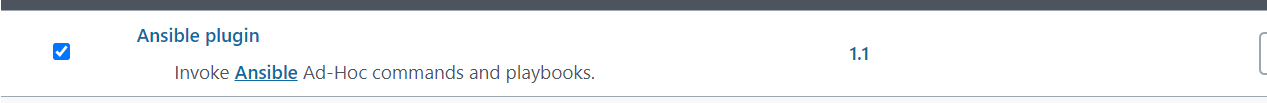
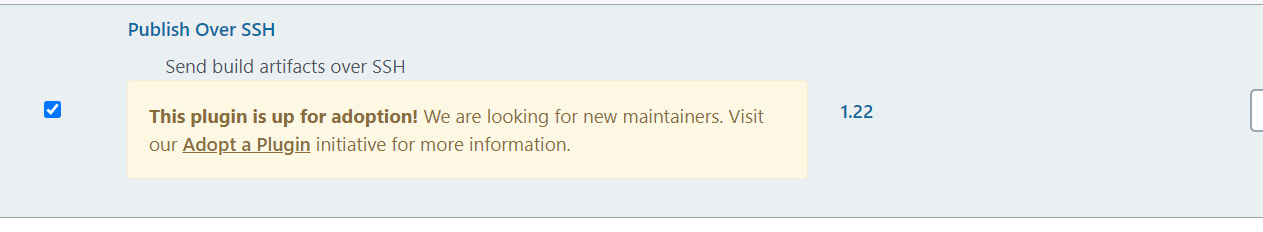
ANSIBLE MEDICLAIM JOB DOCUMENT

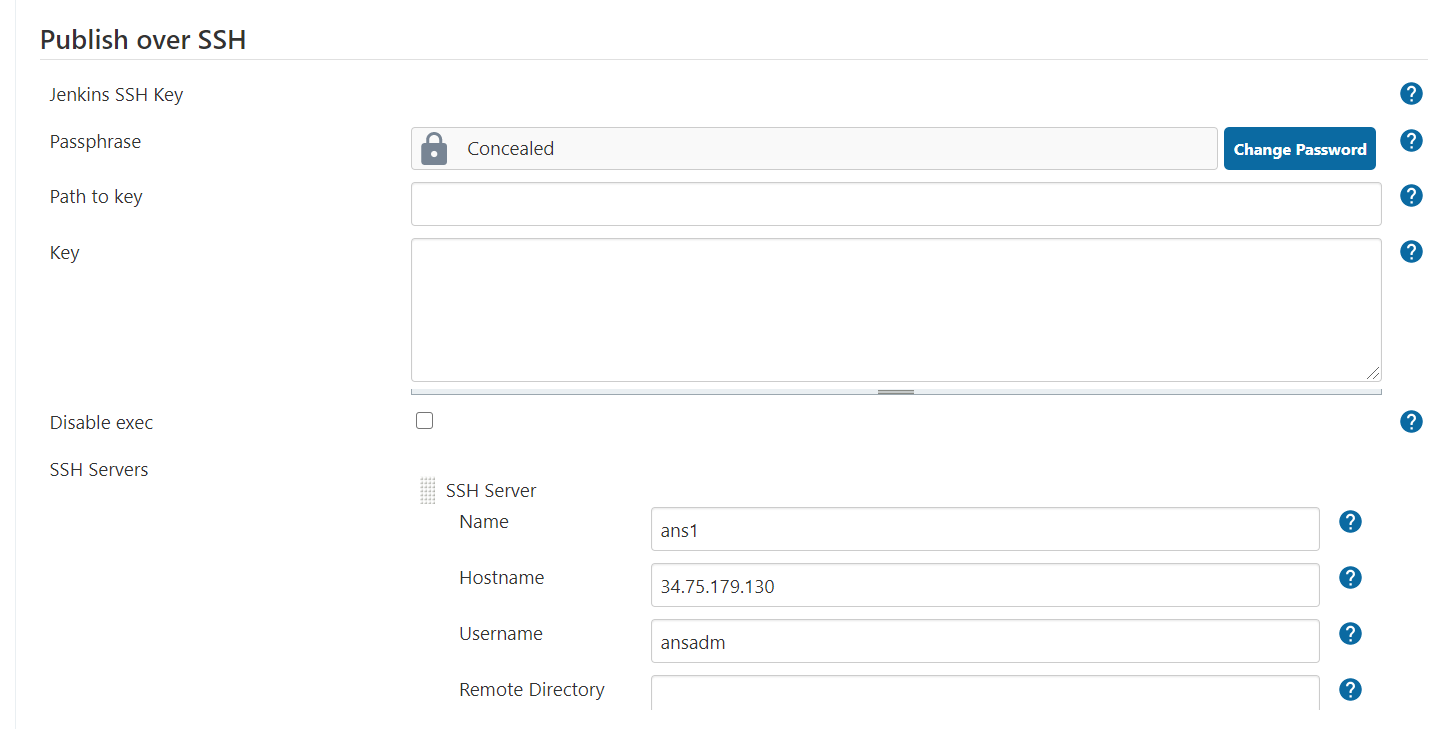
If we want to work on ansible we need to download re required plugin as a name of ANSIBLE PLUGIN in Jenkins.

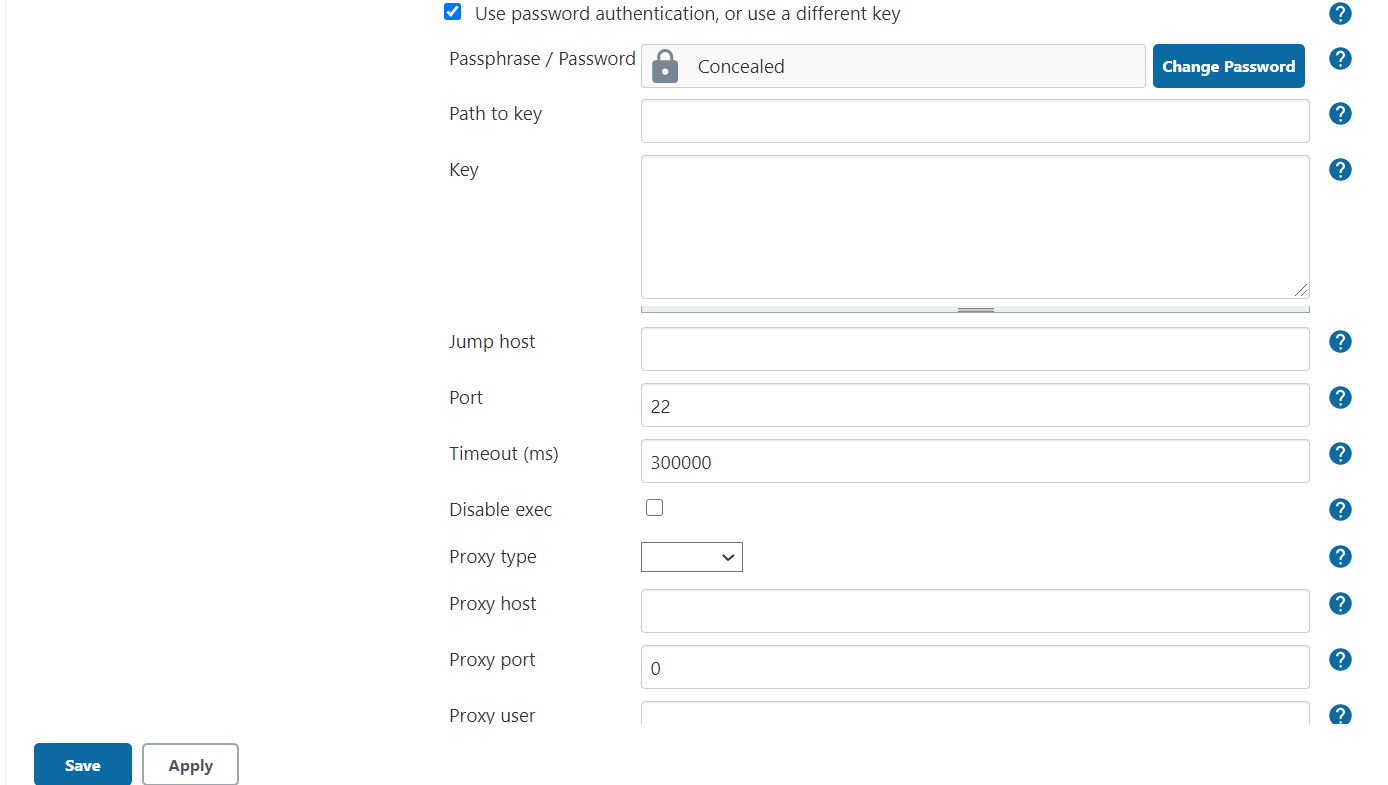


And if we want communicate over a ssh with help one more plugin is called PUBLISH OVER SSS PLUGIN

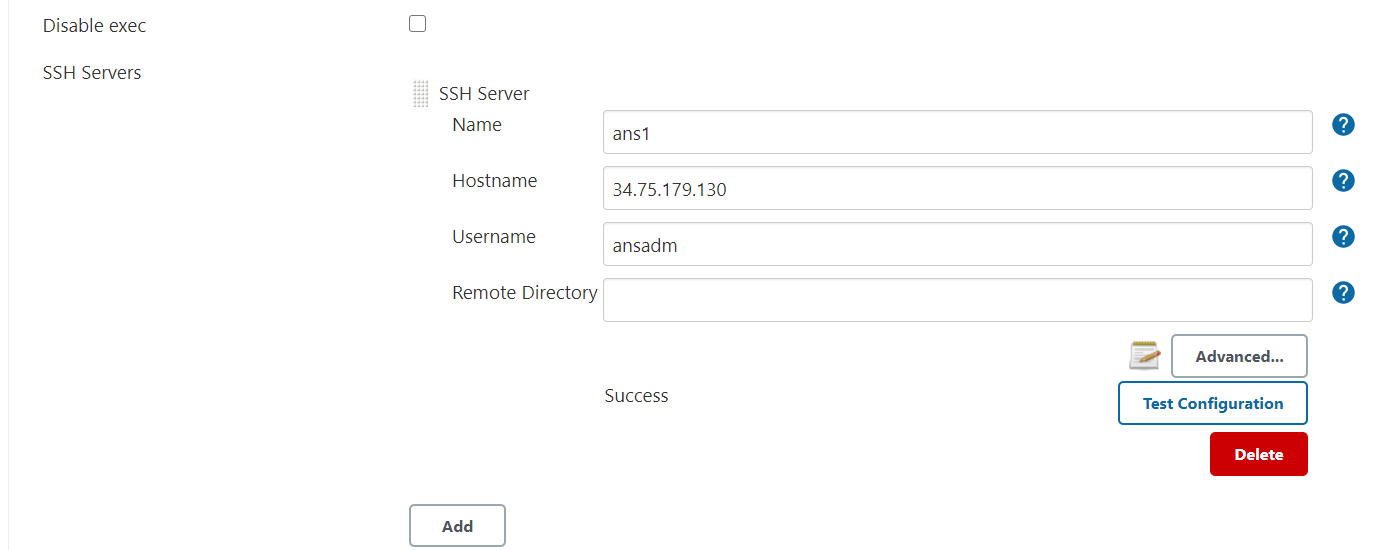


Then after successfully installed that plugins and we need to configure the ssh configuration goto manage Jenkins🡪configure system



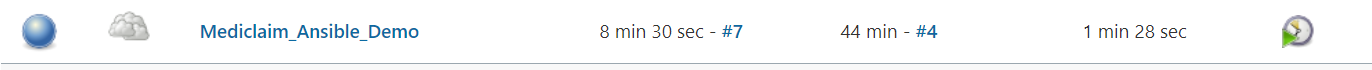


If we want check the connection is success or fail by click on test connection



After adding the configuration of slave machine save and apply

Now create new pipeline job as a name of Mediclaim\_ansible\_Demo as shown below



Go to pipeline script write the script as shown below

pipeline {

agent any

stages {

stage('SCM Checkout') {

steps {

git 'https://github.com/Gkasiraju/mediclaim.git'

}

}

stage('Build Approval'){

steps{

slackSend baseUrl: 'https://hooks.slack.com/services/', channel: 'jenkins', color: 'bad', message: "${env.BUILD\_URL}", tokenCredentialId: 'slackid1', username: 'Buildapprover'

script{

def userInput

try {

userInput = input(

id: 'Proceed1', message: 'Approval',submitterParameter: 'submitter', submitter:'admin', parameters: [

[$class: 'BooleanParameterDefinition', defaultValue: true, description: '', name: 'Please Provide Build Approval']])

}

catch(err) {

def user = err.getCauses()[0].getUser()

userInput = false

echo "Aborted by: [${user}]"

}}

}

}

stage ('Publish Unit Test Report')

{ steps {

sh '/opt/maven/bin/mvn verify -Dmaven.test.skip=true'

junit allowEmptyResults: true, testResults: '\*\*/target/surefire-reports/\*.xml'

}

}

stage ('Build && SonarQube analysis') {

steps {

withSonarQubeEnv('sonar'){

sh '/opt/maven/bin/mvn clean package sonar:sonar -Dsonar.password=admin123 -Dsonar.login=admin -Dmaven.test.skip=true'

}

}

}

stage('Quality Gate'){

steps {

timeout(time: 1, unit: 'HOURS'){

waitForQualityGate abortPipeline: true

}

}

}

stage ('Artifact-Deployment') {

steps {

sh '/opt/maven/bin/mvn clean deploy -Dmaven.test.skip=true'

}

}

stage ('Release') {

steps {

sh 'export JENKINS\_NODE\_COOKIE=dontkillme ;nohup java -jar $WORKSPACE/target/\*.jar &'

}

}

stage('copy files'){

steps {

script {

sshPublisher(publishers: [

sshPublisherDesc(configName: 'ans1',

transfers: [

sshTransfer(

cleanRemote: false,

excludes: '',

execCommand: '',

execTimeout: 120000,

flatten: false,

makeEmptyDirs: false,

noDefaultExcludes: false,

patternSeparator: '[, ]+',

remoteDirectory: '/var/www/sockets',

remoteDirectorySDF: false,

removePrefix: '',

sourceFiles: '\*\*/\*.jar'

)

],

usePromotionTimestamp: false,

useWorkspaceInPromotion: false,

verbose: false

)]

)

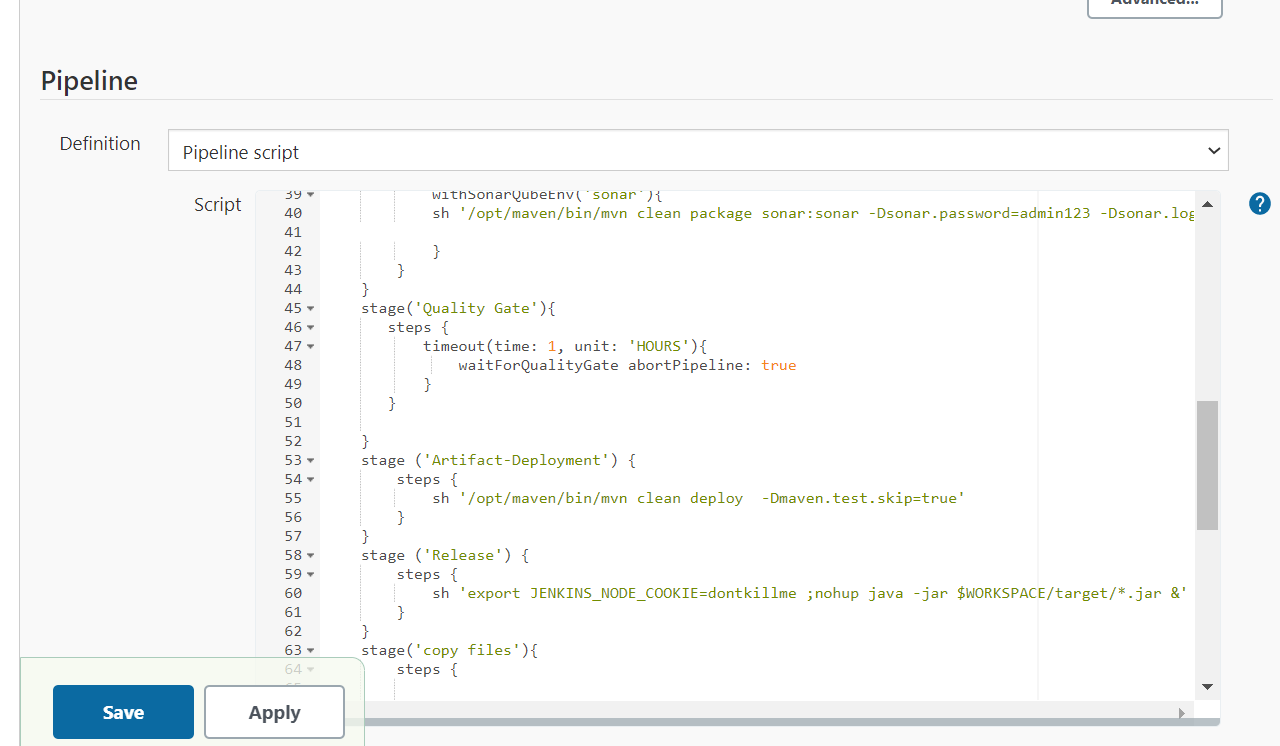
}

}

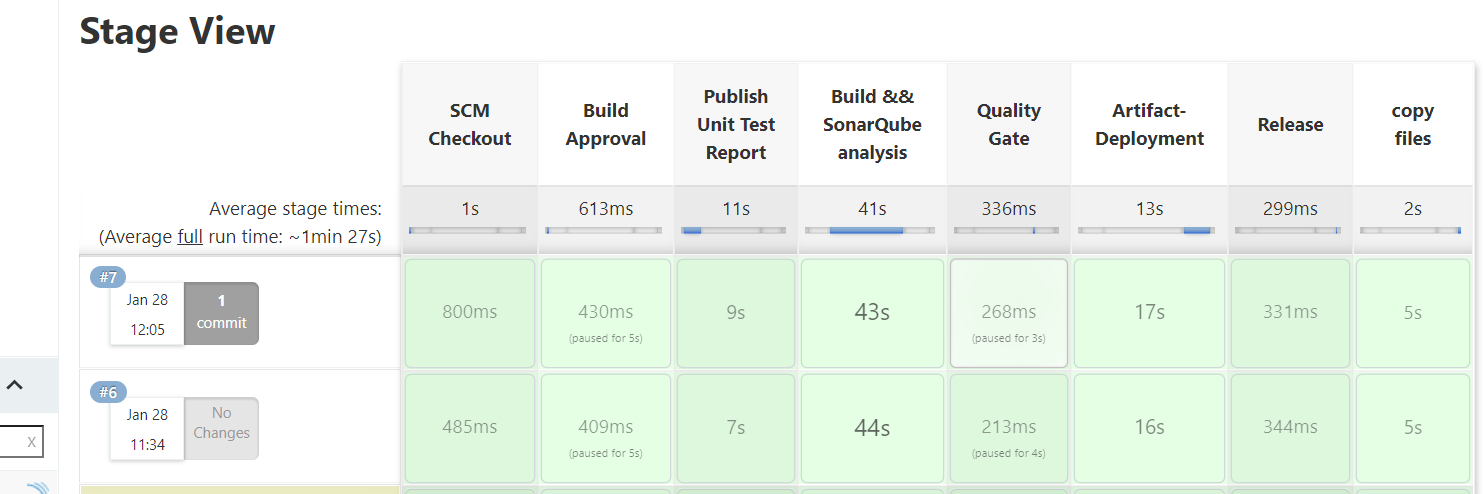
}

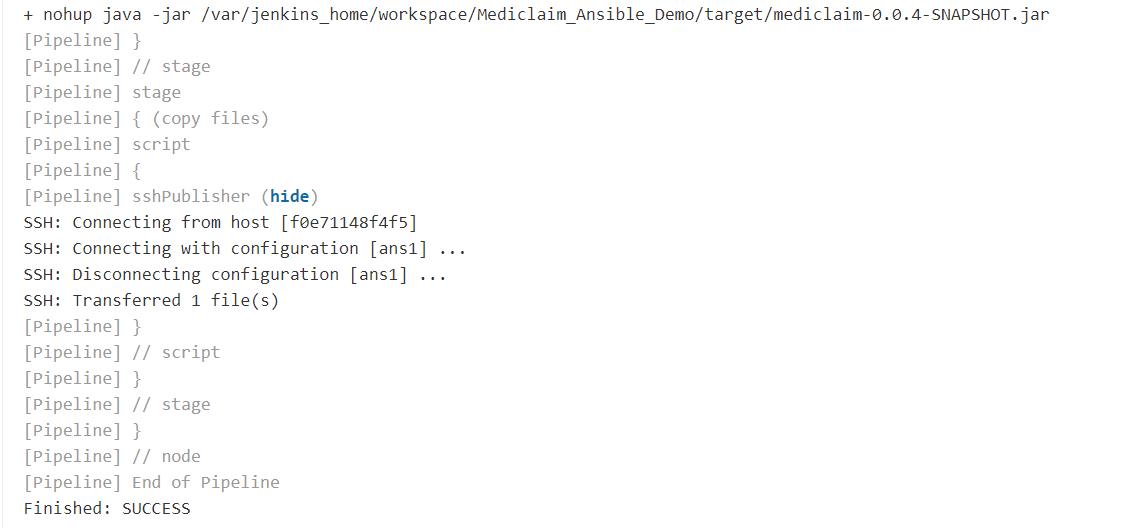
}

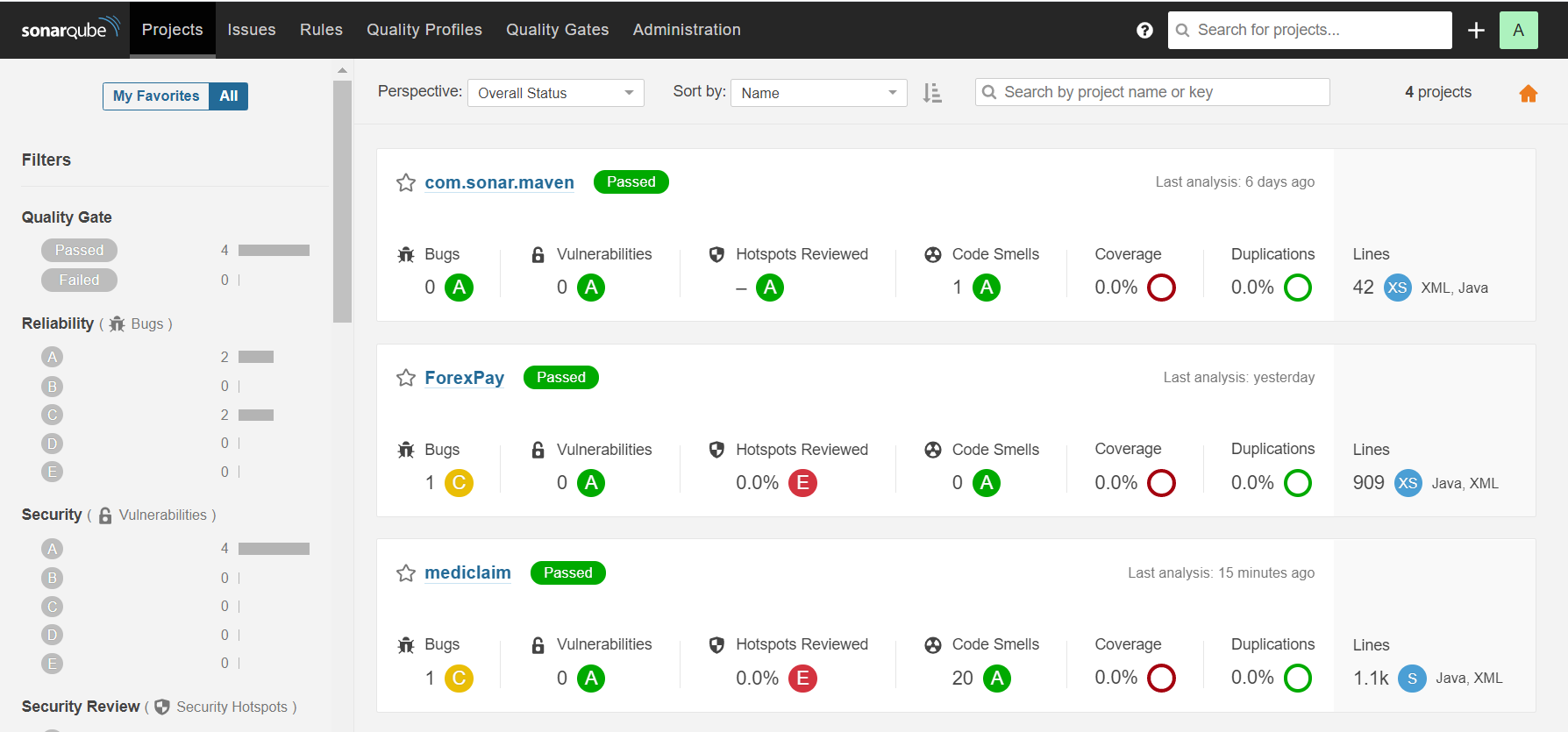
}

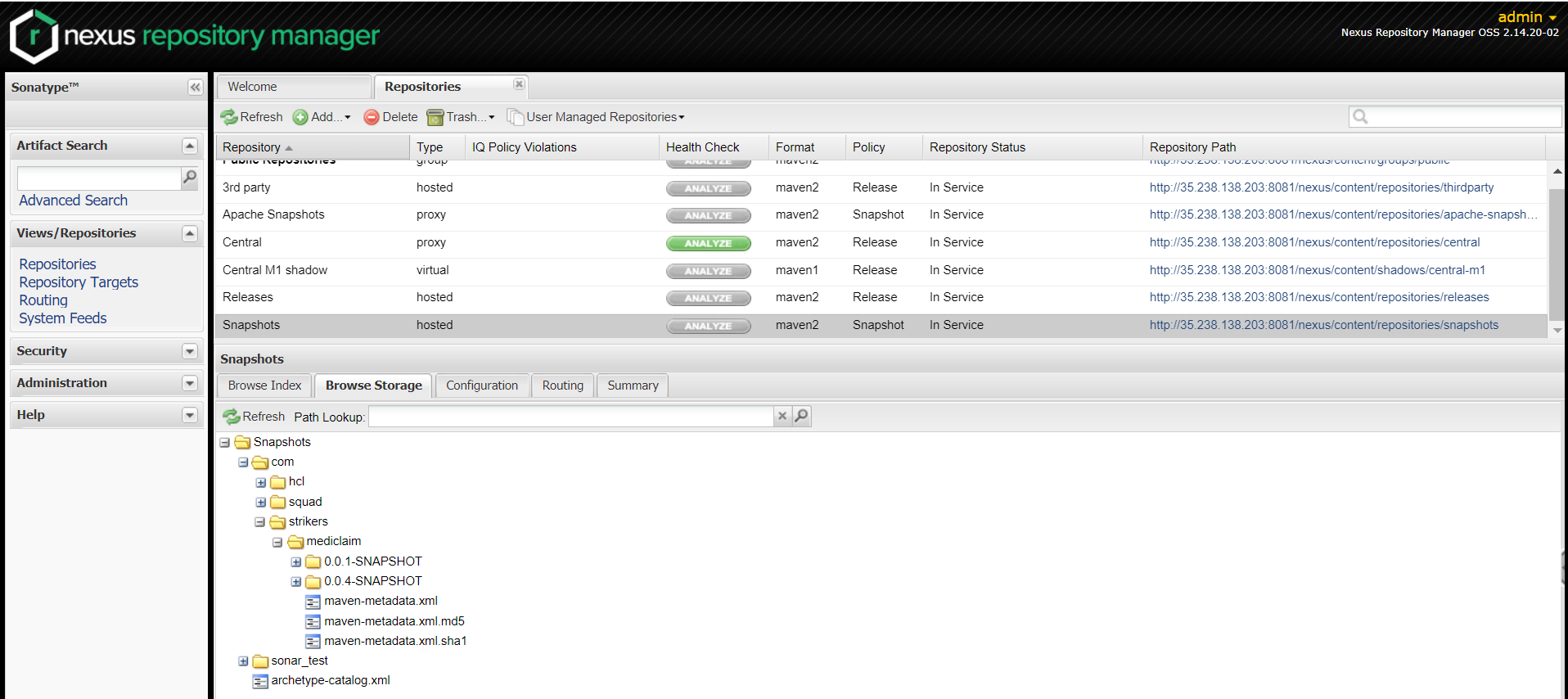


Click on save and apply

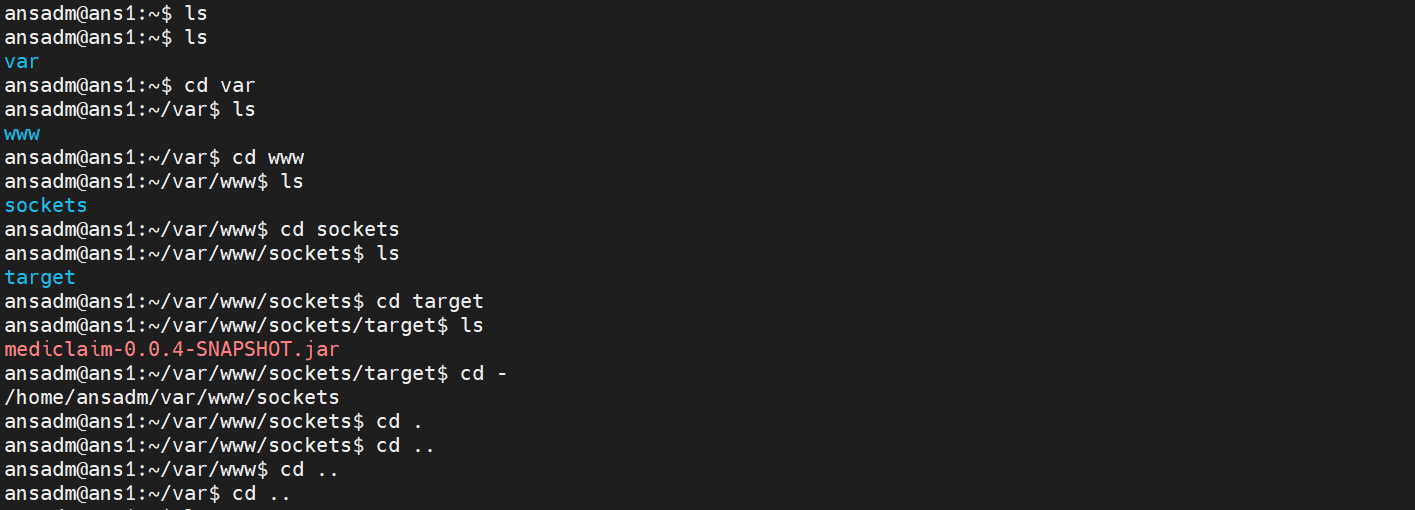






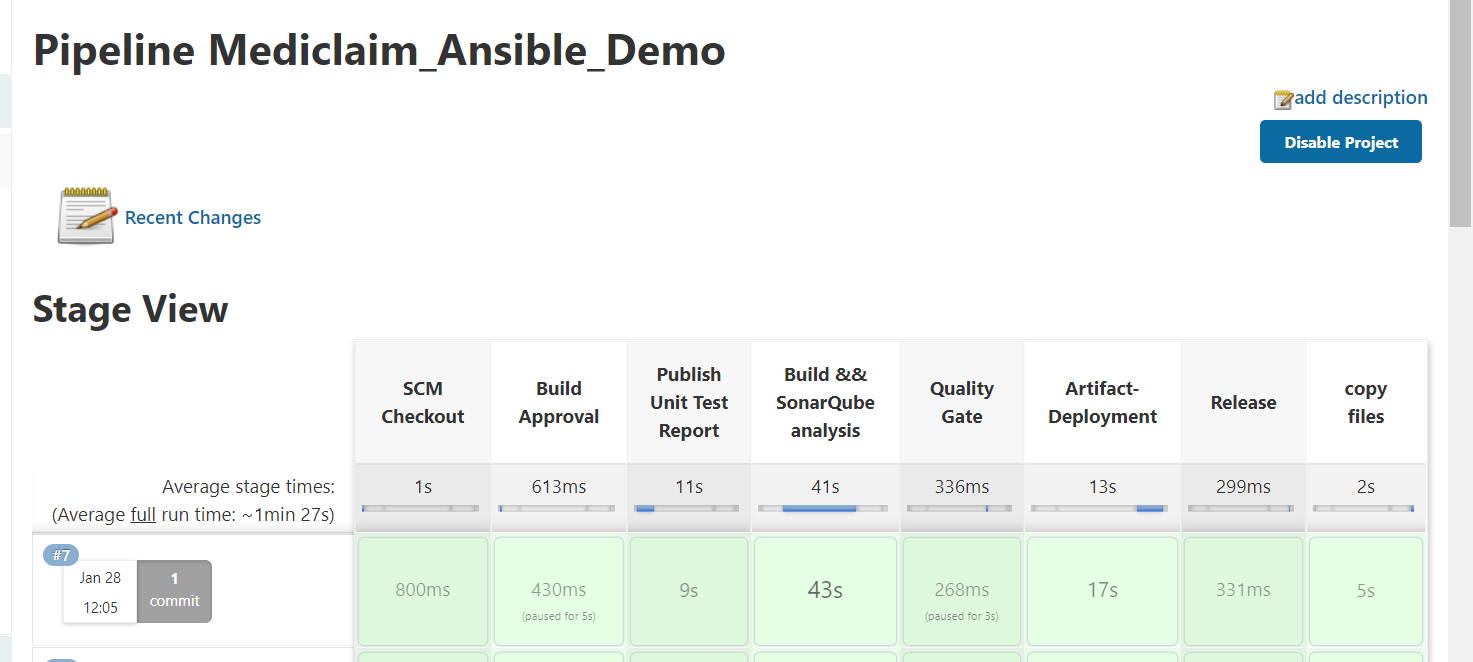


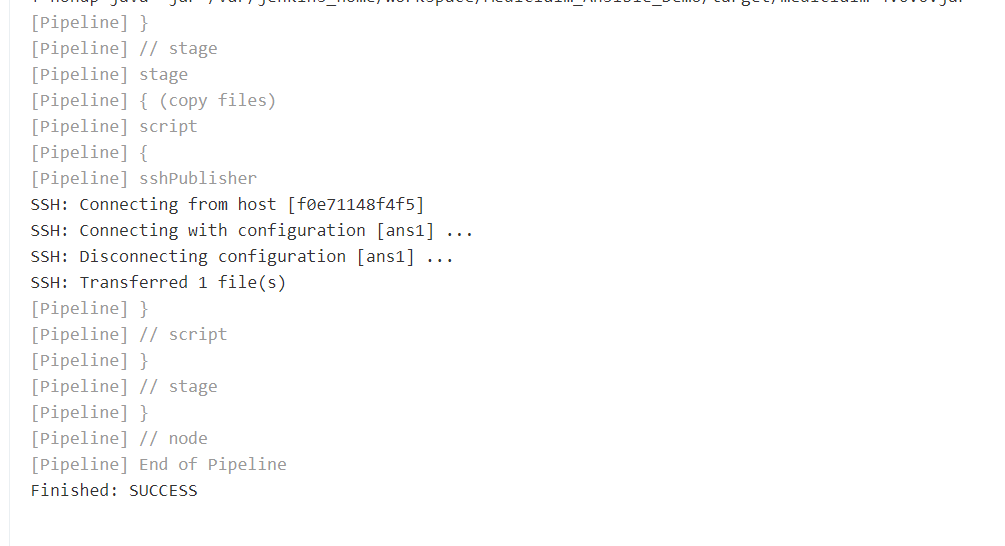
This is in the slave machine that is added in the configuration system



Now we are going to deploy a .jar file of in release time for that we need to change the pom.xml in snapshot to release

Go to your job again re build it.





We can see the mediclaim-4.0.0.jar file in target folder of slave machine

