# JENKINS DECLARATIVE PIPELINES:

## WHAT IS JENKINS FILE?

If we are writing the entire Jenkins pipeline in a text format. It contains the steps that are required for running the Jenkins pipeline.

It is used to create a pipeline for **build and deploy** the code. This Jenkins file uses **groovy syntax**.

These jenkins file will written in 2 ways

1. Declarative pipeline
2. Scripted Pipeline

**Declarative pipeline:** It is a **recent feature** of the jenkins pipeline which helps us to write the pipeline in a easier way. It will starts with the word **pipeline**

**Scripted pipeline:** It is a **traditional way** of writing a jenkins pipeline as a code. It starts with the word **node**

## SINGLE STAGE PIPELINE:

```
pipeline {
    agent any
    stages {
        stage ("stage-1") {
            steps {
                echo "hai this is my first stage"
            }
        }
    }
}
```

**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.

**agent :** it defines that, in which server the pipeline will gets executes.

**stage :** stage in jenkins file contains of unique tasks such as build, test, Deploy etc..

**step :** It tells the jenkins what exactly needs to be done.

ex: executing a build command or linux command etc..

## MULTI-STAGE PIPELINE:

```
pipeline {
  agent any
  stages {
    stage ("MUSTAGA") {
      steps {
        echo "hello Mustafa"
      }
    }
    stage ("devops") {
      steps {
        echo "we are learning devops"
      }
    }
    stage ("aws") {
      steps {
        echo "we are learning aws also"
      }
    }
  }
```

```
}
```

**PIPELINE AS A CODE:** YOU CAN RUN COMMANDS HERE

```
pipeline {
  agent any
  stages {
    stage('CMD') {
      steps {
        sh 'touch file1'
        sh 'pwd'
      }
    }
  }
}
```

**MULTIPLE COMMANDS OVER SINGLE LINE:**

```
pipeline {
  agent any
  stages {
    stage('CMD') {
      steps {
        sh '''
        touch file2
        pwd
        date
        whoami
        '''
      }
    }
```

```
  }
}
```

## ENVIRONMENT VARIABLES:

```
pipeline {
  agent any
  environment {
     name = 'raham'
  }
  stages {
     stage('ENV') {
        steps {
                echo "hai my name is $name"
        }
     }
  }
}
  pipeline {
     agent any
     environment {
        name = 'raham'
     }
     stages {
        stage('ENV1') {
           steps {

              sh 'echo "${name}"'
           }
        }
```

```
        stage('ENV2') {
            environment {
                name = 'shaik'
            }
            steps {

                sh 'echo "${name}"'
            }
        }
    }
}
```

## PIPELINE TO GET A SOURCE CODE TO DEV SERVER:

```
pipeline {
    agent {
        node {
            label 'dev'
        }
    }
    stages {

        stage ("git") {
            steps {
                git "https://github.com/devops0014/devoprepoforpractice.git"
            }
        }
    }
}
```

## PIPELINE WITH PARAMETERS:

### STRING:

```
pipeline {
   agent any
   parameters {
      string (name: "aws", defaultValue: "EC2", description: "i am unsing aws cloud")
   }
   stages {
      stage ("stage-1") {
         steps {
            echo "hai i am using parameters"
         }
      }
   }
}
```

### BOOLEAN:

```
pipeline {
   agent any
   parameters {
      booleanParam (name: "jenkins", defaultValue: true, description: "")
   }

   stages {
      stage('Hello') {
         steps {
            echo ' i am using boolean parameter'
         }
      }
   }
}
```

## CHOICE:

```
pipeline {
   agent any
   parameters {
      choice (name: "branch", choices : ["one", "two", "three", "four"], description: "this is cp")
   }
   stages {
      stage('Hello') {
         steps {
            echo 'Hello World'
         }
      }
   }
}
```

## PIPELINE WITH POST BUILD ACTIONS:

### ALWAYS:

```
pipeline {
  agent any

  stages {
     stage('Hello') {
        steps {
           echo 'Hello World
        }
     }
  }
  post{
     always {
```

```
            echo 'THIS WILL BE PRINTED ANYWAY'
        }
    }
}
```

**SUCCESS:**

```
pipeline {
  agent any

  stages {
    stage('Hello') {
      steps {
        echo 'Hello World
      }
    }
  }
  post{
    success {
        echo 'THIS WILL BE PRINTED IF THE BUILD GETS SUCCESS'
      }
    }
}
```

**FAILURE:**

```
pipeline {
  agent any

  stages {
    stage('Hello') {
      steps {
        echo 'Hello World
```

```
        }
      }
    }
    post{
      failure {
            echo 'THIS WILL BE PRINTED EVEN IF THE BUILD GETS FAILURE'
        }
      }
}
```

**SUCCESS, ALWAYS, FAILURE:**

```
pipeline {
    agent any

    stages {
        stage('Hello') {
            steps {
                ech 'Hello World'
            }
        }
    }
    post {
        always {
            echo "this will gets printed anyway"
        }
        success {
            echo "this will be printed if the build gets success"
        }
        failure {
            echo "this will be printed even if the build gets failure"
        }
```

```
    }
}
```

## INPUT IN JENKINS FILE:

```
pipeline {
    agent any

    stages {
        stage('deploy') {
            input {
                message "can i deploy"
                ok "yes you can"
            }
            steps {
                echo 'our code is deployed'
            }
        }
    }
}
```

## JENKINS FILE TO DEPLOY THE CODE IN TOMCAT SERVER:

```
pipeline {
    agent any
    stages {
        stage ("code") {
            steps {
                git "https://github.com/devops0014/myweb-01.git"
            }
        }
```

```
    stage ("build") {
      steps {
        sh 'mvn clean package'
      }
    }
    stage ("deploy") {
      steps {
        sh "cp /var/lib/jenkins/workspace/pipeline-1/target/*.war /opt/apache-tomcat-9.0.71/webapps"
      }
    }
  }
}
```

## JENKINS POST BUILD ACTIONS:

```
pipeline {
  agent any

  stages {
    stage('Hello') {
      steps {
        eco 'Hello World'
      }
      post {
        always {
          echo "this will be printed anyways"
```

```
        }
      }
    }
  }
  post {
    always {
      echo "This block always runs."
    }
    changed {
      echo "This block runs when the current status is different than the previous one."
    }
    fixed {
      echo "This block runs when the current status is success and the previous one was failed or unstable."
    }
    regression {
      echo "This block runs when the current status is anything except success but the previous one was successful."
    }
    unstable {
      echo "This block runs if the current status is marked unstable."
    }
    aborted {
      echo "This block runs when the build process is aborted."
    }
    failure {
      echo "This block runs when the build is failed."
```

```
    }
    success {
      echo "This block runs when the build is succeeded."
    }
    unsuccessful {
      echo "This block runs when the current status is anything except success."
    }
  }
}
```

## JENKINS WHEN CONDITION:

```
pipeline {
  agent any
  stages {
    stage("Test") {
      when {
        equals(actual: currentBuild.number, expected: 5)
      }
      steps {
        echo "Hello World!"
      }
    }
  }
}
```

Note: In the above pipeline, when the build id is 5, then only stage will gets executed.

## WHEN NOT:

```
pipeline {
    agent any
    stages {
        stage("Test") {
            when {
                not {
                    branch "master"
                }
            }
            steps {
                echo "Test stage."
            }
        }
    }
}
```

## ALL OF:

```
pipeline {
    agent any
    stages {
        stage("Deploy") {
```

```
        when {
            allOf {
                branch "master"
                environment(name: "ENV", value: "production")
                tag "release-*"
            }
        }
        steps {
            echo "Deploy to production."
        }
    }
  }
}
```

## ANY OF:

```
pipeline {
  agent any
  stages {
    stage("NewFeature") {
      when {
        anyOf {
          branch "feature"
          changelog ".*new feature.*"
        }
      }
      steps {
```

```
            echo "Test new feature."
        }
    }
  }
}
```

## BUILD PERIODICALLY:

```
pipeline {
  agent any
  triggers {
    cron "* * * * *"
  }
  stages {
    stage("Test") {
      steps {
        echo "Hello World!"
      }
    }
  }
}
```