GIT

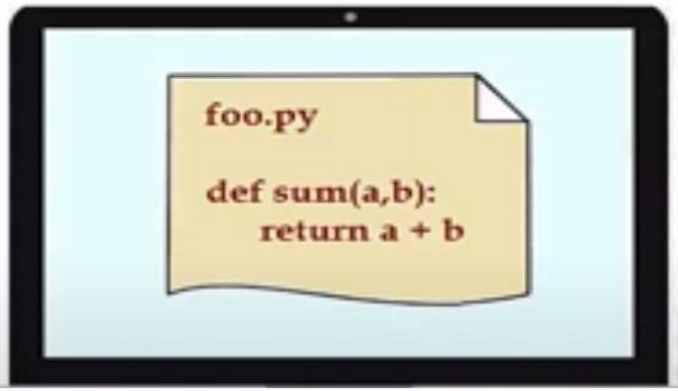
LO 6

Objective

After attending this session, you should be able

- Introduction of GIT
- Installing GIT
- Basic GIT commands
- Undoing/reverting/resetting file changes
- create/merge/delete branches

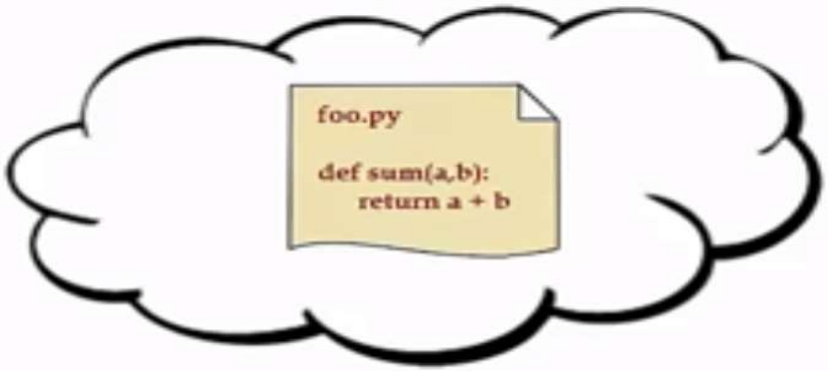
Problem # 1

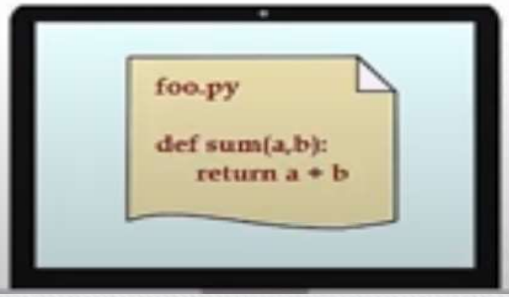


Hard disk failure or Computer crash

Code Lost

Solution



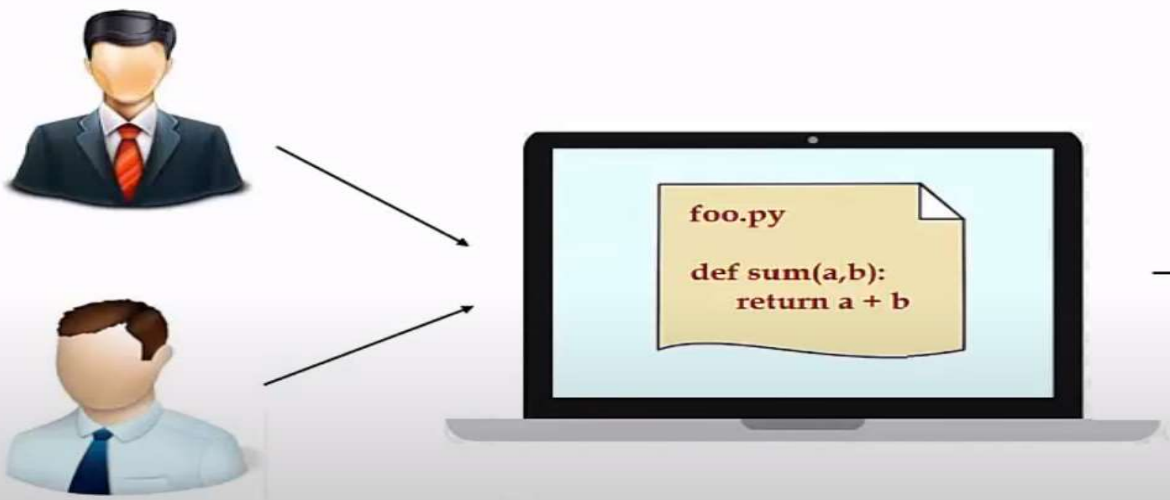


Hard disk failure or Computer crash

Sync it up from cloud

Solutio

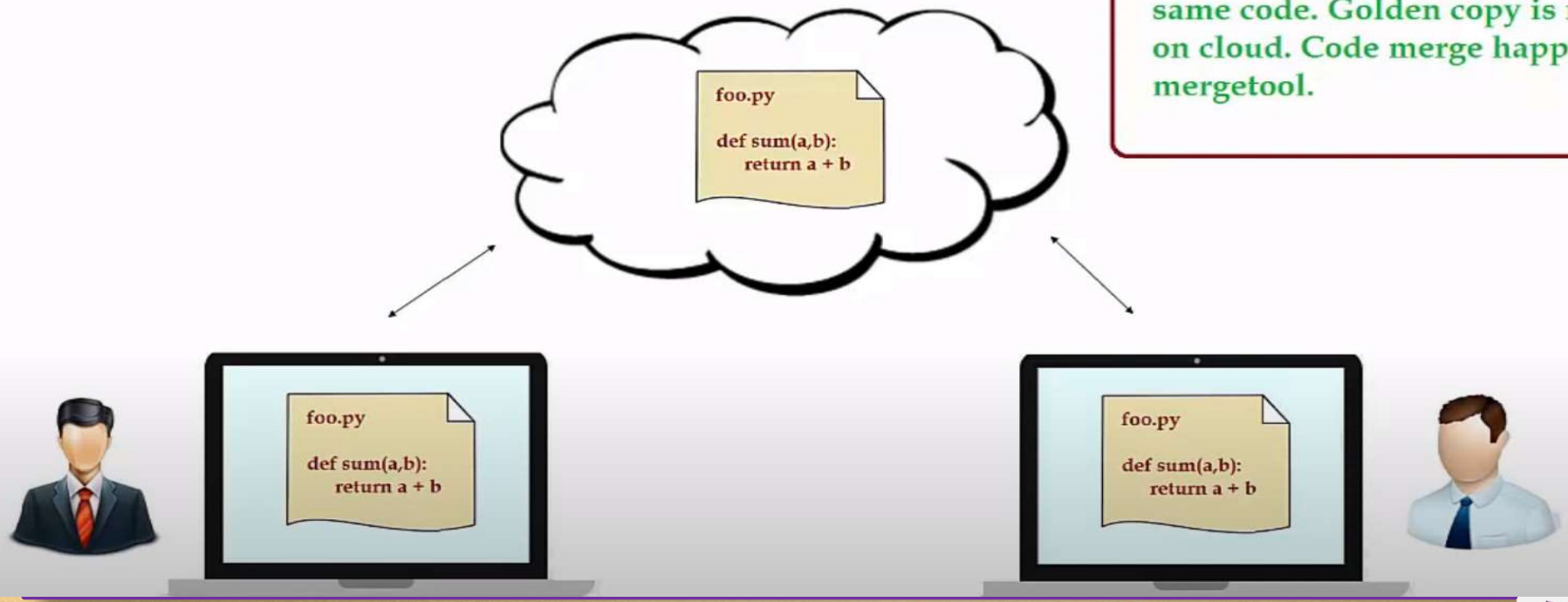
Problem # 2



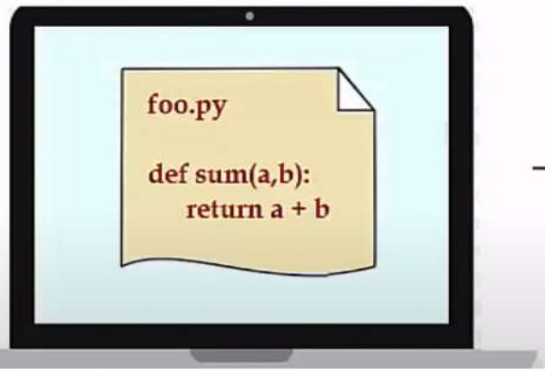
Hard for multiple people to work on same code

Solution

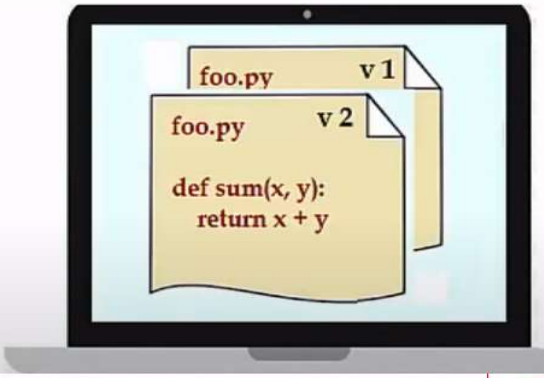
Multiple people work on same code. Golden copy is maintained on cloud. Code merge happens using mergetool.



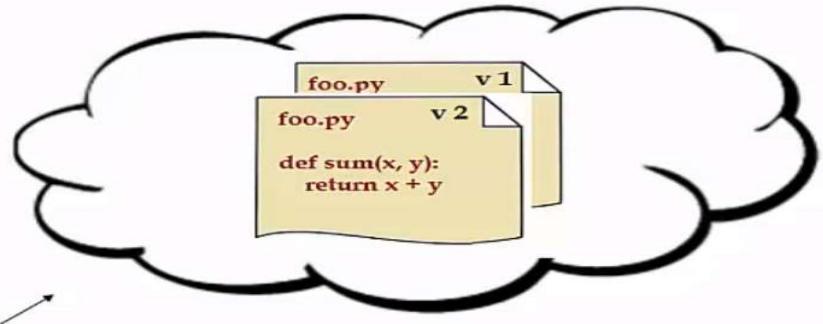
Problem # 3

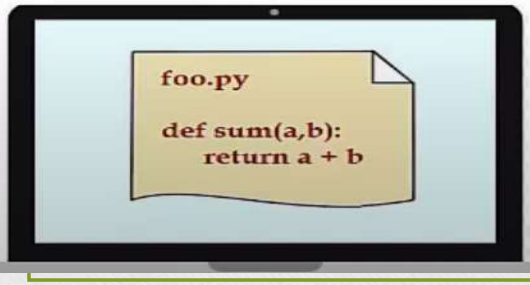


Difficult to track changes
Revert back to previous version



Solution





VCS maintains version history,
you can revert to any of the previous state of your code

Version Control Systems

- GIT
- SVN
- Taravault
- Beastack
- Bitbucket
- Lauchpad

Why Git is by far the best?

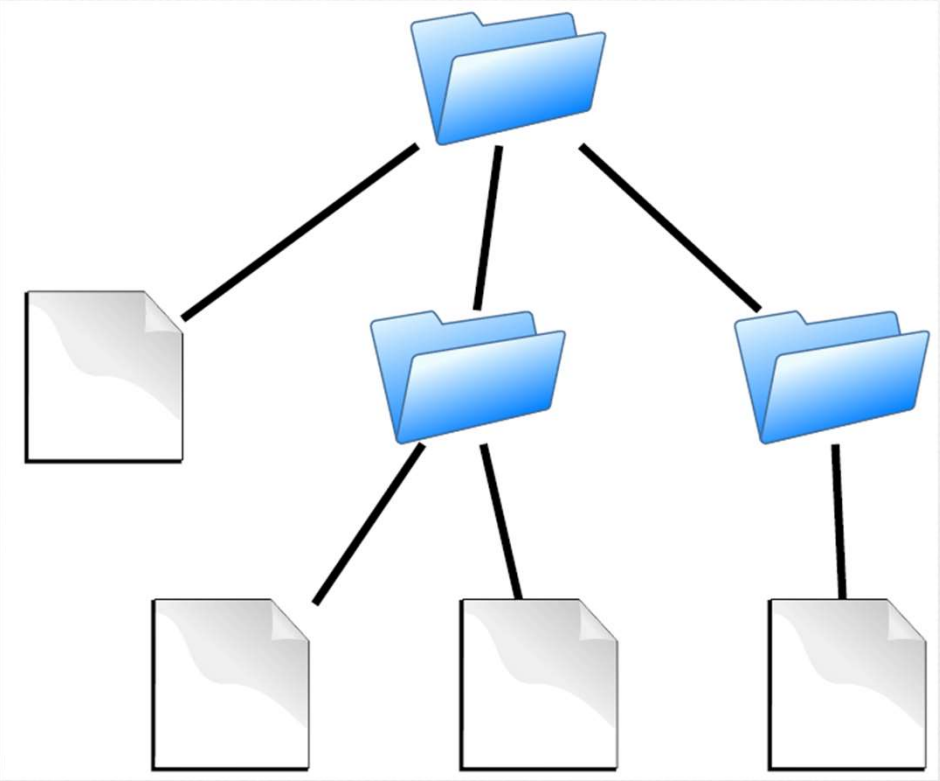
Why Git is by far the best?

- Distributed version control system : version history maintain on local computer and on cloud
- Github: very popular website to host code is based on git

Source code

contains

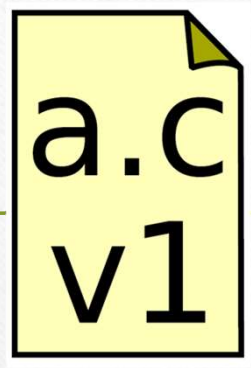
- Directories
- Files



is the substance of a software configuration

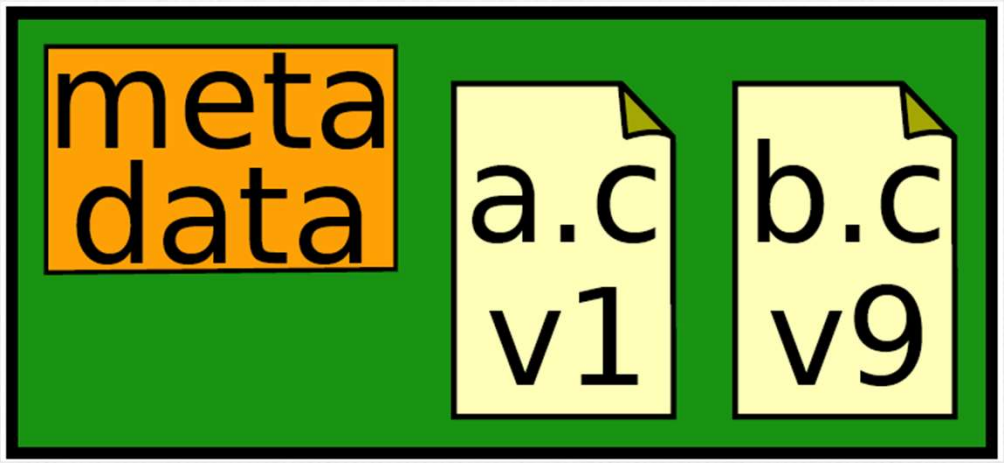
http://edgyu.excess.org/git-tutorial/2008-07-09/intro-to-git.pdf

Repository



Contains

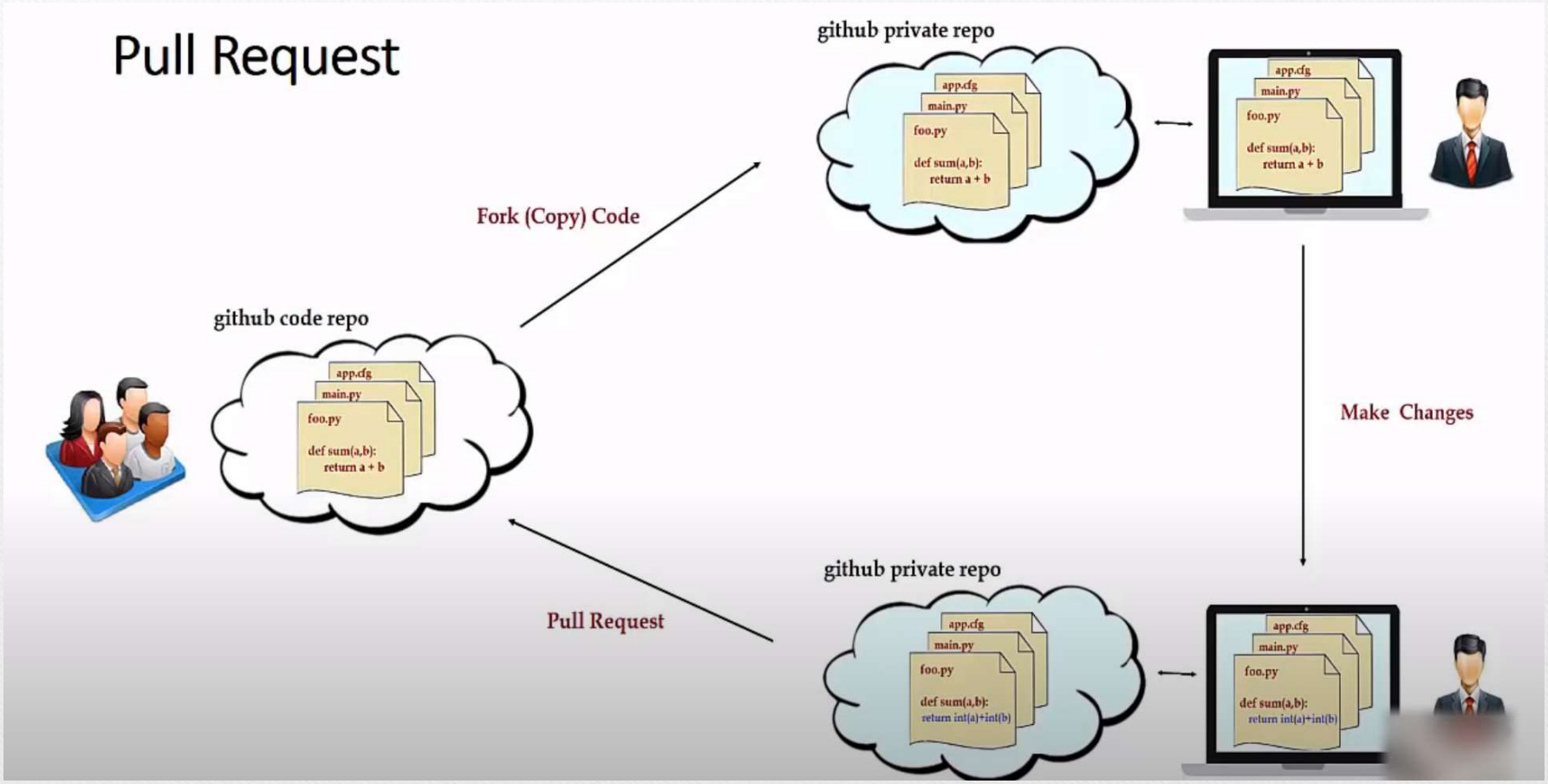
- files
- commits



records history of changes to configuration

http://edgyu.excess.org/git-tutorial/2008-07-09/intro-to-git.pdf

Pull Request

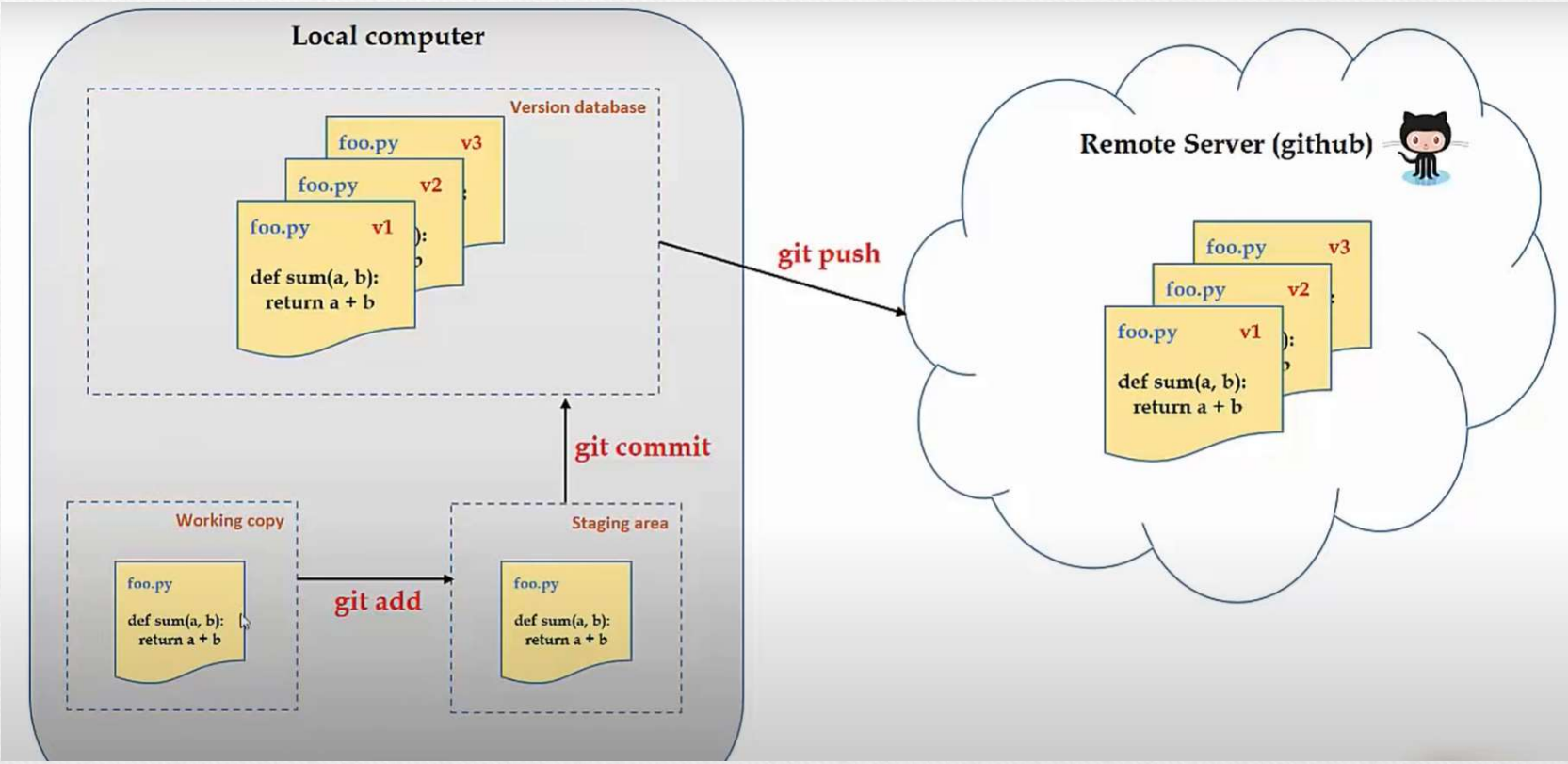


Installing Git

- https://git-scm.com/downloads
- Launching git-bash / git command from command prompt

Basic command: add, commit, push

- Signup on github.com
- Create new repository (new test-public,readme file)
- Edit readme file
- Git clone command (git clone URL)
- Pycharm – new code test program
- Older commands:
  - Git status
  - Git add (staging area)
  - Git commit –m 'first version of hungry code' (still on local computer)
  - Git log (show all commit history)
  - Git push (push all code on github repo)


Local computer
Version database
foo.py v3
foo.py v2
foo.py v1
def sum(a, b):
return a + b
git push
Remote Server (github)
foo.py v3
foo.py v2
foo.py v1
def sum(a, b):
return a + b
git commit
Working copy
foo.py
def sum(a, b):
return a + b
git add
Staging area
foo.py
def sum(a, b):
return a + b

- Undoing file changes (uncommitted changes)
  - Git checkout - - file.py // one file only
  - Git checkout - - .// all files
- Retrieve committed file (committed changes)
  - Git revert commit-id //using git log
- Resetting code changes (very powerful-take to any where in history as per commit-id)
  - Git reset –hard commit-id //be very careful

Branches

- Give provision to do certain experimental changes in code by creating branch
- Create branch
  - Git branch // showing branches
  - Git branch thirsty //thirsty branch create, master default
  - Git checkout thirsty //to activate branch-show in green colour-change code
- Merge branches
  - Git checkout master //change back to master-no have any changes
  - Git merge thirsty //merge thirsty to master branch
- Delete branch (dummy branch)
  - Git checkout –b dummy
  - Git branch
  - git checkout master
  - Git checkout –d dummy

Head is reference to most recent commit in current branch (git show HEAD)

Groups of Git commands

- Setup and branch management
  - init, checkout, branch
- Modify
  - add, delete, rename, commit
- Get information
  - status, diff, log
- Create reference points
  - tag, branch

Summary

- Introduced of GIT
- Installed GIT
- Basic GIT commands
- Undoing/reverting/resetting file changes
- create/merge/delete branches

Himanshu Patel, Instructor
Saskatchewan Polytechnic
email: patelh@saskpolytech.ca
office 100.1, phoenix building