PIPLENE

It is the file in wich we write the comands, that jenkins will execute, one by one, it is written in a domain specific language. It dosnot have exstention, something similar to Dockerfile.

Jenkinsfile\_pipline - это как список команд которые должны быть выполнены jenkinsom, идея в том чтобы не прописывать каждую команду а зделать типа список, что за чем, также есть возможность указывать переменные.

Всё наченается в главной секции называющейся piplene, например:

pipline{

any agent <-- любой node(slave) может выполнять следующие далее инструкции, это можно перенастроить на какойто отдельный. Каждый node имеет

конфигуацию, где можно установить/прописать лэйблы. В job-е в jenkins file we indicate not 'any agent', but:

agent{

label 'указываем лэйбл того нода который мы хотим чтобы выполнил этот файл'

}

stages{ <-- блок где расположены stages

stage('stage\_1'){

...здесь команды которые надо выполнить, для удобочитаемости их можно разбить по этим самым stage-ам

}

stage('stage\_2'){

...

}

}

post { <--optional block (что-то типа finally в java где закрываются ресурсы, такоеже назначение и здесь)

...что-то зделать в конце выполнения всей программы

}

}

So lets create jenkinsFile:

pipeline{

agent any

stages{

stage('stage-1'){

steps{

echo "stage-1"

echo "doing mvn clean"

echo "doing mvn package"

}

}

stage('stage-2'){

steps{

echo "stage-2"

echo "building docker image"

}

}

stage('stage-3'){

steps{

echo "stage-3"

echo "pushing docker image"

}

}

}

post{

always{

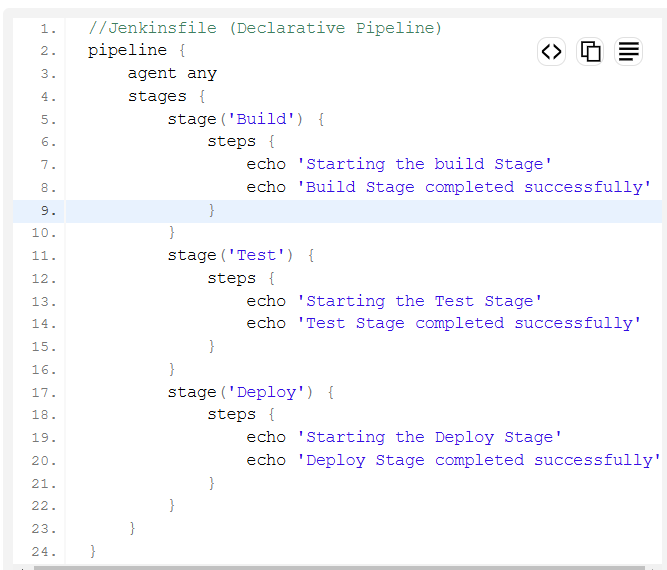
echo "post section"

}

}

}

Пример pipline -a :



Then select new job,give the name, choose pipeline, scrol down and paste the code that we write in a jenkinsfile,save it, press 'build now'.

OVERRIDE VARIABLSE IN JENKINSFILE

Sometimes we may want to override some variables that we use in a file.

pipeline{

agent any

environment {

NUMBER = 3 <-- this is something as global variable. All the stages can refer to this variable.

}

stages{

stage('stage-1'){

steps{ <-- steps is kind of comands that we execute

echo "stage-1"

echo "doing mvn clean"

echo "doing mvn package"

echo "NUMBER = ${NUMBER}" <-- example how to use variable

}

}

stage('stage-2'){

environment {

NUMBER = 100 <--we override variable

}

steps{

echo "stage-2"

echo "building docker image"

echo "NUMBER = ${NUMBER}" <-- printing new value of variable

}

}

stage('stage-3'){

steps{

echo "stage-3"

echo "pushing docker image"

echo "NUMBER = ${NUMBER}" <-- in this case the value vill be original = 3, because blocks of code works as in java

} and we overrided the value in stage #2 not in #3

}

}

post{

always{

echo "post section"

}

}