**EC2 instance to build and deploy an application using docker**

**Phase 1: AWS EC2 Instance and Jenkins Installation**

**Step 1: Launch an EC2 Instance and install Jenkins**

**To install Jenkins first install java**

**sudo apt update**

**sudo apt install fontconfig openjdk-21-jre**

**java -version**

**To install Jenkins**

**sudo wget -O /etc/apt/keyrings/jenkins-keyring.asc \**

**https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key**

**echo "deb [signed-by=/etc/apt/keyrings/jenkins-keyring.asc]" \**

**https://pkg.jenkins.io/debian-stable binary/ | sudo tee \**

**/etc/apt/sources.list.d/jenkins.list > /dev/null**

**sudo apt update**

**sudo apt install Jenkins**

**Phase 2: Install Docker and Configure Permissions**

**sudo apt install docker.io -y**

**sudo systemctl start docker**

**sudo systemctl enable docker**

**sudo usermod -aG docker Jenkins**

**sudo usermod -aG docker ubuntu**

**sudo systemctl restart Jenkins**

**Phase 3: Jenkins Initial Setup and Pipeline Code**

**Take password from admin**

**sudo cat /var/lib/jenkins/secrets/initialAdminPassword**

**Step 4: Create a Sample Jenkinsfile**

**pipeline {**

**agent any**

**environment {**

**// Define the name and tag for your Docker image**

**DOCKER\_IMAGE = "my-app-image:${env.BUILD\_ID}"**

**CONTAINER\_NAME = "my-app-container"**

**}**

**stages {**

**// 1. Checkout Code**

**stage('Checkout') {**

**steps {**

**echo 'Cloning the application repository...'**

**// You would use the 'git' step here for a real repo**

**// Example: git url: 'https://github.com/your-username/your-app.git', branch: 'main'**

**}**

**}**

**// 2. Build Docker Image**

**stage('Build') {**

**steps {**

**echo 'Building Docker image...'**

**// This command assumes a Dockerfile exists in the root of the repo**

**sh "docker build -t ${DOCKER\_IMAGE} ."**

**}**

**}**

**// 3. Test/Verify (using the newly built image)**

**stage('Test') {**

**steps {**

**echo 'Running tests...'**

**// Placeholder for running unit/integration tests inside a container**

**sh 'echo "Tests passed successfully!"'**

**// In a real scenario, this would involve running a container with specific test commands.**

**}**

**}**

**// 4. Deploy Application**

**stage('Deploy') {**

**steps {**

**echo 'Deploying application by stopping old and starting new container...'**

**// Stop and remove the old container to ensure a clean deployment**

**sh "docker stop ${CONTAINER\_NAME} || true"**

**sh "docker rm ${CONTAINER\_NAME} || true"**

**// Run the new container, exposing port 80 on the host (EC2)**

**// You must ensure this port is open in your AWS Security Group.**

**sh "docker run -d --name ${CONTAINER\_NAME} -p 80:8080 ${DOCKER\_IMAGE}"**

**}**

**}**

**}**

**post {**

**always {**

**echo "Pipeline finished with status: ${currentBuild.result}"**

**}**

**}**

**}**

**Step 5: Create the Pipeline Job in Jenkins**

**Step 6 : Trigger the Jenkinstask**