Pipelines as code in Jenkins

“Pipeline-as-code” is the idea to allow everyone involved in DevOps to create and maintain Jenkins pipelines.

In fact, there are two ways of applying this “pipeline-as-code” principle in life: Scripted and Declarative pipelines.

Types of Pipelines are:

1. Declarative Pipeline Script
2. Scripted Pipeline Script

Declarative Pipeline:

1. It is always begins with the word 'pipeline'
2. Declarative pipeline is a relatively new feature that supports the pipeline as code concept. It makes the pipeline code easier to read and write. This code is written in a Jenkinsfile which can be checked into a source control management system such as Git.
3. Since this Groovy script was not typically desirable to all the users, the declarative pipeline was introduced to offer a simpler and more optioned Groovy syntax.
4. Declarative pipelines are a more recent approach to the “pipeline-as-code” principle
5. The “pipeline” block is the main block that contains the entire declaration of a pipeline. In this example, we'll consider only “agent”, “stages”, and “steps” sections:

   pipeline – contains the whole pipeline
   agent – defines the machine that will handle this pipeline
   stages – declares the stages of the pipeline
   steps – small operations inside a particular stage

Syntax:

```
pipeline{

    agent any
    stages{

        stage("Checkout the code from the GitHub Repository by CubeIplKumar"){
            steps{
                git url:'https://github.com/cubeiplKumar/DevOpsClassCodes.git'
            }
        }
        stage("code compile by kumar"){
            steps{
                sh "mvn compile"
            }
        }
         stage("code tesing by kumar"){
            steps{
                sh "mvn test"
            }
        }
        stage("code QA by kumar"){
            steps{
                sh "mvn pmd:pmd"
            }
        }
        stage("code package by kumar"){
            steps{
                sh "mvn package"
            }
        }
        }
}
```

Scripted Pipeline

1. It is always begins with the word 'node'

2. Scripted pipeline is a traditional way of writing the code. In this pipeline, the Jenkinsfile is written on the Jenkins UI instance.

3. Though both these pipelines are based on the groovy DSL, the scripted pipeline uses stricter groovy based syntaxes because it was the first pipeline to be built on the groovy foundation.

***Node Block*:**

Node is the part of the Jenkins architecture where Node or agent node will run the part of the workload of the jobs and master node will handle the configuration of the job. So this will be defined in the first place as

Stage Block:

Stage block can be a single stage or multiple as the task goes. And it may have common stages like

- Cloning the code from SCM
- Building the project
- Running the Unit Test cases
- Deploying the code
- Other functional and performance tests.

Syntax:

```
node {
    stage('Test') {
        git 'https://github.com/cubeiplKumar/DevOpsClassCodes.git'
        sh 'mvn test'
        archiveArtifacts artifacts: 'target/surefire-reports/**'
    }
    stage('Build') {
        sh 'mvn clean package '
        archiveArtifacts artifacts: 'target/*.jar'
    }
    stage('Deploy') {
        sh 'echo Deploy'
    }
}
```