**Step-by-Step: Mainframe DevOps Local Lab**

**PHASE 1 – Local Git/Jenkins Setup**

**Step 1 – Install Jenkins**

1. **Download Jenkins (LTS version):  
   https://www.jenkins.io/download/**
2. **Run it:**

**bash**

**CopyEdit**

**java -jar jenkins.war**

1. **Open in browser:**

**arduino**

**CopyEdit**

**http://localhost:8080**

1. **Follow the on-screen setup to unlock Jenkins.**

**Step 2 – Create Git Repository Folder Structure  
On your local machine:**

**bash**

**CopyEdit**

**mkdir MainframeDevOpsLab**

**cd MainframeDevOpsLab**

**mkdir -p src/cobol src/jcl env/dev env/qa env/prod scripts**

**git init**

**Folder meaning:**

* **src/cobol → COBOL source programs.**
* **src/jcl → JCL job files.**
* **env/dev, env/qa, env/prod → simulate datasets for each stage.**
* **scripts → mock build/deploy scripts.**

**Step 3 – Add Sample COBOL & JCL  
src/cobol/HELLO.cbl**

**cobol**

**CopyEdit**

**IDENTIFICATION DIVISION.**

**PROGRAM-ID. HELLO.**

**PROCEDURE DIVISION.**

**DISPLAY "HELLO FROM MAINFRAME DEVOPS LAB".**

**STOP RUN.**

**src/jcl/HELLO.jcl**

**jcl**

**CopyEdit**

**//HELLOJOB JOB (123),'HELLO',CLASS=A,MSGCLASS=X**

**//STEP1 EXEC PGM=HELLO**

**//SYSOUT DD SYSOUT=\***

**Step 4 – First Git Commit**

**bash**

**CopyEdit**

**git add .**

**git commit -m "CCID12345: Initial COBOL and JCL for HELLO program"**

**PHASE 2 – Simulate Mainframe Build/Deploy**

**Step 5 – Create Mock Build Script  
scripts/build.sh**

**bash**

**CopyEdit**

**#!/bin/bash**

**echo "=== Building COBOL Programs ==="**

**cp src/cobol/\* env/dev/**

**echo "Compiling COBOL program..."**

**sleep 1**

**echo "Link-edit complete"**

**Make it executable:**

**bash**

**CopyEdit**

**chmod +x scripts/build.sh**

**Step 6 – Create Mock Promotion Script  
scripts/promote.sh**

**bash**

**CopyEdit**

**#!/bin/bash**

**from\_env=$1**

**to\_env=$2**

**echo "Promoting from $from\_env to $to\_env..."**

**cp env/$from\_env/\* env/$to\_env/**

**echo "Promotion complete from $from\_env to $to\_env"**

**Make it executable:**

**bash**

**CopyEdit**

**chmod +x scripts/promote.sh**

**PHASE 3 – Jenkins Pipeline**

**Step 7 – Create Jenkinsfile  
In your Git repo root:**

**groovy**

**CopyEdit**

**pipeline {**

**agent any**

**stages {**

**stage('Pre-Move Validation') {**

**steps {**

**echo 'Validating syntax...'**

**sh 'ls src/cobol && ls src/jcl'**

**}**

**}**

**stage('Build to DEV') {**

**steps {**

**sh './scripts/build.sh'**

**}**

**}**

**stage('Approval for QA') {**

**steps {**

**input message: 'Approve promotion to QA?'**

**}**

**}**

**stage('Promote to QA') {**

**steps {**

**sh './scripts/promote.sh dev qa'**

**}**

**}**

**stage('Approval for PROD') {**

**steps {**

**input message: 'Approve promotion to PROD?'**

**}**

**}**

**stage('Promote to PROD') {**

**steps {**

**sh './scripts/promote.sh qa prod'**

**}**

**}**

**stage('Post-Move Validation') {**

**steps {**

**echo 'Running post-deployment checks...'**

**sh 'ls env/prod'**

**}**

**}**

**}**

**}**

**Step 8 – Run in Jenkins**

1. **Create New Item → Pipeline in Jenkins.**
2. **Link it to your local Git repo or paste the Jenkinsfile directly.**
3. **Run the job → approve when prompted → watch simulated build/promotion.**

**PHASE 4 – Real API Integration (Optional)**

**Step 9 – Install Zowe CLI**

**bash**

**CopyEdit**

**npm install -g @zowe/cli**

**Configure:**

**bash**

**CopyEdit**

**zowe profiles create zosmf-profile devProfile --host myzos.host --port 443 --user myid --password mypass --reject-unauthorized false**

**Test:**

**bash**

**CopyEdit**

**zowe jobs submit file src/jcl/HELLO.jcl --rfj**

**PHASE 5 – Management Demo**

**Step 10 – Prepare Demo**

* **Take screenshots of:**
  + **Git commit with CCID in message.**
  + **Jenkins pipeline stages.**
  + **Folder structure with DEV/QA/PROD.**
* **Create 2 diagrams:**
  + **Current State → Endevor only.**
  + **Proposed → Git + Jenkins + API → z/OS.**
* **Add bullet points on benefits:**
  + **Automated builds & promotions.**
  + **Same governance, less manual work.**
  + **Cloud + mainframe release alignment.**

**If you want, I can now package this into a downloadable ZIP with:**

* **Pre-filled folder structure.**
* **Sample COBOL/JCL.**
* **Jenkinsfile.**
* **Scripts ready to run.**

**Then you can just unzip → run → demo without typing all code manually.**