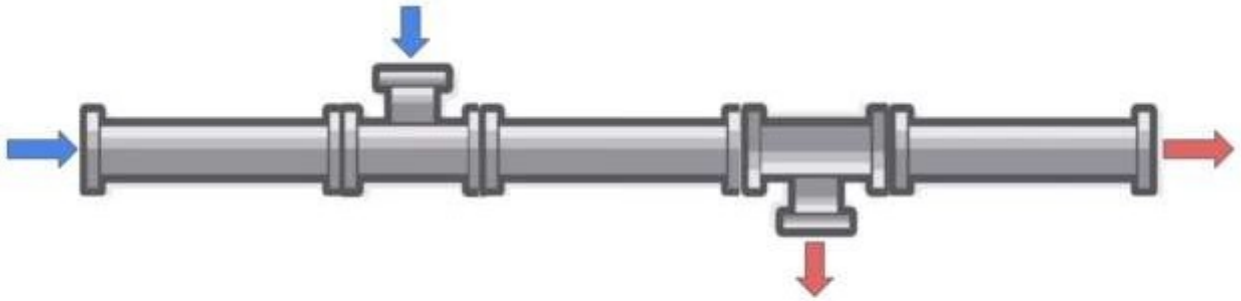
Jenkins Pipeline

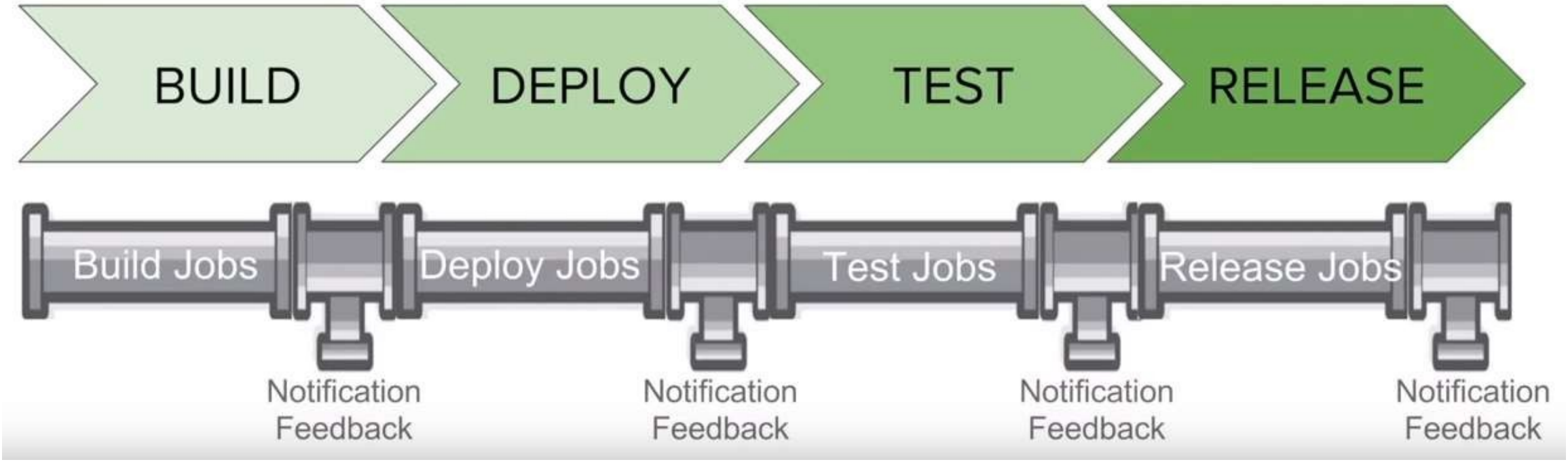
What is Pipeline?

- In Jenkins, a pipeline is a group of events or jobs which are interlinked with one another in a sequence.

Pipe



Pipeline


BUILD
DEPLOY
TEST
RELEASE
Build Jobs
Deploy Jobs
Test Jobs
Release Jobs
Notification Feedback
Notification Feedback
Notification Feedback
Notification Feedback

How to setup BUILD PIPELINE in Jenkins

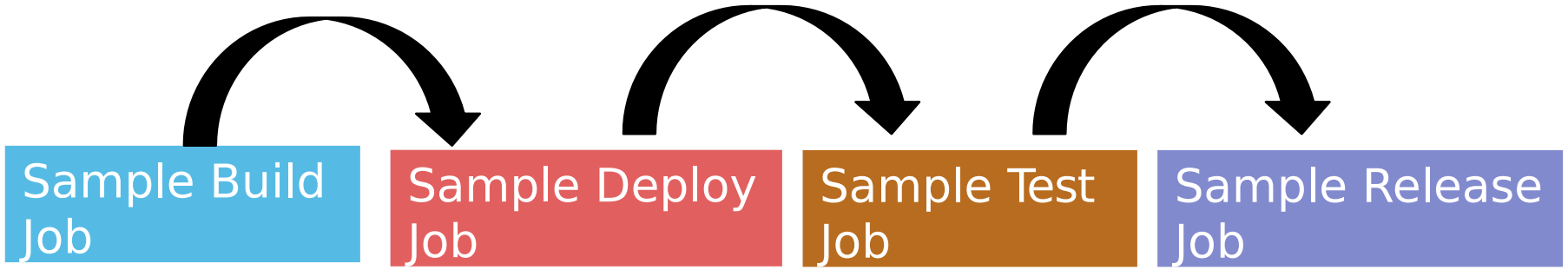
Step1: Chain required jobs in sequence Add upstream/downstream jobs Step2: Install *Build Pipeline Plugin*

Step3: Add

Build Pipeline View Configure the view Step

Step4: Run and Validate

Demo

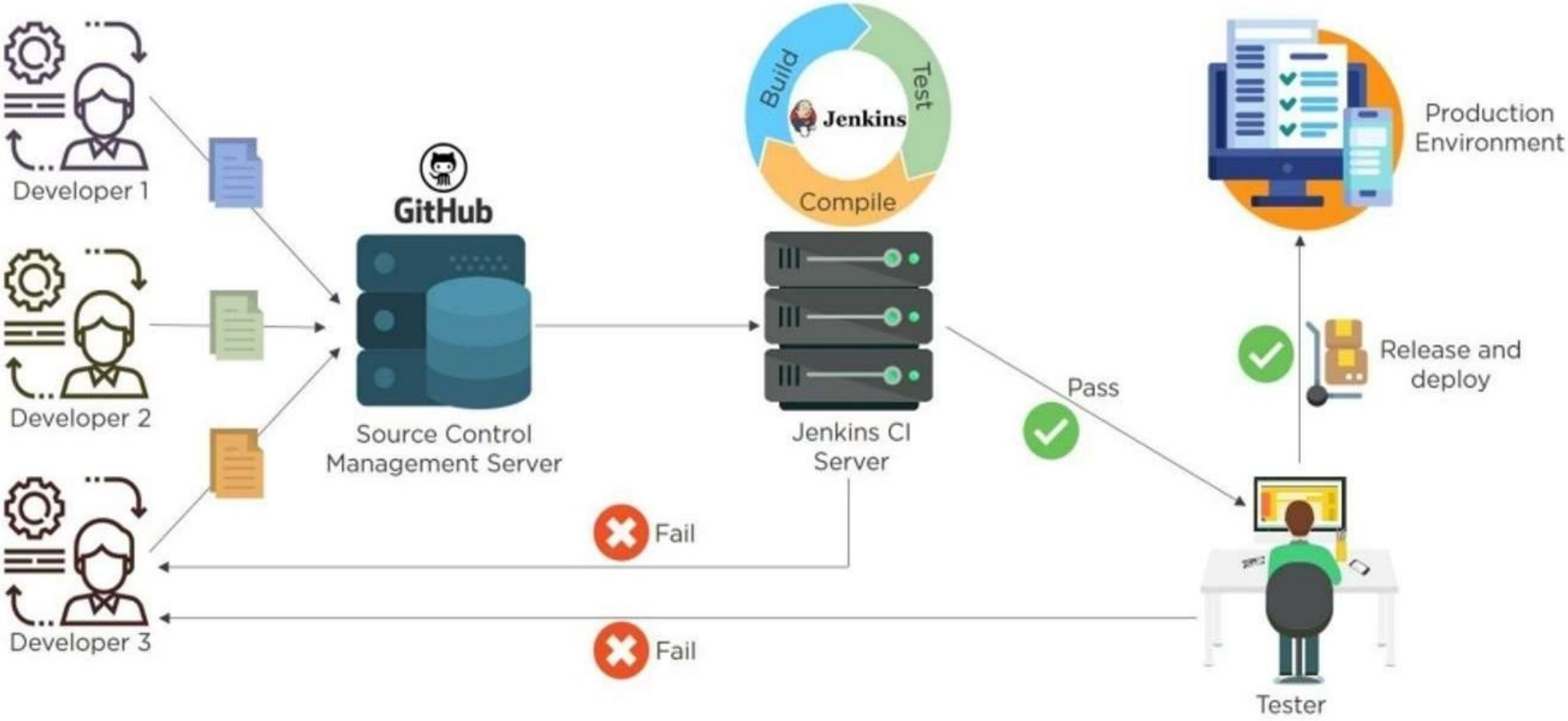

Sample Build Job
Sample Deploy Job
Sample Test Job
Sample Release Job

How to Build Pipelines with Groovy Script

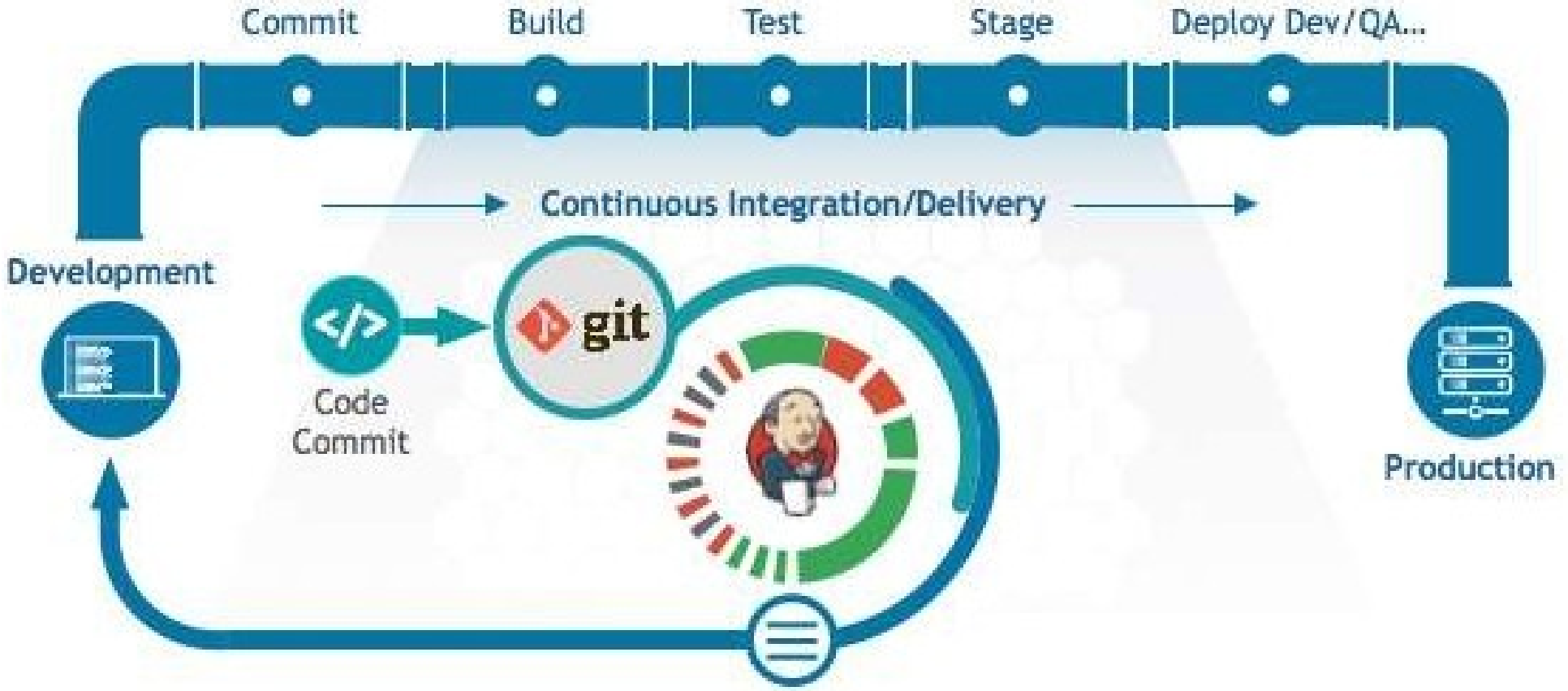
Agend
a

- CI & CD
- What is Jenkins Pipeline?
- How many ways we can create Pipeline?
- Create Jenkins Pipeline using Groovy Script / Jenkins File

CI &


Developer 1
Developer 2
Developer 3
GitHub
Source Control
Management Server
Build
Test
Compile
Jenkins
Jenkins CI
Server
Pass
Fail
Fail
Production
Environment
Release and
deploy
Tester

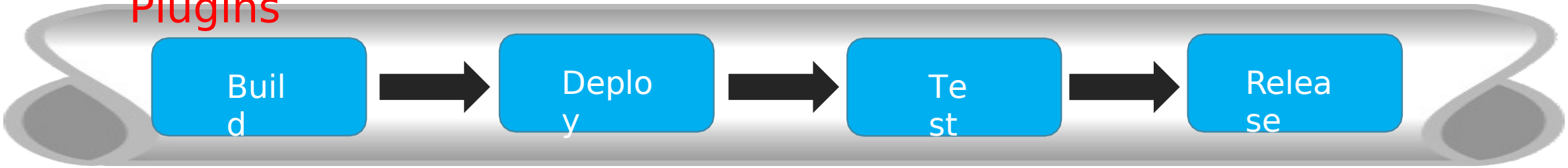
Jenkins Pipeline



How many Ways we can create Pipeline

- We can Create Jenkins Pipeline in 2 Ways
- 1) Using **Build And Delivery Pipeline Plugins**
- **2) Using Groovy Script on the Fly(Here we use Jenkins file)**
  - Scripted
  - Declarative

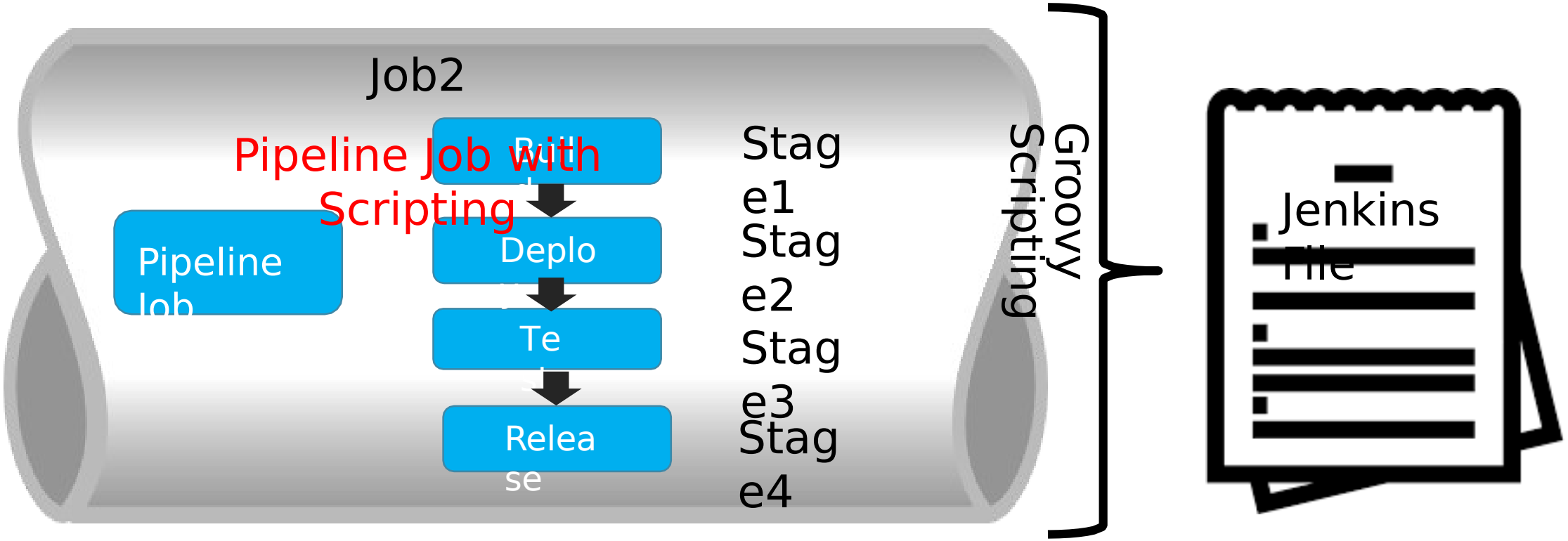
Build And Delivery Pipeline Plugins



Job1

Job 3

Job 4



Types of Pipeline

- Scripted Pipeline
- Declarative Pipeline

Pipeline concepts

- Pipeline
- A Pipeline is a user-defined model of a CD pipeline. A Pipeline's code defines your entire build process, which typically includes stages for building an application, testing it and then delivering it.
- Also, ***a pipeline block is a key part of Declarative Pipeline syntax***.
- Node
- A node is a machine which is part of the Jenkins environment and is capable of executing a Pipeline.
- Also, ***a node block is a key part of Scripted Pipeline syntax.***
- Stage
- A stage block defines a conceptually distinct subset of tasks performed through the entire Pipeline (e.g. "Build", "Test" and "Deploy" stages), which is used by many plugins to visualize or present Jenkins Pipeline status/progress.
- Step
- A single task. Fundamentally, a step tells Jenkins what to do at a particular point in time (or "step" in the process). For example, to execute the shell command make use the sh step: sh 'make'. When a plugin
  extends the Pipeline DSL, that typically means the plugin has implemented a new step.

Scripted Pipeline

Jenkinsfile (Scripted Pipeline)

```
node { 1
    stage('Build') { 2
        // 3
    }
    stage('Test') { 4
        // 5
    }
    stage('Deploy') { 6
        // 7
    }
}
```

1. Execute this Pipeline or any of its stages, on any available agent.
2. Defines the "Build" stage. `stage` blocks are optional in Scripted Pipeline syntax. However, implementing `stage` blocks in a Scripted Pipeline provides clearer visualization of each `stage's subset of tasks/steps in the Jenkins UI.
3. Perform some steps related to the "Build" stage.
4. Defines the "Test" stage.
5. Perform some steps related to the "Test" stage.
6. Defines the "Deploy" stage.
7. Perform some steps related to the "Deploy" stage.

Jenkins file

- Jenkinsfile is a text file that contains the definition of a Jenkins Pipeline and is checked into source control.
- https://jenkins.io/doc/book/pipeline/jenkinsfile/

Declarative Pipeline

Jenkinsfile (Declarative Pipeline)

```
pipeline {
    agent any (1)
    stages {
        stage('Build') { (2)
            steps {
                // (3)
            }
        }
        stage('Test') { (4)
            steps {
                // (5)
            }
        }
        stage('Deploy') { (6)
            steps {
                // (7)
            }
        }
    }
}
```

1. Execute this Pipeline or any of its stages, on any available agent.
2. Defines the "Build" stage.
3. Perform some steps related to the "Build" stage.
4. Defines the "Test" stage.
5. Perform some steps related to the "Test" stage.
6. Defines the "Deploy" stage.
7. Perform some steps related to the "Deploy" stage.