

Parallel Stages

Jenkins has added syntax support for parallel stages. In earlier versions of Declarative Pipeline, the only way to run chunks of Pipeline code in parallel was to use the parallel step inside the steps block for a stage, like this:

/\* .. snip .. \*/

stage('run-parallel-branches') {

steps {

parallel(

a: {

echo "This is branch a"

},

b: {

echo "This is branch b"

}

)

}

}

/\* .. snip .. \*/

While this works, it doesn’t integrate well with the rest of the Declarative Pipeline syntax. For example, to run each parallel branch on a different agent, you need to use a node step, and if you do that, the output of the parallel branch won’t be available for post directives (at a stage or pipeline level). Basically, the old parallel step required you to use Scripted Pipeline within a Declarative Pipeline.

But now with Declarative Pipeline 1.2, Jenkins has introduced a true Declarative syntax for running stages in parallel:

Jenkinsfile

pipeline {

agent none

stages {

stage('Run Tests') {

parallel {

stage('Test On Windows') {

agent {

label "windows"

}

steps {

bat "run-tests.bat"

}

post {

always {

junit "\*\*/TEST-\*.xml"

}

}

}

stage('Test On Linux') {

agent {

label "linux"

}

steps {

sh "run-tests.sh"

}

post {

always {

junit "\*\*/TEST-\*.xml"

}

}

}

}

}

}

}

You can now specify either steps or parallel for a stage, and within parallel, you can specify a list of stage directives to run in parallel, with all the configuration you’re used to for a stage in Declarative Pipeline.

You can also use stage directives, including post, when, agent, and all the others covered in the [Pipeline Syntax reference](https://jenkins.io/doc/book/pipeline/syntax) in your sequential stages, letting you control behaviour for different parts of each parallel branch.

In the example below, we are running builds on both Windows and Linux, but only want to deploy if we’re on the master branch.

pipeline {

agent none

stages {

stage("build and deploy on Windows and Linux") {

parallel {

stage("windows") {

agent {

label "windows"

}

stages {

stage("build") {

steps {

bat "run-build.bat"

}

}

stage("deploy") {

when {

branch "master"

}

steps {

bat "run-deploy.bat"

}

}

}

}

stage("linux") {

agent {

label "linux"

}

stages {

stage("build") {

steps {

sh "./run-build.sh"

}

}

stage("deploy") {

when {

branch "master"

}

steps {

sh "./run-deploy.sh"

}

}

}

}

}

}

}

}