# Multi Branch Pipeline

- We can define pipeline as a code in different branches to handle changes according to
  - different customer
  - differnt type build
  - different environments
- Refer Here for official docs
- Refer Here for repo created in the classroom
- note: use classroom video for screens

## Understanding shared libraries

- Write a declarative pipeline to build maven project
  - stages:
    - scm: git
    - build: maven with goal
    - post
      - display unit tests
      - archive artifacts
- pipeline

```
pipeline {
    agent any
    stages {
        stage('git'){
            steps {
                git url: 'https://github.com/spring-projects/spring-
petclinic.git',
                    branch: 'main'
            }
        }
        stage('build') {
            steps {
                sh 'mvn clean package'
            }
        }
    }
}
```

- We can reuse the pipelines or groovy scripts using shared libraries
- Refer Here
- As discussed in the class we have create a simple pipeline

```
@Library('qt-classroom@v1.1') _
mavenSimple()
```

- Other implementation with parameters

```
@Library('qt-classroom@v1.2') _
maven maven: 'MAVEN_3.9.8',
      branch: 'main',
      url: 'https://github.com/spring-projects/spring-petclinic.git',
      goals: 'package'
```