git config --global user.name "Your Name"

git config --global user.email [youremail@yourdomain.com](mailto:youremail@yourdomain.com)

git init [directory] – (initializing the git repository) create a Git repository from an existing directory

ls – what are the files in the working directory (untracked)

git status – what are the files in the staging area (tracking)

git ls-files – what are the files in the local repo (tracked)

git add – copy the file from directory to staging area

git commit -m “completed” – to commit the file and it will be in local repo( type however you want ‘ex – completed’)

git log –all details like who has created the file, how many times he committed, how many files he comitted, what are the changes he made

git show + commit id – all the commits will be shown

git log -- oneline – Recent action i.e recent committed one will be on top of the history

git tag -a “HTMLfile” +commit id m “adding tag for 1st commit” – to add tag( small ‘a’ means attribute and there should not be any space in attribute

git stash - do not allow to track the changes made(statements) in the existing file which is committed already(i.e stash will hide the new statements or changes of the committed file or existing file

git show - Outputs metadata and content changes of the specified commit

git stash list – shows the all stash ID

git show + stash id – shows which are all changes in stash

git stash pop + stash id – to revert back the stash id

git stash drop + stash id - The command deletes the specified stash( when you have stash conflict, stash will create a backup for lastest stash id, manually you have to drop the backup.

git ignore – .gitignore file tells Git which files to ignore when committing your project to the GitHub repository ( Ex: you created new file say “index.hmtl” and you dont want to track this new file “index.hmtl”. For that again you create a new file .gitignore and in that .gitignore, you put the file “index.hmtl” which you dont want to track.

git rm -- cached +file – To delete the file from the local repo but not from the working directory( if u don’t want the file which is already committed then you use this formula, if you don’t mention the “--cached” the file will be deleted permanently).

git branch – list all the branches in the repository

git checkout -b +branch name – to create a branch( when you are creating the branch, you should be in master otherwise new branch will take previous branch as master, so switch over the master and create a new branch)

git checkout master – to switch over the master

git merge +branch name – to merge the branch file in the master ( while merging the branch in the master also, you should be in master )

**Note:** once merging one file from one branch into master, we cannot merge the file with same file name from another branch into master coz merge id conflict will be there. So, we rename the second file name then the edited name file will merger into master under existing merge id automatically(manually we should not merge the renamed file coz master will take the renamed file and deleted the old file, we should not even the commit the renamed file we should go master and commit)

cat + folder name = This command displays says that “the file is created by whom”, it shows the statements of the file

mv filename1 filename2 – to change or replace the filename1 to filename2

git merge master – merging the master from the branch

git update master – to get update from the master

**7th session:**

In real time, to merge the branch file into the master, we need to raise the PR ticket. Reviewer review the PR ticket and merge the file into Master.

Master file should be in Remote repo only and Branch can be in local repo Remote repo

git remote add origin +github link – to put the master into the remote

git push -u origin +branchname - To push into the remote repository i.e Github (while you are pushing the branch to remote, you should be in branch )

**Note:** when you are using remote repo i.e from local repo to remote repo, we have four commands i.e…

git fetch origin +branchname – if something changes in remote repo (will give metadata from remote repo, we have to use this command in local repo)

git push – Pushing the file from local to remote (command should be used in local repo)

git clone – The git clone command is used to create a copy of a specific repository or branch from different project within a repository. Cloning the entire file or repo from remote (command should be used in local repo)

git pull origin +branchname – will pull modified files from remote repo (command should be used in local repo)

Pull request – Creating PR(Pull request ) ticket for merging the code from branch to Master( this will be done in Remote Repo)

Task : git rebase ( why we use this command, not much use in real time, for interviewer purpose )

**8th session:**

1. Jenkins – we have to configure the cluster [one master (Jenkins Master) and rest (Jenkins slaves) will be the slave nodes].
2. Jenkin Master only will have Dashboard 8080 and slaves will not have any Dashboard, slave will follow the Master.
3. First provisioning the Master (in aws). We need to launch the EC2 server i.e Linux virtual instance (virtual machine) for provisioning the Master.
4. Termius(Terminal) – download Termius (Remote Application) and Install - <https://termius.com/download/windows> [ Putty, SecureCRT, MobaXterm, ~/.ssh/config are Terminal tools ]
5. Create a New Host – Paste Public address of launched EC2 server in the Address box on right top – put some name in Label – Username & Password and then launch the Host.

For AWS Linux, ubuntu, Redhat –

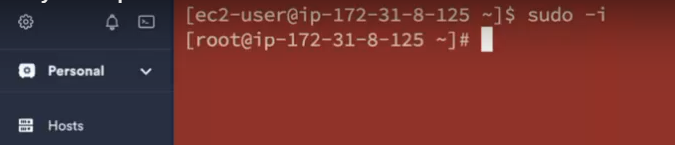
Username: ec2-user

Password: downloaded key pair

1. The below Yellow shaded is private ip ( AWS recommended create and use Host name for private id ) that we can see in AWS under EC2 launched instance.

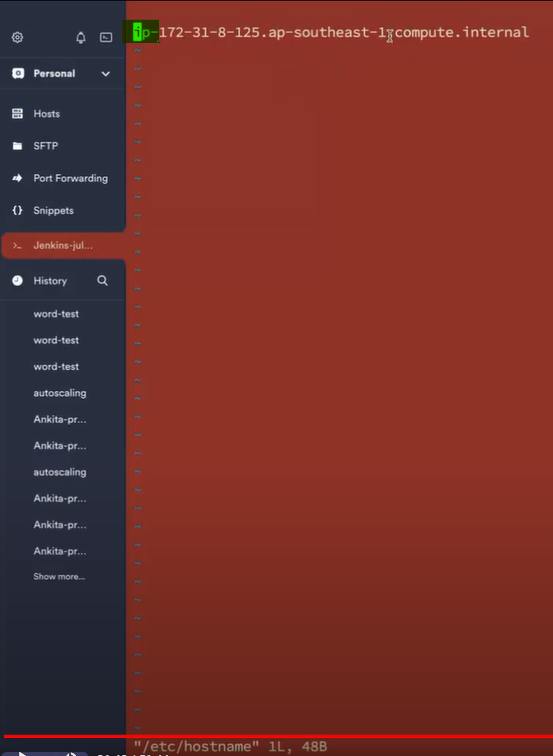


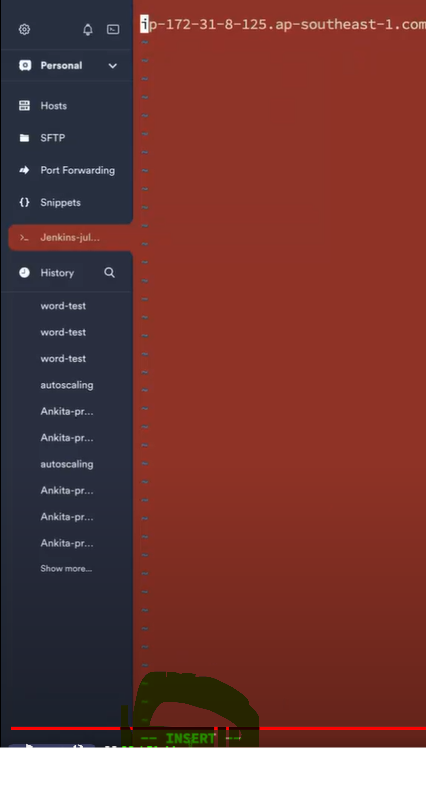
sudo lets you run commands in your own user account with root privileges. su lets you switch user so that you're actually logged in as root. sudo -s runs a shell with root privileges. sudo -i does this as well, but also acquires the root user's environment.



**Note:** IP address in binary code is taken default host name, we need to define the Host name. $ - it means normal user where as # - means root user.

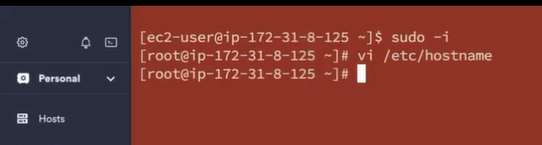
1. How to define the Host name – command is “ vi /etc/hostname “. vi means vim editor. Then press i to make the status to “INSERT” mode and you can define the Host name.





Once it becomes INSERT MODE, then type the Host name whichever you want.

How to save VIM editor: Press Esc + : + wq



Host name still not yet changed, use command integer vaule “init 6” to restart the instance/server. In AWS console also, you can restart the instance by clicking the option “Rebot instance”

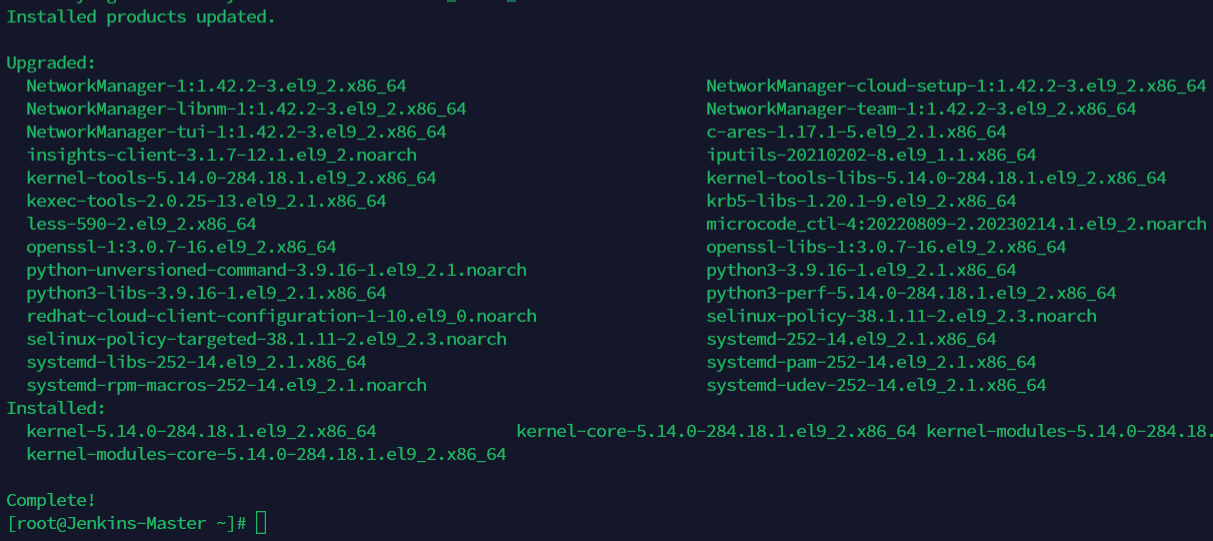
**Integer value:**

init 6 – restart

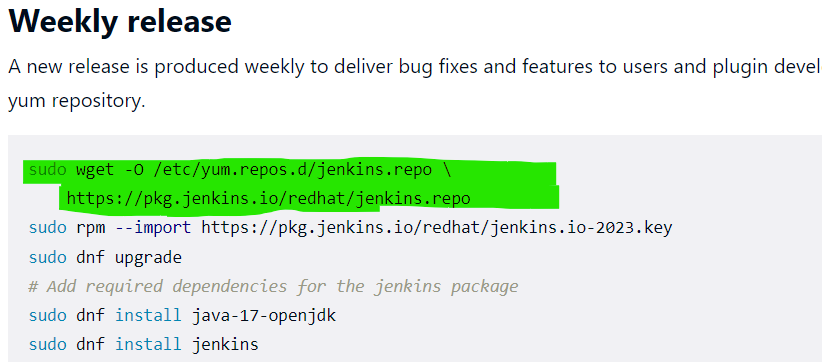
init 0 – terminate

init 1 – stop

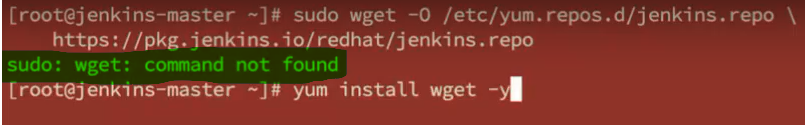
1. How to install Jenkins for Linux - <https://www.jenkins.io/doc/book/installing/linux/>
2. First update the Subscription Management repositories in Tremius. Command is “yum update -y”.
3. Once Subscription management repositories is complete then install Java.

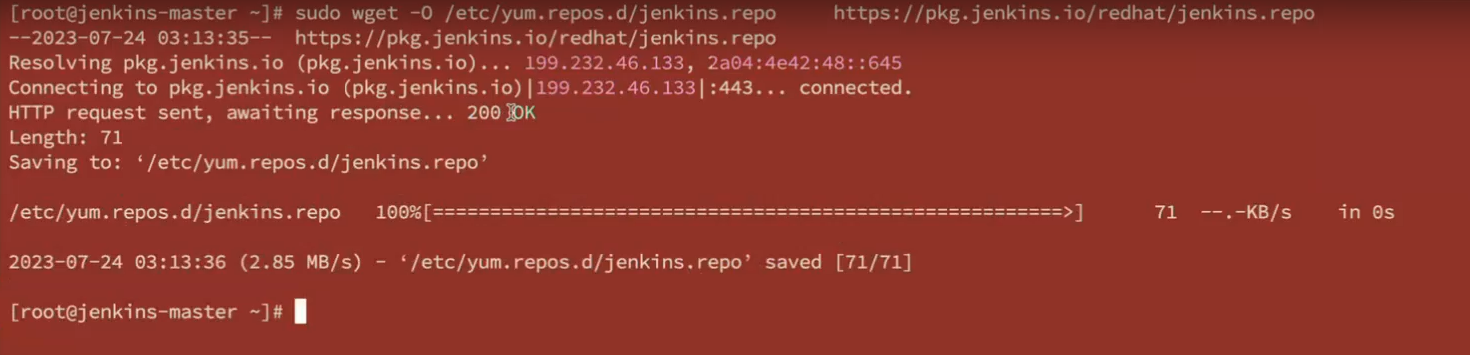


1. Copy and paste below command



1. When you paste this command, if you get like below, then install wget also. Command is “yum install wget -y” and then paste above commands

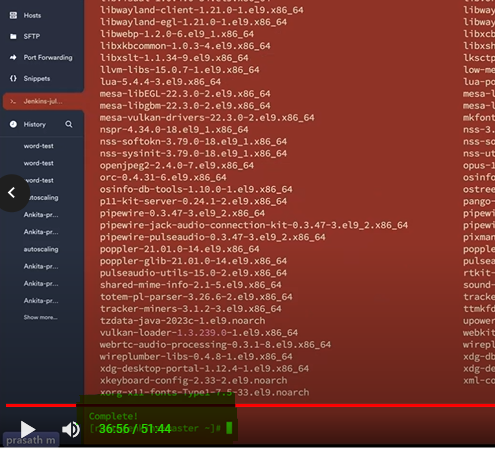




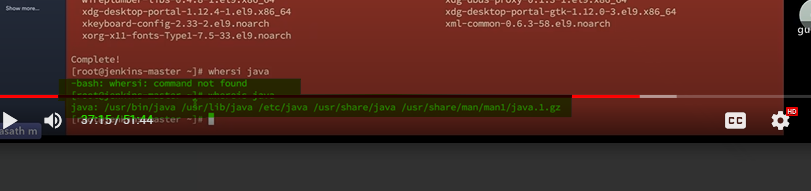
1. Once successfully completed, then paste the single single line command and enter.



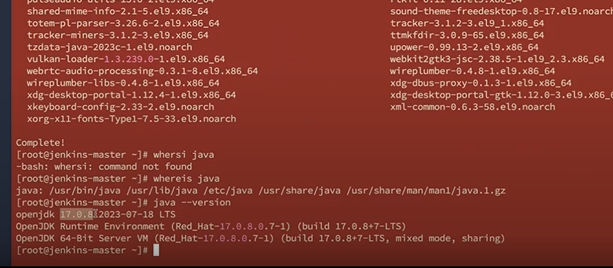
1. Lastly, Java will be installed successfully in below screenshot.



1. To see where is Java or Java path, command “whereis java”

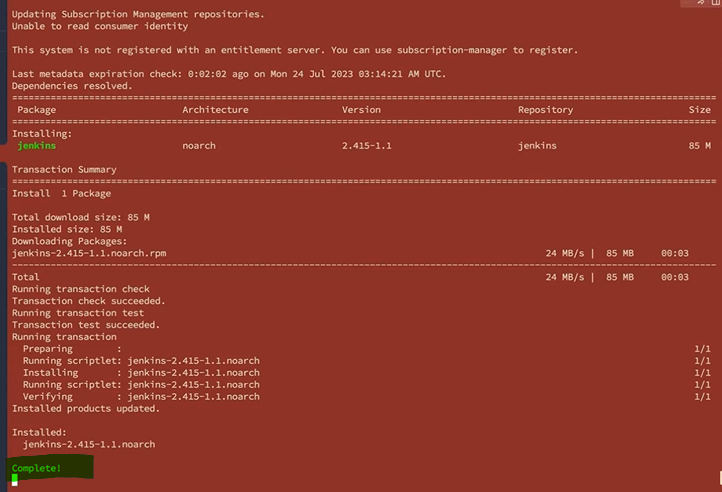


1. To see which java version, command “java --version”.

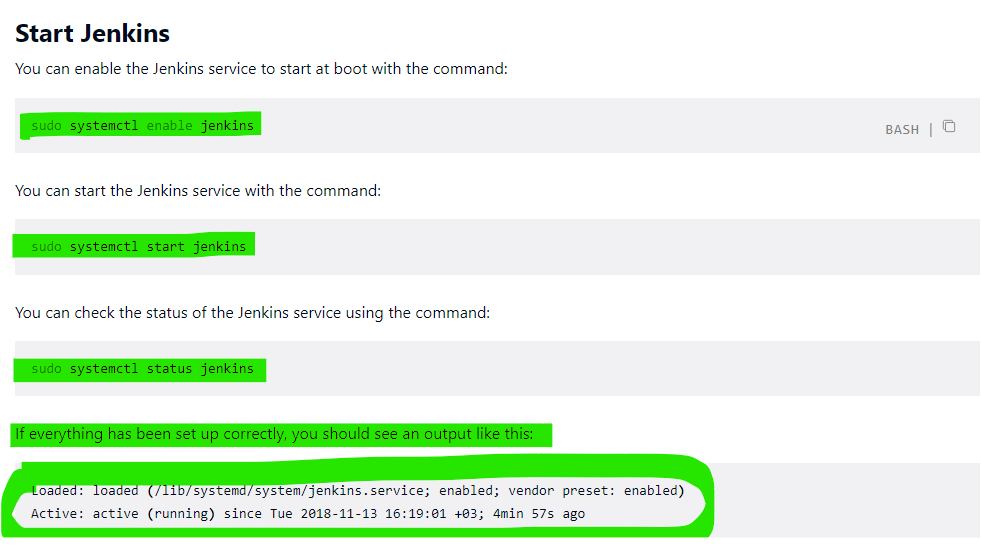


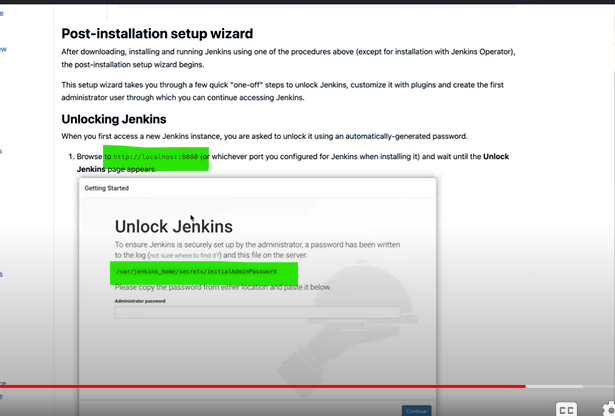
1. And Finally install Jenkins.

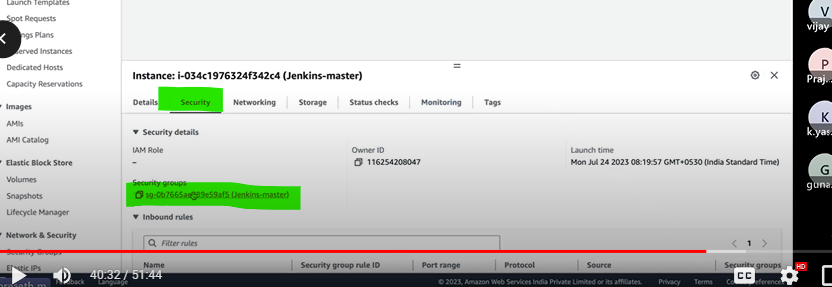


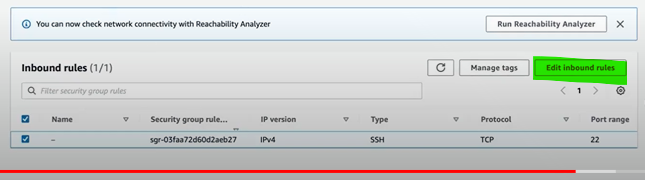


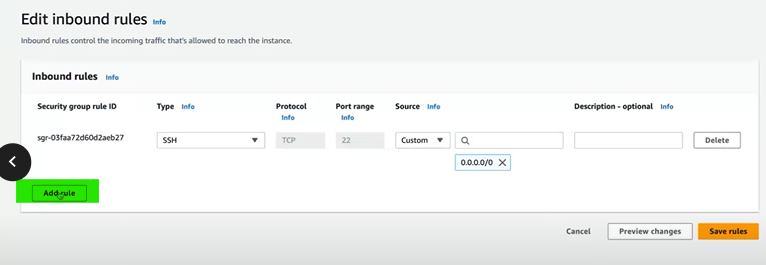
1. To see where is Jenkins or Jenkins path, command “whereis jenkins”
2. To see which Jenkins version, command “jenkins --version”.
3. Then you can start Jenkins enable using below commands

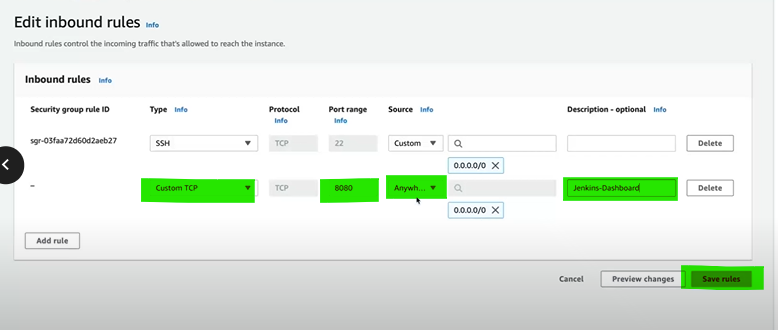


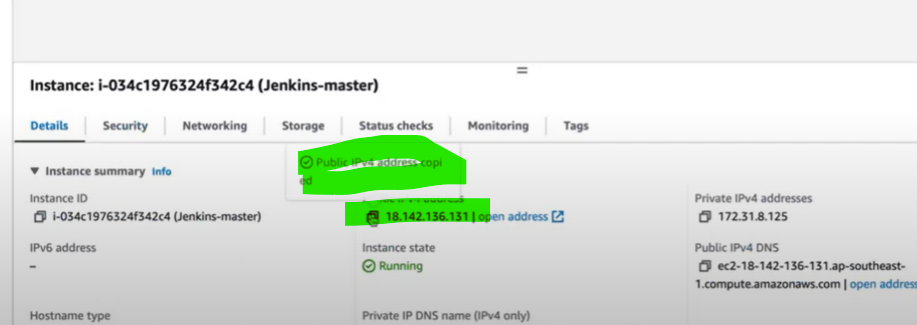


1. We need to open security firewall 8080 (see above screenshot ) in aws.
2. 

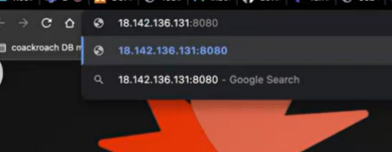




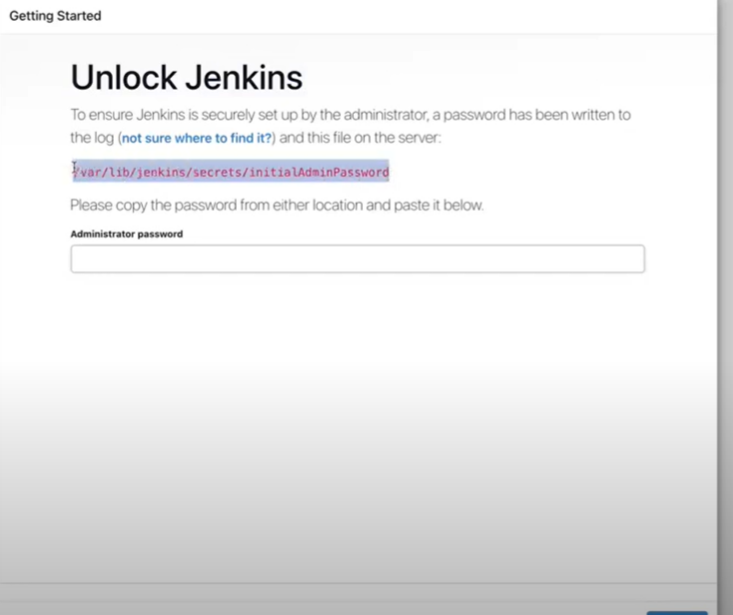


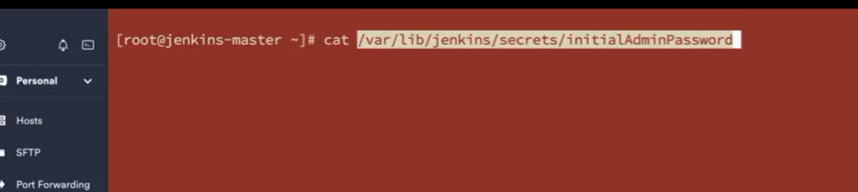


Paste the public address with port 8080 and enter, you will get Jenkins dashboard like below.



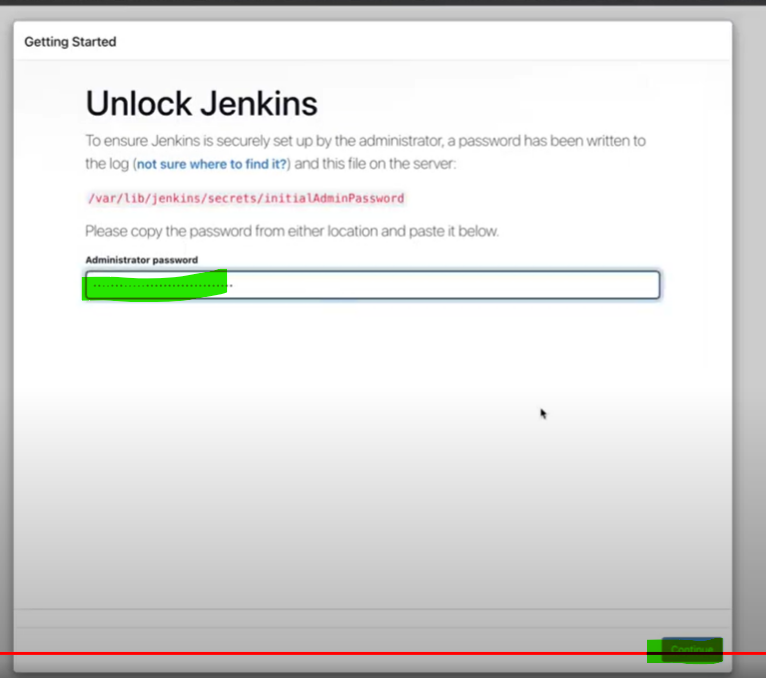
Jenkins dashboard, paste the link which is initial password in termius(command – cat +link), later you can reset the password



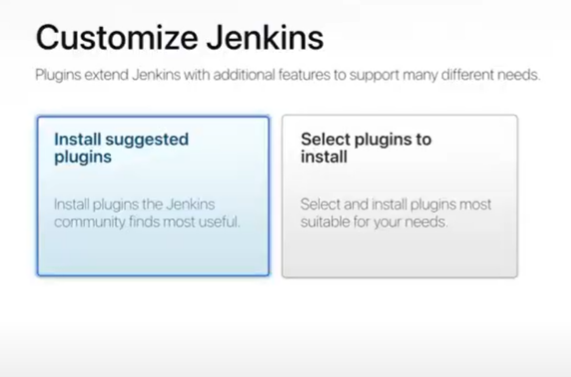




Copy above screenshot command and paste it in the administrator password box like below and continue.



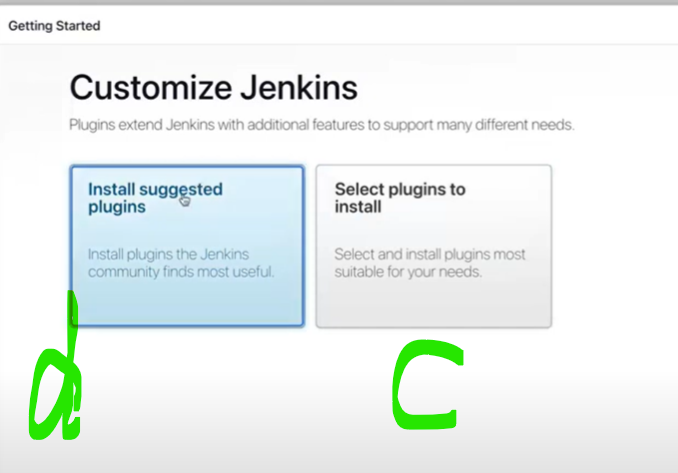
Then you will get like below. Finally, Jenkins has been installed.



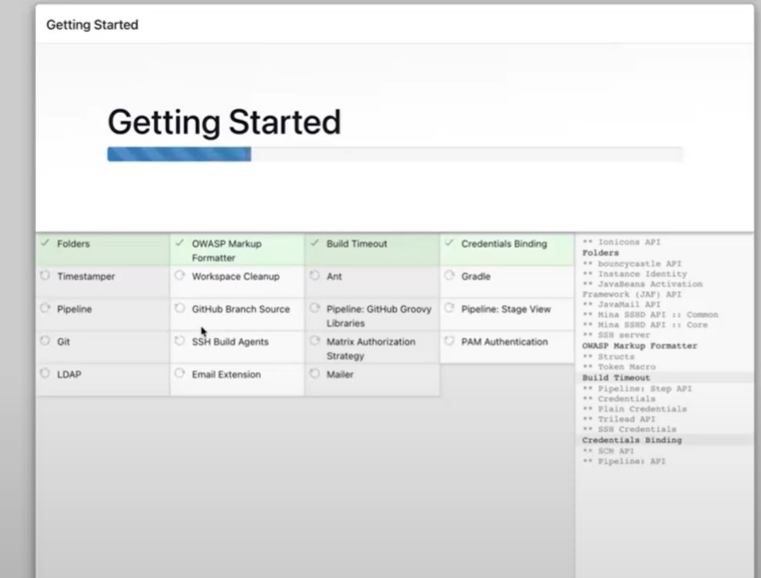
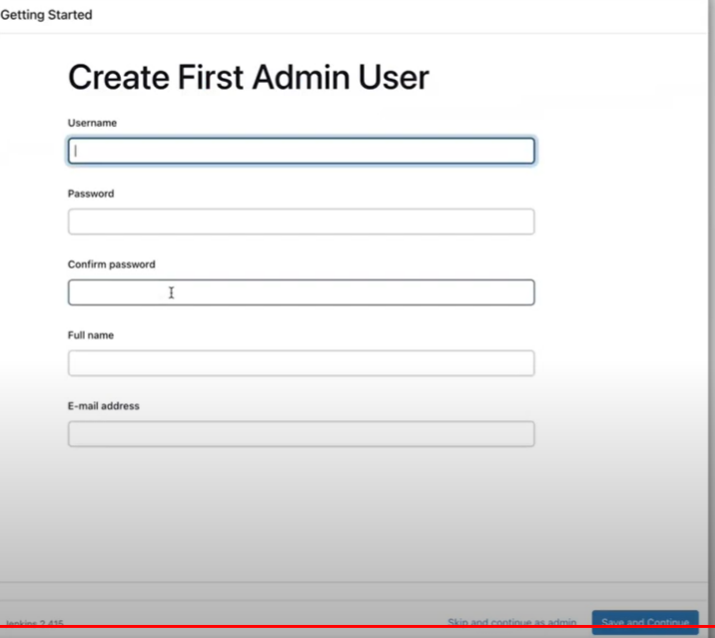
Jenkins is just a platform, if we need to perform in Jenkins, we need plugins which are developed by groovy language. Plugins are used to help Jenkins to communicate with some applications like “Github”.

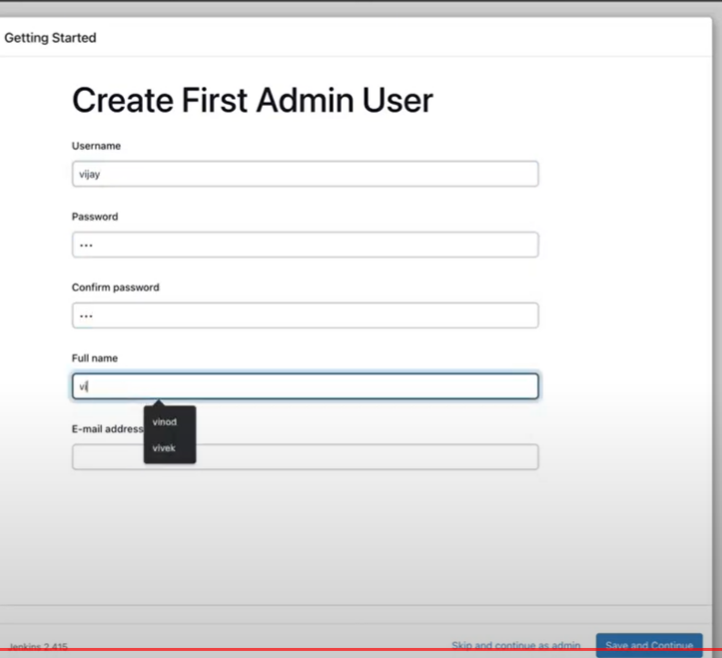
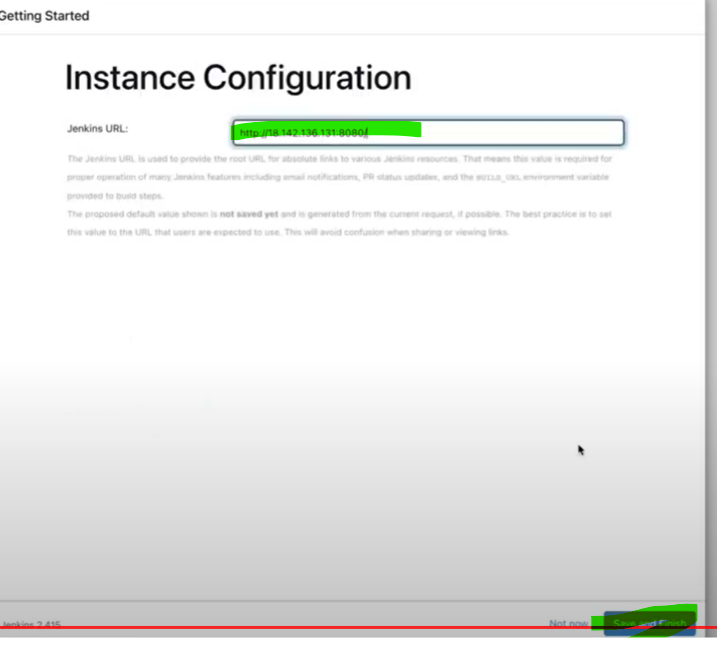
Groovy language will be developed by developer not by Devops engineer.

In Jenkins, we have both Default plugins and also Customized plugins also available.

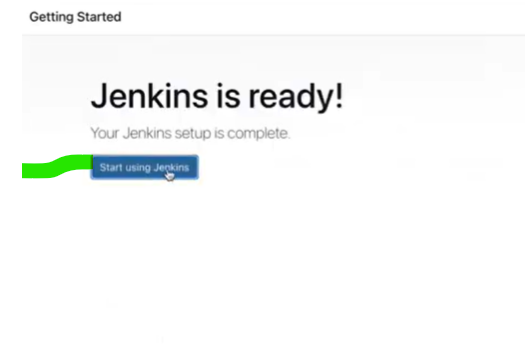


Install the default plugins like below.

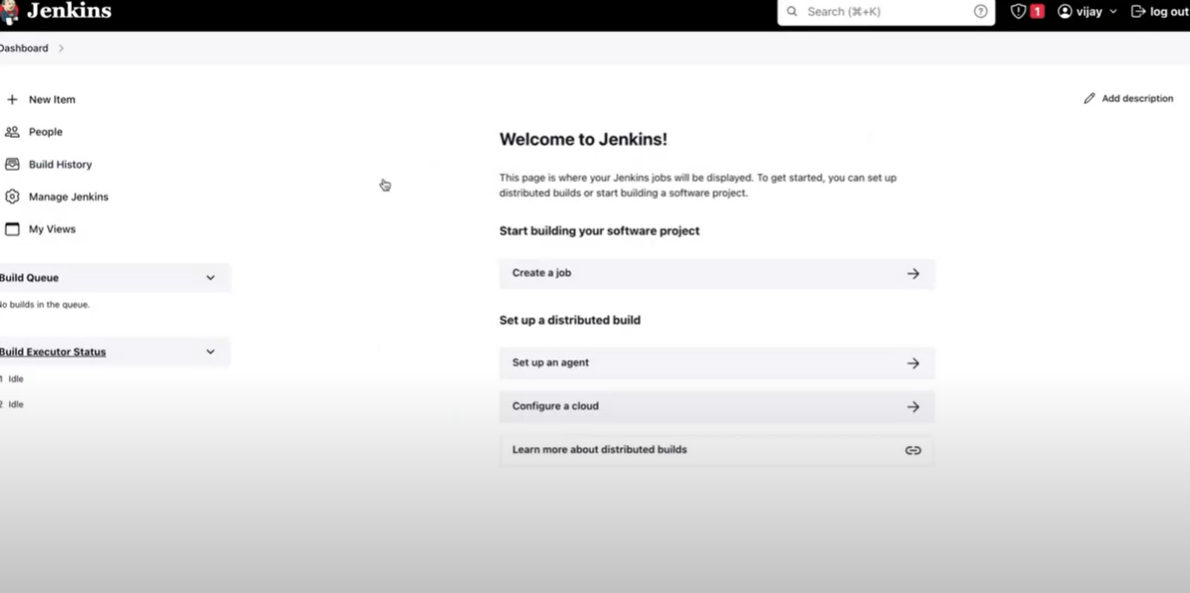
 

Jenkins Master is ready.



Jenkins dashboard.



For customized plugins, follow below steps.

