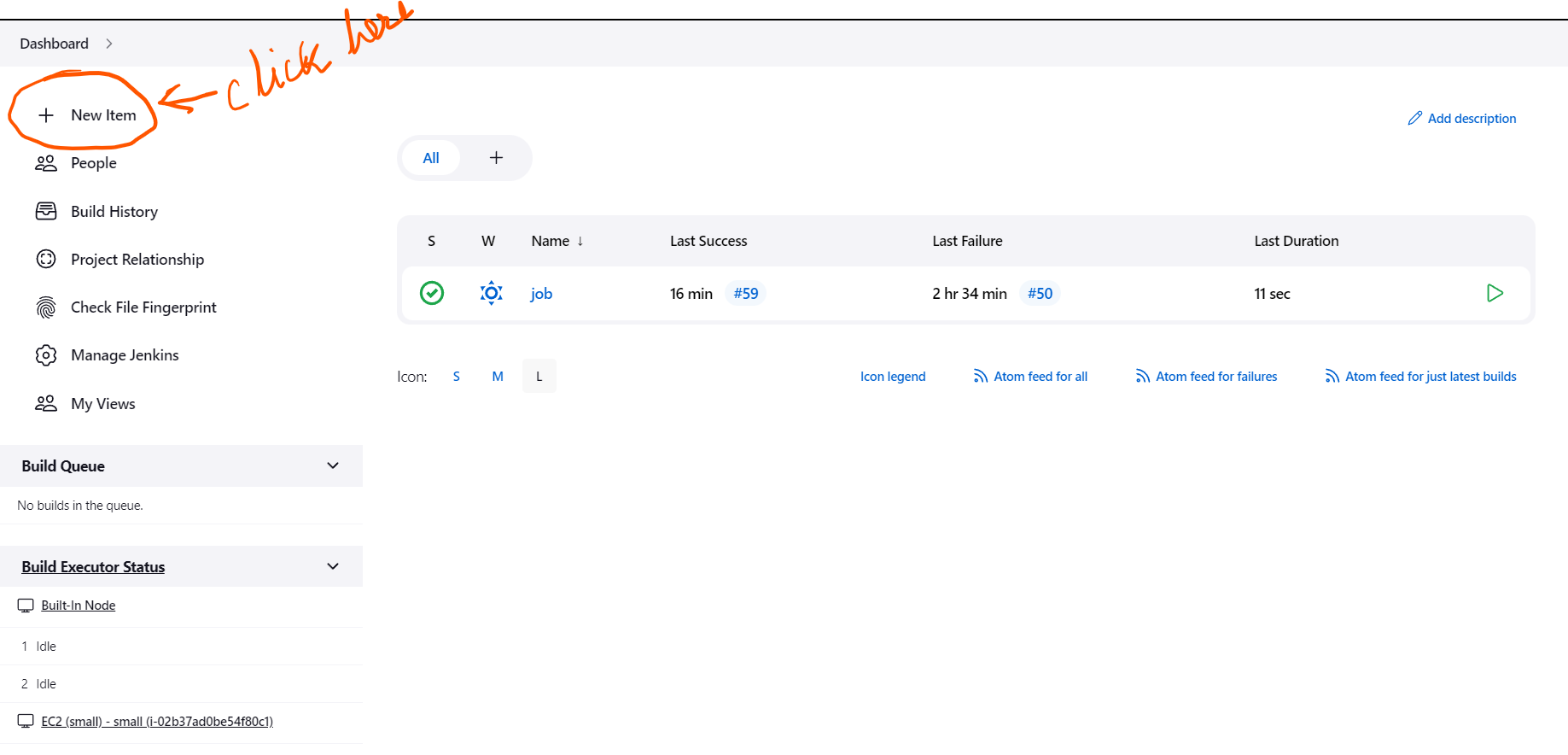
**Parameterized scripted Pipeline**

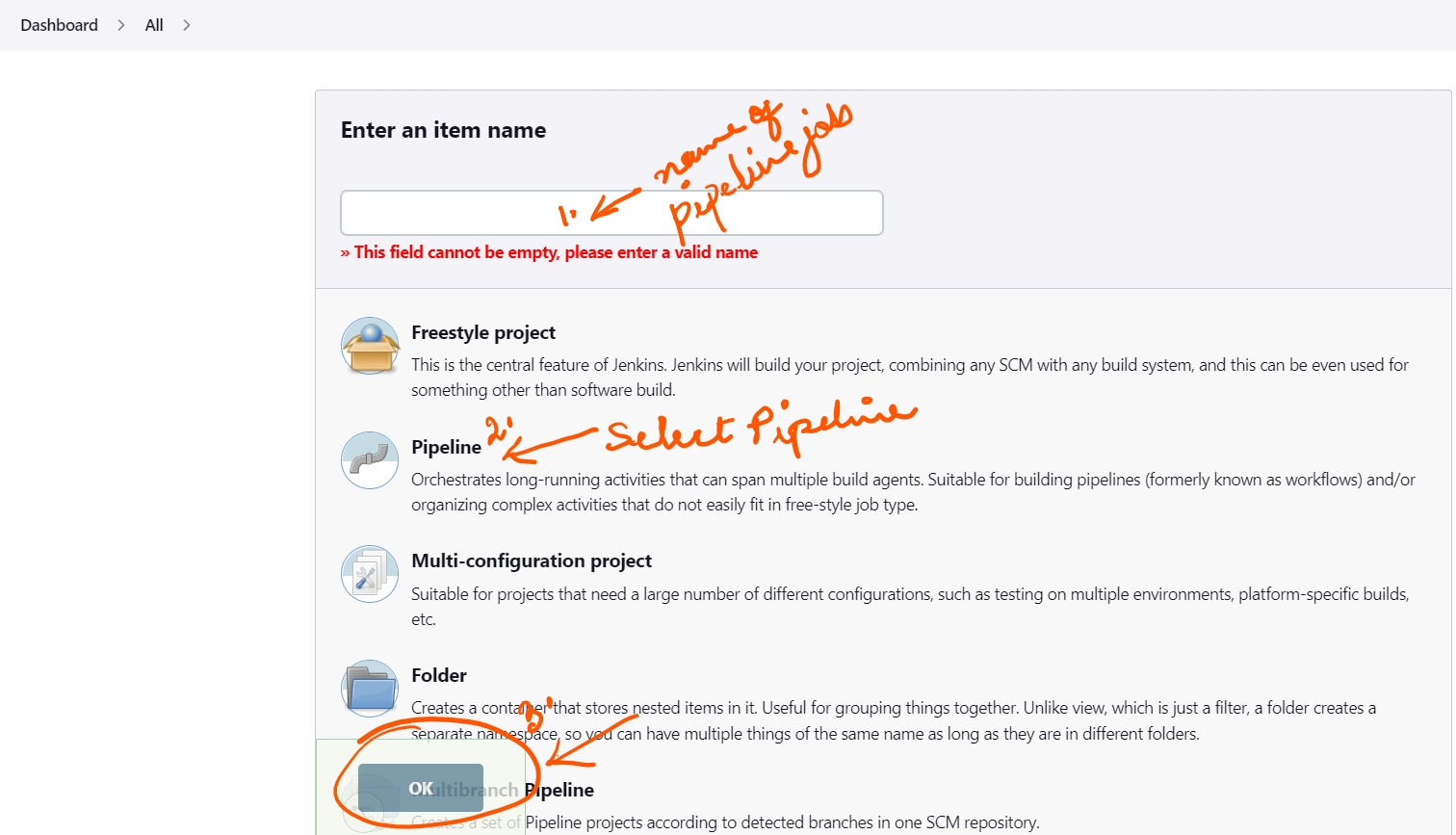
***Use Case: Based on different feature branch a dynamic slave instance of different configuration will be invoked and developers can run their pipeline.***

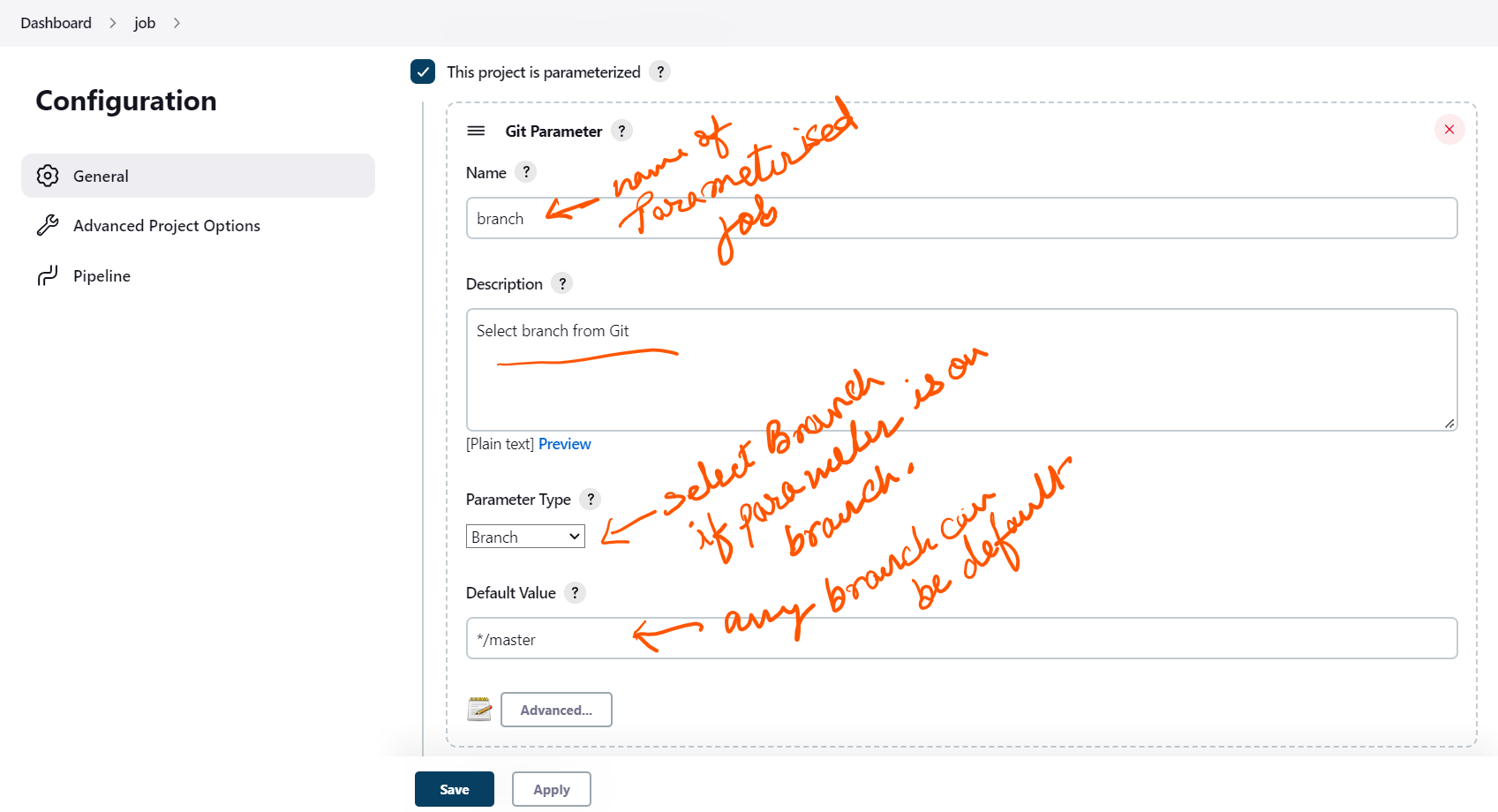
1. ***Create Jenkins-master EC2 instance.***
2. ***Create EC2 slave instance and install git and maven. Create image of this and terminate the instance.***

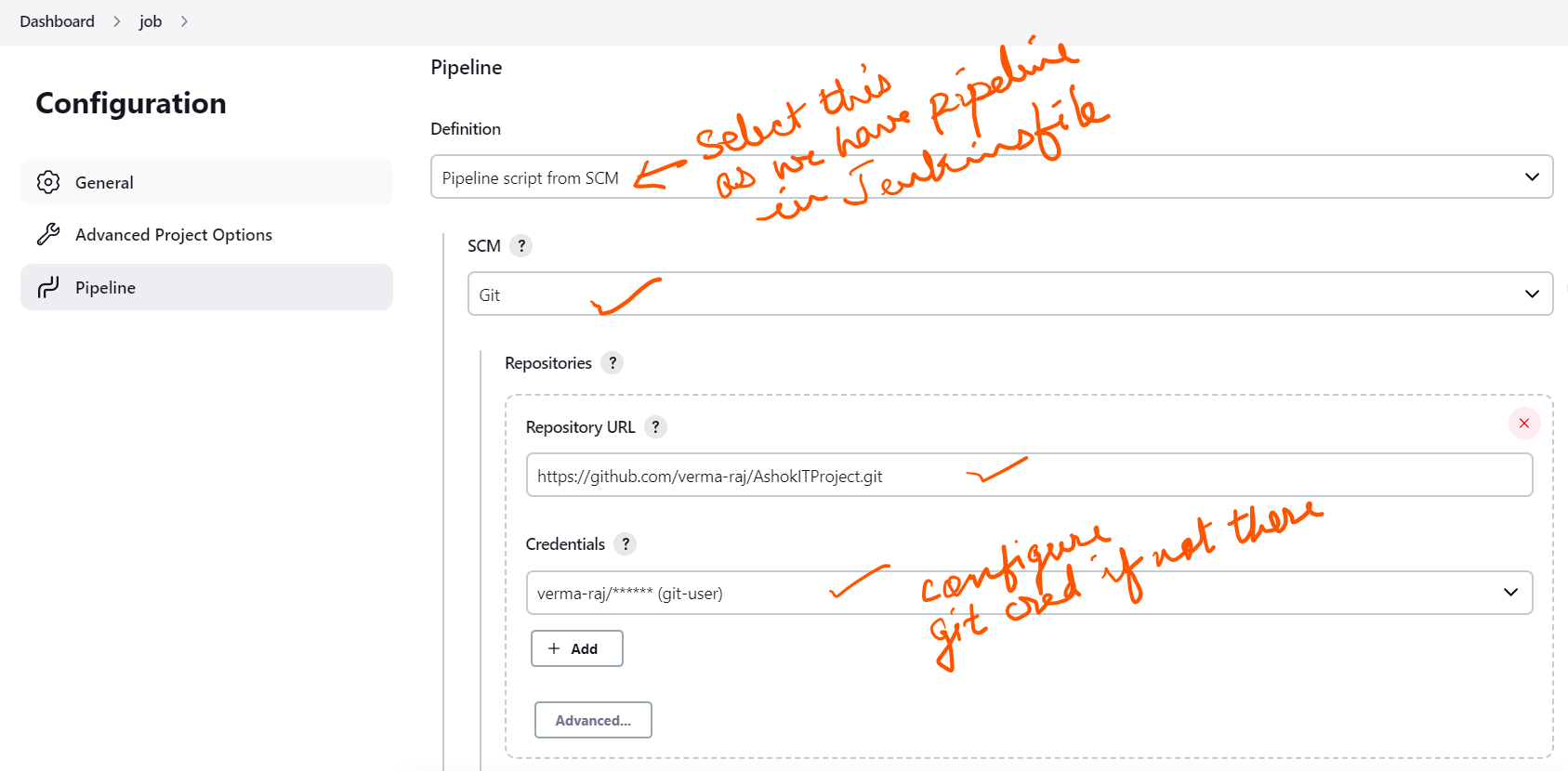
***Below is the way we configure the parameterized pipeline in Jenkins.***

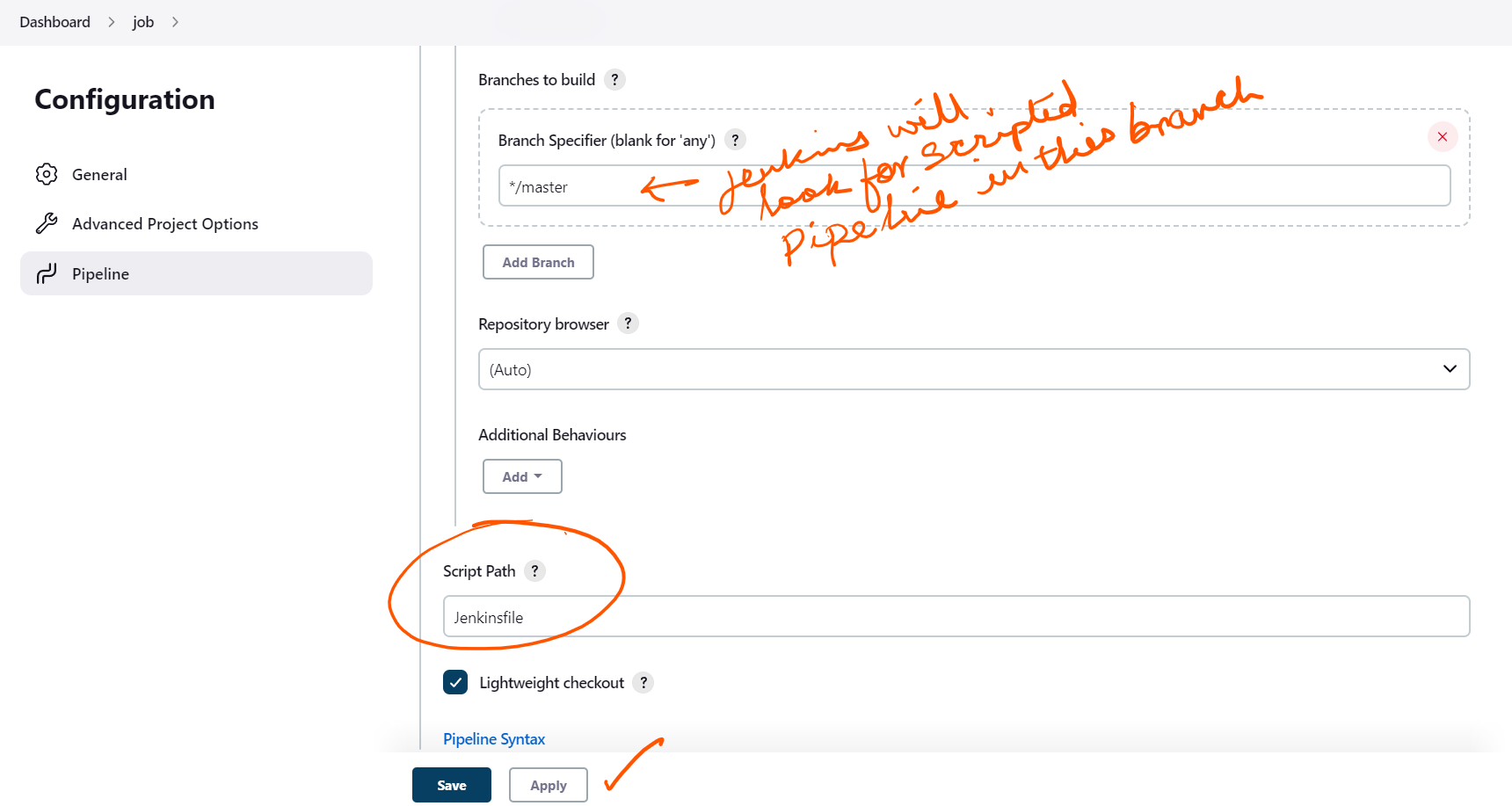
***\*\*\** Install *Git Parameter plugin* for this setup**

******

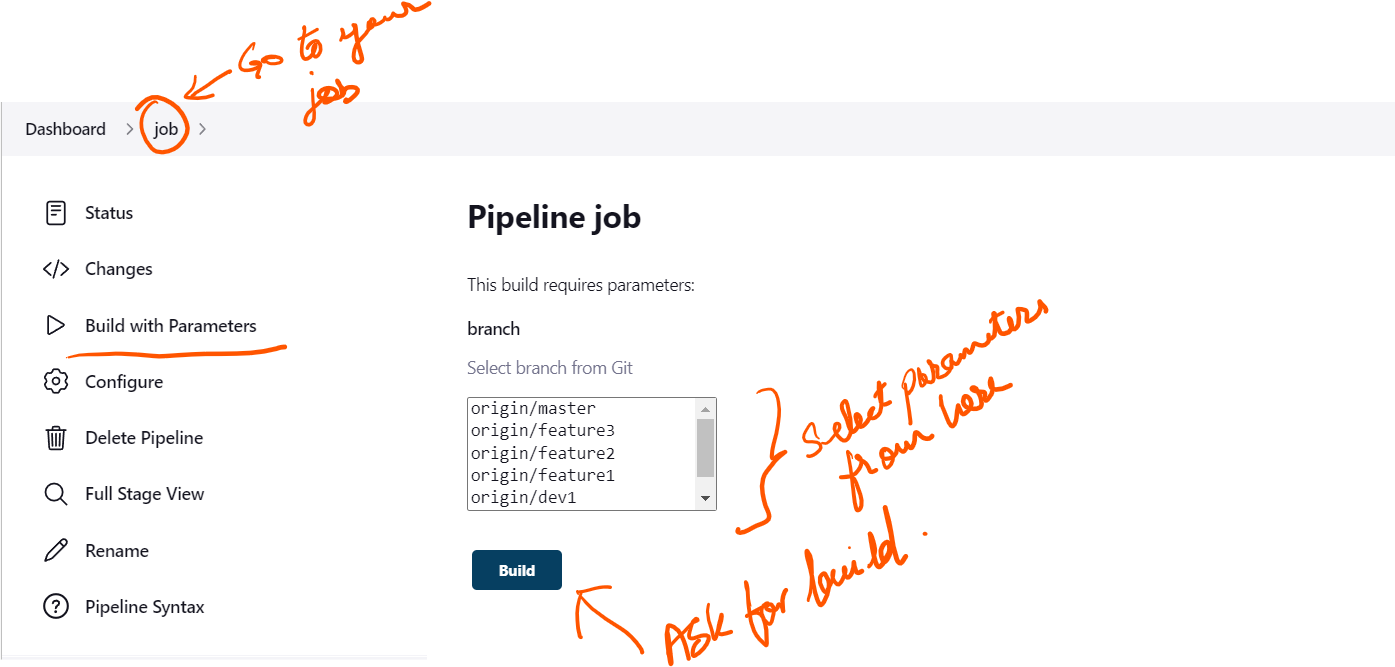
******

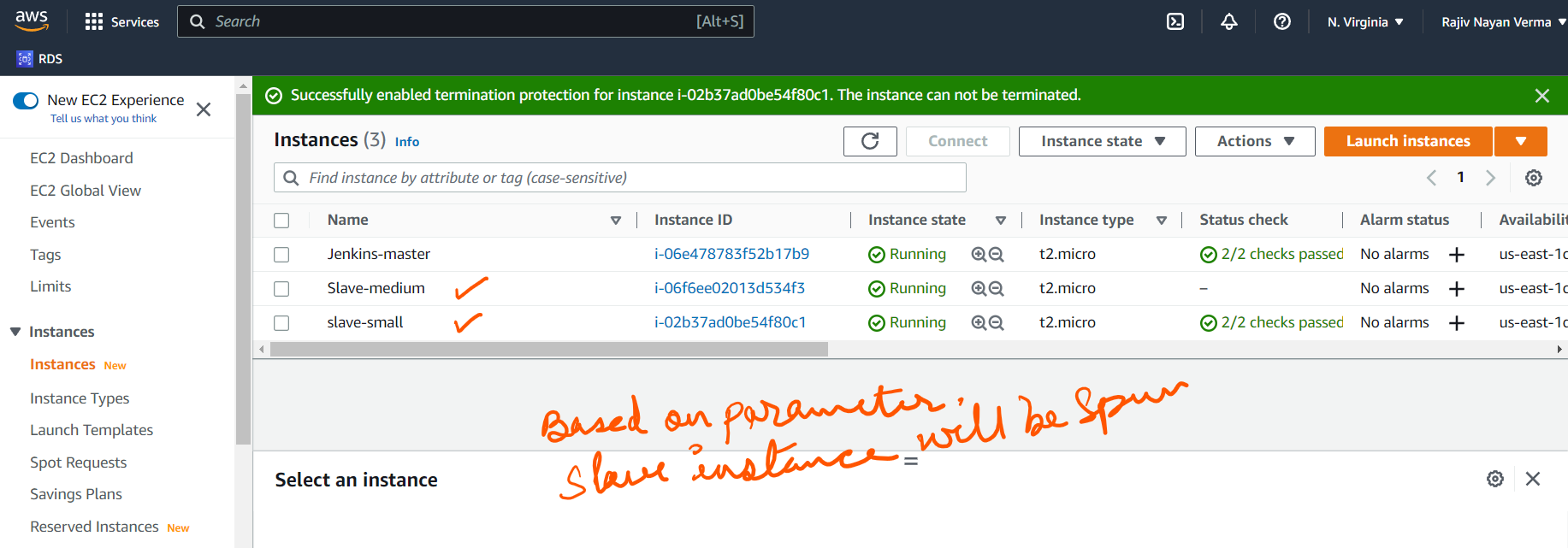
******

******

******

***Now your Jenkins configuration is done. Go to***

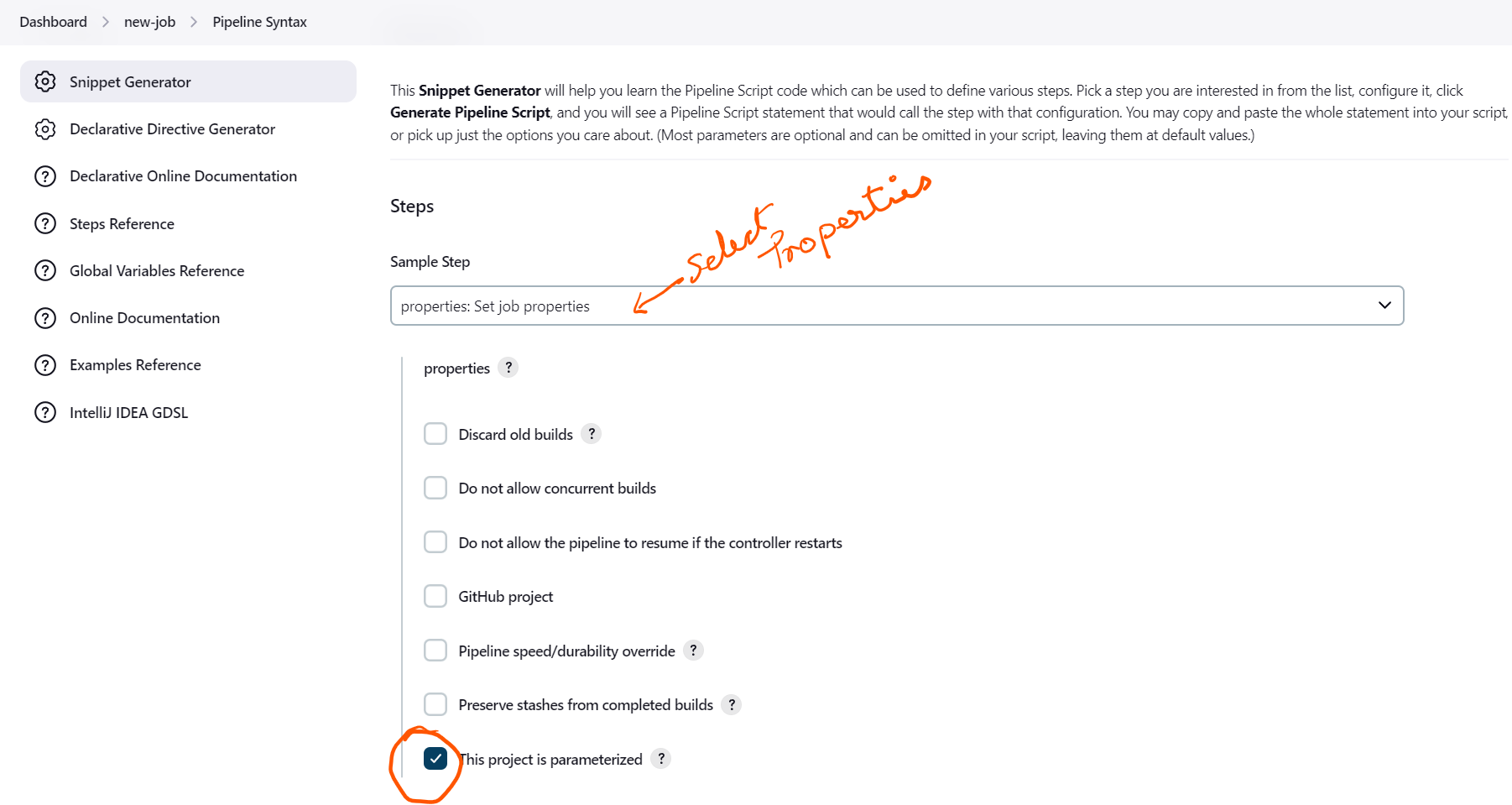
******

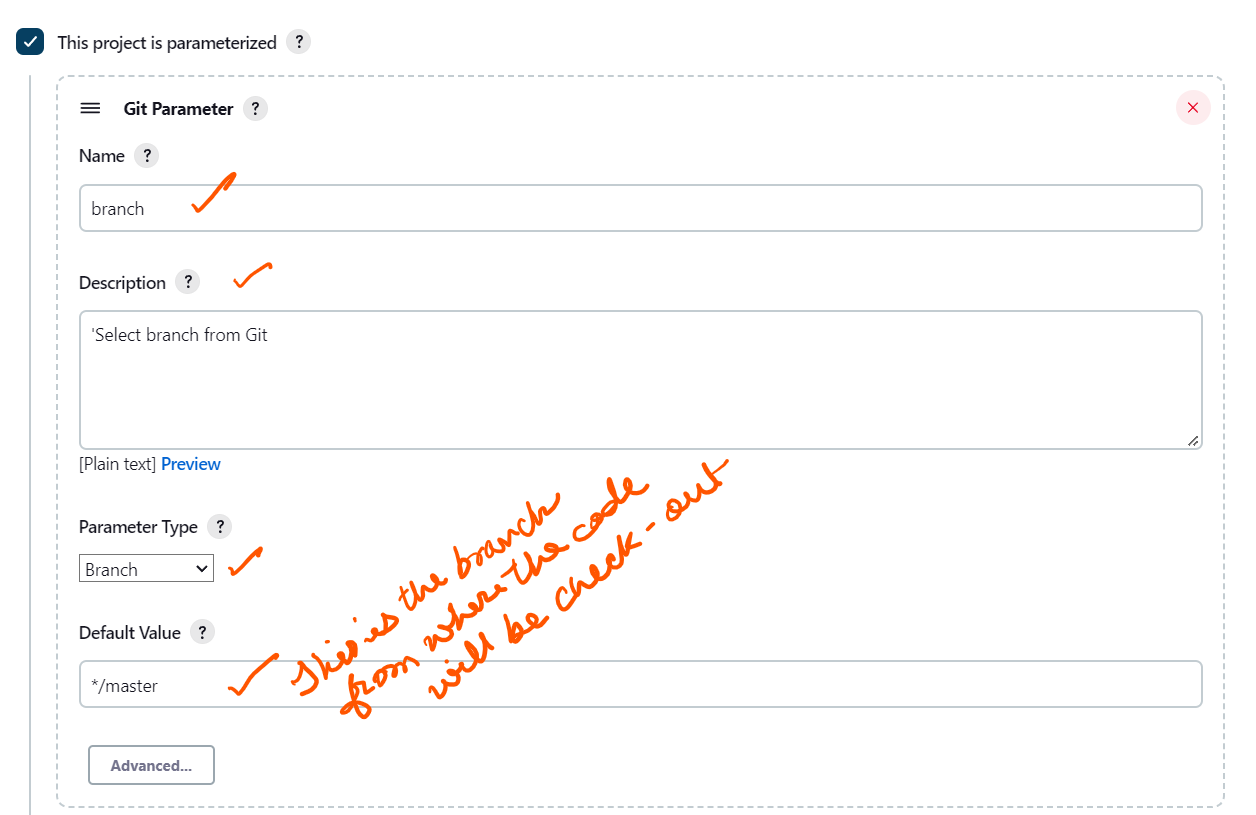
******

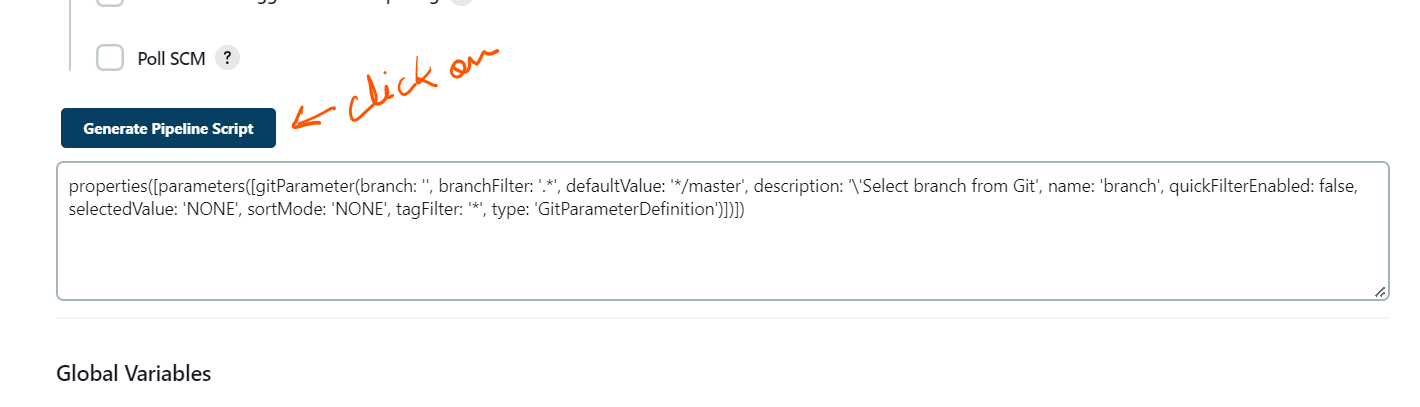
***Note : For master-slave configuration see document dynamic ec2-label launch***

***Follow below steps to generate pipeline script.***

properties([parameters([gitParameter(branch: '', branchFilter: '.\*', defaultValue: '\*/master', description: 'Select branch from Git', name: 'branch', quickFilterEnabled: false, selectedValue: 'NONE', sortMode: 'NONE', tagFilter: '\*', type: 'GitParameterDefinition')])])

******

******

******

***Create Jenkinsfile in you master branch of your repository. script is as below***

properties([parameters([gitParameter(branch: '', branchFilter: '.\*', defaultValue: '\*/master', description: 'Select branch from Git', name: 'branch', quickFilterEnabled: false, selectedValue: 'NONE', sortMode: 'NONE', tagFilter: '\*', type: 'GitParameterDefinition')])])

script {

if (env.branch == 'origin/feature1') {

def thisbranch = env.branch

echo "I only execute on the $thisbranch branch"

echo " This is what i Get - ${params.branch}"

node("small")

{

stage ("git-clone") {

echo " Launching small Instance and cloning GIT from $thisbranch and ${params.branch} "

checkout([$class: 'GitSCM', branches: [[name: params.branch]], extensions: [], userRemoteConfigs: [[credentialsId: 'git-user', url: 'https://github.com/verma-raj/AshokITProject.git']]])

}

stage ("Compiling the code") {

echo "\*\*\*\* Compiling the code\*\*\*\*\*"

sh 'mvn -v'

sh 'mvn clean package'

}

}

}

else if (env.branch == 'origin/feature2') {

def thisbranch = env.branch

echo "I only execute on the $thisbranch branch"

node("medium")

{ stage ("git-clone") {

echo " Launching small Instance and cloning GIT from $thisbranch "

echo " Launching small Instance and cloning GIT from $thisbranch and ${params.branch} "

checkout([$class: 'GitSCM', branches: [[name: params.branch]], extensions: [], userRemoteConfigs: [[credentialsId: 'git-user', url: 'https://github.com/verma-raj/AshokITProject.git']]])

}

stage ("Compiling the code") {

echo "\*\*\*\* Compiling the code\*\*\*\*\*"

sh 'mvn -v'

sh 'mvn clean package'

}

}

}

else {

echo 'I can execute elsewhere but its not configured'

}

}