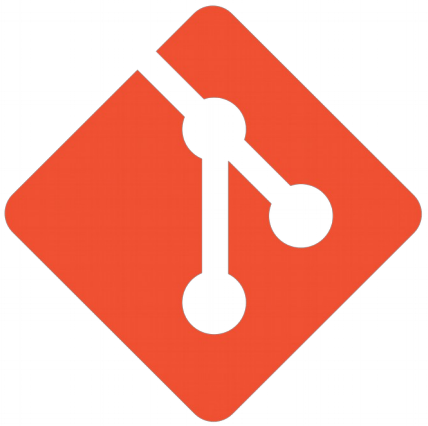


Git

darshan@fsmk.org
dharshan4help@gmail.com

~~Rules~~ Request

- Strictly NO Sir!!
- Listen (All info present online)
- Mobiles – lets not affect session (silent)
- Feel free to ask questions

Some GNU/Linux commands

- ls
- mkdir
- cd
- vi/gedit

Lets not start with Git

?

Lets start with a problem!

Lets build a math library

Assumption

- sin
- cos
- tan

Requirement 1

I just need Addition

Requirement 2

Subtraction

Requirement 3

Multiplication

Requirement 4

Division

Requirement 5

Now I need

- sin
- cos
- tan

(Ooops!! I should have copied)

Solution

Lets store each version in a Folder/File

- Requirement 1
- Requirement 2
- Requirement 3
- Requirement 4

Issue

Which requirement in which file/folder?

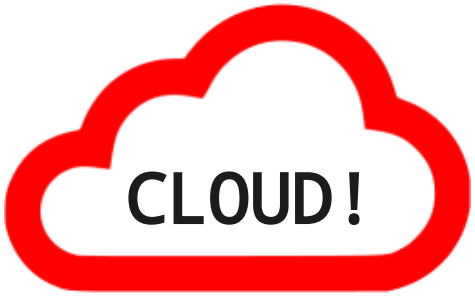
Issue

Disk space!

Issue

share with others ?

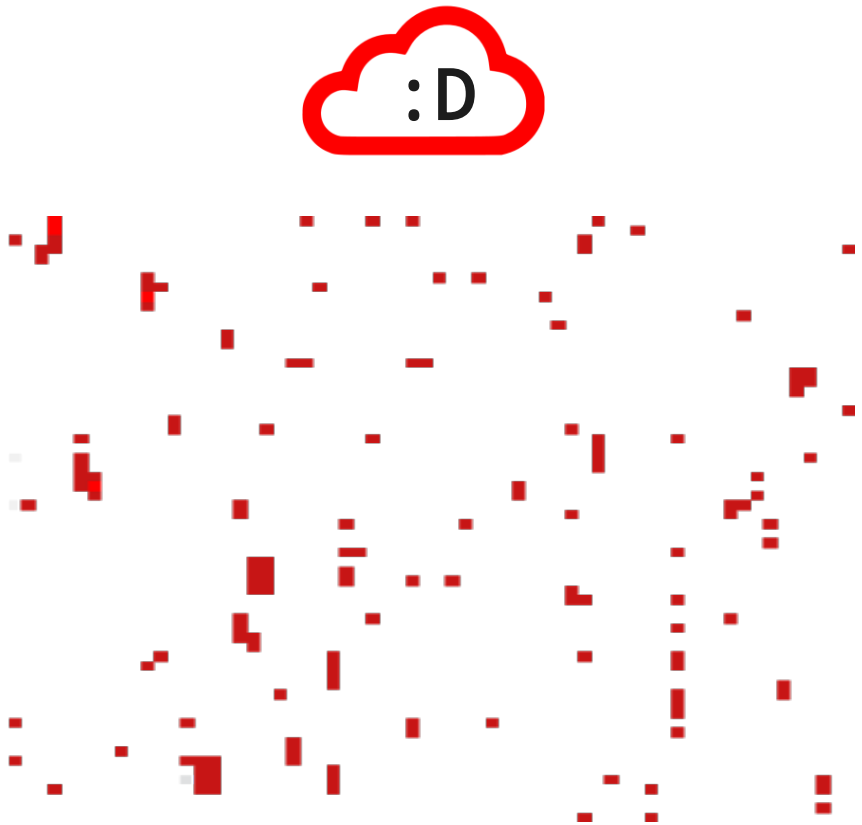
Solution

Lets move it to 

:)

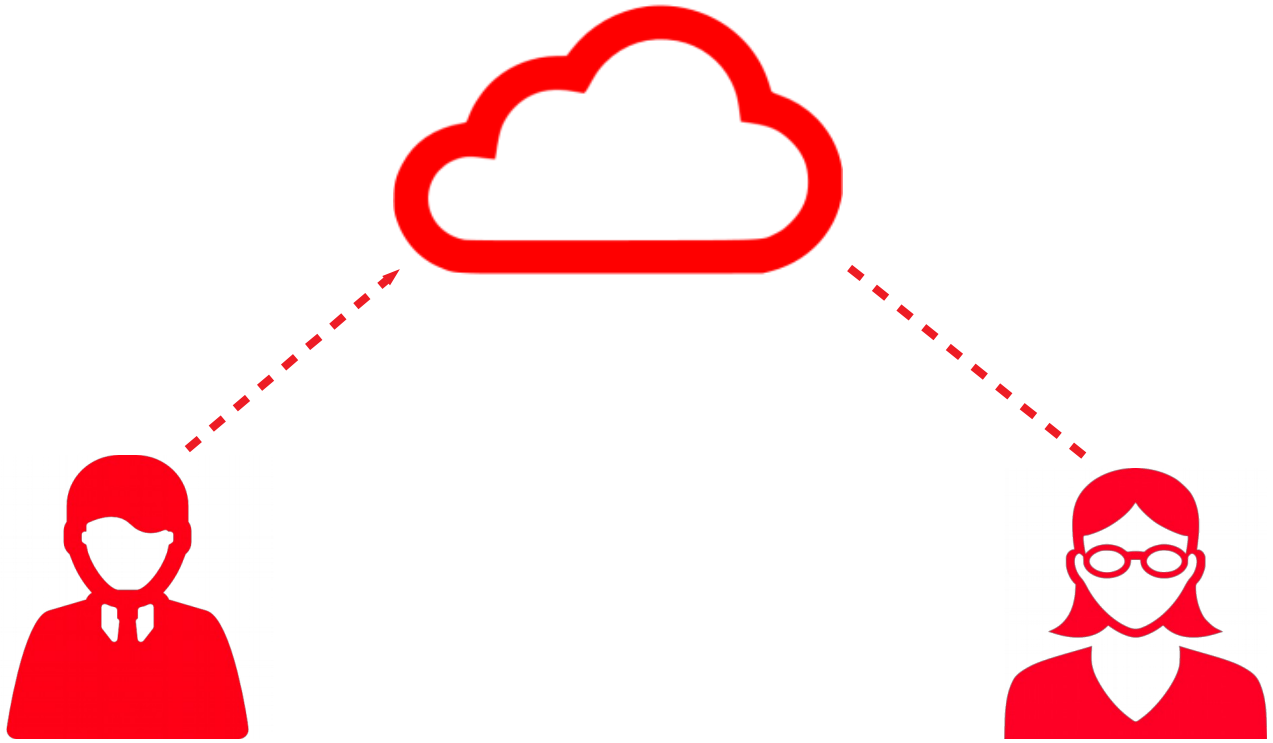
Issue

What if cloud rains ?!!



Issue

Lets collaborate!



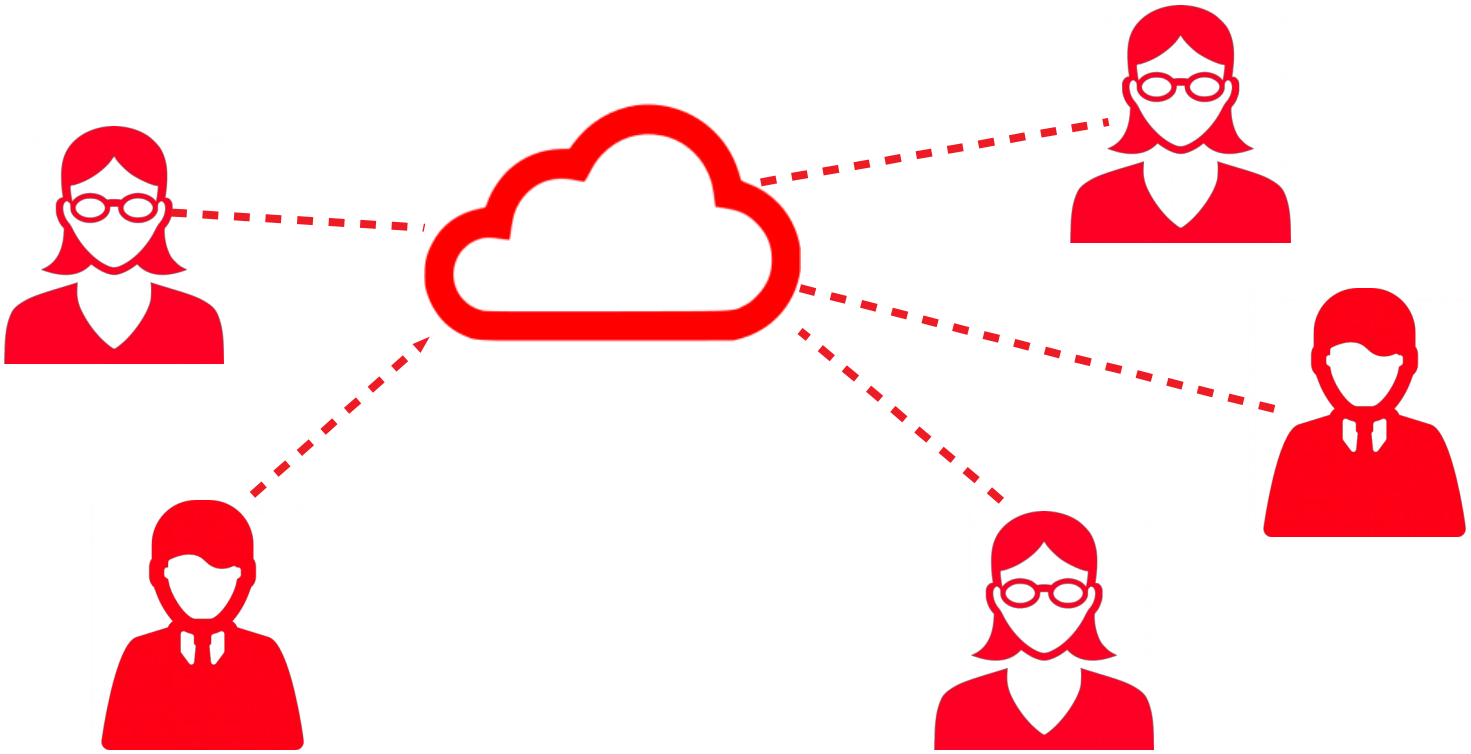
Issue

Both changed same file and same feature

How do you combine both works?

Issue

What if multiple people?



Other issues

- Centralised
- Hackable
- network issue
- Blah Blah issue

Oh NO!!

Saviour – Version Control System

Version control system?

Magician who keeps track of all the versions / modifications in your project folder.

- what changed?
- when it got changed?
- who did the change?

Version control system..

He is little shy, he is not seen directly in your project folder.

But he will be there and tracking all your activities.

You can ask question like, how was my repo last 2 days ago or a week ago or a year ago !!

Version control system?

Wait. Thats not at all.

He also helps in sharing your repository with others and collaborate with your.

And he tracks their changes also.

Version control system

There are multiple VCS available.

how to decide which one to use?

How to decide ?

- Personal preference (like OS, editor)
- Checkout options and decide
- How easy to learn and use
- Free (as in freedom)
- Open source

Git

Git is one of those VCS

Git - history

- Linus Torvalds in 2005
- He is also the creator of linux kernel

Git

- Git was created to handle / maintain linux kernel project
- Linux project has more than 20 mn lines of code
- More than 3000 lines of code added every day
- More than 1000 devs involve in a major version release

- He wanted to collaborate quick and painless.

Git

Will git handle my tiny project ?

Git

- Its a distributed version control system
- There is no central repository, evryone who works on the project has their own repo
- You can interact with git without any network connection

Git

Lets dive in to git

Git - install

```
$ sudo apt install git

$ git –version
```

Git - initiate

Initiating a project with git

```
$ git init my_project
```

Add git to existing project (inside your project)

```
$ git init
```

Git – where is it?

He is very shy!! He is hidden

```
$ ls -a
# .git
```

Git

Lets start the same math library project

```
$ git init math
```

Requirement Assumption

- sin
- cos
- tan

Git - status

At any given time, if you want to seed status of your project

```
$ git status
```

Git - Ask to track file

Git won't track automatically, you have to tell

```
$ git add math.c
```

(Can't git automatically track by default ? Well, some file I don't want to collaborate with others. Like password files or reference files)

Git - stage

Git can track now

Git - status

Is it really tracking? Lets check. Modify some content

$ git status

Ohh!!! You are changed now!! :(

Git - diff

What changed from the time you told me to track you?

```
$ git diff <file_name>
```

Git - commit

Once you are satisified and want to tell git to consider this as version, commit it

```
$ git commit
```

(give a short and meaningfull message)

Git - config

Git is little strict.! He want to know who you are.

Tell git who you are

```
$ git config --global user.name “my name”
$ git config --global user.email “my email”
```

Git - commit

Shortcut command for add and commit with message

```
$ git commit -a -m “My commit message”
```

Git - log

to see my versions/snapshots history

$ git log

- A unique identifier to a commit/snapshot
- Author who committed
- Date and time of commit/version/snapshot taken

Git – diff

We can use git diff to compare 2 commits/snapshots

```
$ git diff <id_2> <id_2>
```

Git - checkout

(git time traveller)

You can go back to your commit/snapshot and look around for changes

```
$ git checkout <version/branch>
```

Git - ignore

If you want git to ignore some file from tracking, then you can add them in .gitignore

```
$ vi .gitignore
```

Git - branch

Horizontal/parallel development

$ git branch

Git - branch

Creating a new branch and

```
$ git branch <branch_name>
```

switching into new branch

```
$ git checkout <branch_name>
```

Shorthand command

```
$ git checkout -b <branch_name>
```

Git - branch

Lets implementation math library in float

Git - branch

Delete a branch

```
$ git branch -D <branch_name>
```

Git - merge

That float implementation looks good. Lets merge it to main

```
$ git merge <branch_name>
```

Once we merge feature branch comitts will appear in your main branch.

Git - conflict

For certain extent git can intelligently merge feature branch history to main history for you.

If there are changes to same file and same line, git will leave option to us which one to use.

Git - conflict

In conflict file we can see which line is conflicting and from which branch.

We can keep either the feature branch changes or main branch feature or both

Once conflicts are resolved, we have to tell git that you are resolved the conflicts. By

```
$ git add <conflct_file>
```

(just) git commit [git will autopopulate message in commit file]

Git

How to collaborate with other?

Git - clone

Copy = copy files

Clone = copy files + git history + add a link remote

You give a copy of your project to your partner with git history.

Now he/she can start working on new feature or fix bug in your project remotely.

Git - clone

you can clone repo locally also.

$ git clone <repo_to_clone> <new_repo_name>

Inside <new_repo> you can see previous work history.

It will also have info on where it is cloned from.

$ git remote

Git - remote

In our main repo we don’t have any remote. We can add one for collaboration.

Lets add a remote link to our main repo

```
$ git remote add <remote_name> <remote_path>
```

Git - push

```
$ git push <remote_name> <branch_name>
```

Push/copy feature_branch to original/remote repo

Once it is success you can go to to your original repo and list branches to see newly pushed branch. You can checkout to that branch and see the changes also.

Git - push

I like the branch and did some changes and committed in the feature_branch. And checkout to master branch.

Meantime person 2 also did some modification, commits and tries to push.

But git will reject. It says there is a change in remote repo for your branch, first pull the changes and then do the update/edit.

Git - pull

So how to pull?

```
$ git pull origin <branch_name>
```

Once we pull remote code and resolve conflict if any. then we are ready to push.

```
$ git push <origin_name> <branch_name>
```

Git – remote, push

Similarly we can do the same from main repo by committing a change. Adding a remote repo and pushing it to remote.

Git – remote service

There are lot of options for hosting remote repo. Each provide different feature.

Most popular are:

GitLab, GitHub and bitbucket

Git - github

Lets create a github or gitlab account and take a tour of

- create account
- search some opensource project
- see commits, contributors, issues, PR, discussion etc, etc.

Git - github

Lets create a repo in remote service (github) and link it to local repo.

- Push
- Pull
- Clone

Git – github pages

Github pages

Websites for you and your projects, directly from your repositories.

https://pages.github.com/

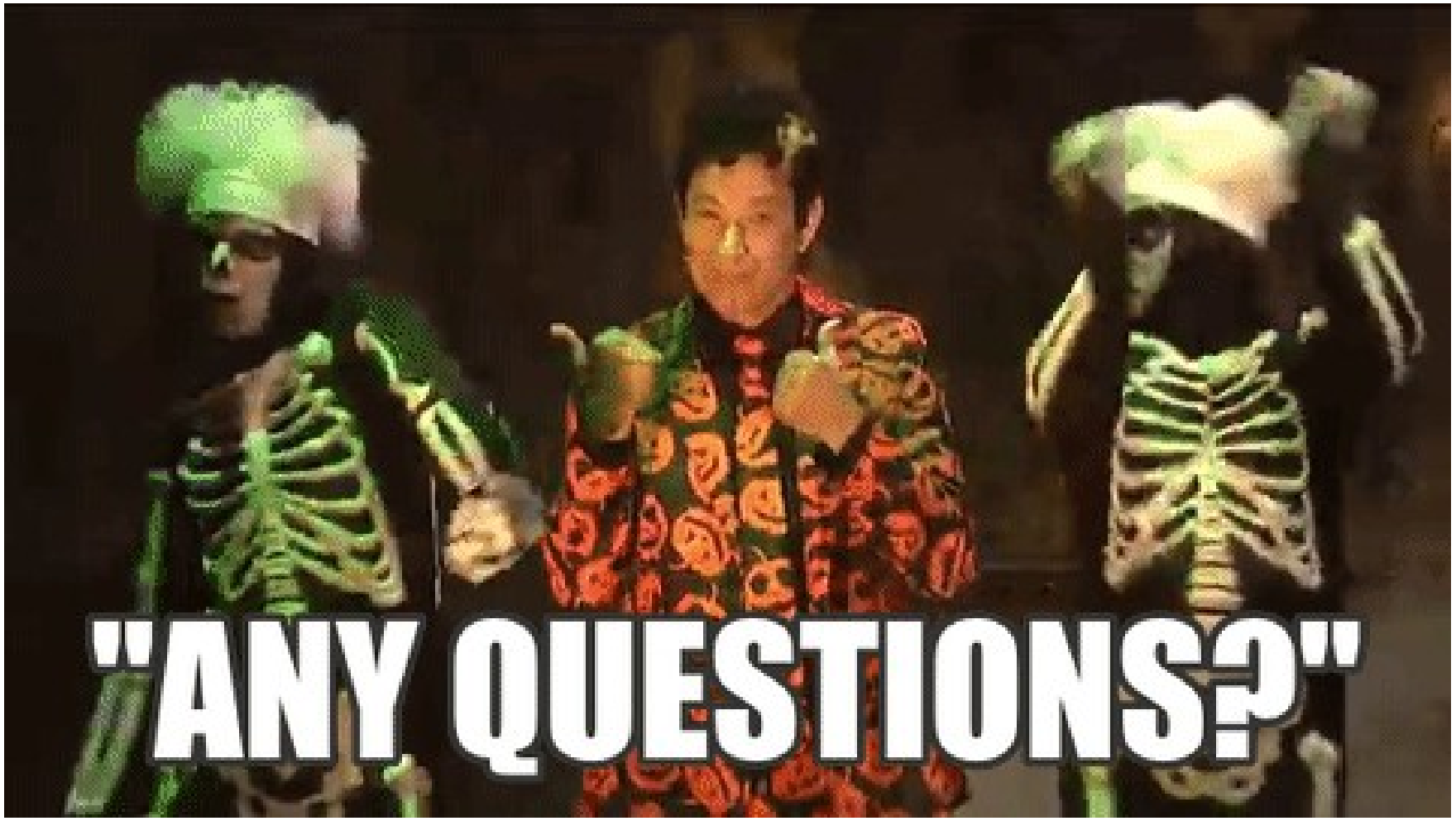
Git - references

https://git-scm.com/docs

https://try.github.io/

https://learngitbranching.js.org/

Questions ?



Thank you