

Got you 👍 — here are your **FINAL WORKING pipelines** + the **brief fixes** we applied.

Simple, clean, and exactly matching your setup.

☑ SCRIPTED PIPELINE (Final)

```
node {

    try {

        stage('Git Clone') {
            git branch: 'main',
            url: 'https://github.com/betawins/hiring-app.git'
        }

        stage('Read Version From POM') {
            VERSION = sh(
                script: "mvn help:evaluate -Dexpression=project.version -q -DforceStdout",
                returnStdout: true
            ).trim()
        }

        stage('SonarQube Integration') {
            withSonarQubeEnv('SonarQube') {
                sh 'mvn sonar:sonar'
            }
        }

        stage('Maven Compilation') {
```

```

sh 'mvn clean install -DskipTests'

}

stage('Nexus Artifactory') {
    nexusArtifactUploader(
        nexusVersion: 'nexus3',
        protocol: 'http',
        nexusUrl: '184.73.134.76:8081',
        groupId: 'in.javahome',
        version: VERSION,
        repository: 'Challenge-Scripted',
        credentialsId: 'Nexus',
        artifacts: [[artifactId: 'hiring', classifier: "", file: 'target/hiring.war', type: 'war']]
    )
}

stage('Deploy On Tomcat') {
    withCredentials([usernamePassword(credentialsId: 'tomcat-cred', usernameVariable: 'USER', passwordVariable: 'PASS')]) {
        sh 'curl -u $USER:$PASS --upload-file target/hiring.war "http://13.218.232.37:8082/manager/text/deploy?path=/hiring&update=true"'
    }
}

slackSend channel: '#jenkins-integration',
tokenCredentialId: 'Viqaas-17',
message: "Build SUCCESS ${env.JOB_NAME} #${env.BUILD_NUMBER}"

} catch (err) {

slackSend channel: '#jenkins-integration',

```

```

tokenCredentialId: 'Viqaas-17',
message: "Build FAILED ${env.JOB_NAME} #${env.BUILD_NUMBER}"}

throw err
}
}

```

Interview one-liner:

👉 “Scripted pipeline provides flexible CI/CD with manual control using Groovy.”

DECLARATIVE PIPELINE (Final)

```

pipeline {
    agent any

    stages {
        stage('Git Clone') {
            steps {
                git branch: 'main',
                url: 'https://github.com/betawins/hiring-app.git'
            }
        }

        stage('Read Version From POM') {
            steps {
                script {
                    VERSION = sh(
                        script: "mvn help:evaluate -Dexpression=project.version -q -DforceStdout",
                        returnStdout: true
                    )
                }
            }
        }
    }
}
```

```
    ).trim()

}

}

stage('SonarQube Integration') {

    steps {
        withSonarQubeEnv('SonarQube') {
            sh 'mvn sonar:sonar'

        }
    }
}

stage('Maven Compilation') {

    steps {
        sh 'mvn clean install -DskipTests'

    }
}

stage('Nexus Artifactory') {

    steps {
        nexusArtifactUploader(
            nexusVersion: 'nexus3',
            protocol: 'http',
            nexusUrl: '184.73.134.76:8081',
            groupId: 'in.javahome',
            version: VERSION,
            repository: 'Challenge-Declarative',
            credentialsId: 'Nexus',
            artifacts: [[artifactId: 'hiring', classifier: "", file: 'target/hiring.war', type: 'war']]]
    }
}
```

```

        )
    }

}

stage('Deploy On Tomcat') {
    steps {
        withCredentials([usernamePassword(credentialsId: 'tomcat-cred', usernameVariable: 'USER',
passwordVariable: 'PASS')]) {
            sh 'curl -u $USER:$PASS --upload-file target/hiring.war
"http://13.218.232.37:8082/manager/text/deploy?path=/hiring&update=true"
        }
    }
}

post {
    success {
        slackSend channel: '#jenkins-integration',
        tokenCredentialId: 'Viqas-17',
        message: "Build SUCCESS ${env.JOB_NAME} #${env.BUILD_NUMBER}"
    }
    failure {
        slackSend channel: '#jenkins-integration',
        tokenCredentialId: 'Viqas-17',
        message: "Build FAILED ${env.JOB_NAME} #${env.BUILD_NUMBER}"
    }
}
}

```

Interview one-liner:

👉 “Declarative pipeline provides structured CI/CD flow with built-in post actions.”

🛠️ ERRORS WE HIT — AND HOW WE FIXED THEM

✖ 1. Slack credentialId Error

Problem:

Invalid parameter credentialId

Fix:

Used:

tokenCredentialId

👉 Slack plugin uses bot token.

One-liner:

👉 “Slack Jenkins step authenticates using tokenCredentialId.”

✖ 2. Git Checkout Failed (master branch)

Problem: Jenkins tried origin/master.

Fix:

git branch: 'main'

One-liner:

👉 “Explicit branch prevents default master checkout failure.”

✖ 3. WAR File Not Found

Problem: Pipeline searched wrong filename.

Your pom.xml:

```
<finalName>hirring</finalName>
```

Fix:

```
target/hirring.war
```

One-liner:

👉 “*finalName* in pom.xml controls actual WAR filename.”

✖ 4. Snapshot vs Release Version

Problem: Nexus repo is RELEASE but pipeline used SNAPSHOT.

Fix:

Read version directly from pom.xml:

```
mvn help:evaluate
```

One-liner:

👉 “*Release repos require stable version from pom.xml.*”

✖ 5. Tomcat Credential Missing

Problem:

```
Could not find credentials entry with ID 'tomcat-cred'
```

Fix:

Created Jenkins credential:

Kind: Username with password

ID: tomcat-cred

One-liner:

👉 “*withCredentials* requires existing Jenkins credential ID.”

⌚ FINAL RESULT (What you built)

REAL CI/CD pipeline:

Git → SonarQube → Maven → Nexus (Release) → Tomcat Deploy → Slack Alert