp bitnami/nginx
```

3. Check:

```
helm list
```

4. Uninstall:

```
helm uninstall myapp
```

LAB 9 – Prometheus + Grafana

1. Add repo:

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts
```

2. Install stack:

```
helm install monitor prometheus-community/kube-prometheus-stack
```

3. Open Grafana:

```
kubectl port-forward svc/monitor-grafana 3000:80
```

4. Open browser:

<http://localhost:3000>

LAB 10 – AWS CodePipeline

1. Open AWS Console → search “CodePipeline”
2. Click “Create pipeline”
3. Give name → Next
4. Select GitHub as source → Connect repo
5. Choose CodeBuild for Build stage
6. Select your deploy target (S3/ECS/Beanstalk)
7. Create pipeline → it runs automatically