Lab 02 - Working With Bash and Git

CS 1XA3

Jan. 15, 2018

Working with Linux Servers: SSH

- SSH (named for Secure SHell) is a protocol commonly used for remote login. You can use it from a command line interface with the following syntax

  ```
  ssh username@server_url
  ```

- Example: I can log into the McMaster Server: mills.mcmaster.ca with my MacID: dalvescb

  ```
  ssh dalvescb@mills.mcmaster.ca
  ```

Try logging into mills.mcmaster.ca with your MacID

Working with Linux Servers: SCP

- SCP (Secure Copy) allows Remote Transfer of files
- To copy a file from a server to */copyto*

```
scp username@server_url:/path/from/file.txt /path/to
```

- To copy a file from a server to local machine

```
scp /path/from/file.txt username@server_url:/path/to/
```

Basic Commands

- File Browsing
  - **cd** */path/to*
    Change Directory to */path/to*
  - **ls**
    List current directory contents
  - **pwd**
    show *Parent Working Directory* (where am I?)
- File Manipulation
  - **cp** *file1.txt file2.txt*
    Copy file1.txt to file2.txt
  - **mv** *file.txt /path/to*
    Move file or directory to */path/to*
  - **mkdir** *Directory1*
    Make a new directory named *Directory1*
  - **rm** *file.txt*
    Removes a file - Warning: no undo

Remark on Manipulating Directories

- When copying or removing directories, you must specify the recursive flag
- **cp** **-r** *dir* */path/to*
  copies a directory *dir* and all its contents to */path/to*
- **rm** **-r** *dir*
  removes a directory *dir* - Warning: no undo

Remark on File Paths

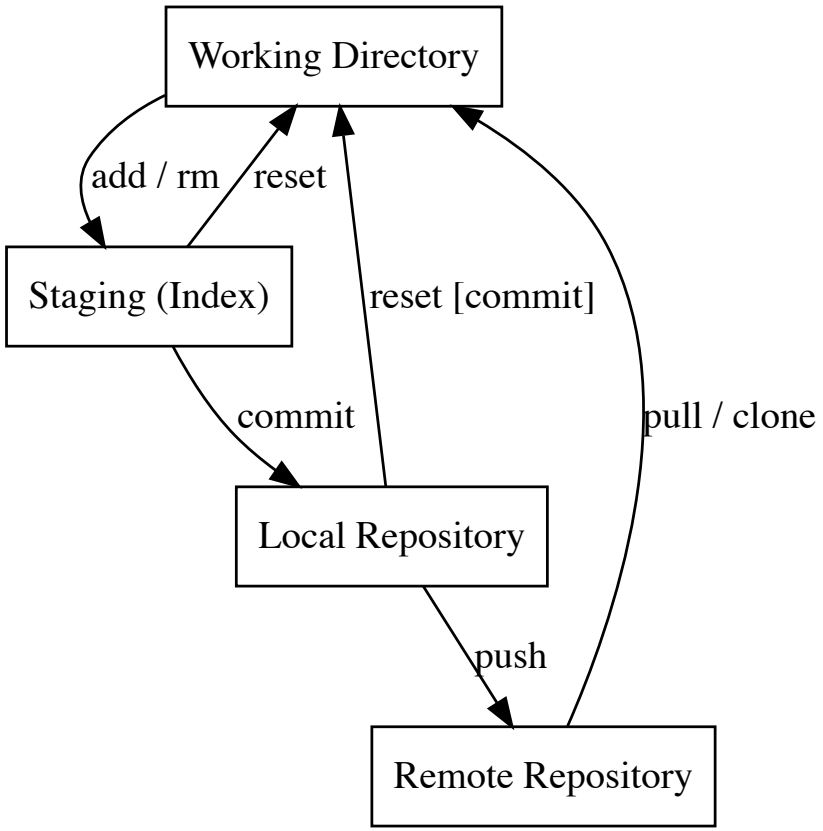
- / is the root directory
- ~ is your home directory
- **cd /path/to** will change the directory to **/path/to**
- **cd path/to** will change the directory to **PWD/path/to**
- **.** specifies the current directory
- **..** specifies the previous directory (try **cd ..**)

Vim: The basics

- Vim operates in two modes
  - Insert Mode: press **a** to enter
  - Command Mode: press **ESC** to enter

- From Command Mode
  - **:q** to quit and **:q!** to force quit without saving
  - **:w** to write (save)
  - **dd** deletes the line the current curser is on
  - **u** for undo
  - **CTRL-r** for redo

Revision Control With Git



- The Working Directory is the local directory you downloaded the repo to
- The Index and Local Repository are files providing version control on your local system
- The Remote Repository is the GitHub server where your repo is kept

Simple Git Commands

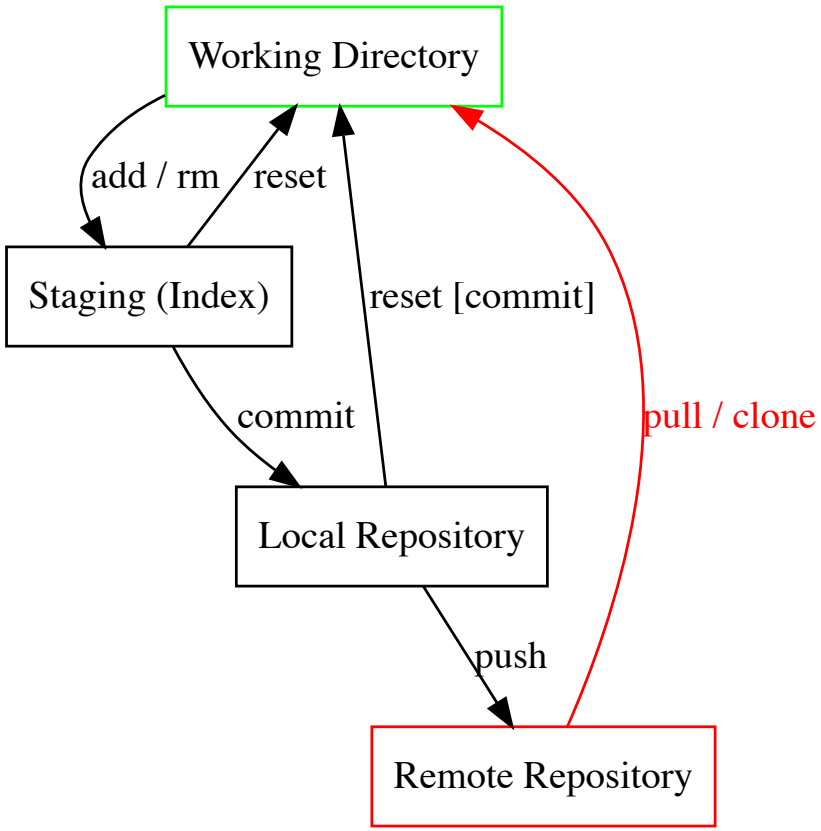
- **git clone** *repo-url*
  Download the code from Remote Repo to current directory
- **git pull**
  Merges code from remote repository to current directory
- **git add** *file* / **git rm** *file*
  First step to adding or removing a file or directory
- **git reset** *file*
  Undo local changes so far (opposite of git add)
- **git commit** *-m "comment"*
  Commit changes to Local Repo
- **git push**
  Push changes in Local Repo to the Remote Repo

Simple Git Commands

These commands are often overlooked by beginners, but are extremely useful

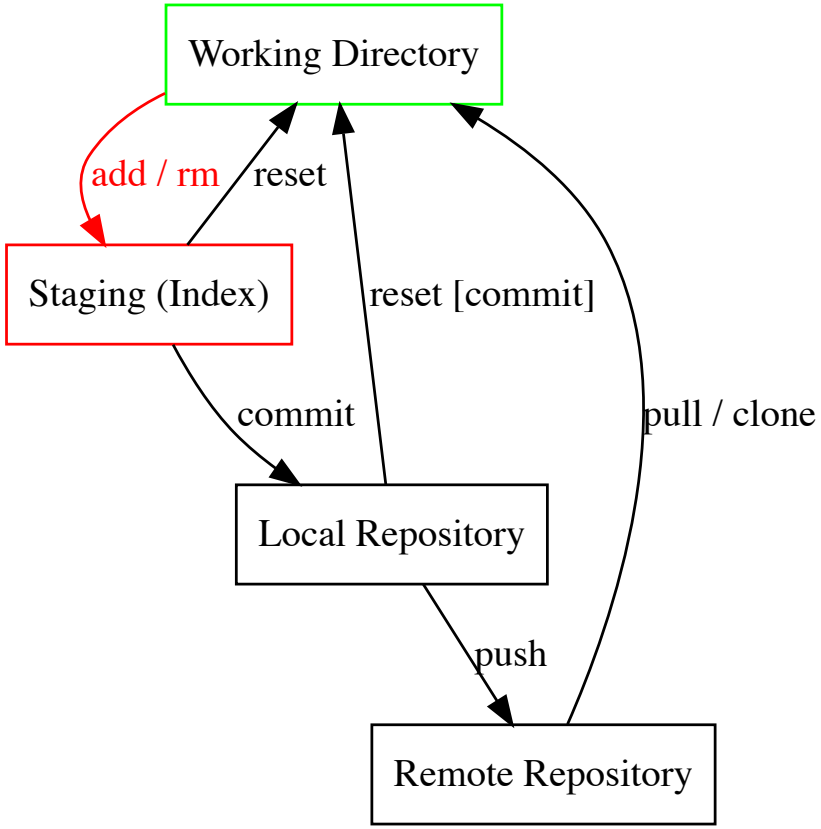
- **git status**
  Shows the current status of staged files
- **git log**
  Displays the log of commits
- **git –help**
  List git commands with descriptions

Git Clone / Pull



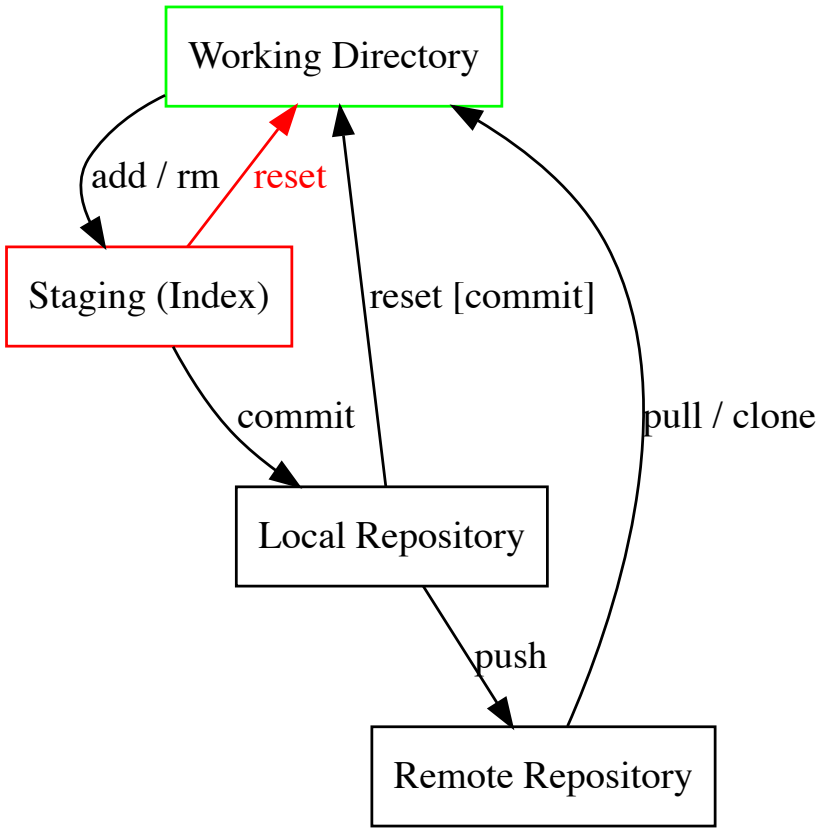
- Use *git clone repo_url* to download the repo to the current working directory
- Use *git pull* to fetch and merge code from the remote repo

Git Add



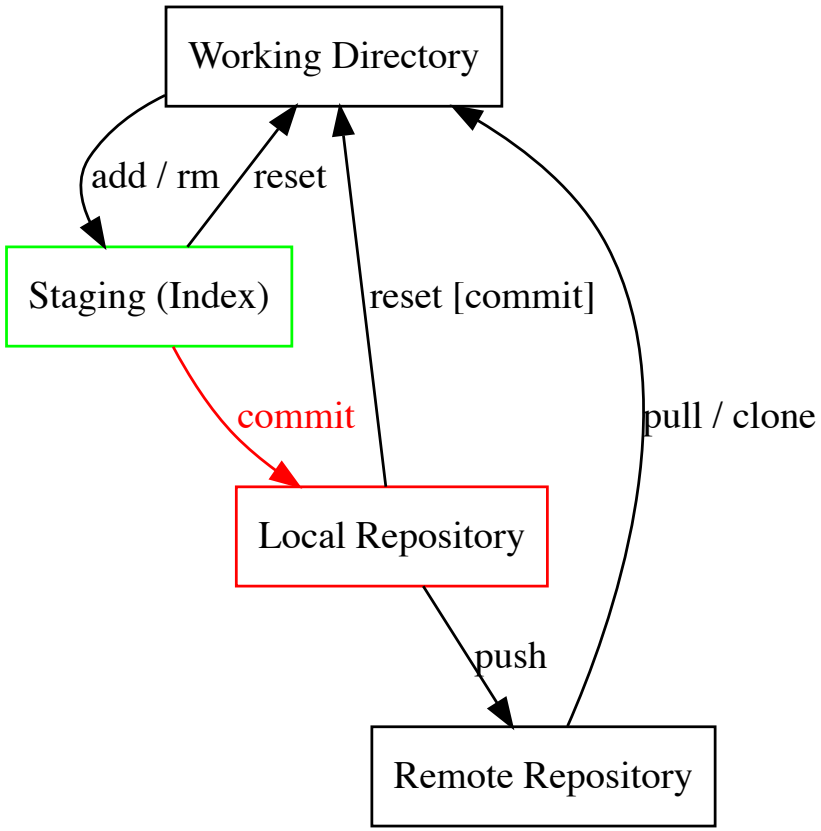
- Create a new directory with *mkdir* and create a README in it
- Use *git add* to stage your new directory and file for committing

Git Reset



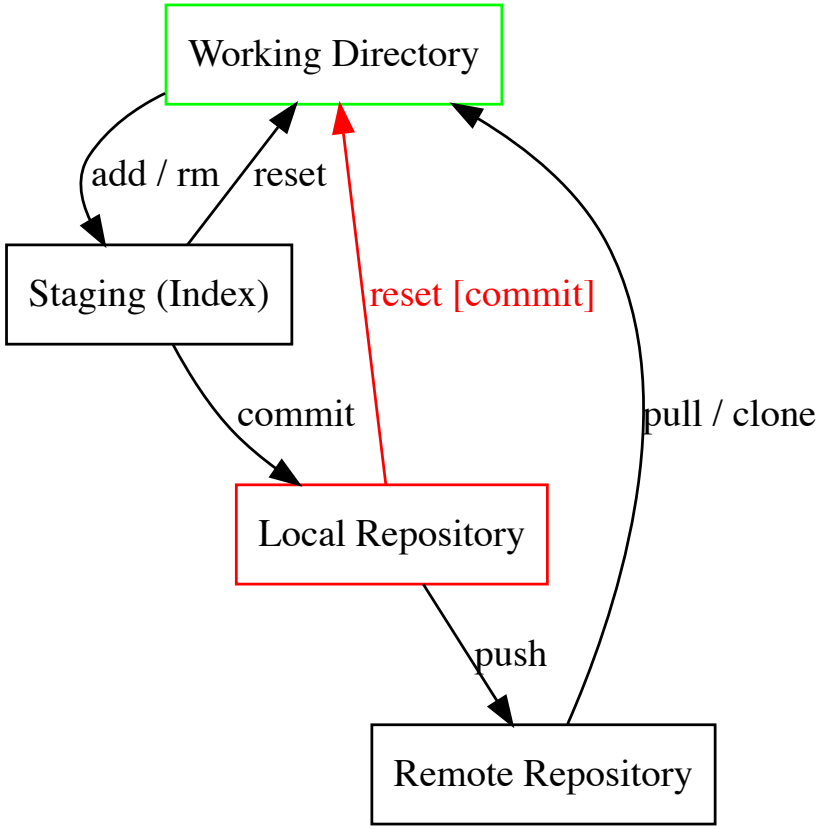
- Screw up staging and don't want to commit?
- Use *git reset* to disregard **git add's / rm's** before the last commit (doesn't change file edits)
- Use *git reset –hard* to change everything back to the last commit (known as **HEAD**)

Git Commit



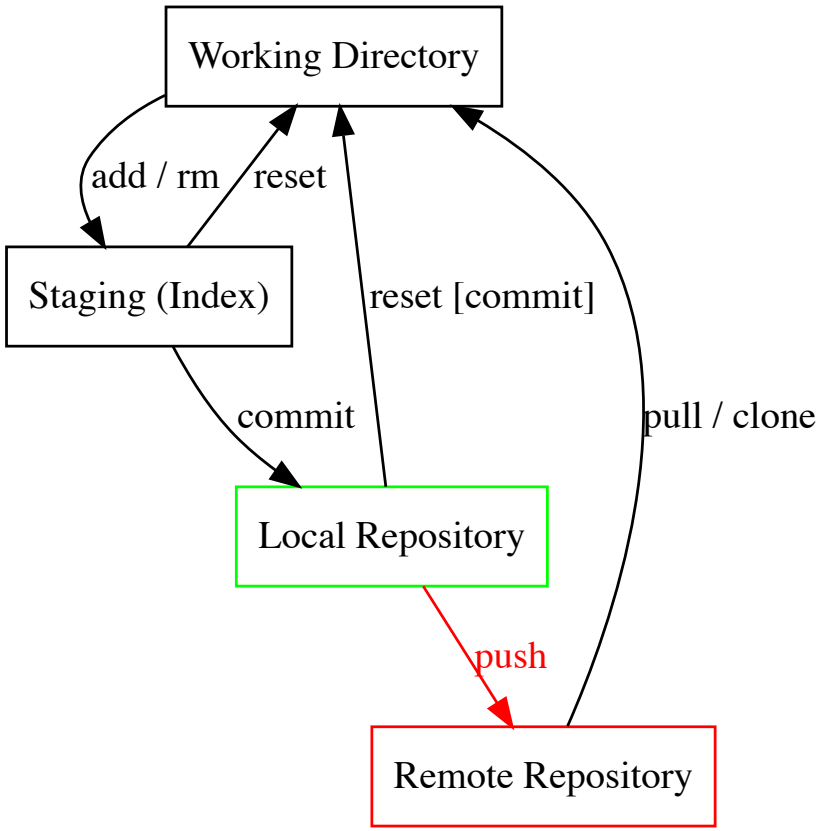
- Use *git commit -m "message"* to commit changes to Local Repository
- **ALWAYS** leave a concise but descriptive Commit Message

Git Reset 2



- Use *git log* to list commits, copy the ID of the commit you want to reset to
- Use *git reset –hard CommitID* to change everything back to specified commit

Git Push



- Use *git push* to send your latest commit to the Remote Repo
- You will be prompted for your GitHub username and password

Some Busy Work

- Add directories Assign1, Assign2, Assign3 with README files inside
- Add a directory Labs
- Inside the Labs directory, add directories
  - Bash1, Bash2
  - Elm1, Elm2
  - Haskell1, Haskell2
- Put a TODO file that lists the date each lab takes place
- Appropriately commit as you go along with descriptive messages
- When you're done, push your changes to GitHub