



Proceso de Desarrollo de Software

Andrés Mauricio Martínez Hincapié

Actividad práctica #4: Pipeline Avanzado con Paralelismo y Deploy Simulado

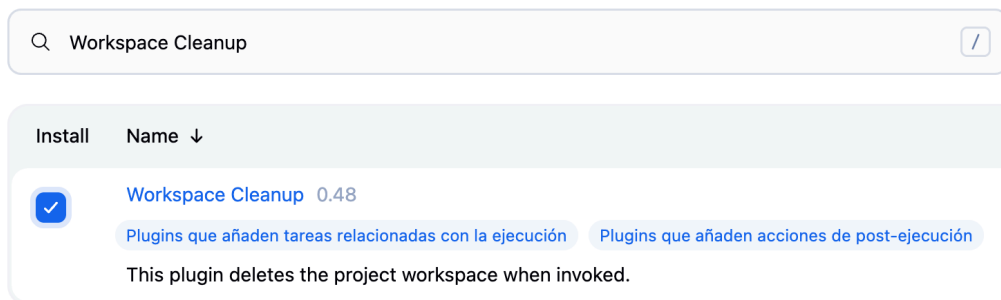
Objetivos:

- Paralelizar pruebas en múltiples entornos.
- Agregar un paso de deploy simulado.
- Configurar notificaciones ante fallos.

Paso 1: Preparación del Entorno

1.1 Instalar Plugins (Si no los tienes)

- Ve a **Jenkins > Manage Jenkins > Plugins > Available:**
 - **HTML Publisher:** Para reportes visuales.
 - **Workspace Cleanup:** Para limpieza post-ejecución.



1.2 Configurar Herramientas

- En **Manage Jenkins > Global Tool Configuration:**
 - Asegúrate de tener Node.js (Node_24) configurado.

Paso 2: Modificar el Jenkinsfile

Reemplaza tu pipeline actual con este código mejorado:

```
pipeline {
```

```
agent any

tools {

    nodejs 'Node_24' // Configurado en Global Tools

}

stages {

    // Etapa 1: Checkout

    stage('Checkout') {

        steps {

            git branch: 'main', url:
'https://github.com/amartinezh/ucp-app-react.git'

        }

    }

    // Etapa 2: Build

    stage('Build') {

        steps {

            sh 'npm install'

            sh 'npm run build'

        }

    }

    // Etapa 3: Pruebas Paralelizadas

    stage('Pruebas en Paralelo') {

        parallel {

            // Pruebas en Chrome

            stage('Pruebas Chrome') {

                steps {
```

```

        script {

            try {

                sh 'npm test -- --browser=chrome
--watchAll=false --ci --reporters=jest-junit'

                junit 'junit-chrome.xml'

            } catch (err) {

                echo "Pruebas en Chrome fallaron: ${err}"

                currentBuild.result = 'UNSTABLE'

            }

        }

    }

}

// Pruebas en Firefox

stage('Pruebas Firefox') {

    steps {

        script {

            try {

                sh 'npm test -- --browser=firefox
--watchAll=false --ci --reporters=jest-junit'

                junit 'junit-firefox.xml'

            } catch (err) {

                echo "Pruebas en Firefox fallaron: ${err}"

                currentBuild.result = 'UNSTABLE'

            }

        }

    }

}

}

```

```
    }  
  }  
  
  // Etapa 4: Deploy Simulado  
  
  stage('Deploy a Producción (Simulado)') {  
    steps {  
      script {  
        // Crear carpeta "prod" y copiar build  
  
        sh 'mkdir -p prod && cp -r build/* prod/'  
  
        echo "¡Deploy simulado exitoso! Archivos copiados a /prod"  
      }  
    }  
  }  
}  
  
post {  
  always {  
    // Publicar reportes HTML (opcional)  
  
    publishHTML target: [  
      allowMissing: true,  
      alwaysLinkToLastBuild: true,  
      keepAll: true,  
      reportDir: 'prod',  
      reportFiles: 'index.html',  
      reportName: 'Demo Deploy'  
    ]  
  }  
}
```

```

// Notificación por email ante fallos

emailExt (

    subject: "Pipeline ${currentBuild.result}: ${env.JOB_NAME}",

    body: """

        <h2>Resultado: ${currentBuild.result}</h2>

        <p><b>URL del Build:</b> <a
href="${env.BUILD_URL}">${env.BUILD_URL}</a></p>

        <p><b>Consola:</b> <a href="${env.BUILD_URL}console">Ver
logs</a></p>

        """,

    to: 'tu-email@example.com',

    mimeType: 'text/html'

)

// Limpiar workspace

cleanWs()

}

}

}

```

Paso 3: Configuración email

1. Permitir acceso en Gmail (aplicaciones menos seguras)

Desde el 2022, Gmail ya no permite usar nombre de usuario y contraseña directamente. Debes usar:

- Una contraseña de aplicación (si usas verificación en 2 pasos)
- O usar un proveedor SMTP externo (menos recomendable para Gmail)

➤ Cómo obtener una contraseña de aplicación:

1. Ve a tu cuenta de Google: <https://myaccount.google.com>
 2. Ve a Seguridad
 3. Activa la verificación en dos pasos
 4. Luego ve a Contraseñas de aplicación
 5. Crea una nueva contraseña para "Correo" y "Otro (Jenkins)"
 6. Copia la contraseña generada (será algo como **abcd efgh ijkl mnop**)
-

2. Instalar plugin de correo en Jenkins (si no lo tienes)

En Jenkins:

1. Ve a Manage Jenkins > Plugin Manager
 2. En la pestaña Available, busca:
 - Email Extension Plugin
 - Mailer Plugin
 3. Instálalos y reinicia Jenkins si es necesario.
-

3. Configurar SMTP en Jenkins

1. Ve a Manage Jenkins > Configure System
2. Busca la sección E-mail Notification
3. Llena los campos así:

SMTP server: smtp.gmail.com
Use SMTP Authentication: ☒
Username: tu_correo@gmail.com
Password: (la contraseña de aplicación que generaste)
Use SSL: ☒

SMTP port: 465

4. En "Reply-To Address" pon también tu correo si quieres recibir respuestas.
5. Haz clic en Test configuration by sending test e-mail (usa un destinatario válido).

Paso 4: Explicación Clave

Paralelización de Pruebas

- Usamos `parallel` para ejecutar pruebas en Chrome y Firefox simultáneamente.
- Cada etapa captura resultados separados (`junit-chrome.xml` y `junit-firefox.xml`).
- Si una suite falla, el pipeline continúa pero marca el build como `UNSTABLE`.

Deploy Simulado

- Se crea una carpeta `prod` y se copia el resultado del build (`build/`) allí.
- El comando `mkdir -p prod && cp -r build/* prod/` es compatible con Linux/macOS.

Notificaciones

- Se envía un email con formato HTML cuando el pipeline termina (éxito o fallo).
- Usamos `cleanWs()` para limpiar el workspace post-ejecución.

Paso 5: Simular un Fallo

Para probar las notificaciones:

1. **Modifica temporalmente un test** para que falle:

```
// En tu suite de pruebas (ej: App.test.js)

test('Test que falla', () => {

  expect(true).toBe(false);

});
```

2. Haz commit y push:

```
git add . && git commit -m "Test fallido" && git push origin main
```

3. Verifica:

- En Jenkins, el pipeline mostrará etapas en rojo/naranja.
- Recibirás un email con el error.

Paso 6: Demo de Resultados

1. Reportes de Pruebas:

- Accede a `http://tujenkins/job/tu-job/lastBuild/testReport/` para ver los resultados de JUnit.

2. Deploy Simulado:

- Revisa la carpeta `prod` en el workspace de Jenkins:

```
docker exec -it <contenedor_jenkins> ls /var/jenkins_home/workspace/tu-job/prod
```

3. Notificaciones:

- Verifica tu bandeja de entrada (incluye enlaces directos al build).
-