***1. create password less authentication with ssh-keygen***

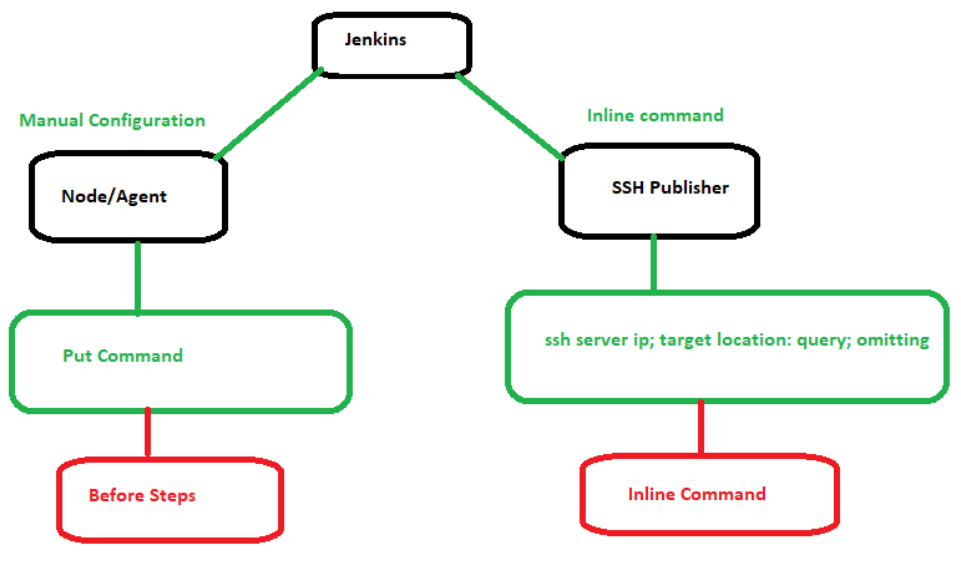
Master (Jenkins server) : 192.168.56.101  
Remote server : 182.48.70.84

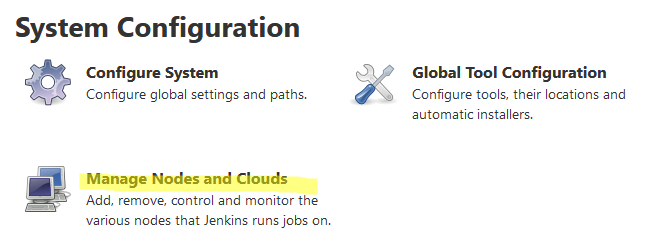
Login master server and run following command:

*ssh-keygen  
 [enter]  
 [enter]  
 [enter]  
  
 cd ~/.ssh  
 ssh-copy-id -i id\_rsa.pub* [*root@182.48.70.84*](mailto:root@182.48.70.84)  
  
Now check the login from master to remote host  
 *ssh 182.48.70.84*

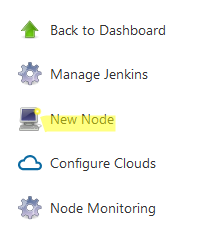
in remote server that file locate with this name:  
 *cd ~/.ssh  
 cat authorized\_keys*

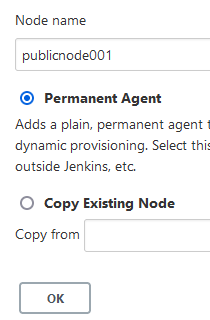
**2. there are 2 ways 2 login in remote server from jenkins**

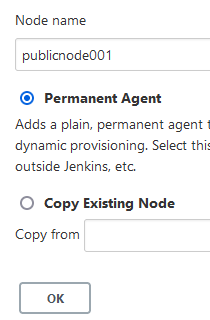
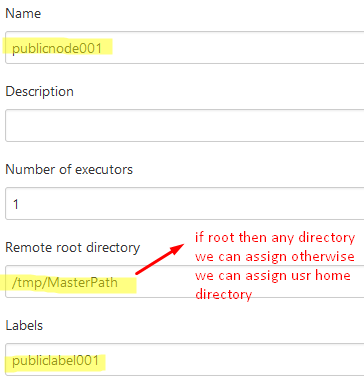
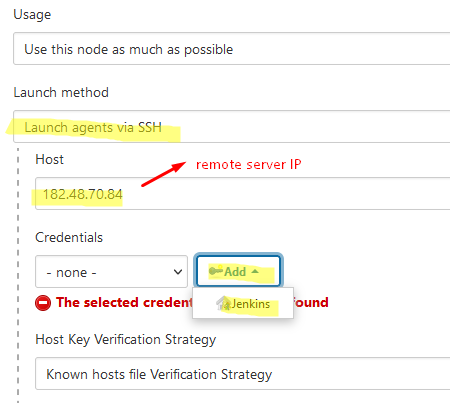
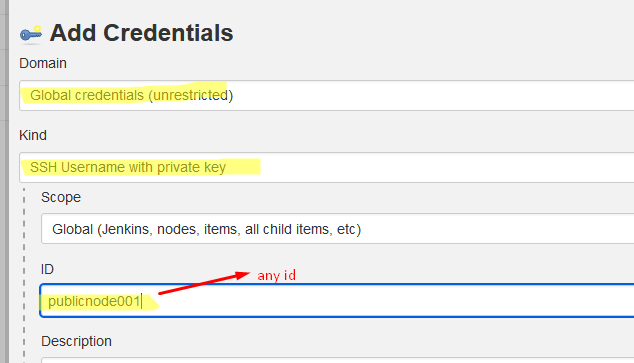
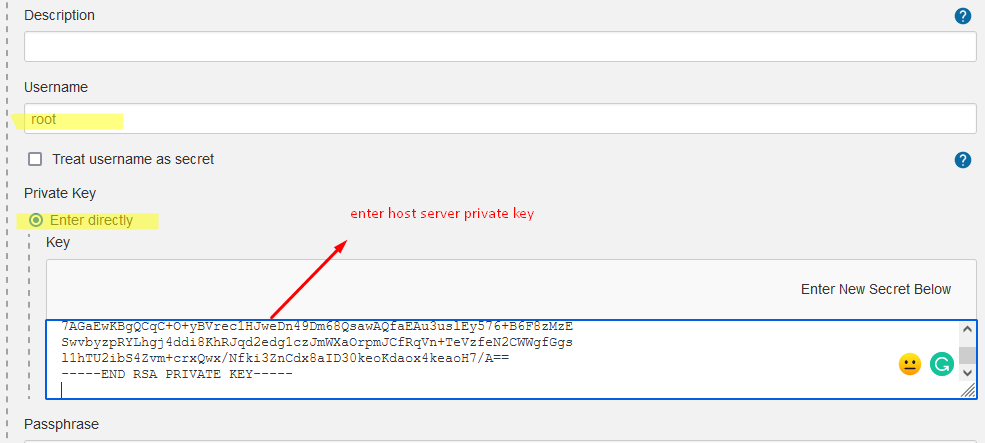


***3. Add node***   
Manage Jenkins -> manage nodes and cloud -> new node   
**name** : node001  
**permanent agent**  
**remote root directory** : /home/mehedi (if other user then give home directory, for root user, we can provide any path)  
**labels**: suppose 10 project is running. Every project seems use 10 nodes/servers. Every node is with same label *label001*. If anything change in label001 then all node will get the changes  
**usage** : use this node as much as possible  
**launch method** : launch via ssh  
  


NEXT >>>>>>>>



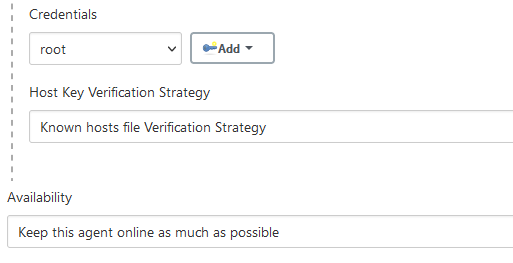


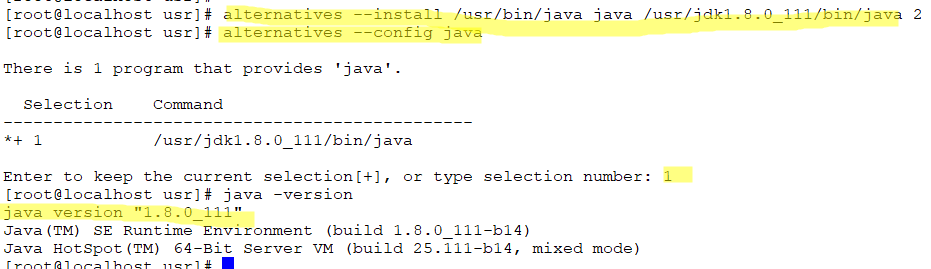
Collect private key from host machine:

cd ~/.ssh

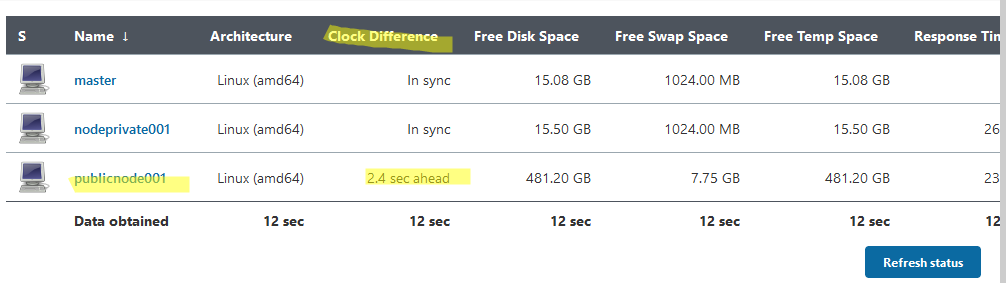
id\_rsa id\_rsa.pub known\_hosts

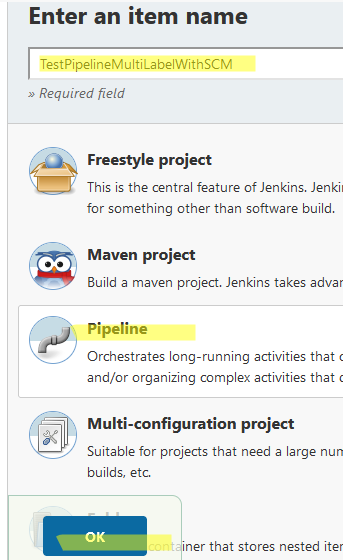
id\_rsa -> private key  
  


***\*\*\*In remote server, must need to install java 8 or above***

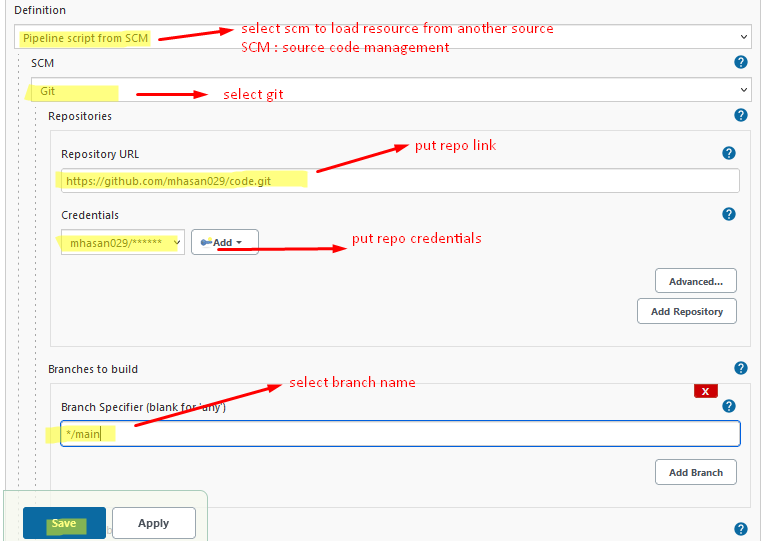
If jdk location : /usr/jdk1.8.0\_111  
then run this command:  
alternatives --install /usr/bin/java java /usr/jdk1.8.0\_111/bin/java 2  
alternatives --config java (press 1)  
java -version  
  


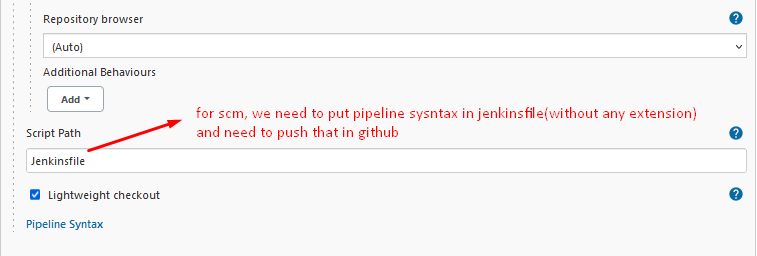
**If connect successfully then it will show like this:**



***4. Now create one pipeline:*** Put the pipeline Project name here   


After creating pipeline project have to configure it .

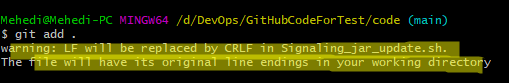




5. task is to write a script, which will stop switch first, then update jar and start switch  
 5.1 script : Signaling\_jar\_update.sh  
 5.2 Jenkinsfile need to edit

|  |
| --- |
| pipeline{  agent none  stages{  stage("memeory of 102 server"){  agent {label 'lebelprivate001'}  steps{  sh 'ip addr'  sh 'free -m'  }    }      stage("Signaling updated with new jar"){  agent {label 'labelpublic001'}  steps{  sh 'chmod 777 /tmp/MasterPath/workspace/TestPipelineMultiLabelWithSCM/Signaling\_jar\_update.sh'  sh 'sh /tmp/MasterPath/workspace/TestPipelineMultiLabelWithSCM/Signaling\_jar\_update.sh'  }  }  }  } |

5.3 need to put Jenkins file, signaling jar and script in laptop git clone directory  
 5.4 then need to push git hub  
 -if get this error:  
 LF -> Linux format  
 CRLF -> Windows format



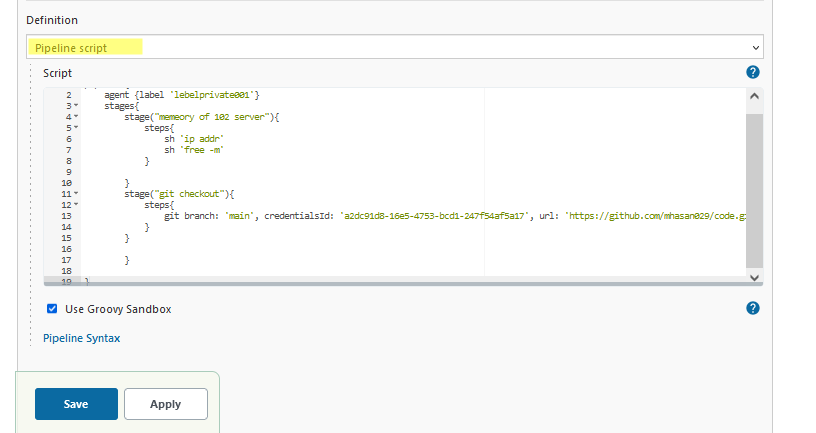
For troubleshot, we need to set this value clone wise or global wise:

|  |
| --- |
| #git config --list | grep crlf  core.autocrlf=true ///this is result (to check current status)  #git config --global core.autocrlf false //this is for global setting  #git config --global core.eol lf //this is for global setting  Now you can also switch single repos to crlf (in the working directory!) by running  #git config core.eol crlf |

5.5 Build the pipeline project  
 -Please note if we up service sh command then it will auto stop, need to run service with service command

**###Same task we can do over inline pipe script:**

1. Configure the pipeline
2. Select pipeline syntax



And put scrip:

|  |
| --- |
| pipeline{  agent none  stages{  stage("memeory of 102 server"){  agent {label 'lebelprivate001'}  steps{  sh 'ip addr'  sh 'free -m'  }  }    stage("git checkout to 102 server"){  agent {label 'lebelprivate001'}  steps{  git branch: 'main', credentialsId: 'a2dc91d8-16e5-4753-bcd1-247f54af5a17', url: 'https://github.com/mhasan029/code.git'  }    }        stage("Signaling updated with new jar"){  agent {label 'labelpublic001'}  steps{  sh 'chmod 777 /tmp/MasterPath/workspace/TestPipelineMultiLabelWithSCM/Signaling\_jar\_update.sh'  sh 'sh /tmp/MasterPath/workspace/TestPipelineMultiLabelWithSCM/Signaling\_jar\_update.sh'  }  }  }  } |

3. build and check