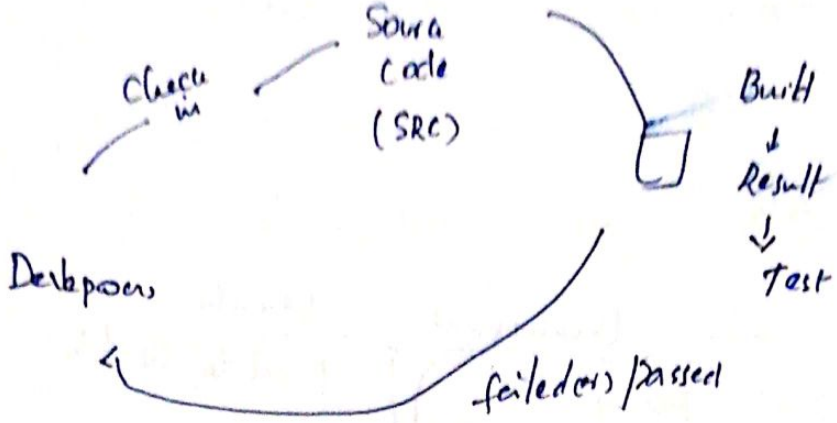
Meeting ID- 926296765

25th Sep 2019

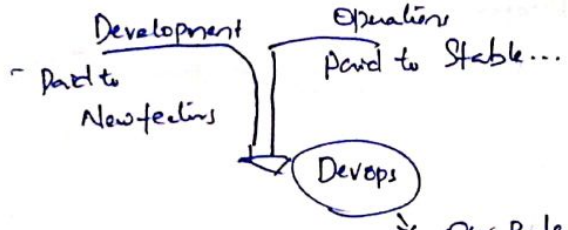
⇒



⇒ Operations Part

- Server
- Install OS
- Runtime - JRE / Python / Nodejs
- Web Servers
- Application Service.
- Load Balancing
- DB service.
- Eg 2000 Clients - 1 Huge server
  10000 Clients - 5 Huge Servers
  100000 Clients - Lot of Infra..

→ operations Team Roles -



⇒ def of Devops

** ↓ is a S/w Engineering Culture and practice that aim at ~~bridging~~ unifying S/w development and S/w operation

* => Generic devops is a set of practices that automate processes b/n S/w dev teams and infra (ops) teams so they can build, test and publish S/w more quickly and reliably.

** → Devops practices

(i) Version Controlling → Developers
(ii) Automated Build - Build teams
(iii) Automated Testing - QA Team
(iv) CI - CD (Automation)
(Continous Integration - continous development)
(v) Automated Infra (Scripting Tools)
↳ Here comes the Role of Python.
(vi) Monitoring

Role.

Any task - CLI (100% Automation) 25th Sep 2019
↳ Command line Interface

→ Pre-Requisites: 1. Infrastructure → Aws → Google Cloud - free.
2. Linux Admin {CentOS 7 ~ Redhat} {Ubuntu}
→ file & Directory mgt
→ Text Editor (vi - Visual editor)
→ Archives
→ utilities
→ user mgt
→ file permissions
→ package mgt → Service mgt.

> Aws - Charges
→ FREE TIER / 1 Year
{ T_2 . Micro machine - 1GB RAM
1GB CPU }

> Google Cloud - Completely free
Get $300
- FREE TIER / 1 Year / INR 20,000 will be Credited
{ You can use as much as 4 CPUs & 16GB RAM }

> Azure

> System Activities

> LAMP Stack (web development platform)
L - Linux
A - Apache web Server
M - My SQL Database Server
P - PHP
(Programming language)

> Content management System
- Cms App

→ Devops
CODE - BUILD - Test - Release. RESULT → Deploy

> Version control System → History copy.
> developers / Infra Automation
> Git / Github { Client / Server }
> Source code (src)

> Build Tools { Soure code = Executable Code }
> Java application → Build on Infra Servers
{ CODE - Team 1 }
{ CODE - Team 2 }
{ CODE - Team 3 }
$T_1 + T_2 + T_3$ { Common App Ex
> C { make } → Changes Code into executable
> Java { ant - maven - gradle }
> NodeJs { ng - npm }
> Python { pybuilder }

Eg {make Install - make build}

Java
Eg {ant run - mvn Install}

Node.js
Eg {npm Install - npm start - npm run}

Python
Eg {pyb Install - pyb start}

> Testing. → Selenium (Functional Testing)

Tools → 1) Sonar Qube.

Contains
> Dash board
> Bugs
> Valerabilities
> Best Practises
> Duplicate Code.

> mvn Sonar: Sonar
{ Bug & Vulnerabilities }

> mvn Verify.

> Releases
> Sonatype Nexus { Registry }
> Holds all the releases of your application
> Dependency mgt
> Application Server (TomCat)

** => Infra → Virtulization (Vms) { Ecom:: OS + JRE + WS + DB + App + LB - 4 Hours } 20 env → 80 hours

> Hypervisors { Virtual Box } { Virtulation } → working like { Typical Admin } { Heavy weight }
> Vagrant { Infra As CODE } - Reuseability
20 Env → 2 hours.
Good for { Dev / QA env }

> Configuration mgt Tools
{ Production Env }
> Pull based - CHEF **
> Push Based - Ansible **

> Containers - { Light weight } { 20 mins }
> Docker
> Kubernetes

> CI - CD

Continuous Integration - development.

> JENKINS

> Projects

> Pipelines

↓

Checks

Code - Build - Test - Release - Deploy.

{ Java, Python, Nodejs - Ruby }

30th Sept 2019

→ Cloud computing:

Offerings

> 1 On Premise

> 2 IAAS:

> Infrastructure as Service.

> 3 PAAS:

> Platform as Service.

> 4 SAAS

> S/w as Service.

IAAS (EC2) → Virtualization Take care by Cloud.

(Virtual machines)

You manage:
- Applications
- Data
- Runtime
- Middleware.
- O/S

Managed by Cloud Service provider:
- Virtualization
- Servers
- Storage
- Networking

- PAAS : (platform as Service)
(app service)
eg : Google drive

you manage { Application
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking

managed by Cloud Service provider.

- SAAS :
(O365)
office

Application
Data
Runtime
middleware
O/S
Virtualization
Servers
Storge
Networking

managed by Cloud Service provider.

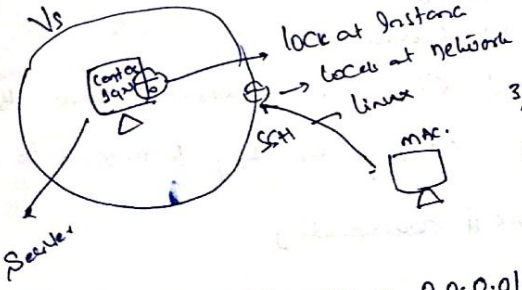
⇒ AWS Account Setup :-

https://aws.amazon.com/free.

⇒ https://cloud.google.com/free

AMI - Amazon machine Image

Vpc - default n/w.
(Virtual private Cloud)



3389 windows port
3306 - Database port

Ip range → 0.0.0.0/0

↓
Can be accessible with internet

→

=> other tools for putty → git bash

In putty → we need to change the Security key
9Am.pem
↓ changes
9Am.ppk

putty gen → load pem
Save pvt key
↓
ppk.

⇒ Git bash:
↓
To Go with SSH Client.
CLI Console.
↓ Go directly where .pem key placed.
Change the key permission to 400.

~/Downloads
~/Downloads > Chmod 400 9am.pem

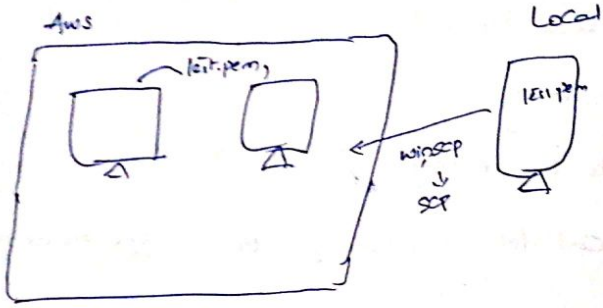
> SSh -i "9am.pem" root@ec2-54-224 ...>
asks us to login as "Centos" instead of "root"
>
as root access is disabled

[Centos@ip-172-31-46-258 ~] $ uname
Linux
$ logout
Exits from machine.
$ Cat /etc/os-release → provides the os details)

⇒ Linux Admin :-

1st → Nov
3rd Oct 2019

Remote machines



> To copy the files from local to remote server A.

SAME PEM

> Scp -i file-to-auth.pem file-to-be-sent.zip

Centos@ DNS-A: /home/Centos..

> Eg

Scp -i test.pem test.pem Centos@ec2-203.x...

↑ name may be change : /home/Centos/..

Eg Scp -i test.pem Documentary.Out @

Centos@ec2.203 ... : /home/Centos

>

> Scp -i file-to-auth.pem file-to-be-sent.zip

Centos@ DNS-B : /home/Centos

⇒ Different PEM file

Scp -i file-to-auth-A.pem file-to-auth-B.pem

Centos@ DNS-A : /home/Centos

=> Aws → Google Cloud Server:

→ on Aws Server Create the SSh key using.

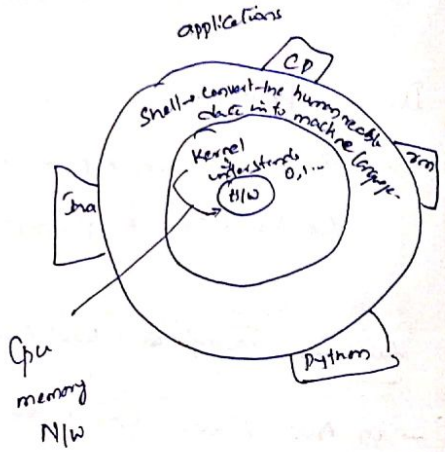
=> SSh-keygen

=> Cat

→ Copy the public key to google cloud machine

→ So we can connect to gc machine using SSh.

Linux Architecture:



Vi editor

> Vi

> Command mode - delete / search / copy

> Insert mode - Modify / Insert

> Vi <new file>

i - insert mode

esc - exit insert mode.

:w - write

:q - quit

:wq - write and quit

> Cntrl + u - page up

> Cntrl + d - page down.

> Beginning of file - gg

> End of file - G

> :se nu {Set number}

> : line-number {:38}

> x - remove character

> 5x - 5 characters to be removed.

> dw - delete the word

> 3dw - delete the 3 words

> dd - delete line

> 5dd - delete the 5 lines

> Cntrl + R - Redo the operations

> u - undo.

> yw + p - word copy and paste

> yy + p → copy line.

> 5 yw + P - {Copy paste 5 words}

> 5yy + p - {Copy 5 lines}

> ~~/Goto~~ /pattern → eg Kantas... ↓ to search

> File & Directory operations :

Note > Comand --help
> man Command

1 > Cp <Source file> <destination path>

Eg Cp test.pem test

> Eg Cp test.pem test.pem-old.

2 > Cp -r <source directory> <destination directory>

3 > rm <file>

4 > rm -r <directory>
rm -rf <directory>
↳ force fully.

5 >
mv - move / Remove

mv <Sorce location> <destination-Should be ~~same~~ other-path>

> rename

mv <Src-location> <destination-Should be Same-path>

> Archives :

4th oct 2019

> Winrar / 7zip → in windows

> In built { tar utility }

> Archive using tar

** > tar cvf my.tar files
↳ for creating tar file
c - Create
V - Verbose (showing on Screen)
f - file.

> Extract using tar

*. > tar xvf my.tar
↓
Extract

** > Wget <link>
{ download files from remote locations }

> To Install application

~~yum~~ > yum -y install filename

> View content inside the tar file

>tar tf my.tar

> zip/unzip

>Sudo yum -y install zip unzip

Eg zip new.zip files*.

unzip -l new.zip

↓
List

unzip new.zip

> Curl

Eg Curl http://localhost.

To test the pages are loading in local machine.

⇒ Curl https://www.google.com

> grep:

grep <pattern> file.txt

Eg Sudo Systemctl status firewalld | grep active

> | → pipe. → used to connect the Commands.

> wc → Word Count

wc -l → lines

Eg wc -l httd.conf.

wc httd.conf

92 254 2177 httpd.conf

↓ ↓ ↓

lines words Characters in file.

> head httpd.conf.

> head file.txt

{displays top section of file default is 10

> tail file.txt

{displays bottom section of file, default is 10 lines}

> find files

> locate { We need to install }

> find.

Eg locate httpd.conf
/etc/httpd.conf
/etc/... , /home/ninja/httpd.conf
Sudo yum -y install mlocate
> Sudo updatedb

Eg find <location> -name httpd.conf

> Sudo find / -name httpd.conf
/etc/httpd.conf
/home/ninja/httpd.conf

⇒ USERS :

> id <user>

User types

1 > Root user

> Sudo su - { switch to root user }

2 > System users

> will be added, when we install Certain Services.

Eg install apache Web Server, will get apache user.

Eg : install jenkins Server, will get jenkins user

3> Normal users

> which an admin might create, using user add command

> User info present in /etc/passwd file

> Sudo Vi /etc/passwd

1 2 3 4 5 6 7
myadmin : x : 1003 : 1004: : /home/myadmin : /bin/bash

1. username
2. password {/etc/shadow}
3. user id
4. group id
5. Comments
6. Home directory
7. default Shell.

> Adding user

> Sudo useradd <username>

> Sudo useradd myadmin

> Set the password

> Sudo passwd <username>

> Sudo passwd myadmin

⇒ Algorithms used to encrypt in /etc/shadow.

1 is MD5

2 is Blowfish

5 is SHA-256 → Aws and Azure use this algorithm.

6 is SHA-512

> Switch to other user.

> Su <username>

> Su myadmin.

> Modifying the user.

> usermod -aG <group> <username>

append Secondary group.

> Sudo usermod -aG google-users myadmin

> Sudoers file

> Sudo visudo

> Sudo usermod -l <new name> <existing name>

Eg usermod -l devi myadmin
Testadmin

> usermod --help ↵

> usermod -u 2003 testadmin ↵

To change the user id.

> groupmod --help ↵

10th oct 2019

> System admin Activities → MAMP → mac os.

Stacks Available: → WAMP → Windows.

> LAmp Stack (web platform platforms)

Linux Apache web server

PHP

MySQL DB Server

> Py Stack

> os

> web Runtime

> DB Layer

> MEAN Stack.

Apache Web Server :-

> Pop up msgs are load into this web server.

- files will expose on internet.
- its like Servlet to transmit data in web page.

> Web Server: A Server S/w that turns your server to transmit the data over internet

> Any web application on internet should be hosted on web Server.

> parameters of Apache:

> package name - ~~httpd~~ httpd (Centos) / apache2 (Ubuntu)

> Port - 80 / 443 {http / https}

> main Config - /etc/httpd/conf/httpd.conf

> Document Root - /var/www/html.

=> Logs /var/log/httpd (of application hosted)

rpm -qi httpd

rpm -qa | grep httpd.

=> installing Apache:

yum -y install httpd.

=> rpm -qi httpd

for checking

=> Systemctl Status httpd

Sudo netstat -ntpl | grep 80

for Starting

Sudo Systemctl Start httpd

netstat -ntpl | grep 80.

httpd -t → To test the configuration file is OK or not.

=> Loading web page.

Vim /var/www/html/index.ht

<html>
< 2

→ To change the http. port we need to make changes in the configuration file.

/etc/httpd/conf/httpd.conf

Changin Listen option

as 9090, Now http will run on 9090 port.

⇒ # Sudo getenforce

Strict

Sudo Setenforce @ # Sudo Setenforce 0

To change Permissive mode

(or) Changin file

⇒ Vim /etc/Selinux/Config.

> Set SELinux {Security enhanced linux}

> Aws - Security Groups

- Allow inbound rule for, Specific port i.e port 80 for http.

> open ALL ports {Stop the instance, Once lab is done}

> Gcp - firewall rules

> Google Cloud page.

> Hamburger menu > Networking > Vpc ~~content~~ > firewall rules > Networks

Create firewall rule > Name (all) > Targets > Sorce Ip range (0.0.0.0/0) > Protocall and ports (allow all) > Create.

> Azure → NSG rules. (Network Security group)

⇒ .php files will not run without having Runtime.

≥ webtatic.release ≠3 Should bin

⇒ Repository, (/etc/yum/inventory/)

⇒ S/w Repository: is a storage location from which your system retrieves and installs OS updates and applications. Each repository is a collection of s/w hosted on a remote Server and intended to be used for installing and updating s/w Packages on Linux Systems. Repositories contain thousands of Programs.

17th oct 2019

Virtual hosting

↓ Refers

<VirtualHost *:8080>

DocumentRoot /var/www/html/Pizza

</VirtualHost>

⇒ Need to change the /etc/httpd/conf/httpd.conf file.

netstat -ntpl

Listen portnumber

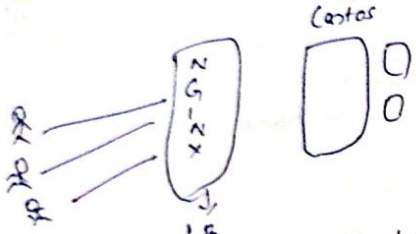
Eg Listen 9081

⇒ Linux Based

- Webserver
- Load Balancer

Reverse + Proxy

NGINX ↓do load Balancer.



> efficient distributing the network traffic.

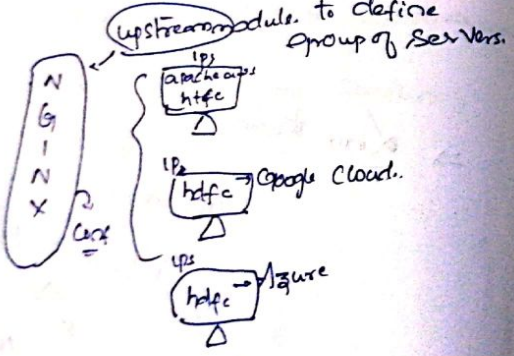
> forward proxy - protection at client side.

> VPN's

> Reverse proxy - Backend Server protection

> Nginx ~~xx~~ → Good one.

Eg



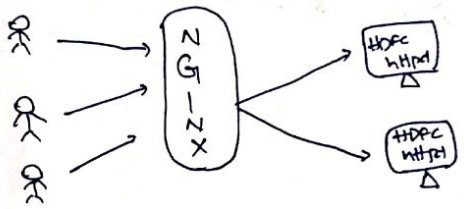
14th oct 2019

> EPEL Repo

- Extra packages for enterprise Linux.

epel. repo

→ Load Balancer



IAAS → Infrastructure as Service:

- Is one of cloud computing offerings.

Nginx parameters

> Package - nginx

> Port - http/80 && https/443

> Document Root - /usr/share/nginx/html

> main config - /etc/nginx/nginx.conf →

{ upstream }

→ Test config → # nginx -t

Sudo systemctl start nginx

** Backend upstream is a collection of web1 and web2

#

```
upstream backend {
    server 35.232.10.88;
    server 35.222.10.87;
}
```

M+

> L + A + P

↓ ↓ ↓

Linux Apache PHP

{ my SQL → Install DB }

{ DB is connected with application - wordpress }

{ Wordpress - PHP }

→

15th oct 2019

my SQL :

** Repository :

mysql-community-release.el7-5.rpm

ls /etc/yum.repos.d/

⇒ mysql port - 3306

Sudo netstat -nlpt | grep 3306

Sudo systemctl start mysqld

⇒ log in to db

> my sql -u root -p

⇒ Setup Credentials

> my sql_secure_installation

⇒ login mysql.

> mysql -u root -p ↵

mysql> Show databases;

mysql> Show tables:

mysql> Create database wp:

↳ Creating database

mysql> exit

⇒ (Wordpress, Drupal) Are the Examples of Content management Systems

17 Oct 2019

Devops

Version Control Systems:

CODE – BUILD – TEST – RELEASE – DEPLOY

↓

Eg ecomm Application

↓ 15 developers working

5D → CoD → USA

5D → Netbanking → INDIA.

5D → Credit Cards → Australia

- managing the development part.

VCS (Version Control Systems):-

⇒ Also known as revision control or source control, is the mgt of changes to documents, Computer programs, Web sites, files and any other Text data.

⇒ Changes are usually identified by a number or letter code, termed as "revision number".

> VCs are essential for the organization multi-developments.

Why we need Vcs:-

- made a change to code, realised it was a mistake and wanted to revert back.
- Lost code and didn't have a backup of the code.?
- had to maintain multiple versions of a product.

→ Advantages - Vcs

- Backup
- Collaboration.
- Storing Versions.
- Restoring Previous Versions.
-

=> Version Control / Revision Control / Source Control is a Sfw that helps Software developers to work together and maintain a complete history of their work.

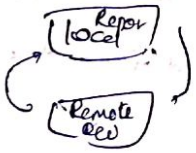
- Vcs as kind of "database"
- Snapshot of our Complete project at any time we want.
- It differed form the previous one Version

Types of Vcs

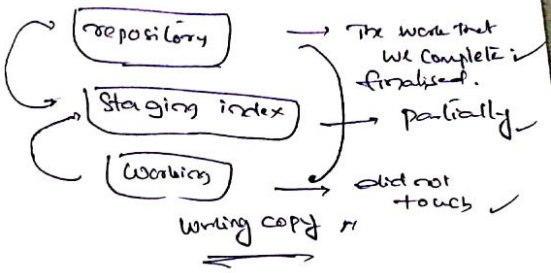
1. Centralized Version Control System.
   - Typical Server-Client model.
   - Uses a central Server to store all file, with collaboration.
   - disadvantages
     - Single point of failure (SPOF)
2. Distributed Version Control System.
   - Eg. Remote Server → local Server model.
   - (DVCS)
   - We can Commit Changes, Create branches, and perform other operations when you are Offline.

Typical Vcs working

- A typical Vcs use something called two tree architecture



→ Three architecture will be in distributed Vcs.



21st Oct 2019

Release no → commit-id → # change we done.

cid

Repo 2

Repo 3.

⇒ Normal directory can be changed into Repository by using `# git init`

eg mkdir test

mkdir website. To know it > git status > cd test > git status

⇒ Git Setup :

> Sudo yum -y install git

> Config

> git config --global user.email

devendraprasad.ch@gmail.com

{Git hub Account email}.

> git Config --global user.name devendra

> git Config --global color.ui true.

{ Enables

Creating Git

mkdir test.
mkdir website
cd website

1. # git init
initialized empty Git repository in /home/.../.git/
git Status

2. # git Clone website://... .com.

git Status

Configuring our own git :-

git Config --global user.mail dbabug@dxc.com

git Config --global devi

git Config --global Color.ui true

git Status

on branch master → is the pointer where Changes are taking place

Initial Commit

Changes to be Committed

Eg Touch dummy.html

git add index.html.

git Commit -m "update page index.html".

git Status

git log

git Show 777... → Unique.
≈ 48 digits code.
↓ code generated in gitlog file.
Reference code of Change

git log --one line.
all logs in one line.

git log --one line --grep title
Searching title.

git log --online --grep author
Searching using Author

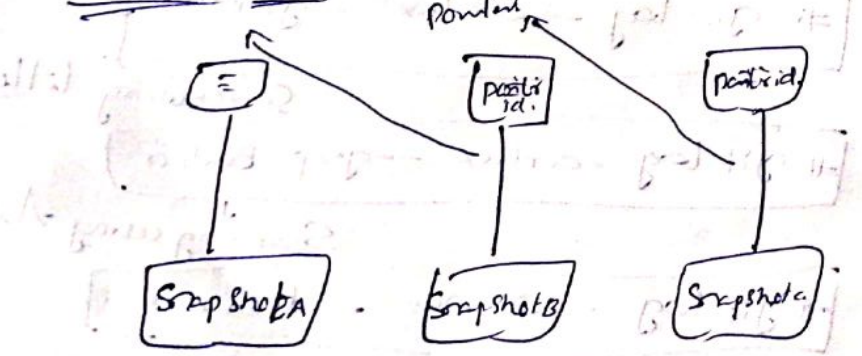
git log --online. --decorate

- Branch will allows us to isolate to different location that doesnot effect to current running application,
- So we need to issolate to make the modifying, fixing the bugs.

⇒ Branch allows the developer to branch out from the original code base and isolate from others.

⇒ master (main branch)

⇒ A branch in git is simply a lightweight movable pointer to one of these Commits



git checkout testing
// Switching to the Commit parts.

git branch

⇒ Head is a pointer to the latest Commit made.

** → Head moves forward when a Commit is made.

git log --online --decorate

3b5d101 (HEAD, master) title update.

⇒ # git merge master
// When the Current Changes are happening.

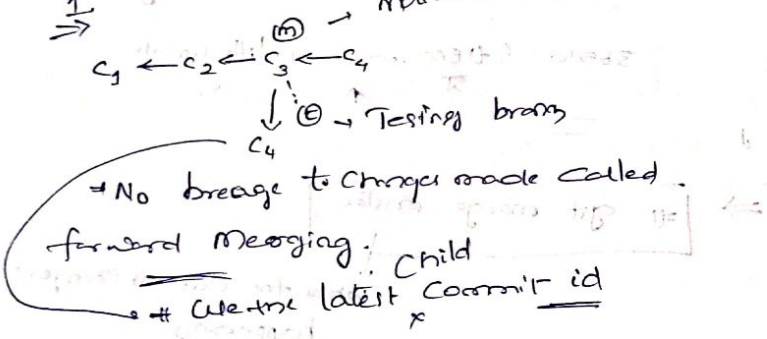
git Checkout master
// Switch to master branch

git merge testing
// Now Telling Branch Merge to master.

=> Mergings are of Two types

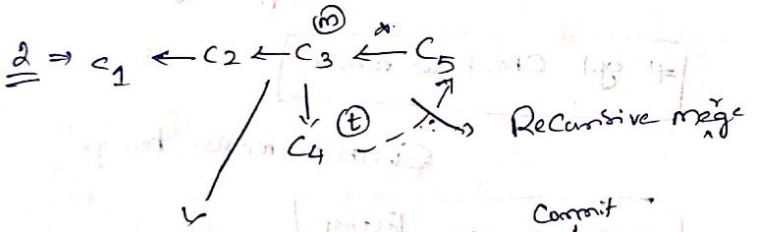
- fast forward mergings
- Recurise (Curise) mergings.

1 =>



* No breage to changes made called forward Merging. Child
* Always latest Commit id

2 =>



* Here Create the new Commit C-id

=> Both fast forward and Recurise merge Come under "Automatic Merging"

=> # git reflog

↳ will show how your head logs moved from initial id

Eg for recurise merging

```
# git branch
# git Checkout Testing
# git Status
# git add .
# git Commit -m "contact.html".
```

↳

```
# git Checkout master
# git Status
# git add.
# git Commit -m "------>".
```

Gives new Commit id

↳

```
# git reflog
```