

Types of Projects in Jenkins

- Freestyle
- Scripted Pipelines
- Declarative Pipelines

Pipeline as code

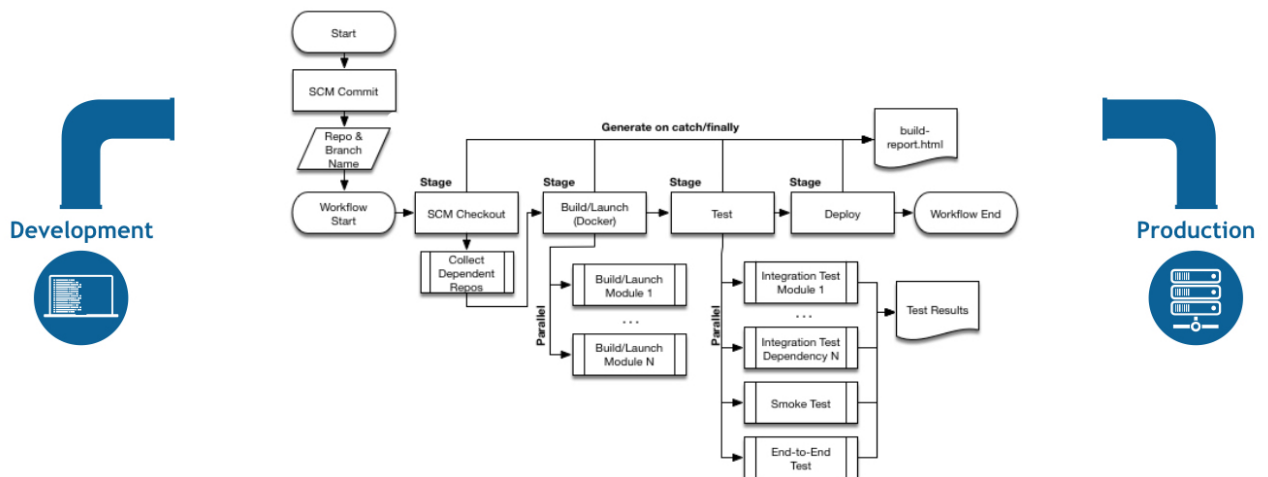
- We express the CI/CD Pipeline as Code and store it in version control system.
- We have history of all of the changes made.
- We can have different pipelines per branch
- [Refer Here](#) for Pipeline concepts

Scripted Pipeline

- This was offered by Jenkins initially as Pipeline as Code offering.
- Scripted Pipelines are in Groovy Language (Java Based Language)
- We can still use this if your CI/CD steps require lot of customizations.
- [Refer Here](#) for scripted Pipeline fundamentals

Declarative Pipeline

- This was offered from learnings from Scripted Pipelines
- This is expressed in Jenkins DSL (Domain Specific Language)
- This approach has easier learning curve for the long time users of Jenkins
- This is straight forward and supports most of the CI/CD Pipelines in almost all technologies
- [Refer Here](#) for declarative Pipeline fundamentals



Approach for writing Pipelines

Scripted Pipelines

- We create a file with name `Jenkinsfile` in some repository
- The syntax will be this

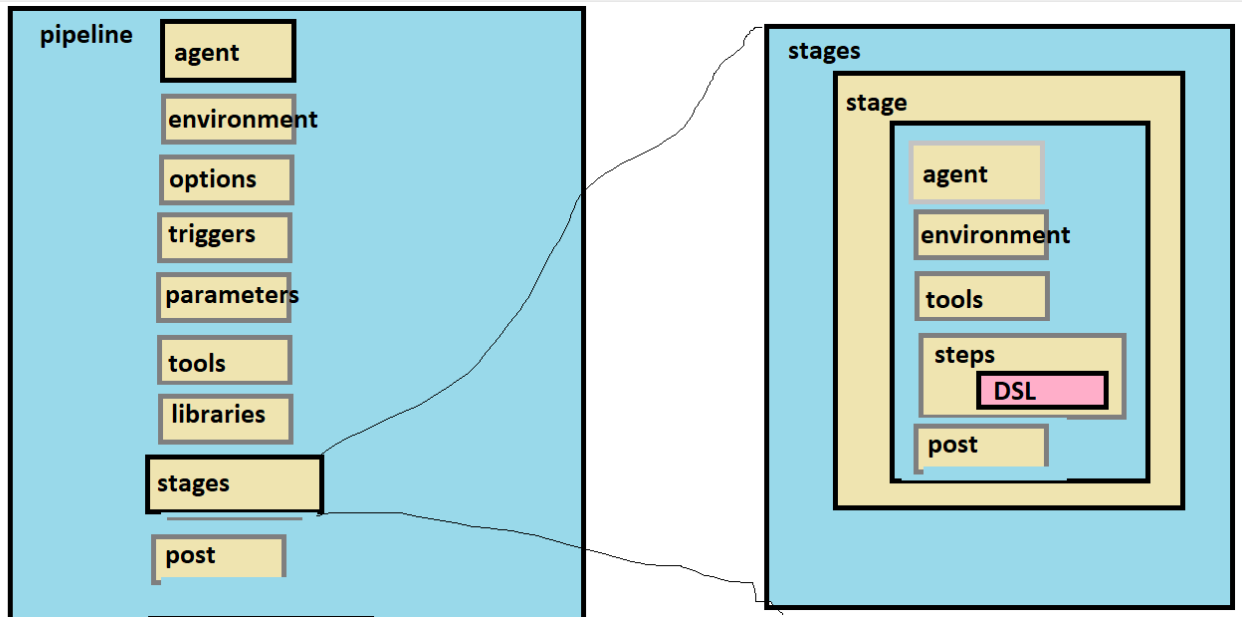
```
node {
  stage('Build') {
    // steps
  }
  stage('Test') {
    // steps
  }
  stage('Deploy') {
    // steps
    // groovy
  }
}
```

Declarative Pipelines

- We create a file with name **Jenkinsfile** in some repository
- syntax

```
pipeline {
  agent any
  stages {
    stage('Build') {
      steps {
        //
      }
    }
    stage('Test') {
      steps {
        //
      }
    }
    stage('Deploy') {
      steps {
        //
      }
    }
  }
}
```

- Syntax Overview [Refer Here](#)



- [Refer Here](#) for Pipeline steps
- Agents: [Refer Here](#)
- Environments [Refer Here](#)
- Triggers [Refer Here](#)

Lets create a hello-declartive-pipeline

- refer classroom video for screenshots
- [Refer Here](#) for the Jenkinsfile

Dashboard > spc-declarative-1 > Configuration

Configure

- General
- Advanced Project Options
- Pipeline**

Pipeline

Definition

Pipeline script from SCM ✓

SCM ?

Git

Repositories ?

Repository URL ?

https://github.com/dummyrepos/spring-petclinic-aug24.git ✓

Credentials ?

- none -

+ Add

Save Apply

Dashboard > spc-declarative-1 > Configuration

Configure

- General
- Advanced Project Options
- Pipeline**

Pipeline

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main ✓

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add

Script Path ?

Jenkinsfile ✓

Lightweight checkout ?

Save Apply

Exercise:

- Create a declarative pipeline to
 - clone any repo in **GIT** stage
 - execute **dotnet restore test.sln** in **BUILD** stage
 - print environmental variables in **FUN** stage