

### Lab3: Continuous Integration and Delivery using Jenkins

Creacion del jenkinsfile multibranch

```
// Jenkinsfile (para multibranch CICD)
```

```
@Library('epamlab3') _
```

```
pipeline {  
    agent any  
    environment {  
        // por defecto, la variable PORT será seteada en 'script' según BRANCH_NAME  
        IMAGE_NAME = "myapp-${env.BRANCH_NAME}-${env.BUILD_NUMBER}"  
    }  
  
    stages {  
        stage('Prepare') {  
            steps {  
                script {  
                    if (env.BRANCH_NAME == 'master') {  
                        env.APP_PORT = '3000'  
                    } else if (env.BRANCH_NAME == 'dev') {  
                        env.APP_PORT = '3001'  
                    } else {  
                        env.APP_PORT = '3001' // default  
                    }  
                    echo "Branch: ${env.BRANCH_NAME} -> PORT=${env.APP_PORT}"  
                }  
            }  
        }  
  
        stage('Checkout') {  
            steps {  
                checkout scm  
            }  
        }  
    }  
}
```

```
        }

    }

stage('Build') {
    steps {
        // Ajusta a tu build: ejemplo Node.js
        sh """
            if [ -f package.json ]; then
                npm ci
                npm run build || true
            fi
        """
    }
}
```

```
stage('Test') {
    steps {
        sh """
            if [ -f package.json ]; then
                npm test --silent || echo "tests may have failed"
            fi
        """
    }
}
```

```
stage('Build Docker Image') {
    steps {
        script {
            sh "docker build -t ${IMAGE_NAME} -f ./Dockerfile ."
        }
    }
}
```

```

        }

stage ('Push and Deploy') {
    steps {
        dockerPipeline(
            image: "${IMAGE_NAME}",
            triggerDeploy: true
        )
    }
}
}
}

```

```

post {
    always{
        echo "Pipeline finalizado para ${env.BRANCH_NAME}"
    }
}
}

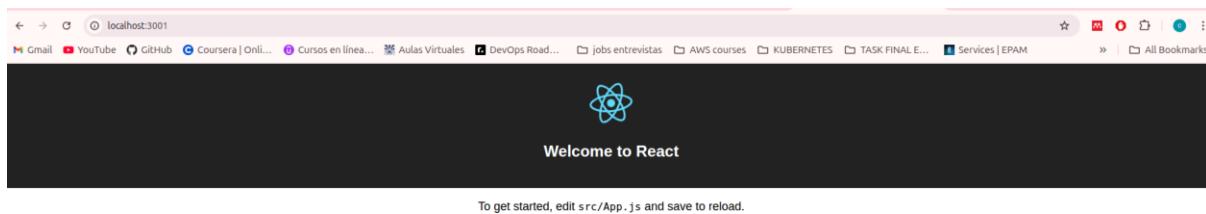
```

The screenshot shows the Jenkins CICD dashboard. On the left, there's a sidebar with various Jenkins management links like Status, Configure, Scan Multibranch Pipeline Now, Scan Multibranch Pipeline Log, Multibranch Pipeline Events, Delete Multibranch Pipeline, Build History, Favorite, Open Blue Ocean, Rename, Config Files, Pipeline Syntax, and Credentials. The main area is titled 'CICD' and shows 'Branches (2)'. It lists two branches: 'dev' and 'master'. Each branch entry includes a status icon (green circle with a checkmark), a weather icon (cloud with sun or rain), the branch name, the last success time ('2 min 34 sec #4'), the last failure time ('6 min 2 sec #3'), and the last duration ('35 sec'). There are also 'More' and 'Star' icons next to each branch entry.

S	W	Name	Last Success	Last Failure	Last Duration
		dev	2 min 34 sec #4	6 min 2 sec #3	35 sec
		master	2 min 34 sec #4	6 min 10 sec #3	55 sec

Escenarios de app web con localhost:3000 main branch y localhost:3001 dev branch

Repositorio de GitHub: <https://github.com/cmlavaut/epamlab3.git>



Configuración del dockerPipeline.groovy para utilizarlo en sharedLib Jenkins

```
def call(Map config) {  
    pipeline {  
        agent any  
  
        stages {  
  
            stage('Login Docker Hub') {  
                steps {  
                    withCredentials([usernamePassword(credentialsId: 'dockerhub-credentials',  
usernameVariable: 'USER', passwordVariable: 'PASS')]) {  
                        sh "echo $PASS | docker login -u $USER --password-stdin"  
                    }  
                }  
            }  
  
            stage('Push Image') {  
                steps {  
                    sh "docker push ${config.imageName}"  
                }  
            }  
        }  
    }  
}
```

```
}

stage('Trigger Deploy Pipeline'){

    when { expression { return config.triggerDeploy } }

    steps {

        script {

            def branch = env.BRANCH_NAME

            if (branch == "master") {

                build job: "Deploy_to_master", parameters: [string(name: 'IMAGE_NAME', value: config.imageName)]

            } else if (branch == "dev") {

                build job: "Deploy_to_dev", parameters: [string(name: 'IMAGE_NAME', value: config.imageName)]

            }

        }

    }

}

}
```