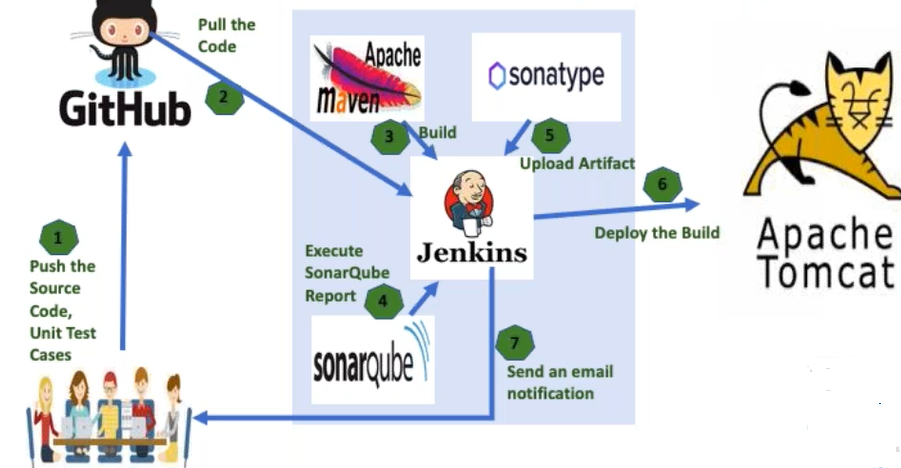
**JENKINS -Pipeline Project**

We can crate pipeline project two ways

1. Scripted Way

2. Declarative Way



**Benefits of pipeline project:**

We can customize CI/CD flow our own way and we can integrate many tools like Anisible,cheff,docker etc

It is very flexible compare to free style project

**Scripted way**

Start with node, in between whatever script going to write inside node flower bracket ({ ) and flower bracket close(}) this node is going to represent

By default Jenkins we are using going called as “**master**“

By default, it is going to run in the master node only.

node

{

}

(OR)

node(‘master’)

{

}

1. Each step going to use as stage

* Get the code from the github repository

node

{

stage('CheckOutCode')

{

git credentialsId:'11ce487b-58ae-4f93-be6e-a05064eee261', url:'https://github.com/sunildevops77/maven.git'

}

stage('Build')

{

sh "mvn clean package"

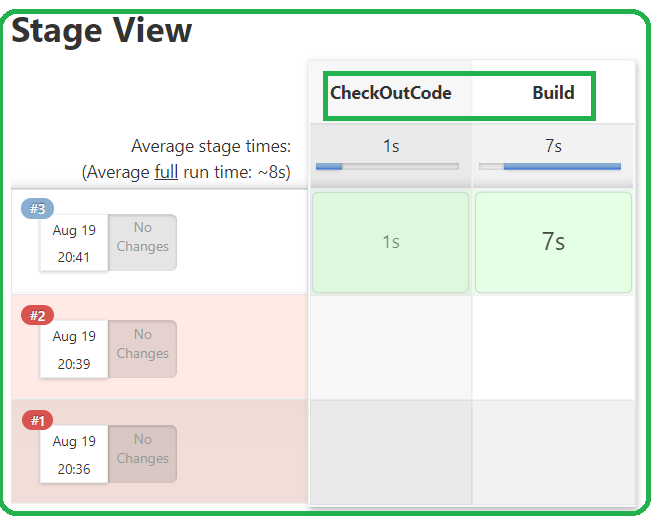
}

}

**Note:**

If Jenkins running on Linux machine, we have to use ***sh***

If Jenkins running on window machine, we have to use ***bat***

******

**Possible error in build stage**

1.groovy.lang.MissingPropertyException: No such property: ‘CheckOutCode’ for class: groovy.lang.Binding

**2.Script return exit code :127**

# mvn : Command not found

**Solution:** Jenkins not install in Linux server it was install in Jenkins server

Install under below path(maven home directory)



Command is available under ‘**bin’**



To get maven home directory to bin path for that one we need to defined variable in script level

We can declare the variable using ***def***

def <anyName> = tool name: < ***Whatever name we have configure tool name in the Jenkins (global tool configuration***)

def mavenHome = tool name: “**maven3.6.2**”

**tool name:** This key word, exactly we have to give.

This variable ***mavenHome*** contains

**/var/lib/jenkins/tools/hundon.taks.Maven\_MavenInstallation/maven3.6.2**

Example:

node

{

stage('CheckOutCode')

{

git credentialsId:'11ce487b-58ae-4f93-be6e-a05064eee261', url:'https://github.com/sunildevops77/maven.git'

}

stage('Build')

{

**def mavenHome = tool name: “maven 3.6.2” //Local Declaration**

**sh "${mavenHome}/bin/mvn clean package"**

}

(To get ***mavenHome direcoty*** path)

}

Example:

node

{

**def mavenHome = tool name: “maven3.6.2” //Global Declaration**

stage(**'CheckOutCode'**)

{

git credentialsId:'11ce487b-58ae-4f93-be6e-a05064eee261', url:'https://github.com/sunildevops77/maven.git'

}

stage('**Build'**)

{

**sh "${mavenHome}/bin/mvn clean package"**

}

(To get ***mavenHome direcoty*** path)

stage(‘**ExecuteSonarQubeReport**’)

{

sh “${mavenHome}/bin/mvn sonar:sonar”

}

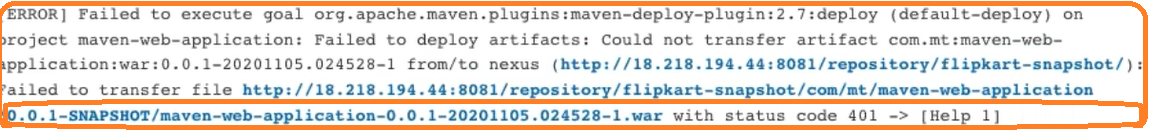
stage (‘**UploadArtifactIntoNexus**’)

{

sh “${mavenHome}/bin/mvn deploy”

}

**401 error code**

****

* Un-authorized exception
* We need verify respective server credential configure correctly or not in setting.xml file.
* Here, we are storing artifacts in nexus server in this case we need to configure nexus server details in setting.xml file

***<server>***

***<id>nexus</id>***

***<username>admin</username>***

***<password>admin</password>***

***</server>***

**Note:**

Artifact what bases it will go to ***SNAPSHOT or RELEASE*** repository?

Based on version tag, in pom.xml file

*If version tag contains key word like SNAPSHOT, then it will going to upload*

*in to the SNAPSHOT repository.*

***<version>0.0.1-SNAPSHOT</version>***

If version tag contains without SNAPSHOT key word and only version number then it will be going to upload in to the release’s repository.

***<version>0.0.1</version>***

**Deploying application in TOMCAT server**

Deploy application nothing but copying .war file into TOMCAT server(webapps)

Directory

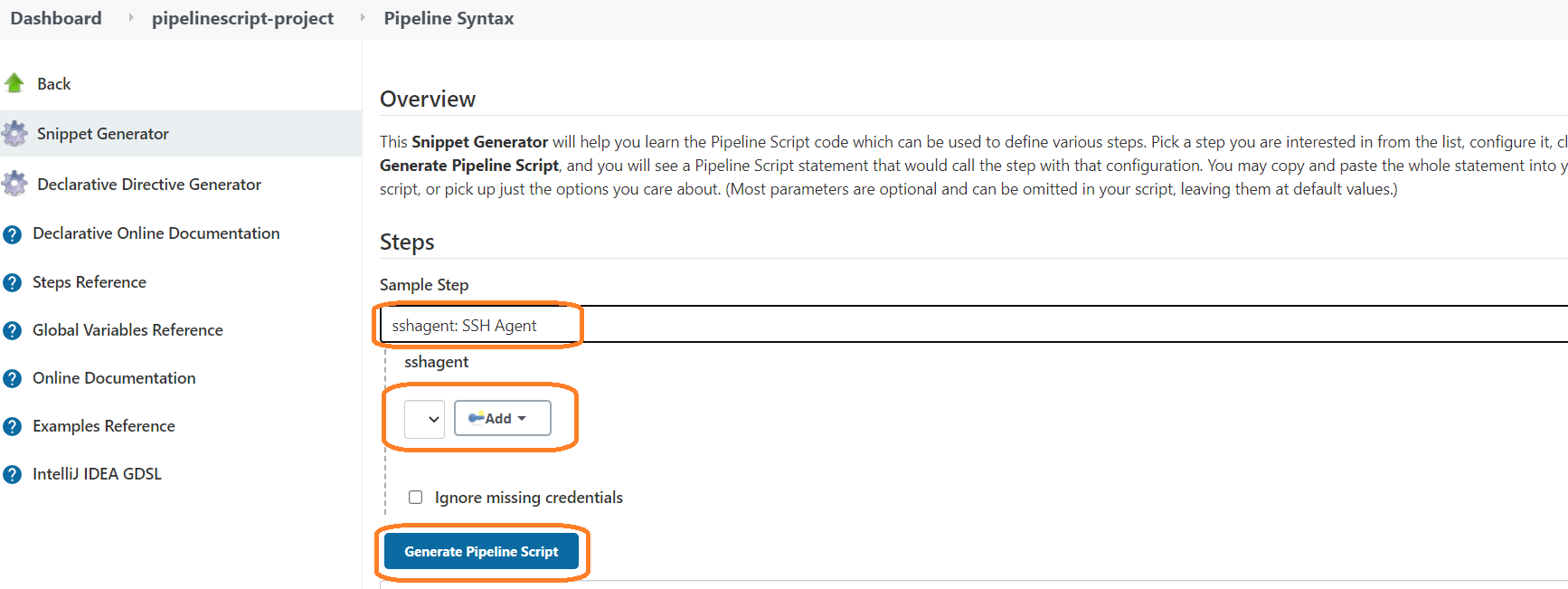
.war file available in Jenkins server and TOMCAT is different server

In order go copy file from one server to another server we use SCP command

While executing command scp command will ask server 2 credentials and Jenkins wont remember server 2 details for that we need to install plug called

SSN-Agent plugin

SCP <FileName> username@hostname: /<destination path>



Once SSH-Agent plug install done, we could see plugin name in drop down list

We have to select plug in and need to give server-2 credentials (i.e need to upload .pem file and ) and generate pipeline script

We need to update generated pipeline script in deployment stage

*sh "scp* ***-0******StritHostKeyChecking=no*** *target/maven-web-applilcation.war ec2-user@<publicIP of TOMCAT>:/opt/apache-tomcat-9.0.39/webapps/*

***-0******StritHostKeyChecking=no:* it wont ask are you sure want to continue or not.if we give.**

**stage(‘DeployeApplicationToTomcat”)**

{

sshagent([‘<token>’])

{

sh "scp -0 StritHostKeyChecking=no target/maven-web-applilcation.war ec2-user@<publicIP of TOMCAT server>:/opt/apache-tomcat-9.0.39/webapps/

}

}

**Sending Mail**

Stage(‘SendEmailNotification’)

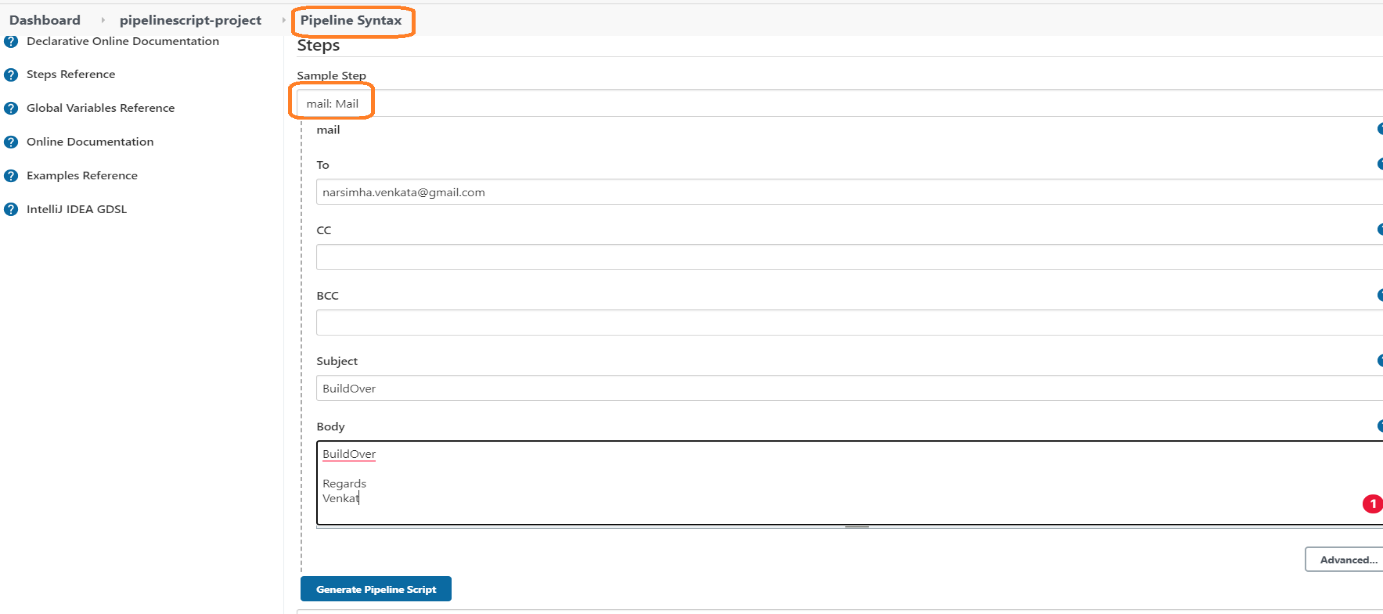
{

mail bcc: '', body: '''BuildOver

Regards

Venkat''', cc: '', from: '', replyTo: '', subject: 'BuildOver', to: 'narsimha.venkata@gmail.com'

}

****