

# Lab: Create and use a resource file

In this exercise you will:

- Create a `resources` file.
- Create a custom step that loads the `resources` file.
- Modify the existing Pipeline to use the custom step.

## Task: Create a `resources` file

Copy the contents of `jenkins/build.sh` in the `pipeline-lab` repository to create a `resources/scripts/build.sh` file in the `shared-library` repository:

1. Navigate to the `pipeline-lab` Gitea repository.
2. Select the `jenkins` directory.
3. Select `build.sh` and copy its contents to the clipboard.
4. Navigate to the `shared-library` Gitea repository.
5. Select **New File** to the right of the directory name to create a new file.
6. Name the file `resources/scripts/build.sh`.
7. Paste in the `build.sh` contents from the clipboard.
8. Add a commit message and select **Commit Changes**.

## Task: Create a custom step that loads the `resources` file

Use the Gitea editor to create the custom step:

1. Navigate to the `shared-library` Gitea repository.
2. Select the `vars` directory.
3. Select **New File** to the right of the directory name to create a new file.
4. Name the file `runLinuxScript.groovy`.
5. Paste in the following content:

```
def call(Map config = [:]) {
    def scriptcontents = libraryResource "scripts/${config.name}"
    writeFile file: "${config.name}", text: scriptcontents
    sh """
        chmod a+x ./${config.name}
        ./${config.name}
    """
}
```

6. Add a commit message and select **Commit Changes**.

## Task: Modify the existing Pipeline to use the custom step

1. Navigate to the `pipeline-lab` Gitea repository.
2. Select `Jenkinsfile`.
3. Select the pencil icon in the tool bar to enter edit mode.
4. Replace every line of

```
sh './jenkins/build.sh'
```

in the Jenkinsfile with

```
runLinuxScript(name: "build.sh")
```

5. Commit the changes. The job should start automatically once the commit has completed.
6. Verify that the pipeline ran successfully and waits for input. You can select **Abort** as your input for this exercise.

## Solution

### Jenkinsfile

```
@Library('shared-library') _
pipeline {
    agent none
    stages {
        stage('Fluffy Build') {
            parallel {
                stage('Build Java 8') {
                    agent {
                        node {
                            label 'java8'
                        }
                    }
                    steps {
                        runLinuxScript(name: "build.sh")
                    }
                    post {
                        success {
                            stash(name: 'Java 8', includes: 'target/**')
                        }
                    }
                }
                stage('Build Java 7') {
                    agent {
                        node {
                            label 'java7'
                        }
                    }
                    steps {
                        runLinuxScript(name: "build.sh")
                    }
                    post {
                        success {
                            postBuildSuccess(stashName: "Java 7")
                        }
                    }
                }
            }
        }
        stage('Fluffy Test') {
            parallel {
                stage('Backend Java 8') {
                    agent {
                        node {
                            label 'java8'
                        }
                    }
                    steps {
                        unstash 'Java 8'
                        sh './jenkins/test-backend.sh'
                    }
                    post {
                        always {
                            junit 'target/surefire-reports/**/TEST*.xml'
                        }
                    }
                }
                stage('Frontend Java 8') {
```

```
agent {
  node {
    label 'java8'
  }
}
steps {
  unstash 'Java 8'
  sh './jenkins/test-frontend.sh'
}
post {
  always {
    junit 'target/test-results/**/TEST*.xml'
  }
}
}
stage('Performance Java 8') {
  agent {
    node {
      label 'java8'
    }
  }
  steps {
    unstash 'Java 8'
    sh './jenkins/test-performance.sh'
  }
}
stage('Static Java 8') {
  agent {
    node {
      label 'java8'
    }
  }
  steps {
    unstash 'Java 8'
    sh './jenkins/test-static.sh'
  }
}
stage('Backend Java 7') {
  agent {
    node {
      label 'java7'
    }
  }
  steps {
    unstash 'Java 7'
    sh './jenkins/test-backend.sh'
  }
  post {
    always {
      junit 'target/surefire-reports/**/TEST*.xml'
    }
  }
}
stage('Frontend Java 7') {
  agent {
    node {
      label 'java7'
    }
  }
  steps {
    unstash 'Java 7'
```

```

        sh './jenkins/test-frontend.sh'
    }
    post {
        always {
            junit 'target/test-results/**/TEST*.xml'
        }
    }
}
stage('Performance Java 7') {
    agent {
        node {
            label 'java7'
        }
    }
    steps {
        unstash 'Java 7'
        sh './jenkins/test-performance.sh'
    }
}
stage('Static Java 7') {
    agent {
        node {
            label 'java7'
        }
    }
    steps {
        unstash 'Java 7'
        sh './jenkins/test-static.sh'
    }
}
}
}
stage('Confirm Deploy') {
    when {
        branch 'master'
    }
    steps {
        input(message: 'Okay to Deploy to Staging?', ok: 'Let\'s Do it!')
    }
}
stage('Fluffy Deploy') {
    when {
        branch 'master'
    }
    agent {
        node {
            label 'java7'
        }
    }
    steps {
        unstash 'Java 7'
        sh "./jenkins/deploy.sh ${params.DEPLOY_TO}"
    }
}
}
parameters {
    string(name: 'DEPLOY_TO', defaultValue: 'dev', description: '')
}
}

```