Version Control System
Git

Bharat Bhushan Verma

Version Control Systems

- Version Control System (VCS) / Source Control Management (SCM)
- File Naming (XYZv1.txt, ABC_final.docx)
- MS Word: Track Changes, Redo/Undo CMD + Z/CTRL + Z
- Wikis (Rolling Back feature)

History of Version Control Systems

- 1972: AT&T releases Source Code Control System (SCCS) with Unix
  - Original Version with new changes
- 1982: Revision Control System (RCS) – Open Sourced
  - Latest Version with change history
- 1990: Concurrent Versions System (CVS)
  - Multiple files and Multiple users managed together
- 2000: Apache Subversion (SVN) / Bitkeeper SCM
  - Track images and other files, collectively track directory
  - Bitkeeper had distributed version control
- 2005: Git
  - Replaced Bitkeeper for Linux kernel code
  - Open source and distributed version control
  - 2018 Git is bought by Microsoft

How Popular is Git

- More than 200 million repositories
- Around 50 million users
- Quora pushed 140 versions of website within 24 hours.
- Widely used version control system
- Alternatives: SVN, Bitbucket, Perforce, Mercurial

What is Distributed Version Control

- No Central / Master Repository
- Different users have their own version of repository
- Changes are change sets and versions of a document
- No network access required
- Forking of Projects
- All repositories are equal

Install Git

- Mac has preinstalled version of Git. If not, then you can use
  - Installer https://git-scm.com
  - Homebrew https://brew.sh
    - brew install git
- pwd
- which git
- Git –version

Basic Configuration

- System Level: /etc/gitconfig — git config --system
- User Level: ~/.gitconfig — git config --global
- Project Level: project_name/.git/config — git config

- Set username: git config –global user.name “ABC XYZ”
- Set username: git config –global user.email ABC@XYZ.com
- Git config –list
- Git config user.name
- Git config user.email
- Cat .gitconfig on home directory

Git Help

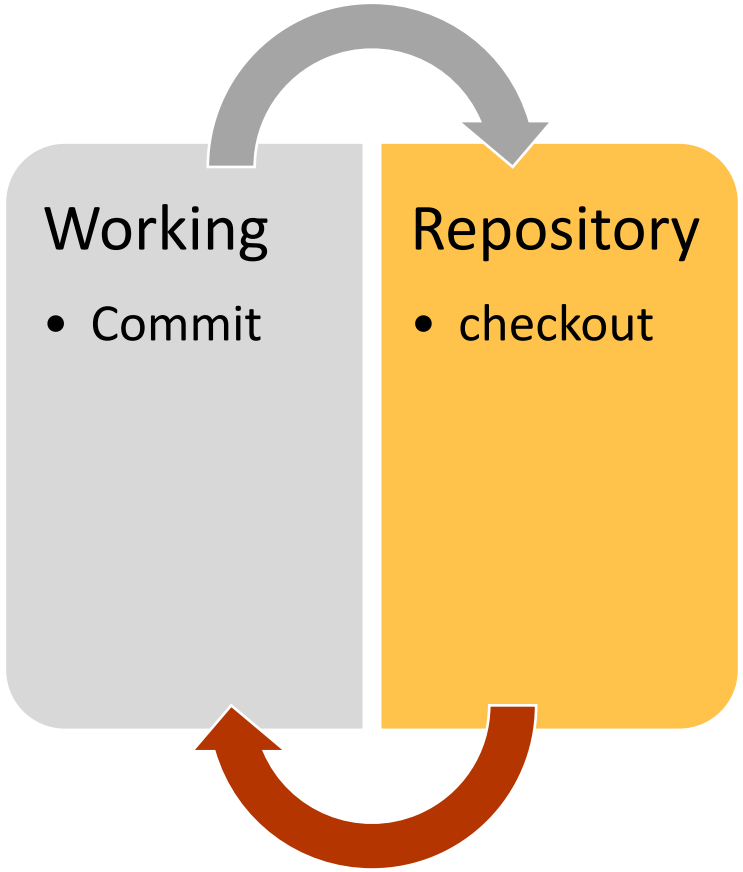
- Git help
- Git help log
- Man git-log
- https://git-scm.com/docs/git-help
- Pro Git https://git-scm.com/book/en/v2
- Videos: https://git-scm.com/videos
- Tutorials: https://git-scm.com/doc/ext
- Community: https://git-scm.com/community

Get Started

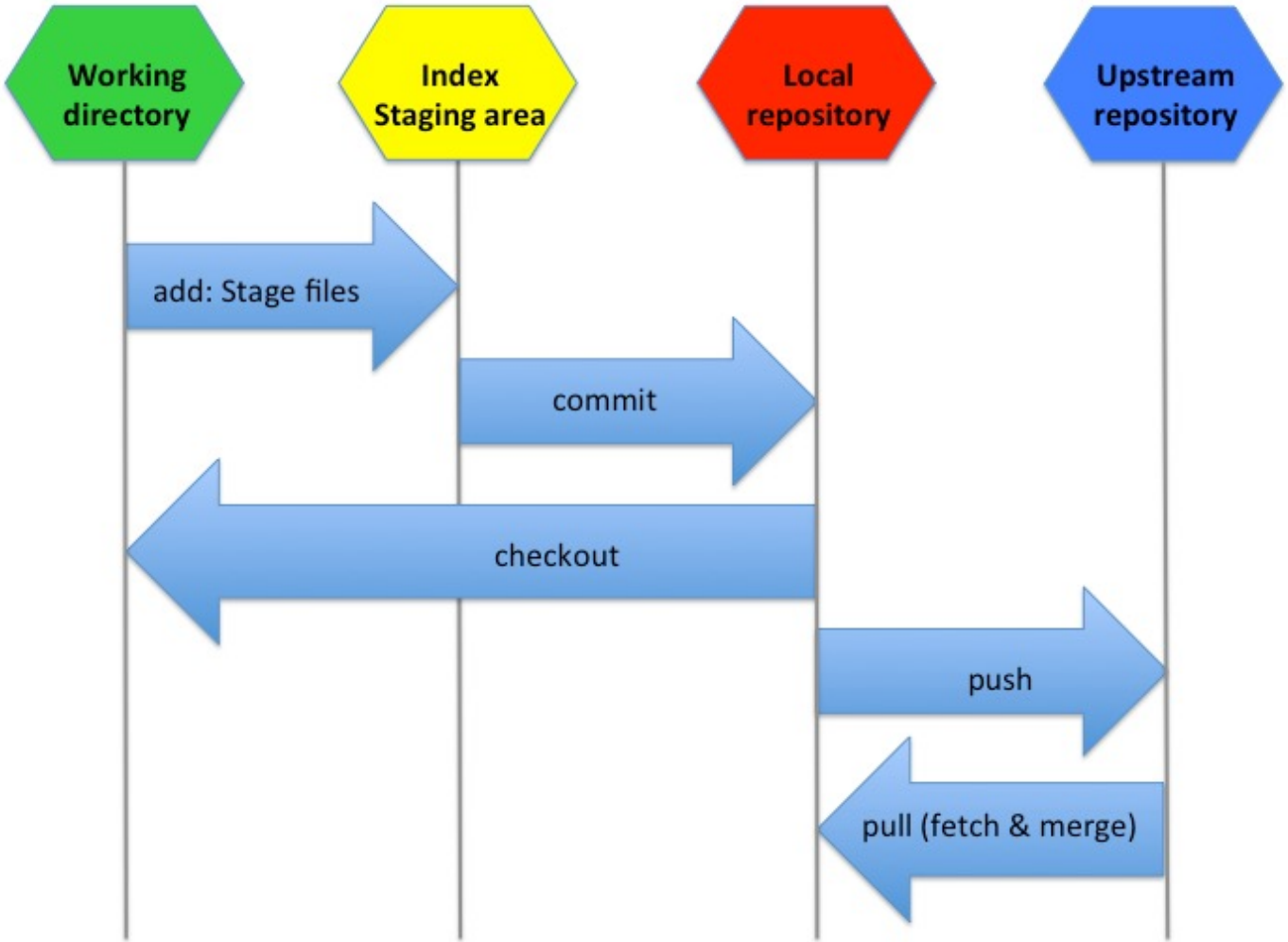
- Git init (Inside the Project Directory, say MBA6693)
- Ls –la
- Ls –la .git
- Cat .git/config

- Create first_file.txt with text “This is my first file” Make Change
- Git status Check Change
- Git add . Add Change
- Git commit –m “Initial Commit” Commit Change
- Git log [-m 5] / [--since=2020-01-25] Check History
- [--grep=“Init”] / [--author=“Name”]

Two Tree Architecture



Three Tree Architecture



Hash Values (SHA-1)

- Git generates a unique hash value for each change set
- An algorithm like SHA-1 converts data into a simple number
- Same data will always equal same checksum (hash value)
- This ensures data integrity
- Changing data will change checksum
- Checksum is a 40-character hexadecimal string
- One cannot change any metadata without changing SHA values.
- Subsequent SHA values depend on previous values as well.

HEAD pointer

- Pointer to current branch in repository
- Last state of repository, what was checked out
- Points to parent of next commit where writing commit takes place
- Ls –la .git
- Cat .git/HEAD
- Cat .git/refs/heads.master
- Git log

Add files

- Create second_file.txt “This is a second file”
- Create third_file.txt “This is a third file”
- Untracked files are not known to git repository
- Git add second_file.txt
- Git status
- Git commit –m “Add second file”
- Git log
- Git Status
- Add and commit third file to repository

Edit files

- Modify first file
- Git status
- Git add first_file.txt
- Git status
- Modify second and third file
- Git status
- Git add second_file.txt
- Git status
- Commit the changes in first and second file
- Git status -> git log
- Commit the third file as well.

View Changes

- Modify first file again by add a few new lines.
- Git status
- Git diff
- Make minor modification in the first line again of first file.
- Git diff
- Make some changes in third file.
- Git diff

Staged Changes

- Git add first_file.txt
- Git diff (Difference between staging and working)
- Git status
- Git diff –staged (Difference between repository and staging)
- Git diff –cached
- Git add third_file.txt
- Git status
- Again check both diff commands
- Git commit –m “Minor changes in first and third file”
- Git status

Undo Changes

- Delete the last two lines of first_file.txt and close the file.
- Git diff
- Git status
- Git checkout – first_file.txt
- Git status

Unstage files

- Modify second_file.txt which is already committed
- Git add second_file.txt
- Git reset HEAD second_file.txt
- Modify third_file.txt
- Git checkout -- .

Using GitHub Desktop

- Download Github Desktop from https://desktop.github.com/
- Install it
- Create a repository on your machine and add/create some files.
- Publish it in remote repository
- Check your github.com account.