

Docker (Ubuntu):

- Install:

```
sudo apt update
sudo apt install -y ca-certificates curl gnupg lsb-release
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.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 -y docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

- Common:

```
docker build -t shamchaudhari/devops-sample:1.0 .
docker run -p 3000:3000 shamchaudhari/devops-sample:1.0
docker ps, docker logs <container>, docker exec -it <id> /bin/sh
```

Kubernetes (kubectl/minikube):

- kubectl apply -f k8s/deployment.yaml
- kubectl get pods,svc,ingress
- kubectl set image deployment/devops-sample-deployment devops-sample=shamchaudhari/devops-sample:1.0:<tag>

Ansible:

- ansible-playbook -i ansible/inventory.ini ansible/playbook.yml
- Install collection: ansible-galaxy collection install community.docker

Terraform:

- terraform init && terraform plan -var "key\_name=<key>" && terraform apply -var "key\_name=<key>"

Jenkins:

- Create pipeline job with Jenkinsfile
- Add Docker Hub creds and kubeconfig as credentials

Helpful tips:

- Replace placeholders (your\_server\_ip, key\_name, Docker Hub creds)
- Use CI to tag images with build numbers
- Use remote Terraform state (S3 + DynamoDB) for teams