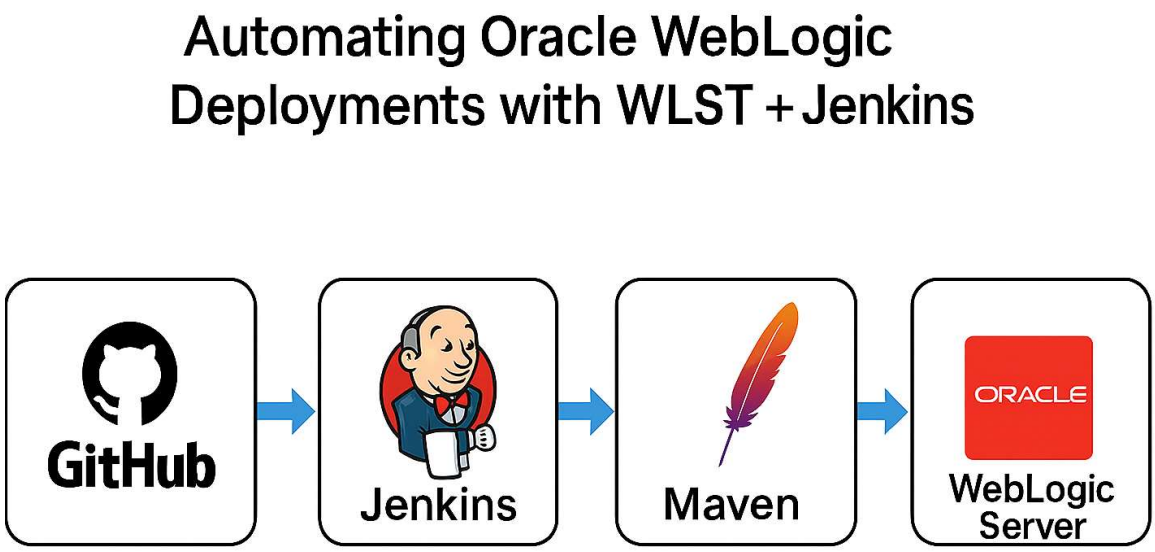
CI/CD for Oracle WebLogic: Automated Deployments with Jenkins, Maven & WLST

🛠 Project Objective:

To design and implement a fully automated deployment solution for **Oracle WebLogic Server** using **WLST (WebLogic Scripting Tool)** integrated with **Jenkins CI/CD pipelines**, enabling faster, error-free, and repeatable application deployments to Managed Servers without manual console intervention.

The automation covers:

- ✅ Fetching latest build artifacts from Maven
- ✅ Seamless undeployment & redeployment
- ✅ Automatic activation of changes
- ✅ Managed Server restart
- ✅ Works for DEV, QA, PROD



🎯 **Goal:** Reduce deployment time, minimize downtime, and improve release reliability by integrating middleware operations into modern DevOps workflows.

📌 Background / Problem Statement

- Manual deployments through the WebLogic console were **time-consuming** and prone to **human errors**.
- Deployment steps varied between environments (DEV, QA, PROD), leading to **inconsistent application behavior**.
- No automated rollback or quick redeployment option was available in case of build issues.
- **Difficult to integrate middleware changes into a CI/CD workflow** because WebLogic deployment scripts were not directly compatible with standard GitHub Actions or Jenkins jobs.
- Source code was version-controlled in **GitHub**, but there was **no direct automation link** between committing new code, building it, and deploying to WebLogic — this meant the deployment process remained a manual, disconnected step.
- Maven was used for building the WAR file, but **build artifacts were not automatically passed to WebLogic** for deployment; instead, they had to be downloaded and uploaded manually via the console.

🛠 Tools & Technologies

- Oracle WebLogic Server 12c or 14c
- WLST (WebLogic Scripting Tool)
- Python (Jython for WLST)
- Jenkins CI/CD
- Maven (for build artifacts)

1️⃣ WLST Python Script

- Connect to Admin Server
- Undeploy old app
- Deploy latest WAR from Maven build
- Activate changes automatically
- Restart Managed Server

📓 Key WLST Script Snippet (Highlight)

Create a WLST script deploy.py on server which called inside jenkins pipeline

```
# deploy.py
import sys
from java.lang import Thread

# Arguments
adminUrl = sys.argv[1]
adminUser = sys.argv[2]
adminPass = sys.argv[3]
warPath   = sys.argv[4]
appName   = sys.argv[5]

# Target managed server
targets = "Maven"  # Change here to your managed server name

# Connect to AdminServer
connect(adminUser, adminPass, adminUrl)
```

```
# Start edit session
edit()
startEdit()

# Undeploy existing app
try:
  print("Undeploying '%s' from target(s): %s" % (appName, targets))
  undeploy(appName, targets=targets, timeout=60000)
  Thread.sleep(5000)
except Exception, e:
  print("Undeploy skipped or failed: %s" % e)

# Deploy new WAR
try:
  print("Deploying WAR: %s" % warPath)
  deploy(appName=appName, path=warPath, targets=targets, upload='true', redeploy='true')
  print("Deployment complete.")
except Exception, e:
  print("Deployment failed: %s" % e)
  cancelEdit('y')
  disconnect()
  sys.exit(1)

# Save & activate changes
save()
activate(block='true')
```

```
print("Changes activated successfully.")


# Restart managed server
try:
  print("Restarting target server: %s" % targets)
  shutdown(targets, 'Server', force='true')
  Thread.sleep(5000)
  start(targets, 'Server')
  Thread.sleep(10000)
  print("Server restarted successfully.")
except Exception, e:
  print("Server restart failed: %s" % e)


disconnect()
print("WLST deployment finished for %s" % targets)
```

Create a New Pipeline Job

1. Go to Jenkins Dashboard → **New Item**
2. Select **Pipeline** → Name it WebLogic-AutoDeploy
3. Click **OK**
4. Poll SCM → */1 * * * * (Every Minute it will chek the Deployment change in GitHub Repository)
5. Definition → Script (Mentined Below) → Save → Apply
6. Click **Build Now** to test
7. If configured correctly, Jenkins will fetch the code, build the WAR, and deploy it automatically to your WebLogic Managed Server

🔄 Jenkins Pipeline Integration

```
pipeline {
  agent any
  environment {
    WLST_PATH = '/data1/Middleware/oracle_common/common/bin/wlst.sh'
    ADMIN_URL = 't3://192.168.80.128:7001'
    ADMIN_USER = 'weblogic'
    ADMIN_PASS = 'weblogic1'
    SCRIPT = '/data1/deploy.py'
  }

  triggers {
    // Poll Git every 1 minute
    pollSCM('*/1 * * * *')
    // OR use webhook (disable above if using webhook)
  }

  stages {
    stage('Checkout from Git') {
      steps {
        git branch: 'main',
            url: 'https://github.com/vinayakshewale99-dev/Test.git'
      }
    }
```

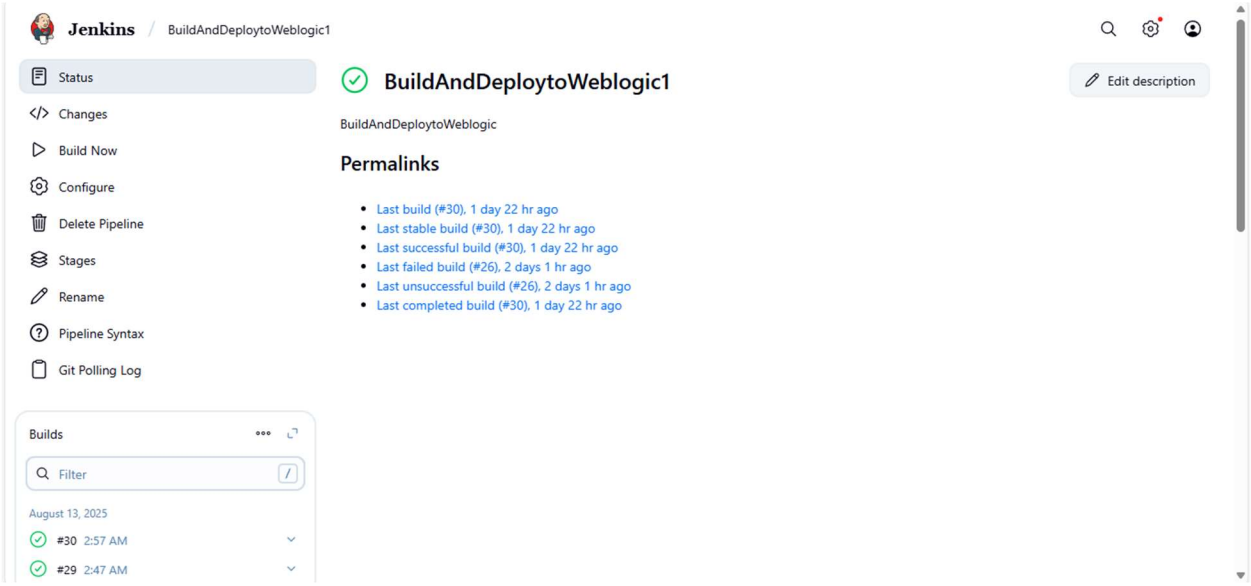
```
    stage('Build WAR with Maven') {
      steps {
        sh 'mvn clean package'
      }
    }

    stage('Deploy WAR to WebLogic') {
      steps {
        script {
          def warFile = "/root/.jenkins/workspace/BuildAndDeploytoWeblogic1/webapp/target/webapp.war" //maven build path
          def appName = "webapp"

          sh """
            ${WLST_PATH} ${SCRIPT} ${ADMIN_URL} ${ADMIN_USER} ${ADMIN_PASS} ${warFile} ${appName}
          """
        }
      }
    }
  }
}
```

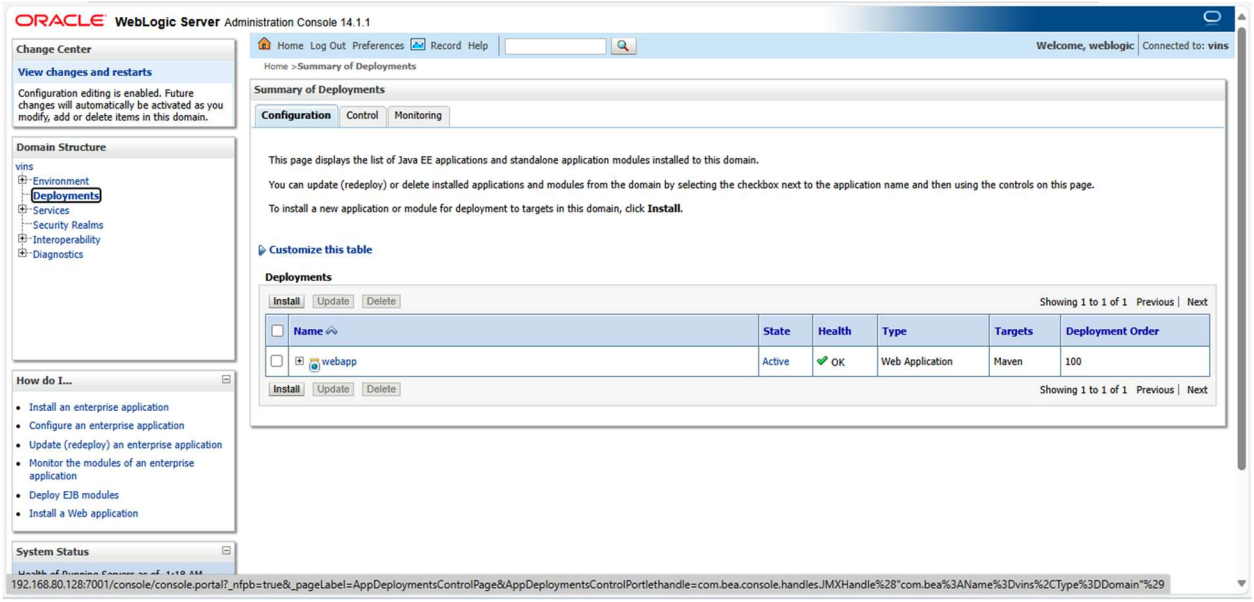
Key Features

- **Fully Automated** deployment (no console login needed).
- **Environment Agnostic** – works for DEV/QA/PROD with config changes.
- **Rollback Ready** – old deployment can be re-triggered if needed.
- **Improved Uptime** – minimal downtime during redeploy.

📈 Results & Benefits

- Deployment time reduced from **20+ mins → 3 mins**
- Consistency across environments
- Instant rollback in case of build issues
- CI/CD-ready for middleware




ORACLE WebLogic Server Administration Console 14.1.1
Change Center
View changes and restarts
Configuration editing is enabled. Future changes will automatically be activated as you modify, add or delete items in this domain.
Domain Structure
vins
Environment
Deployments
Services
Security Realms
Interoperability
Diagnostics
How do I...
Install an enterprise application
Configure an enterprise application
Update (redeploy) an enterprise application
Monitor the modules of an enterprise application
Deploy EJB modules
Install a Web application
System Status
Home Log Out Preferences Record Help
Welcome, weblogic Connected to: vins
Home >Summary of Deployments
Summary of Deployments
Configuration Control Monitoring
This page displays the list of Java EE applications and standalone application modules installed to this domain.
You can update (redeploy) or delete installed applications and modules from the domain by selecting the checkbox next to the application name and then using the controls on this page.
To install a new application or module for deployment to targets in this domain, click Install.
Customize this table
Deployments
Install Update Delete
Showing 1 to 1 of 1 Previous | Next
Name State Health Type Targets Deployment Order
webapp Active OK Web Application Maven 100
Install Update Delete
Showing 1 to 1 of 1 Previous | Next
192.168.80.128:7001/console/console.portal?_nfpb=true&_pageLabel=AppDeploymentsControlPage&AppDeploymentsControlPortlethandle=com.bea.console.handles.JMXHandle%28"com.bea%3AName%3Dvins%2CType%3DDomain"%29