

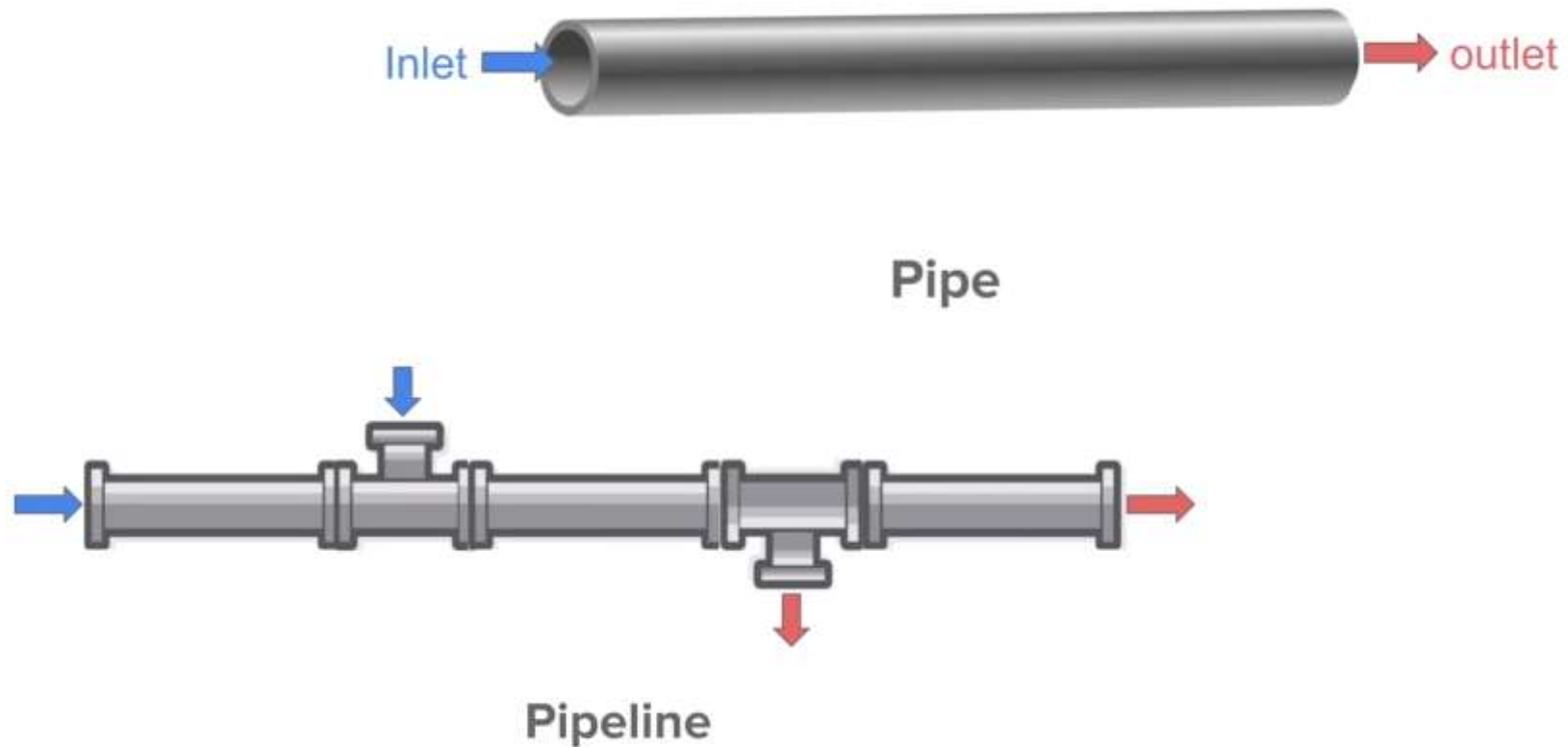
# Jenkins Pipeline

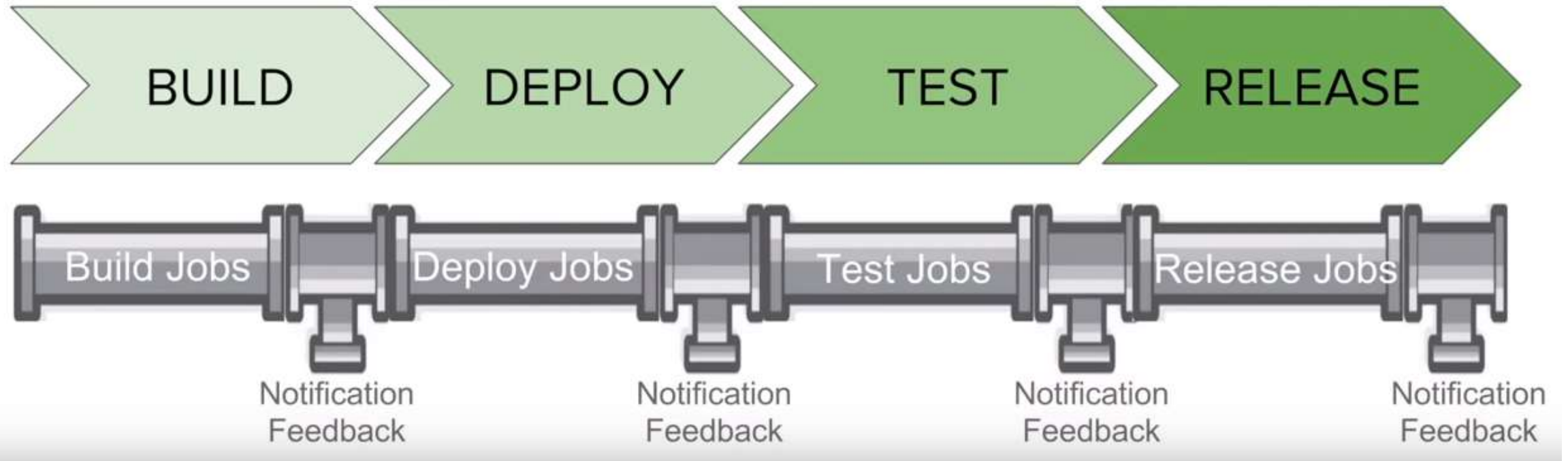
---

# What is Pipeline?

---

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





# How to setup BUILD PIPELINE in Jenkins

---

Step1: Chain required jobs in sequence Add upstream/downstream jobs

Step2: Install *Build Pipeline Plugin*

Step3: Add

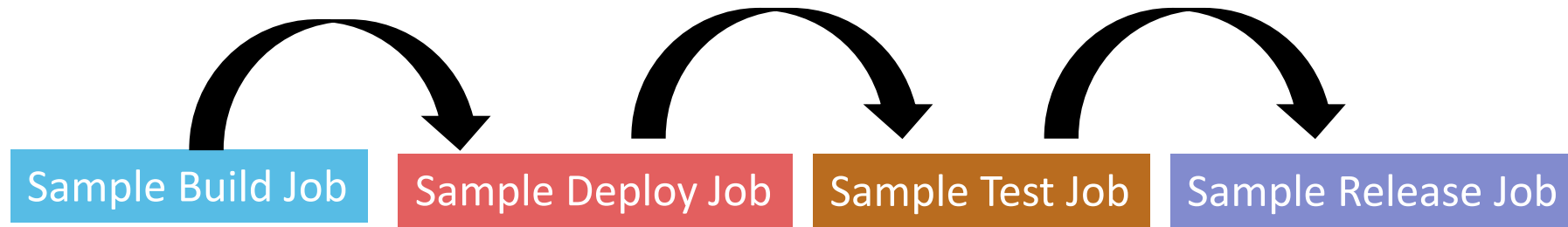
- Build Pipeline View

- Configure the view Step

Step4: Run and Validate

# Demo

---



# How to Build Pipelines with Groovy Script

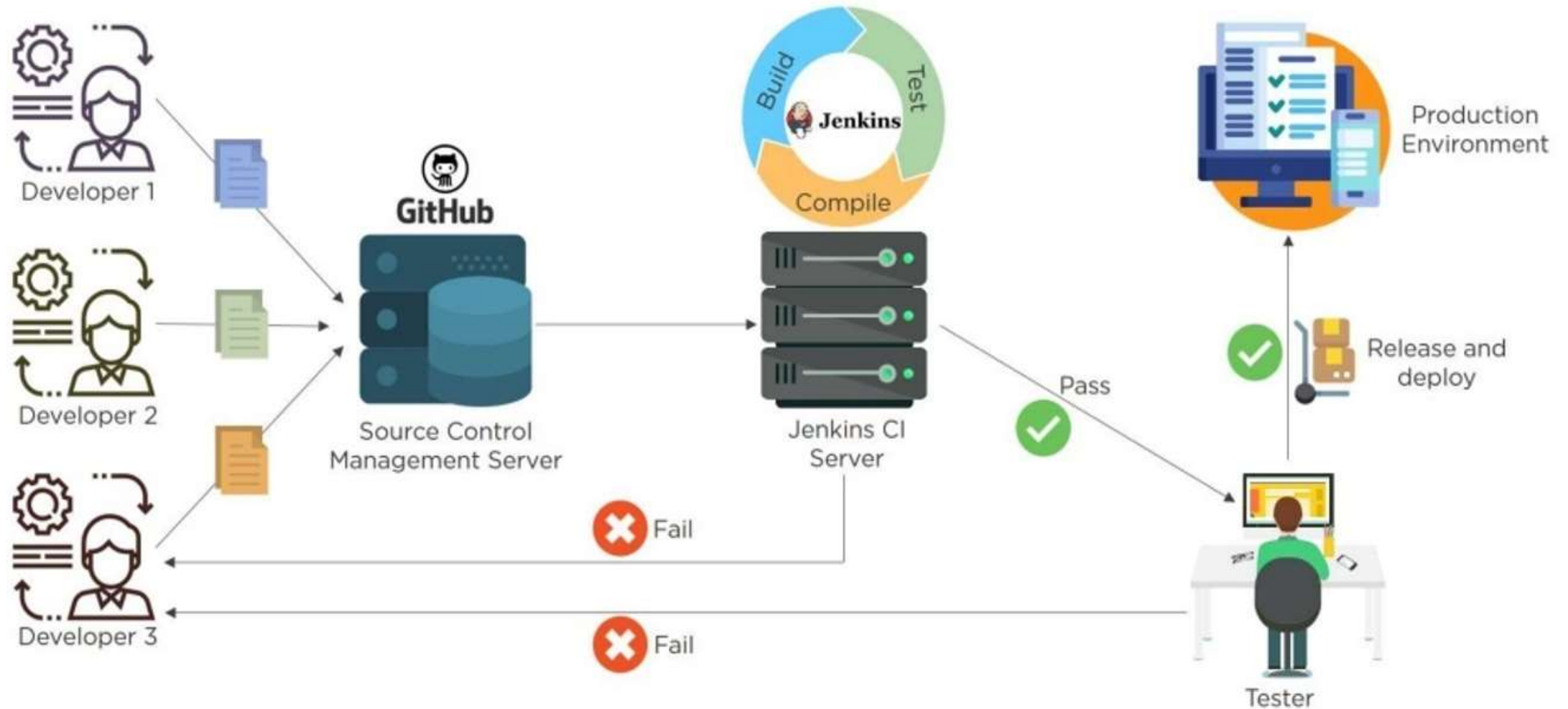
---

# Agenda

---

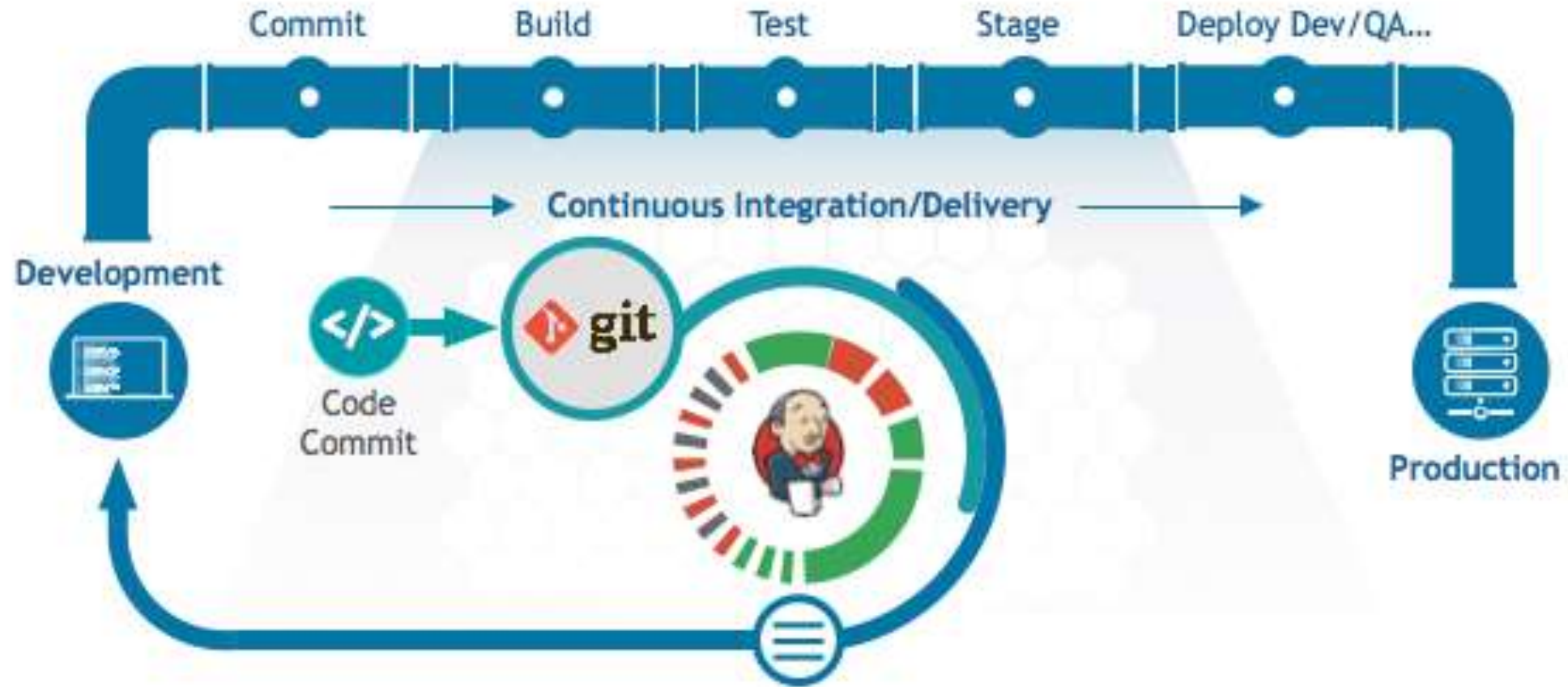
- CI & CD
- What is Jenkins Pipeline?
- How many ways we can create Pipeline?
- Create Jenkins Pipeline using Groovy Script / Jenkins File

# CI & CD





# Jenkins Pipeline

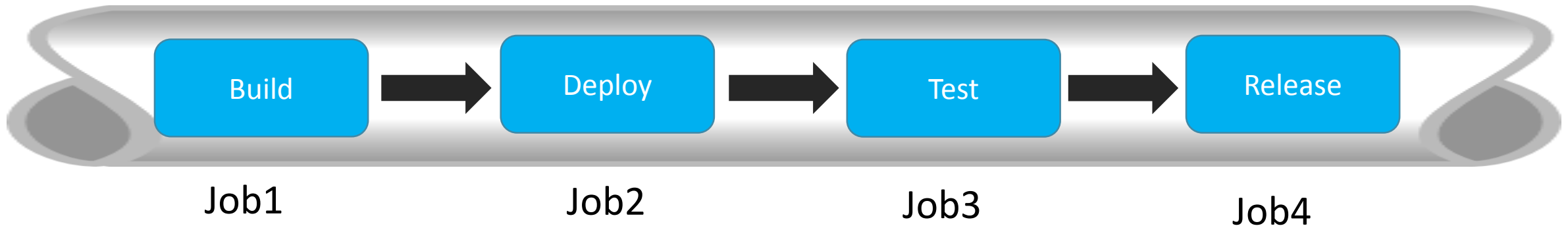


# How many Ways we can create Pipeline

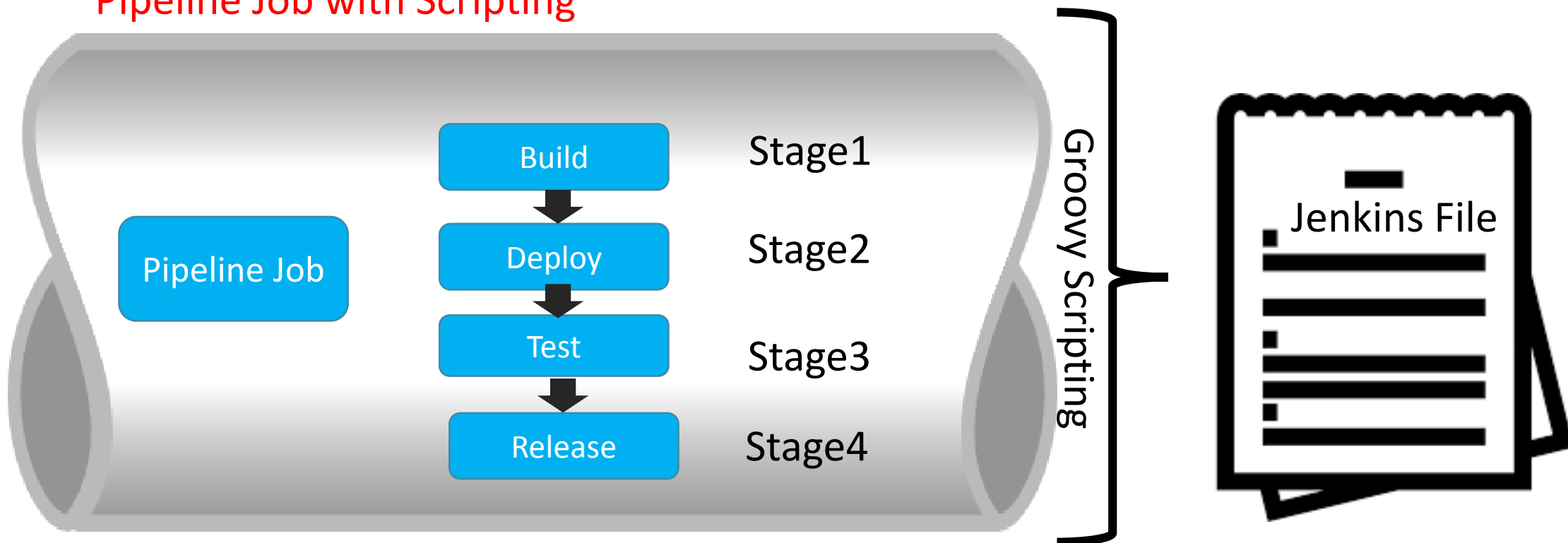
---

- 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



## Pipeline Job with Scripting



# 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 { ❶  
    stage('Build') { ❷  
        // ❸  
    }  
    stage('Test') { ❹  
        // ❺  
    }  
    stage('Deploy') { ❻  
        // ❼  
    }  
}
```

- ❶ Execute this Pipeline or any of its stages, on any available agent.
- ❷ 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.
- ❸ Perform some steps related to the "Build" stage.
- ❹ Defines the "Test" stage.
- ❺ Perform some steps related to the "Test" stage.
- ❻ Defines the "Deploy" stage.
- ❼ 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 ❶  
    stages {  
        stage('Build') { ❷  
            steps {  
                // ❸  
            }  
        }  
        stage('Test') { ❹  
            steps {  
                // ❺  
            }  
        }  
        stage('Deploy') { ❻  
            steps {  
                // ❼  
            }  
        }  
    }  
}
```

- ❶ Execute this Pipeline or any of its stages, on any available agent.
- ❷ Defines the "Build" stage.
- ❸ Perform some steps related to the "Build" stage.
- ❹ Defines the "Test" stage.
- ❺ Perform some steps related to the "Test" stage.
- ❻ Defines the "Deploy" stage.
- ❼ Perform some steps related to the "Deploy" stage.



Thank you

---