End-to-End CI/CD Pipeline Setup with Jenkins, Ansible, Docker, and AWS EKS

# 1. Provision EC2

Launch 1 EC2 instance → Ubuntu 22.04 LTS, instance type t3.medium (2 vCPU, 4 GB RAM).  
Security group: open ports 22 (SSH), 8080 (Jenkins), 80 (HTTP), 443 (HTTPS).

# 2. Update Packages

sudo apt update -y && sudo apt upgrade -y

# 3. Install Dependencies

Java (Jenkins requirement):  
sudo apt install openjdk-17-jre -y  
java -version  
  
Git:  
sudo apt install git -y  
git --version  
  
Docker:  
sudo apt install docker.io -y  
sudo systemctl enable docker  
sudo systemctl start docker  
sudo usermod -aG docker $USER  
newgrp docker  
docker --version  
  
kubectl:  
curl -LO "https://dl.k8s.io/release/v1.31.1/bin/linux/amd64/kubectl"  
chmod +x kubectl  
sudo mv kubectl /usr/local/bin/  
kubectl version --client  
  
eksctl:  
curl --silent --location "https://github.com/eksctl-io/eksctl/releases/latest/download/eksctl\_$(uname -s)\_amd64.tar.gz" -o eksctl.tar.gz  
tar -xzf eksctl.tar.gz -C /tmp  
sudo mv /tmp/eksctl /usr/local/bin  
eksctl version  
  
Ansible:  
sudo apt install ansible -y  
ansible --version

# 4. Install Jenkins

curl -fsSL https://pkg.jenkins.io/debian/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 binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null  
sudo apt-get update  
sudo apt-get install jenkins -y  
sudo systemctl enable jenkins  
sudo systemctl start jenkins  
sudo systemctl status jenkins  
  
Access Jenkins: http://<EC2-PUBLIC-IP>:8080

# 5. Jenkins User Setup

sudo su - jenkins  
ssh-keygen -t rsa -b 4096 -C "jenkins@apache"

(press enter 3 times)  
cat ~/.ssh/id\_rsa.pub  
  
Add this public key to GitHub → Repo → Settings → Deploy Keys → Add Key (Write Access).  
here type exit  
Grant Jenkins sudo rights:  
sudo visudo → add:  
jenkins ALL=(ALL:ALL) NOPASSWD: ALL

# 6. Ansible Setup

Create apache-setup.yml:

vi apache-setup.yml  
- hosts: all  
 become: yes  
 tasks:  
 - name: Install Apache  
 apt:  
 name: apache2  
 state: present  
 update\_cache: yes  
 - name: Start and Enable Apache  
 service:  
 name: apache2  
 state: started  
 enabled: yes  
  
Create hosts.ini:

vi hosts.ini  
[web]  
localhost ansible\_connection=local  
  
Run playbook:  
ansible-playbook -i hosts.ini apache-setup.yml

# 7. Docker Build & Push

git clone https://github.com/akshu20791/apachewebsite.git  
cd apachewebsite  
  
docker build -t apache-website:v1 .  
docker run -d -p 8081:80 apache-website:v1  
curl http://localhost:8081  
  
docker login -u sai2080  
docker tag apache-website:v1 sai2080/apache-website:v1  
docker push sai2080/apache-website:v1

# 8. AWS CLI & EKS Setup

Install AWS CLI:  
sudo apt install unzip curl -y  
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86\_64.zip" -o "awscliv2.zip"  
unzip awscliv2.zip  
sudo ./aws/install  
aws --version  
  
Configure AWS:  
aws configure  
access key:

Secret key:

Region:

Format: json  
Create EKS Cluster:  
eksctl create cluster --name Kumar-eks --region us-east-1 --nodes 2 --node-type t3.medium

# 9. Kubernetes Deployment

vi deployment.yml

apiVersion: apps/v1  
kind: Deployment  
metadata:  
 name: apache-website  
spec:  
 replicas: 2  
 selector:  
 matchLabels:  
 app: apache-website  
 template:  
 metadata:  
 labels:  
 app: apache-website  
 spec:  
 containers:  
 - name: apache-website  
 image: <your-dockerhub-username>/apache-website:v1  
 ports:  
 - containerPort: 80  
---  
apiVersion: v1  
kind: Service  
metadata:  
 name: apache-service  
spec:  
 selector:  
 app: apache-website  
 ports:  
 - protocol: TCP  
 port: 80  
 targetPort: 80  
 type: LoadBalancer  
  
kubectl apply -f deployment.yml  
kubectl get pods  
kubectl get svc  
  
Copy EXTERNAL-IP and open in browser.

# 10. Jenkins Integration with EKS

sudo mkdir -p /var/lib/jenkins/.kube  
aws eks update-kubeconfig --name sai-eks --region us-east-1 --kubeconfig /var/lib/jenkins/.kube/config  
sudo chown -R jenkins:jenkins /var/lib/jenkins/.kube  
sudo chmod 600 /var/lib/jenkins/.kube/config  
sudo su - jenkins  
kubectl get nodes --kubeconfig=/var/lib/jenkins/.kube/config

# 11. Jenkins Pipeline

Jenkinsfile:  
pipeline {  
 agent any  
 stages {  
 stage('Checkout') {  
 steps {  
 git branch: 'master', url: 'https://github.com/sai-2080/apachewebsite.git'  
 }  
 }  
 stage('Ansible Install Apache') {  
 steps {  
 sh 'ansible-playbook -i hosts.ini apache-setup.yml'  
 }  
 }  
 stage('Docker Build & Push') {  
 steps {  
 withCredentials([usernamePassword(credentialsId: 'dockerhub', usernameVariable: 'DOCKER\_USER', passwordVariable: 'DOCKER\_PASS')]) {  
 sh """  
 echo $DOCKER\_PASS | docker login -u $DOCKER\_USER --password-stdin  
 docker build -t apache-website:v1 .  
 docker tag apache-website:v1 $DOCKER\_USER/apache-website:v1  
 docker push $DOCKER\_USER/apache-website:v1  
 """  
 }  
 }  
 }  
 stage('Deploy to K8s') {  
 steps {  
 sh 'kubectl --kubeconfig=/var/lib/jenkins/.kube/config apply -f deployment.yml'  
 }  
 }  
 }  
}