

# Declarative Pipeline Quick Reference

Jenkins Pipeline enables users to build complete continuous delivery (CD) pipelines as part of their application code. The steps to build, test and deliver each application become part of the application itself, stored in a Jenkinsfile. Declarative Pipeline syntax offers a simple, predefined hierarchy to make the creation of pipelines and the associated Jenkinsfiles accessible to users of all experience levels. In its simplest form, a pipeline runs on an agent and contains stages, while each stage contains steps that define specific actions.

```
pipeline {
  agent {
    label ''
  }
  stages {
    stage('Build') {
      steps{
        sh 'mvn install'
      }
    }
  }
}
```

In addition, Declarative Pipeline syntax also provides the ability to control all aspects of the Pipeline execution environment in a straightforward format. For example, building a Java application with Maven in a Docker container that only archives and tests the “Master” branch and times out after six hours.

<https://jenkins.io/doc/book/pipeline/>

```
pipeline {
  agent {
    docker {
      label 'docker-node'
      image 'maven'
      args '-v /tmp:/tmp -p 80:80'
    }
  }
  environment {
    GIT_COMMITTER_NAME = 'jenkins'
  }
  options {
    timeout(6, HOURS)
  }
  stages {
    stage('Build') {
      steps {
        sh 'mvn clean install'
      }
    }
    stage('Archive') {
      when {
        branch '*/master'
      }
      steps {
        archive '**/target/**/*'
        junit '**/target/surefire-reports/*.xml'
      }
    }
  }
  post {
    always {
      deleteDir()
    }
  }
}
```



## DECLARATIVE PIPELINE SYNTAX - REQUIRED

- » **pipeline** - Contains the entire Jenkins Pipeline definition
- » **agent** - Defines the agent used for the entire Pipeline or a stage
  - **label** - Existing Jenkins node label
  - **docker** - Requires Docker-enabled node
    - **image** - Run inside specified Docker image
    - **label** - Existing Jenkins node label
    - **args** - Arguments for Docker container
  - **dockerfile** - Use a local Dockerfile
    - **filename** - Name of local Dockerfile
    - **label** - Existing Jenkins node label
    - **args** - Arguments for Docker container
- » **stages** - Contains Pipeline stages and steps
- » **stage** - A specific named “stage” of the Pipeline
  - **steps** - One or more build steps that define the actions in the stage. Contains one or more of the following:
    - Any build step or build wrapper defined in Pipeline e.g. **sh**, **bat**, **timeout**, **echo**, **archive**, **junit**, etc.
    - **parallel (optional)** - Execute steps in parallel. May not be used with other steps
    - **script (optional)** - Execute Scripted Pipeline block
  - **when (optional)** - Runs stage conditionally
    - **branch** - Stage runs when branch name matches
    - **expression** - Boolean expression
- **agent**, **environment**, **tools** and **post** may also optionally be defined in **stage**

## DECLARATIVE PIPELINE SYNTAX - OPTIONAL

- » **environment** - A sequence of “key = value” pairs to define environment variables
  - **credentials(<id>)** - Bind credentials to variable
- » **options** - Options for the entire Pipeline. For example:
  - **skipDefaultCheckout** - Disable auto checkout SCM
  - **timeout** - Sets timeout for entire Pipeline
  - **buildDiscarder** - Discard old builds
  - **disableConcurrentBuilds** - Disable concurrent Pipeline runs
- » **tools** - Installs pre-defined tools to be available on PATH
- » **triggers** - Triggers for Pipeline based on schedule, polling, etc.
- » **parameters** - Pipeline parameters that are prompted for at run time
- » **post** - Defines actions to be taken when pipeline or stage completes based on outcome. Conditions execute in order:
  - **always** - Run regardless of Pipeline status
  - **changed** - Run if the result of Pipeline has changed from last run
  - **success** - Run if Pipeline is successful
  - **unstable** - Run if Pipeline result is unstable
  - **failure** - Run if the Pipeline has failed

Created by CloudBees