CICD (Continuous Intergration Continuous Delivery)  
A white paper with writing on it

Description automatically generated with low confidence

**Maven** is used to create Artifact(setup file/exe file)

* It has 7 stages of lifecycle. 5th stage is package.

Download oracle vm

Download vagrant

Downlaod vagrant file given by institute --🡪 copy that vagrant file and place it on desktop inside a new folder

Putty download….putty.org

1. Git download

sudo apt-get update

sudo apt-get install -y openjdk-8-jdk

java -version

git --version

mvn --version

1. Jekins download ----- wget <https://get.jenkins.io/war-stable/2.346.2/jenkins.war>

java -jar jenkins.war

<http://35.154.216.32:8080/>

tomcat - cd /etc/tomcat9

<tomcat-users>

<user username="komal" password="komal" roles="manager-script"/>

</tomcat-users>

The other way to download & use jenkins

Create new instance 🡪 New instance 🡪 select ubuntu 🡪and select all needed

Once created new instance 🡪 connect to that instance 🡪 in gitbash connect this instance 🡪install java/git/maven in new instance 🡪 now go to chrome 🡪 jenkins installation on ubuntu(<https://www.jenkins.io/doc/book/installing/linux/>)

Then follow below commands

* curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

* echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

* sudo apt-get update
* sudo apt-get install jenkins

Now copy the ip address of new instance and paste it in chrome <http://43.205.139.29:8080/> enter

Go to gitbash give command ---

sudo cat/var/lib/jenkins/secrets/initialAdminPassword and enter

now copy the password paste in chrome 🡪 install plugins 🡪 create a admin user

**CICD Process – Freestyle project**  
step1 – Login into Jenkins

Download process

Step2 – create a job (Name: Development and then select Freestyle project ----OK)(create a new item)

Step3 – Click on Source code management and then select the Git-----------------------

Repository URL: <https://github.com/intelliqittrainings/maven> (this is the location of the code which developer placed in git site) and then click on Save and then run

To check the status click on # to check the output

Step4 – cd /home/ubuntu/.jenkins/workspace/Development (to check the folders)

Once the development code is created follow the below steps – **Creating a Artificat - Build**

Step 5 – Again click on development ---🡪 select Configure --🡪select option **Build** on top menu

Step 6 – Add Build step (click on drop down) 🡪 select **Invoke top-level Maven targets** from drop down

Step 7 – Now give **package** in goals section of Invoke top-level Maven targets.

Step 8 – click on Apply & Save

To check the logs click on last success 🡪 console output 🡪 here in the bottom you can see the job is success

After this step the war/ear/jar file gets created you can see in console output screen

**Deploy to Container (QA server)**

Below steps are to install plugins if they are not installed

Manage Jenkins 🡪 Manage Plugins 🡪 Available Session 🡪 Search for **Deploy to Container 🡪 select it and select option install without restart**

Step 9 - Again click on development ---🡪 select Configure --🡪 select option **Post Build Actions** on top menu

Step 10 – click on **Add post-build action** drop down menu 🡪 select **Deploy war/ear to** **container**(this option comes only after you install deploy to container plugin) option from drop down menu

Step 11 – Give the name of the WAR/EAR files **--🡪(\*\*/\*.war**) 🡪 in **Context path** option give a name by which tester would test the code (eg: testapp)

Step 12 – Click on **Add Container** drop down 🡪 select **tomcat9 \* Remote**

Step 13 – click on credentials 🡪 Add (Jenkins) 🡪 give username and password of tomcat 🡪 click on Add 🡪 select those credentials

Step 14 – for Tomcat URL -🡪 Go to QA server and then click on QA server 🡪 Copy private IP of QA server (as all the servers like Jenkins/QA/Prod are on server we can user private IP)

Step 15 – Paste the private IP in tomcat URL option ([http://**172.31.15.17:8080**](http://172.31.15.17:8080))

Step 16 – click on Apply & Save

Java Based programs are successfully installed in QA server. How to check it

Go to QA server take its public IP 🡪 paste the public IP in URL (<http://3.110.48.159:8080/testapp/>)

**Continuous Testing**

Step 17 - Click on new item (now we will create a new item for testing) create a job (Name: Testing and then select Freestyle project ----OK)(create a new item)

Step 18 - – Click on Source code management and then select the Git 🡪 Repository URL: https://github.com/intelliqittrainings/FunctionalTesting.git (this is the location where testers are uploading the code in git site) and then click on Save and then run

Console output - /home/ubuntu/.jenkins/workspace/test

**Selenium Testing**

Step 19 – Copy the location where the code is placed i.e /home/ubuntu/.jenkins/workspace/test

Step 20 – in test job open the configuration page 🡪 go into build section

Step 21 – click on **Add Build step** dropdown 🡪 click on **Execute shell** option

Step 22 – in Command give **java -jar /home/ubuntu/.jenkins/workspace/test/testing.jar** (the location of the selenium code)

Step 23 – click on Apply & Save and then run the test job

**To link all the stages follow below process (upstream/downstream configuration)**

Step 24 – go to development job 🡪 go to configure page 🡪 go to post build action 🡪 click on **Add post build action** dropdown

Step 25 – select option **Build other projects 🡪** give the name of the project i.e **Testing**

Step 26 – click on Apply & Save and then run the development job

Step 27 – to check the logs go to development job 🡪 console output to check success

**Deploying to the PROD env**

Before Deploying to the PROD env first we need to install the plugin **Copy Artifact**

**Go to Manage Jekins 🡪 Manage plugins 🡪 search for Copy Artifact 🡪 click on install without restart option**

Step 28 - go to development job 🡪 go to configure page 🡪 go to post build action 🡪click on **Add post build action** dropdown

Step 29 – select option **Archive the artificats 🡪** give the name in Files to archive option i.e **\*\*/\*.war**

Step 30 – click on Apply & Save

Step 31 - go to Testing job 🡪 go to configure page 🡪 go to **Build** section 🡪 click on **Add Build step** dropdown 🡪 select **Copy artifacts from another project 🡪** give the project name i.e **development**

**To Deploy in prod server**

Step 31 – go to post build action section 🡪 click on **Add post-build action** drop down menu 🡪 select **Deploy war/ear to** **container** option from drop down menu

Step 32 – Give the name of the WAR/EAR files **--🡪(\*\*/\*.war**) 🡪 in **Context path** option give a name by which client will access the application i.e prodapp

Step 33 – Click on **Add Container** drop down 🡪 select **tomcat9 \* Remote**

Step 34 - select those credentials

Step 35 – Give the private IP of Prod server in URL (**http://172.31.11.91:8080)**

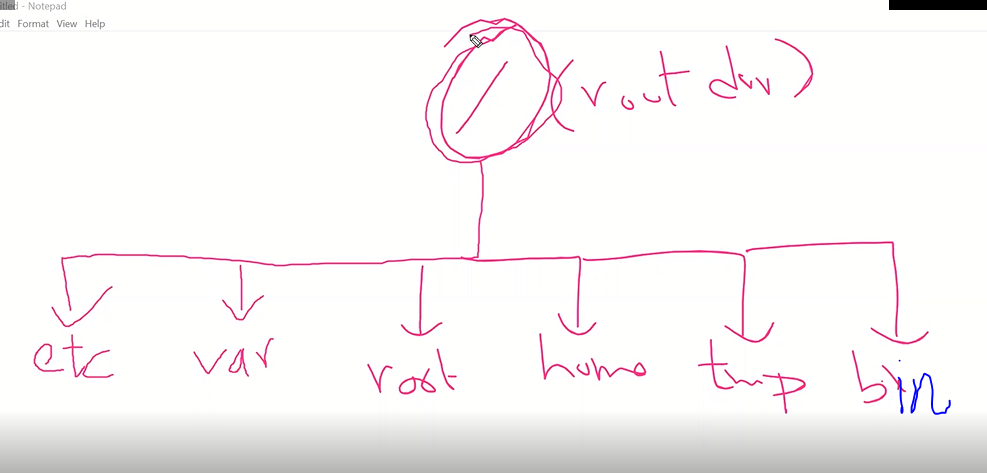
Step 36 - click on Apply & Save and then run the development job

Step 37 – take the public ip of prod server and paste it in chrome (http://13.235.70.138:8080/prodapp/)

**Linux Commands**

In linux forward slash (**/**) is used for root or parent user/director.

The default working directory for user is home and default working directory for admin is root



To install tree command – sudo apt-get install tree

**A single dot represent current working and double dot represent parent directory**

To check the default user - whoami

To check default home directory for current user – echo $HOME

To go into root account – sudo -su root

To check the user – whoami

To come out – exit

touch creates empty file

To create a file – touch filename(file1)

To check the list – ls

To create multiple files – touch filename(file2) filename(file3) filename(file4)

To store some data inside a file – cat > filename(file5) after giving enter you will go inside the file5

I am Komal

To come out of cat command – ctrl+D

To see the contrnt of the file – cat filename(file5)

To append the data in the same file – cat >> filename(file5)

To create directory/folder – mkdir directoryname (dir1)

To go inside the directory – cd directory name(dir1)

To check in which directory you are working – pwd (present working directory)

To come outside the directory – cd ..

To create multiple directories inside a single directory – mkdir -p d1/d2/d3/d4

To check the structure of directories – tree d1

pwd

ls

cd d1

pwd

cd /d2/d3/d4

pwd

touch file2

cd ..

To come to main directory – cd ../../..

To remove the file – rm file1

To be interactive remove the file – rm -i file2 //r stands for recurssive

To remove all the files starting with same letter – rm file\*

To remove directory – rm -r dir1(directory name)

To delete the director – rm -r d1

To see the long listing of all the files – ls -l

To craete the hidden file we use dot before the file name - touch .file5

and if you give ls you will not able to see file5 in the list

To see the hidden files – ls -a

To see the latest/newer files on the top – ls -lat

To see older files on the top – ls -latr

man ls

To see the date – date

To install calender – sudo apt-get install ncal

To check the calender – cal

To copy the files – cp srcfile dstfile

To copy the file to folder cp file1 /tmp

* cd /tmp
* ls

To create files under directory – touch d1/f1

* tree d1

To create files under directory inside directory – d1/d2/f2

* tree d1

To copy entire directory – mkdir -p d1/d2/d3/d4

* tree d1
* cp -r d1 /tmp

To cut paste the file – mv file1 /tmp

To rename the file – mv file2 file3

To link the files we have 2 types 1) soft link 2) hard link

**Softlink** (if we delete the original file the copy of the file gets deleted)

cat > filename(file)

I am komal

cat file

ln -s file file4

ls -l

**Hardlink** (if we delete the original file the copy of the file will not get deleted)

cat > filename(file)

I am komal

cat file

**ln file file4**

**ls -l**

To check the inode number – ls -li

**TEXT PROCESSING TOOLS**

To see content of the file in controlled manner – less /etc/passwd(filename)

To come out of the less use – q

To see top 10 lines we use – head /etc/passwd(filename)

To see specific number of top lines – head number(3) /etc/passwd(filename)

To see last 10 lines we use – tail /etc/passwd(filename)

To see specific number of bottom lines – tail -3 /etc/passwd(filename)

grep is used for searching a string in a file – grep spool filename(file1)

To see the line number where the spool word/string is available – grep -n spool filename(file1)

Exception the line number where the spool word/string is not available - grep -nv spool filename(file1)

Cut is used for cutting the data in column format

d stands for delimiter

f stands for fields

to get specific columns from a file – cut -d “ ” -f columnnumber(1,2) filename(file1)

wc stands for word count

to check the number of words in file – wc filename(file1)

to check number of lines in file – wc -l filename(file1)

to check number of words in file – wc -w filename(file1)

to check the count – wc -c filename(file1)

to check number of lines and words in file – wc -lw filename(file1)

sort does alphatecial sort not numeric to sort the data – sort filename(file1)

to do numeric sort – sort -n filename(file1)

to reverse the numeric sort – sort -nr filename(file1)

**Redirection and piping commands**

Sending an output of the command into a file is called Redirection

data > filename(file1)

cat filename(file1)

**OUTPUT:**

ubuntu@ip-172-31-46-43:~$ date > file3

ubuntu@ip-172-31-46-43:~$ cat file3

Thu Aug 4 17:45:45 UTC 2022

In linux any number of commands can run at a single time – cal;date

**OUTPUT**:

ubuntu@ip-172-31-46-43:~$ cal;date

August 2022

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31

Thu Aug 4 17:49:07 UTC 2022

(>) – to store data in the file

(2>) - to store error data in the file

(&>) - to store data & error in the file

To store multiple commands in a file – (cal;date) > file4

**OUTPUT:**

ubuntu@ip-172-31-46-43:~$ (cal;date) > file4

ubuntu@ip-172-31-46-43:~$ cat file4

August 2022

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31

Thu Aug 4 17:52:18 UTC 2022

To store the error message - mkdir d1/d2/d3 2> file5

**OUTPUT:**

ubuntu@ip-172-31-46-43:~$ mkdir d1/d2/d3 2> file5

ubuntu@ip-172-31-46-43:~$ cat file5

mkdir: cannot create directory ‘d1/d2/d3’: No such file or directory

To store error message and data – (data;cal) &> file6

**OUTPUT:**

ubuntu@ip-172-31-46-43:~$ (data;cal) &> file6

ubuntu@ip-172-31-46-43:~$ cat file6

Command 'data' not found, did you mean:

command 'dat' from deb liballegro4-dev (2:4.4.3.1-2)

command 'date' from deb coreutils (8.32-4.1ubuntu1)

Try: sudo apt install <deb name>

August 2022

Su Mo Tu We Th Fr Sa

1 2 3 4 5 6

7 8 9 10 11 12 13

14 15 16 17 18 19 20

21 22 23 24 25 26 27

28 29 30 31

**Piping** (|) – when a output of a command becomes input of other command

**OUTPUT:**

ubuntu@ip-172-31-46-43:~$ ls -i | wc -l

6

**OUTPUT**:

ubuntu@ip-172-31-46-43:~$ cat file1

hjghxfh

ghzjhx

komal

134256

jhdf5654d76gsidi

ubuntu@ip-172-31-46-43:~$ head -3 file1 | tail -1

komal

ubuntu@ip-172-31-46-43:~$ tail -2 file1 | head -1

134256

ubuntu@ip-172-31-46-43:~$ head -3 file1 | tail -2 | grep komal

komal

**PERMISSIONS**

3 people has access to the file normally OWNER/GROUP/OTHERS

R stands for recursive

r – 4

w – 2

x – 1

the default value for folder is 664

the default value for directory is 775

ubuntu@ip-172-31-46-43:~$ ls -l file1

-rw-rw-r-- 1 ubuntu ubuntu 45 Aug 4 17:12 file1

ubuntu@ip-172-31-46-43:~$ chmod 730 file1

ubuntu@ip-172-31-46-43:~$ ls -l file1

-rwx-wx--- 1 ubuntu ubuntu 45 Aug 4 17:12 file1

**USER ADMINISRTRATION**

User Administration can be provided by different plugins like Role-based Authorization or Matrix or Cloud based

**Creating user accounts and giving them privleges**

How to create users …

First Login to jenkins

Manage Jenkins 🡪 Manage users(security) 🡪 click on Create User

**For first user**

give all the necessary details like Username/password (komaldev/komaldev) etc etc 🡪 create user button

**For second user**

give all the necessary details like Username/password (komaltest/komaltest) etc etc 🡪 create user button

How to give specific authorities

Manage Jenkins 🡪 Manage Plugins 🡪Available section 🡪 search for Role-based Authorization Strategy 🡪 select it 🡪Install without restart

Once you install Role-based Authorization Strategy plugin

Manage Jenkins 🡪 click on Configure Global Security(security) 🡪 Authorization 🡪 click on drop down 🡪 select Role Based Authorization option 🡪 click on Apply & Save

To check if the role based strategy is working logout from the session

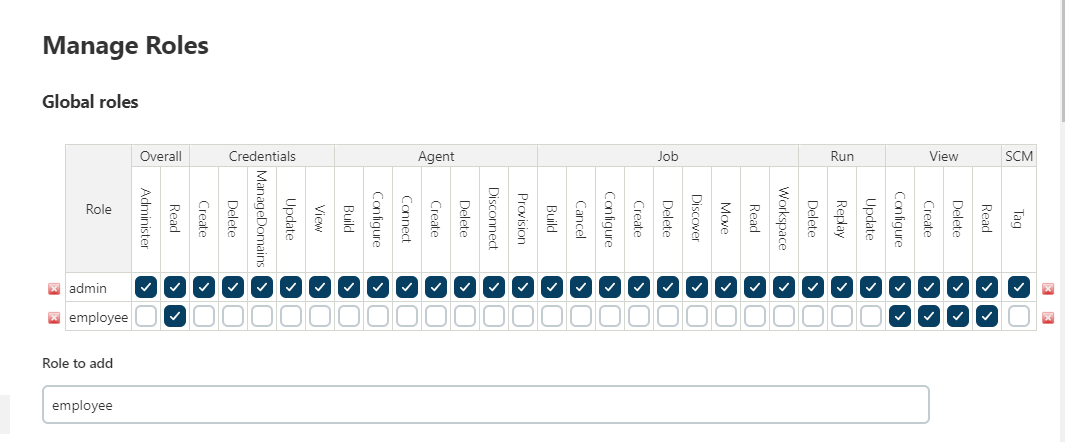
Try to login from the userid (komaldev or komaltest) it will give error as access Denied

**Restrictions**

Login to Jenkins using admin id

Go to Manage Jenkins 🡪 Manage & Assign roles (security) 🡪click on Manage roles 🡪 in Global roles 🡪Role to add 🡪 give any name (employee) and click on Add

Once you add the name(employee) give the authority



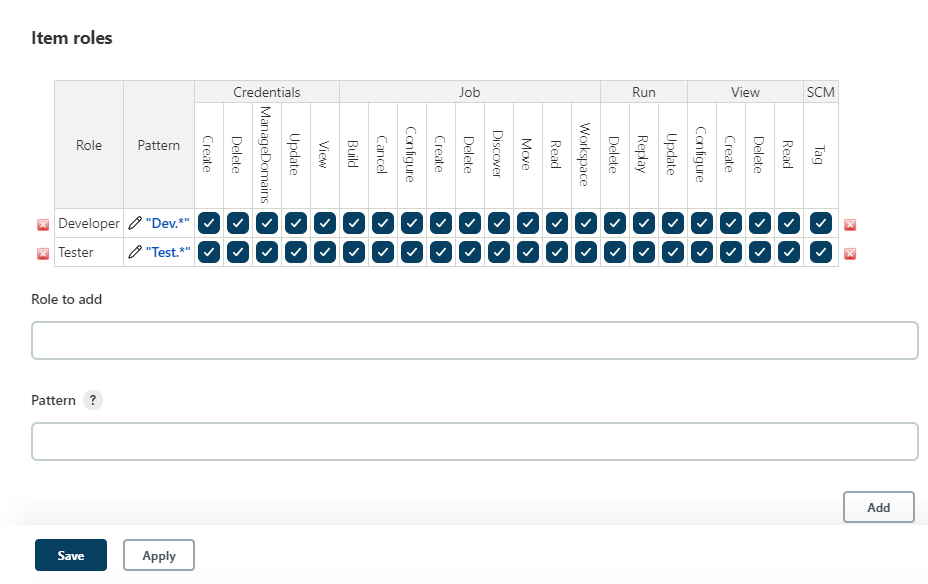
For Developer

in item roles 🡪Role to add 🡪 give any name (Developer) and give the Pattern (Dev.\*)click on Add

A picture containing text

Description automatically generated

For tester



click on Apply & Save

Now you go to the Assign roles

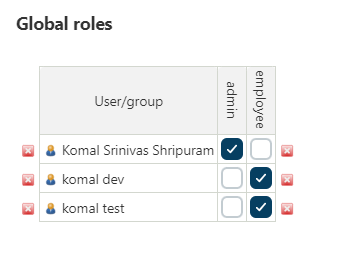
For first user

In Global roles 🡪 Add 🡪 user/group to add 🡪 give the name (komal dev) 🡪 click on ADD

For 2nd user

In Global roles 🡪 Add 🡪 user/group to add 🡪 give the name (komal test) 🡪 click on ADD

Once you add users, select the authority as employee as mentioned in screenshot



Now in item roles

For first user

In item roles 🡪 Add 🡪 user/group to add 🡪 give the name (komal dev) 🡪 click on ADD

For 2nd user

In item roles 🡪 Add 🡪 user/group to add 🡪 give the name (komal test) 🡪 click on ADD

Assign the authority as mentioned in below screenshot

Graphical user interface, application

Description automatically generated

click on Apply & Save ----- Logout from admin user

Logging in from specific users you will not see manage jenkins option

Now login from first user(komal dev) you can see only development jobs

login from second user(komal test) you can see only testing jobs

**MASTER SLAVE ARCHITECTURE OF JENKINS (passwordless connection)**

Commands in red color

Outputs in green color

If there are 2 servers call it as server1 & server2 and you want to connect server1 with server2 using password connection

* create 2 instances in aws
* name it as server1 & server2
* now we want to connect from server1 to server2
* take the ssh command of server2 and connect in gitbash
* [ now in gitbash type command – whoami

ubuntu@ip-172-31-37-96:~$ whoami

ubuntu

* now create the password for server2 – sudo passwd ubuntu

ubuntu@ip-172-31-37-96:~$ sudo passwd ubuntu

New password:

Retype new password:

passwd: password updated successfully

* now to send the password authentication to yes

ubuntu@ip-172-31-37-96:~$ sudo vim /etc/ssh/sshd\_config

* change the PasswordAuthentication field to yes from no once done (esc :wq)
* now restart the ssh service

ubuntu@ip-172-31-37-96:~$ sudo service ssh restart

* once password is set give exit and comeout of the session
* take the ssh command of server1 and connect in gitbash
* now you connect to server1
* give ssh username@private ipaddress of server2

ubuntu@ip-172-31-45-219:~$ sudo ubuntu@172.31.37.96

sudo: ubuntu@172.31.37.96: command not found

ubuntu@ip-172-31-45-219:~$ ssh ubuntu@172.31.37.96

The authenticity of host '172.31.37.96 (172.31.37.96)' can't be established.

ED25519 key fingerprint is SHA256:eejGbpgqjkTRgdMUC4KM/iM2leZiw4oOIhW93bMPeT0.

This key is not known by any other names

Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

Warning: Permanently added '172.31.37.96' (ED25519) to the list of known hosts.

ubuntu@172.31.37.96's password:

Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86\_64)

* exit

Passwordless Connection

* ssh-keygen

ubuntu@ip-172-31-45-219:~$ ssh-keygen

Generating public/private rsa key pair.

Enter file in which to save the key (/home/ubuntu/.ssh/id\_rsa):

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in /home/ubuntu/.ssh/id\_rsa

Your public key has been saved in /home/ubuntu/.ssh/id\_rsa.pub

The key fingerprint is:

SHA256:pRJpOQNT8VOq7xroEiZZvCpFS+lBKSjhCp1jvHyrUuQ ubuntu@ip-172-31-45-219

The key's randomart image is:

+---[RSA 3072]----+

|o. .o.o. . |

|=oo. o + o |

|o=\*. B + . |

|ooOo . = + |

|.Oo+. o S |

|o E. o o |

| \* .o . . |

|+ .o o |

|..... ... |

+----[SHA256]-----+

* Public and private key are generated

ubuntu@ip-172-31-45-219:~$ cd .ssh

ubuntu@ip-172-31-45-219:~/.ssh$ ls

authorized\_keys id\_rsa id\_rsa.pub known\_hosts known\_hosts.old

exit

* Once private & public key generated
* To copy the id – ssh-copy-id server2name@server2 private ip address

ubuntu@ip-172-31-45-219:~$ ssh-copy-id ubuntu@172.31.37.96

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ubuntu/.ssh/id\_rsa.pub"

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

ubuntu@172.31.37.96's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ubuntu@172.31.37.96'"

and check to make sure that only the key(s) you wanted were added.

* Now ID/pwd is generated you can direxctly opne server2 from server1 without password

ubuntu@ip-172-31-45-219:~$ ssh-copy-id ubuntu@172.31.37.96

/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/ubuntu/.ssh/id\_rsa.pub"

/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed

/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

ubuntu@172.31.37.96's password:

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'ubuntu@172.31.37.96'"

and check to make sure that only the key(s) you wanted were added.

ubuntu@ip-172-31-45-219:~$ ssh ubuntu@172.31.37.96

Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-1011-aws x86\_6]

MSA – Disturbuting the work load to other machines which are slave machines

* Start your jenkins/QA/Prod servers in aws
* Connect to jenkins in gitbash
* Open jenkins in chrome with the public ipaddress of the jenkins server :8080
* Now create a new instance in aws call it as slave1
* Connect slave1 in gitbash
* Repeat all the steps from above []
* Once all the steps are done
* Give the command - sudo apt-get update
* sudo apt-get install -y openjdk-8-jdk
* java -version
* now to copy the slave.jar from master sever to slave follow below steps(take the ipaddress of jenkins server i.e master server)
* wget <http://172.31.8.54:8080/jnlpJars/slave.jar>
* chmod u+x slave.jar
* ls -l
* mkdir myworkspace
* cd myworkspace
* pwd
* go to Jenkins 🡪 Manage Jenkins 🡪Manage nodes & clouds (System Configuration)
* create new node 🡪 slave1 (name) 🡪 select Permanent agent 🡪create
* once you create there are few other details
* give remote root directory (/home/ubuntu/myworkspace)
* Labels: myslave
* Usage: only build jobs
* Launch method: Lauch via execution
* Launch command: ssh ubuntu@private\_ipaddress\_of\_slave java -jar slave.jar
* ssh ubuntu@172.31.44.135 java -jar slave.jar
* once it is done click on save
* Go tp testing job 🡪 configuration page 🡪 in general section 🡪 select the option restricit where this project can be run
* Give the label name 🡪 myslave -> save 🡪 go to dashboard
* Run the development job, you can see development job will run on built in node and testing in slave
* Connect the slave in gitbash
* ls
* cd myworkspace
* cd test/
* ls
* in console output also you can see the job

**GIT GLOBAL INFORMATION TRACKER**

3 stages –

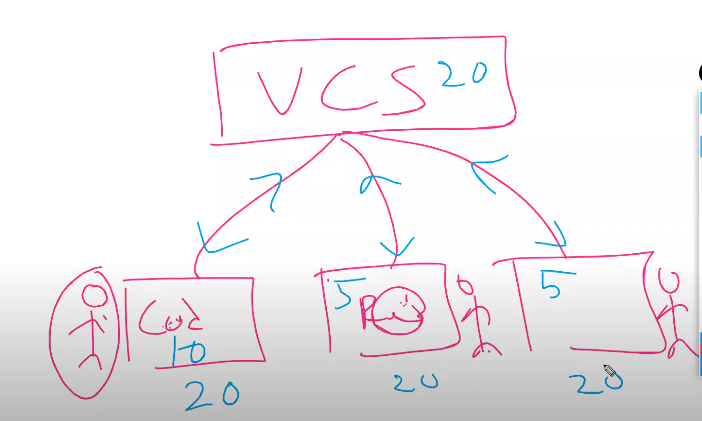
working directory (untracked files),

staging area (indexed files),

Local repository (Committed files)

It is the version controller of the system or it is the code repository where developer place the code

1. Centralized version control



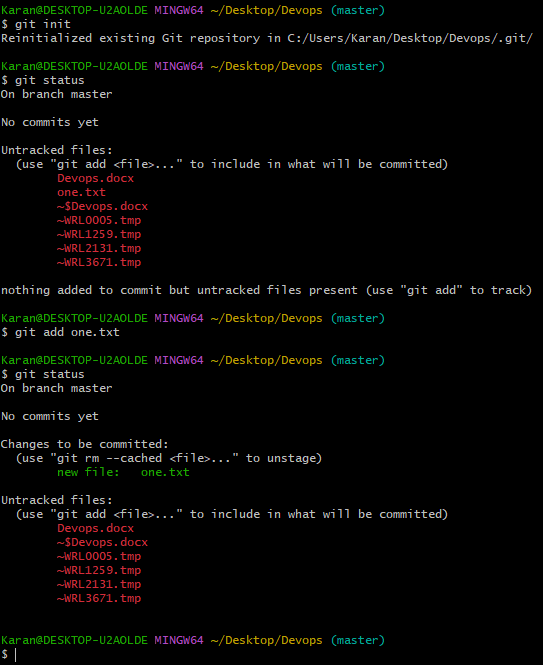
1. Distributed version control

GIT commands

To configure username & email

* git config –global user.name “komal”
* git config –global user.email [komal.shripuram1798@gmail.com](mailto:komal.shripuram1798@gmail.com)
* Text

  Description automatically generated
* Text

  Description automatically generated
* git init
* git status
* ls
* git add filename
* git status
* 
* git add filename1 filename2 (to commit 2 files at a time)
* 
* git add .
* git status
* git rm --cached ts.txt
* git status

GIT BRANCHES

It is feature of GIT, it is possible to segregate the code.

To create the code in organized manner is called branch.

1 To see the list of local branches

git branch

2 To see the list all branches local and remote

git branch -a

3 To create a branch

git branch branch\_name

4 To move into a branch

git checkout branch\_name

5 To create a branch and also move into it

git checkout -b branch\_name

6 To merge a branch

git merge branch\_name

7 To delete a branch that is merged

git branch -d branch\_name

This is also called as soft delete

8 To delete a branch that is not merged

git branch -D branch\_name

This is also known as hard delete

When you create a branch, upto that point of the branch whatever is the parent branch commit history get copied into newly created branch