[day-2] ①

→ Rollback ???

Version control system.

(Notepad)

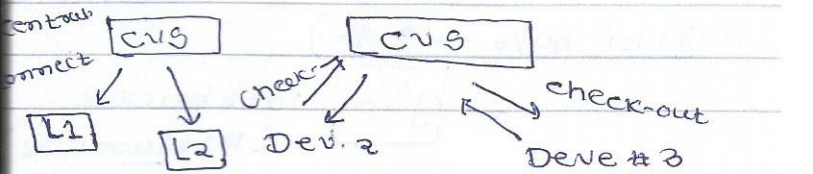
* Without VCS.
  - Downtime of Application.
  -

* With VCS
  - order code back & reApply.
  - Source code ⟷ Versioning.

* Local db (Versioning)

→ created then no Recovery.

* Centralized Version Control



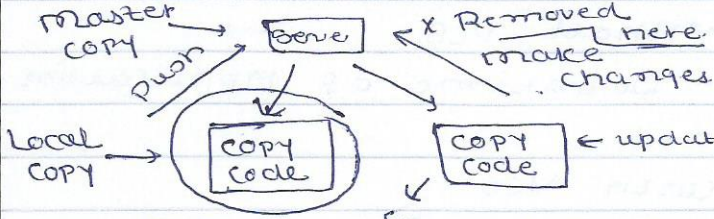
Tortoise → SVN (storing it to server)
Client ←

↳ Central System Burnt | down

↳

* third generation

Distributed Version control



→ update the changes locally.

Git

↓ Fetch

Build code

* source code
* compile

* git (80% ~ 90%)

← git Bitbucket (Atlassian)

Amazon web Service ③

↳ Compute

↳ Server EC2

(elastic computing Cloud)

↙

Launch a instance

(Vm, virtual machine)

↙

Ubuntu — type (free

t2. micro

[1 instance] (1CPU, 1Gb)

↙ (4Gb)

↙

Add storage {8gb}

(32gb free)

↓

Tage [Adding Name,

GitPractice]

↓

configure Security Group.

— data center (Firewall)

* ssh — TCP — port

(traffic 22

can come)

Add Rule

→ diff Service | option

All trafic All - 0-65535

[start ↙ end ↘]

* key-pair (user-name password)

Amazon — Ubuntu (user name)
— ec2-user (other server)

↙

no have password to connect with them.

* key-pair

↳ create new key pair

↙

edu (demo)

↙

(download key pair)

→ Git Practice — green instance.

↙

Public IP - outside

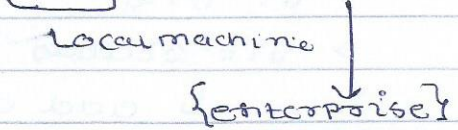
Private IP - [one AZ to other AZ]

Gitbash (bash software)
↳ Linux command on window machine]

~% ssh -i Download/edu.pem
ubuntu@3.12.155.21
(Public IP)

Chmod 400 edudemo.pem
↳ only i can access

> sudo su (super user)
root@ip :~ cat /etc/os-rele
↙ console display. ↘ version



git (local)
github (server (sharing with)]

Github enterprise (hosted in your data center)

Virtual machine ↔ version or making update of all software – {git --version}

> git --version
> git init
> touch one two three
> git status.
> git add one two three
> git commit -m "Initial commit".

> vi one
> git status
↳ add changes to git
> git add one (staging)
> commit (commit)

> git log — commit XYZABC
Author: —
Date: —
comments.

Commit XYZABC
Author: —
Date: —
second commit

[Stage% temp. area ( ]

> git conf --global help.
autocorrect 1.
↙
nearest command to states.

> git config --global
code.editor "vim"

git > git — user.name
user.email
'email@github.com')

* user - user level
* project - @ some project (-)
* system level
  - ↓ every body
  - ↳ documents only

[git config --global ...]
↑ user level

[git config --system ...]
[git config ** ...]
/ ↑ project
dont give

* changeset

L3 [New Layer]

ssh -i file host: public ip
user@ip
ubuntu

git diff Head~1..Head

* Revert – Create new commit id
* Rebase – Removes unnecessary commits
* git reset --hard baba (Removing comp.)
  - soft
    (Staging area)
    Not Removing comp.)
* Parallel Development.
  git checkout -b b1 maste
  *b1 (new Branch)
  master
* git merge b1
  <<< Head
  ===
  >> b1 → {Resolve conflicts}

Stash → {temp. Shelves}

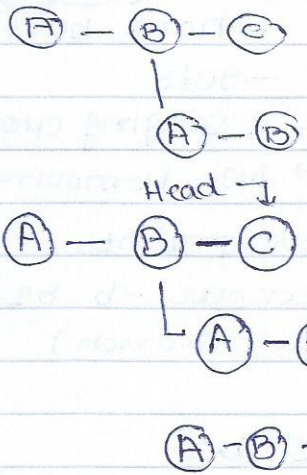
> git status.

> git stash show stash@{0}

stash@{0}

* merge → new commit

*



* git remote (github.com)

git remote add origin

[git push origin https://~ --delete b1] - "

- clone (download) local (1)

* pull | Fetch (know changes)
  - Read (download changes)

(fetch + merg)

* rm fileone
  - ↑ git doesn't know

Red

git rm fileone
- ↑ git knows fileone removed
- green fileone

* git ignore
  - .gitignore → { ravi.txt, java.ext, abc.class,
* Removes or untracked by git status.
* ↳ cat .gitignore
* more .git
* .gitignore → ignore new files

- git new empty folder is not tracked.
- git tag --a version -m "this V1.2.23.1 new hotfix"

→ git tag --a V1.2.23.1 -m "this."
↳ Recent commit id.

Commit 4fcd... (tag: V1.2.23.1)

→ git tag --a V1.2.23.0 -m "old x43 ...

→ git log

→ git tag
→ git show V1.2.23.0
(commit or tag)

↳ git log (all commits)

↳ git show (specific commit)

* new file (Not going to this Release)

* new file is ignore or not staged.

* touch welcome.html index.h

↳ stash (new change, temp space---)

↳ git stash show stash@{1},

↳ git stash apply stash@{1}

- git push --tags (need to push tag sep to)

code at specific level

v1.2.23.1 (tag, push your data.)

Fork - copy (main → another account)

* two github account

* dark lauching.