<https://e.printstacktrace.blog/jenkins-scripted-pipeline-vs-declarative-pipeline-the-4-practical-differences/>

SAMPLE SCRIPTED & DECLARATIVE PIPELINE

node {

stage("Build") {

echo "Some code compilation here..."

}

stage("Test") {

echo "Some tests execution here..."

echo "1"

}

stage("Deploy") {

echo "Some tests execution here..."

echo 1

}

}

pipeline {

agent any

stages {

stage("Build") {

steps {

echo "Some code compilation here..."

}

}

stage("Test") {

steps {

echo "Some tests execution here..."

echo "1"

}

}

stage("Deploy") {

steps {

echo "Some tests execution here..."

echo 1

}

}

}

}

pipeline {

agent any

stages {

stage('Checkout') {

steps {

git credentialsId: 'GitHubAccess', url: 'git@github.com:mavrick202/DevOps-B14-Batch-Repo.git'

}

}

stage("Build") {

steps {

**echo** "Some code compilation here..."

**echo** "${env.BRANCH\_NAME}"

**sh** "ls -al"

}

}

stage("Test") {

steps {

**echo** "Some tests execution here..."

**echo** "${env.BUILD\_NUMBER}"

}

}

stage("Deploy") {

steps {

**echo** "Some tests execution here..."

**echo** "${env.BUILD\_ID}"

}

}

stage("Verify") {

steps {

**echo** "Some tests execution here..."

**echo** "${env.JENKINS\_HOME}"

**echo** "1"

}

}

}

}

Testing WHEN Condition in DeclarativePipeline:



pipeline {

agent any

options {

timestamps()

}

stages {

stage("Build") {

options {

timeout(time: 1, unit: "MINUTES")

}

steps {

sh 'printf "\\e[31mSome code compilation here...\\e[0m\\n"'

}

}

stage("Test") {

when {

environment name: "ENVT", value: "TEST"

}

options {

timeout(time: 2, unit: "MINUTES")

}

steps {

echo "This Is An Test Environment"

}

}

stage("Dev") {

when {

environment name: "ENVT", value: "DEV"

}

options {

timeout(time: 2, unit: "MINUTES")

}

steps {

echo "This Is An Dev Environment"

}

}

stage("Prod") {

when {

environment name: "ENVT", value: "PROD"

}

options {

timeout(time: 2, unit: "MINUTES")

}

steps {

echo "This Is An Prod Environment"

}

}

}

}

DECLARATIVE USING DOCKER AGENT:

pipeline {

agent {

docker { image 'sreeharshav/pack-terr-jenkinsci-ssh-slave:v1' }

}

stages {

stage('Packer Build AMI') {

steps {

sh 'pwd'

sh 'ls -al'

sh 'terraform version'

sh 'packer version'

}

}

stage('Deploy EC2 Server') {

steps {

sh 'pwd'

sh 'ls -al'

sh 'terraform version'

sh 'packer version'

}

}

stage('Build Docker Image') {

steps {

sh 'pwd'

sh 'ls -al'

sh 'terraform version'

sh 'packer version'

}

}

stage('Push Image to Docker Hub') {

steps {

sh 'pwd'

sh 'ls -al'

sh 'terraform version'

sh 'packer version'

}

}

stage('Deploy to Docker Host') {

steps {

sh 'pwd'

sh 'ls -al'

sh 'terraform version'

sh 'packer version'

}

}

stage('Check WebApp Rechability') {

steps {

sh 'pwd'

sh 'ls -al'

sh 'terraform version'

sh 'packer version'

}

}

}

}