Version control concepts & GIT Basics

Introduction to version control

Enables

- Multiple people to simultaneously work on a single project.
- One person to use multiple computers to work on a project.

Integrates

- Work done simultaneously by different team members.

Gives access

- To historical versions of project
- If one makes a mistake, can roll back to previous version.

Why Do We Need A Version Control System (VCS)?

- Backup and Restore
- Synchronization
- Undo
- Track Changes
- Track Ownership
- Sandboxing
- Branching and merging

Repositories and working copies

Working copy

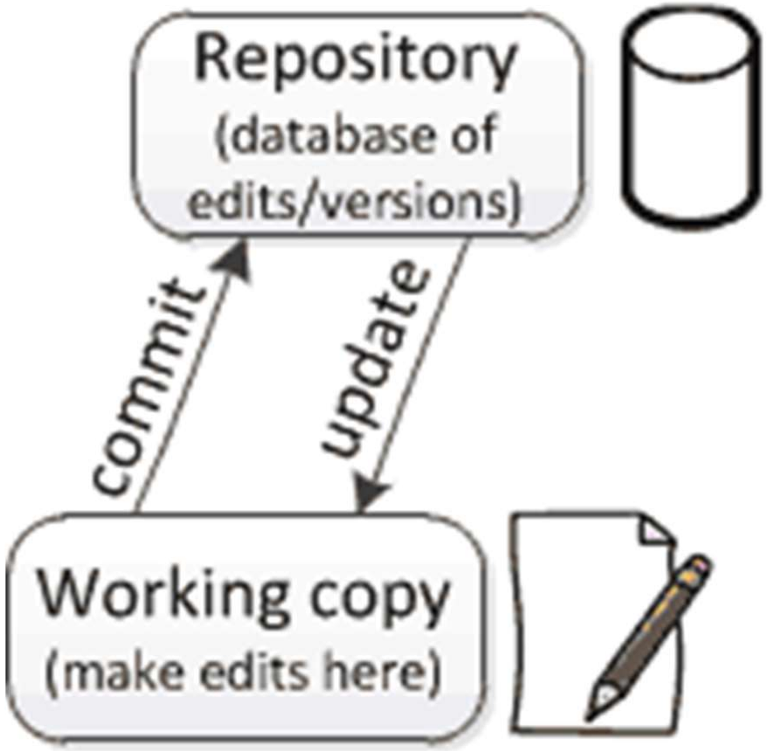
- Personal copy of all the files.
- We changes this copy, without affecting our teammates.

A repo/repository is

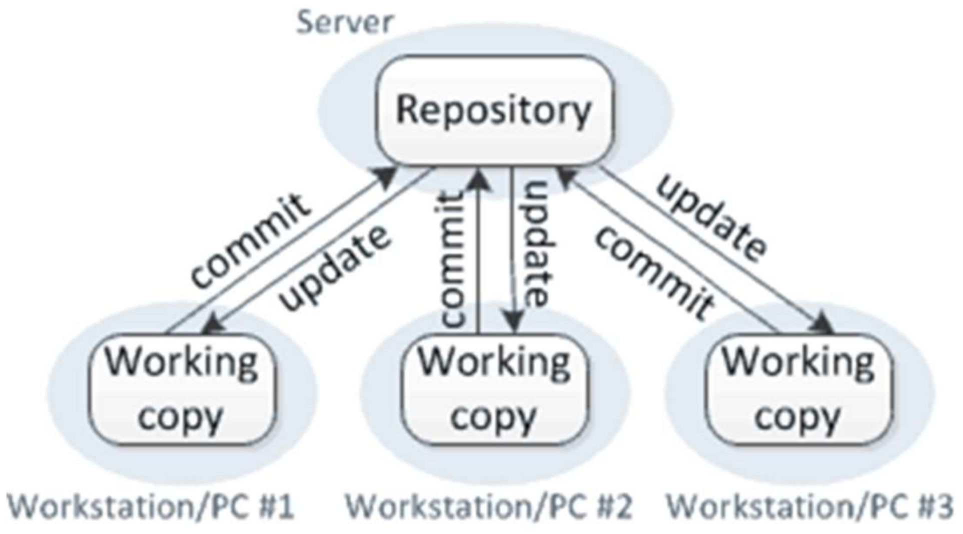
- A database of all the edits and historical versions (snapshots) of project.

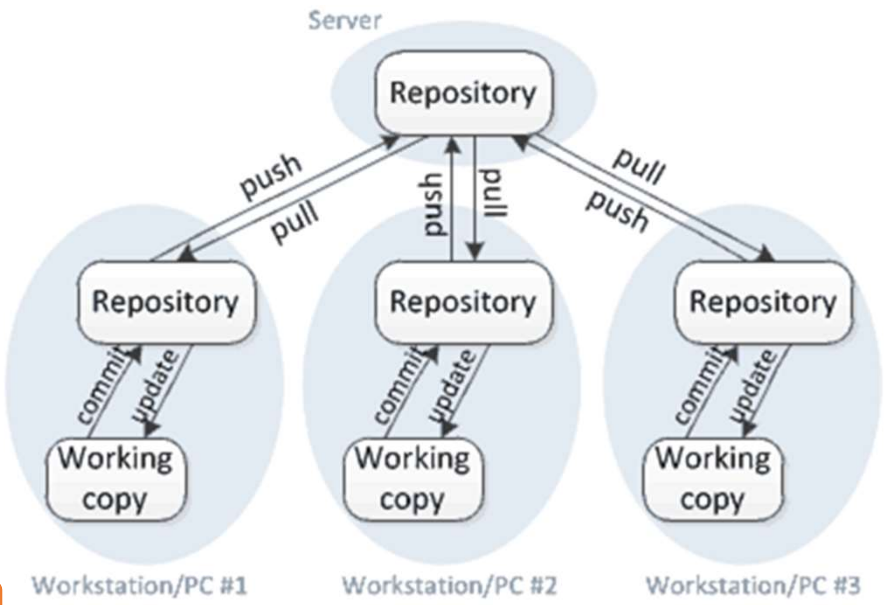
Commit changes to repo

- When we are happy with changes



Centralized vs Distributed version control





Subversion/CVS - Just one repository

Central server is must.

Spoke and hub structure

Local changes are not versioned

Need to communicate with server at each check in/ checkout.

GIT/ Mercurial - Multiple repositories

Can be used Offline

Full history of repository lives on every user’s machine

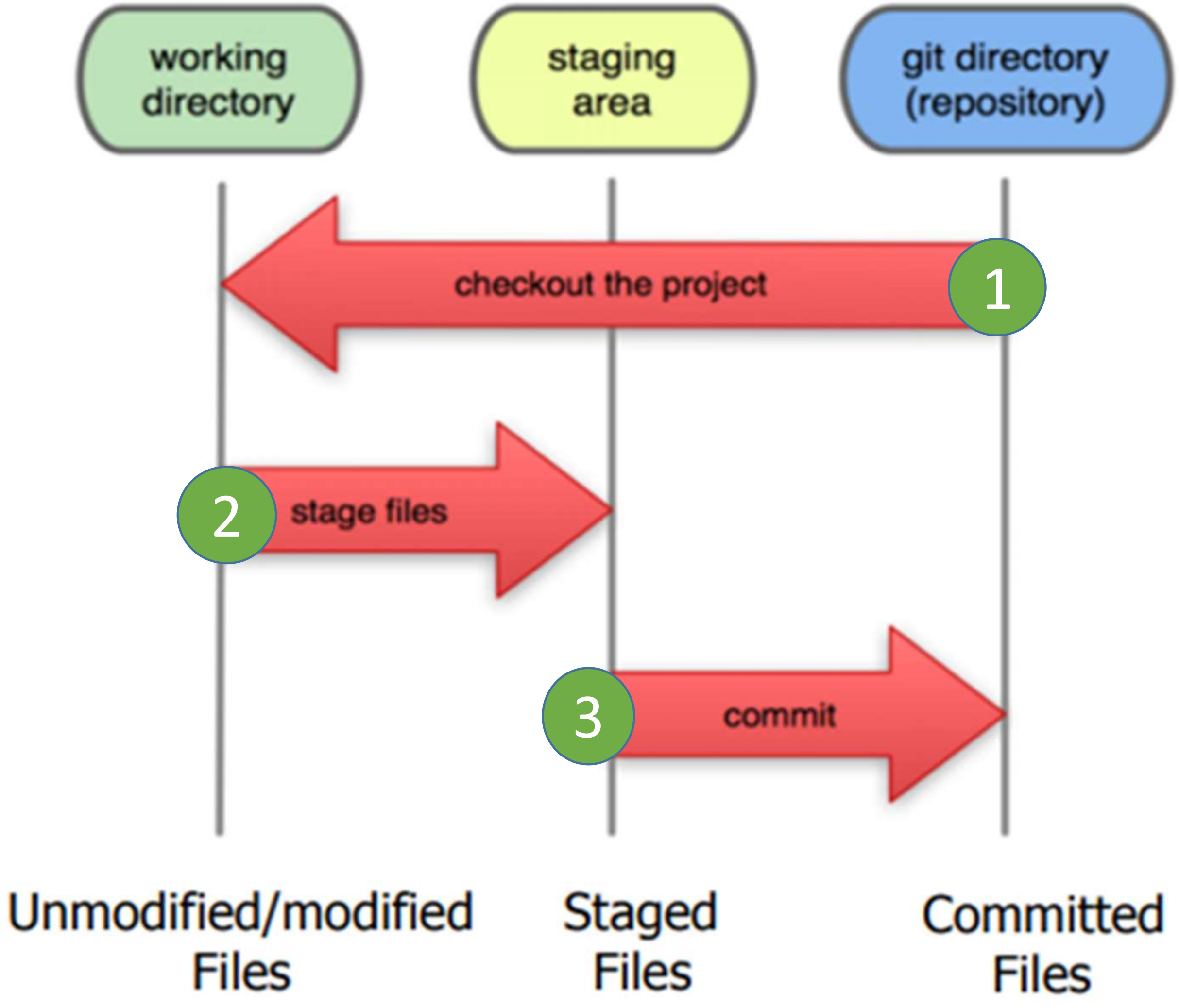
Peer to Peer model

Many operations are local

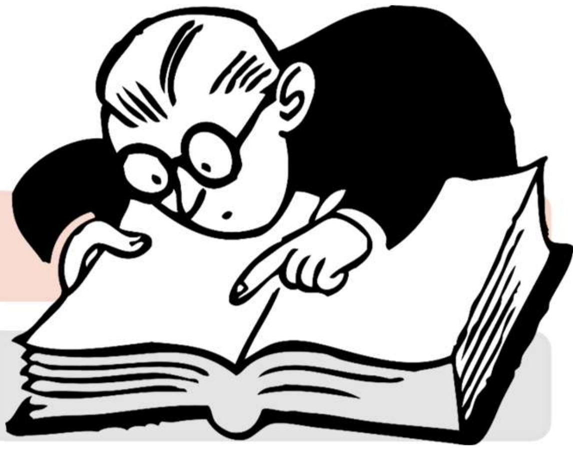
About Git

Created by	• Linus Torvalds - creator of Linux, in 2005
Goals:	• Speed • Support for non-linear development • (thousands of parallel branches) • Fully distributed • Able to handle large projects efficiently
Git website	• http://git-scm.com/

Local git operations

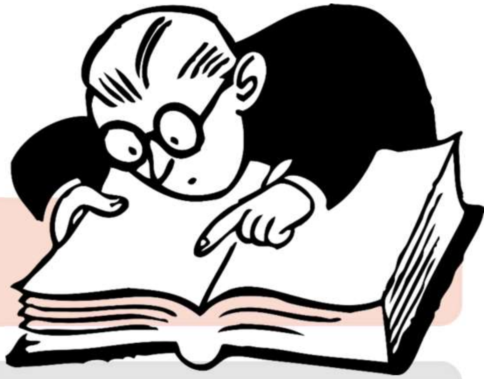


Terminology



Repository	• The database storing the files.
Server	• The computer storing the repo.
Client	• The computer connecting to the repo.
Working Copy	• Our local directory of files, where we make changes.
Master	• The repository's main branch.
Clone	• Copies an existing git repository, normally from some remote location to local environment.
Commit	• Submitting files to the repository (the local one); in other VCS it is often referred to as "checkin"

Terminology



fetch or pull	• Is like "update" or "get latest" in other VCS.
Push	• Used to submit the code to a remote repository
Remote	• "remote" locations of repository, normally on some central server.
SHA	• Every commit or node in the Git tree is identified by a unique SHA key.
Head	• Is a reference to the node to which our working space of the repository currently points.
Branch	• A particular label on a given node.

Workstation Setup

Visit

- git-scm.com/downloads.

Detailed information

- http://git-scm.com/book/en/Getting-Started-Installing-Git

First thing

- git config --global user.name “My Name"
- git config --global user.email myemail@gmail.com

Let’s get started: Create a new Git Repository

Create a new directory

- mkdir mygitrepo
- cd mygitrepo

Initialize repository

- git init

Check status of repository

- git status

Create and commit file

- $ touch hello.txt
- $ echo Hello, world! > hello.txt

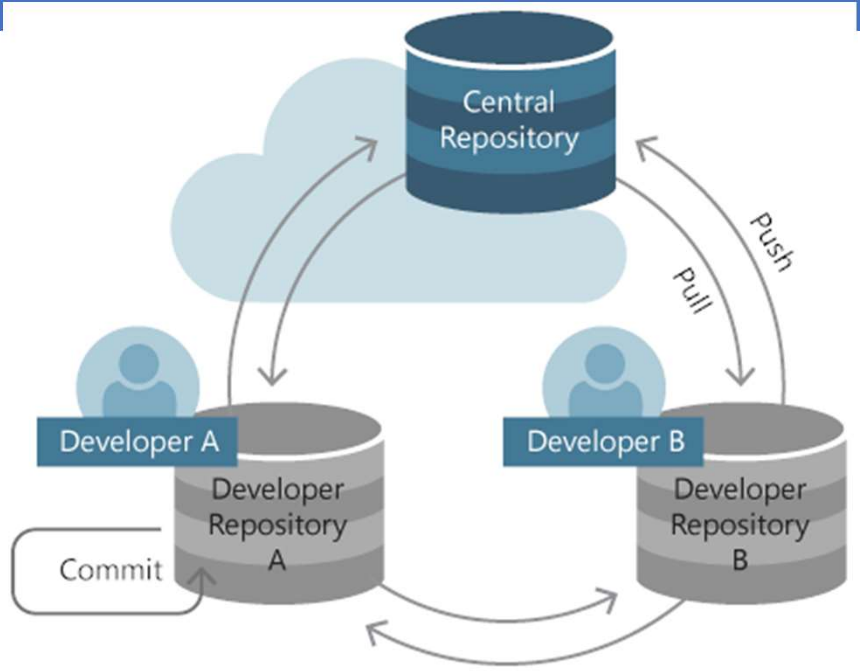
Lets get started: Create a new Git Repository

"register" the file for committing

```
$ git add hello.txt
```

Check status

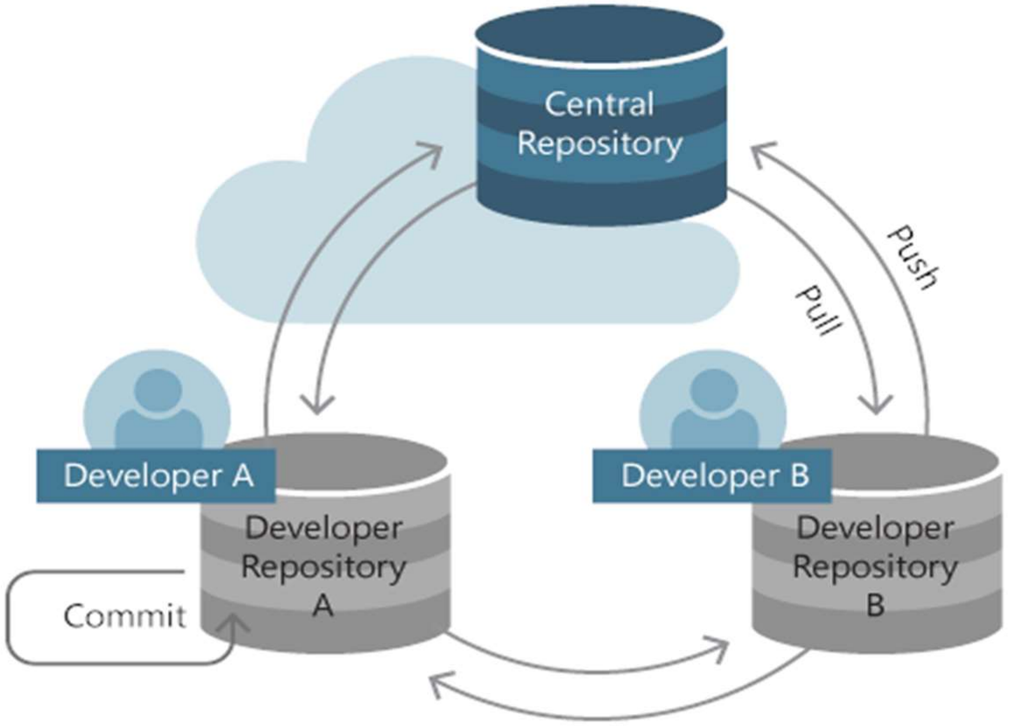
```
$ git status
# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file:   hallo.txt
#
```

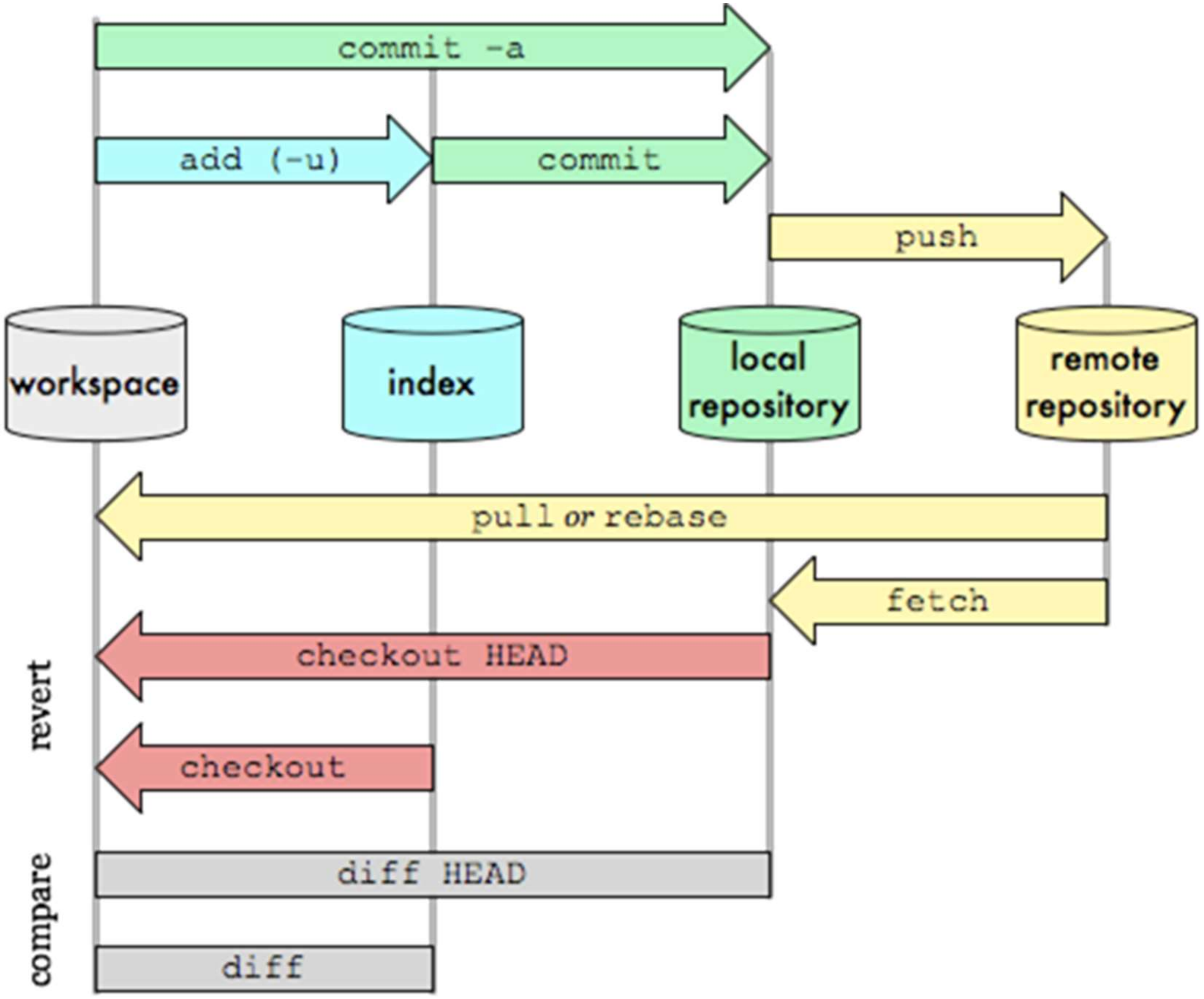


Lets get started: Create a new Git Repository

Commit

- $ git commit -m "Add my first file"
- 1 file changed, 1 insertion(+)
- create mode 100644 hallo.txt




commit -a
add (-u)
commit
push
workspace
index
local repository
remote repository
pull *or* rebase
fetch
revert
checkout HEAD
checkout
compare
diff HEAD
diff

Git commands

Command	Description
git clone url [dir]	Copy a Git repository so we can add to it
git add file	Adds file contents to the staging area
git commit	Records a snapshot of the staging area
git status	View the status of our files in the working directory and staging area
git diff	Shows diff of what is staged and what is modified but unstaged
git help [command]	Get help info about a particular command
git pull	Fetch from a remote repo and try to merge into the current branch
git push	Push our new branches and data to a remote repository

CLONING EXISTING PROJECTS

Syntax

- git clone http://github.com/matthewmccullough/hellogitworld.git

Clone performs several subtasks:

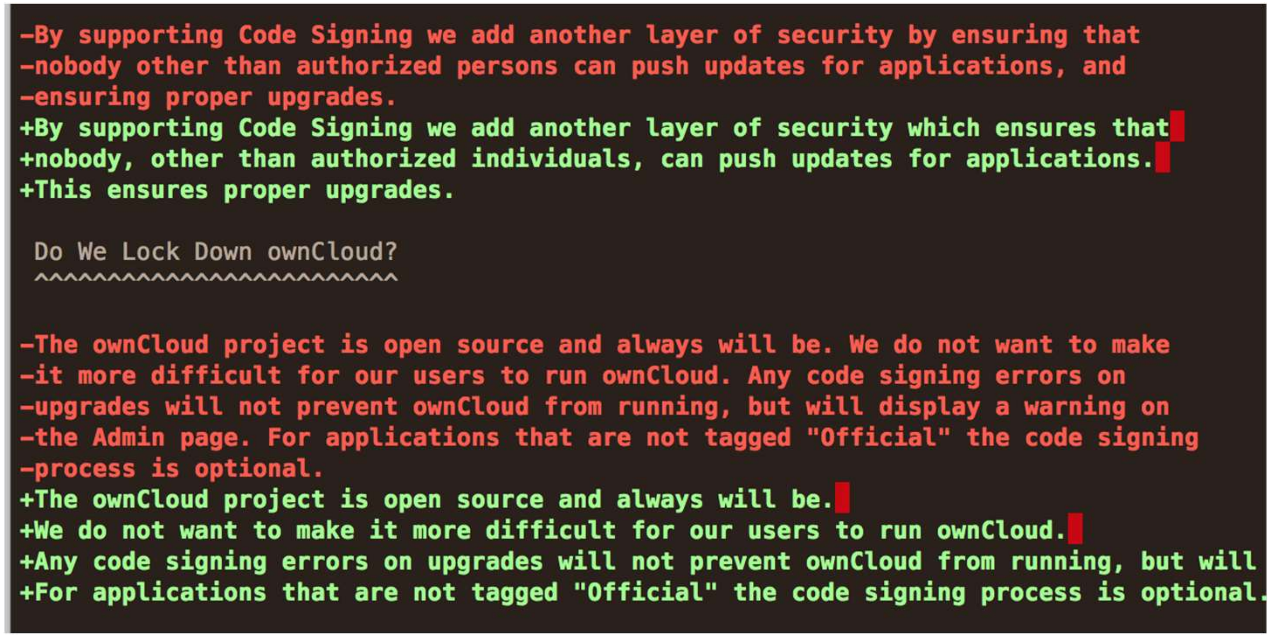
- Sets up a remote named origin that points to the location
  - http://github.com/matthewmccullough/hellogitworld.git
- Asks this location for the contents of its entire repository
- Git copies those objects to the requestor's local disk
- Switches to a branch named master

Ready

- The local copy of this repo is now ready to have edits made, branches created, and commits issued – all while online or offline.

DIFF

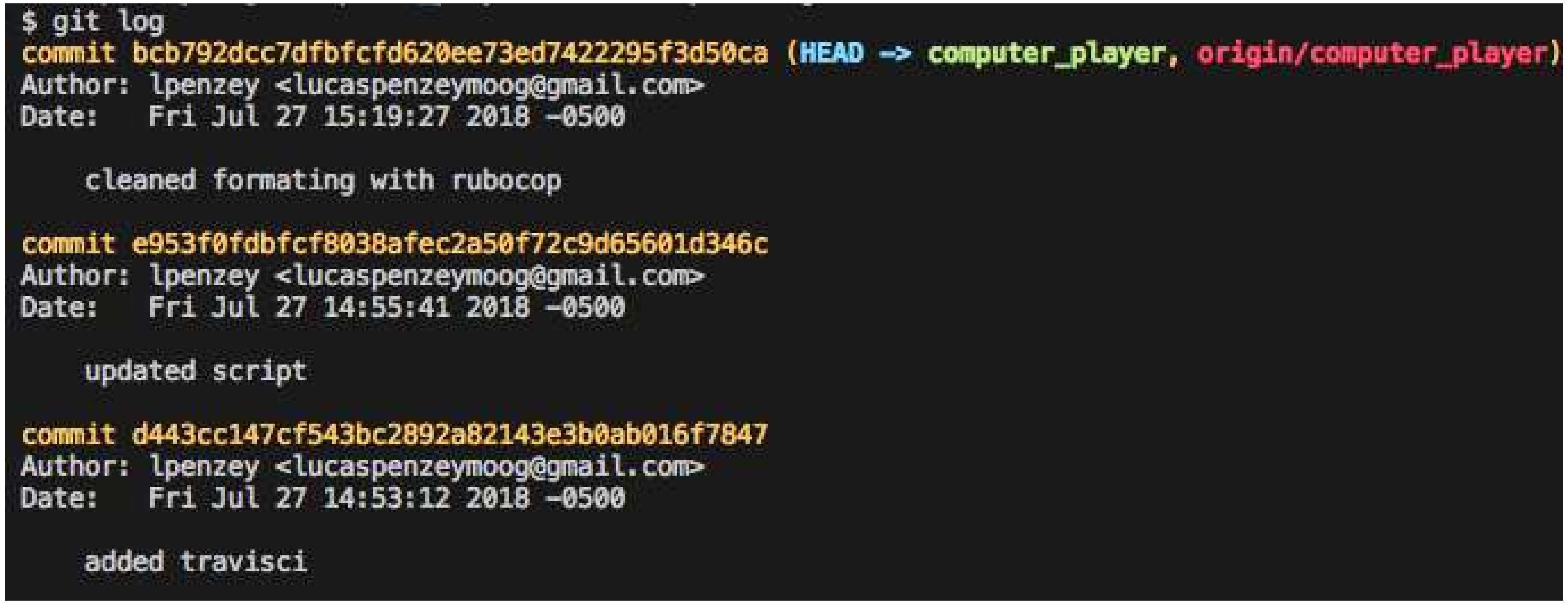
- Difference between edited and committed files
  - git diff



```
-By supporting Code Signing we add another layer of security by ensuring that
-nobody other than authorized persons can push updates for applications, and
-ensuring proper upgrades.
+By supporting Code Signing we add another layer of security which ensures that
+nobody, other than authorized individuals, can push updates for applications.
+This ensures proper upgrades.

 Do We Lock Down ownCloud?
 ^^^^^^^^^^^^^^^^^^^^^^^^^

-The ownCloud project is open source and always will be. We do not want to make
-it more difficult for our users to run ownCloud. Any code signing errors on
-upgrades will not prevent ownCloud from running, but will display a warning on
-the Admin page. For applications that are not tagged "Official" the code signing
-process is optional.
+The ownCloud project is open source and always will be.
+We do not want to make it more difficult for our users to run ownCloud.
+Any code signing errors on upgrades will not prevent ownCloud from running, but will
+For applications that are not tagged "Official" the code signing process is optional.
```

LOG

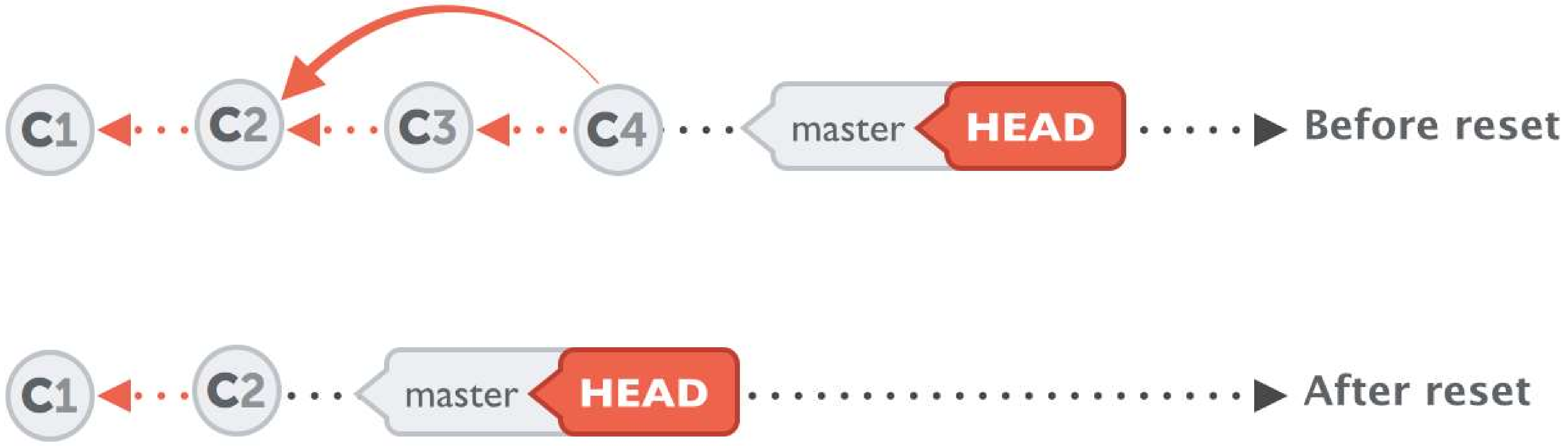
- List of changes
  - git log
  - git log --since=yesterday
  - git log --since=2weeks



```
$ git log
commit bcb792dcc7dfbfcfd620ee73ed7422295f3d50ca (HEAD -> computer_player, origin/computer_player)
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Fri Jul 27 15:19:27 2018 -0500

    cleaned formating with rubocop

commit e953f0fdbfcf8038afec2a50f72c9d65601d346c
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Fri Jul 27 14:55:41 2018 -0500

    updated script

commit d443cc147cf543bc2892a82143e3b0ab016f7847
Author: lpenzey <lucaspenzeymoog@gmail.com>
Date:   Fri Jul 27 14:53:12 2018 -0500

    added travisci
```

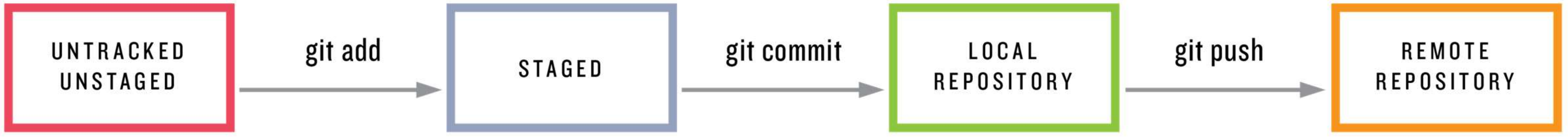
ABORTING

- Abort current uncommitted changes
  - git reset --hard



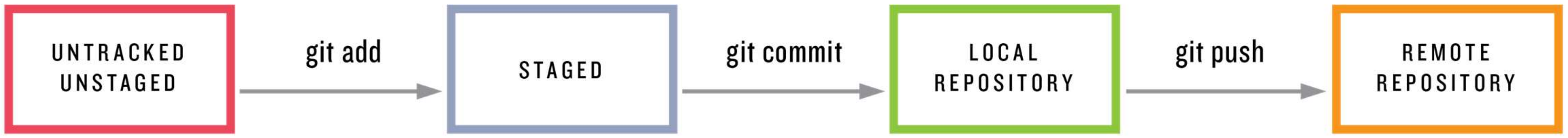
ADDING (STAGING) ✚

- To put files into next commit
  - git add .



COMMITTING

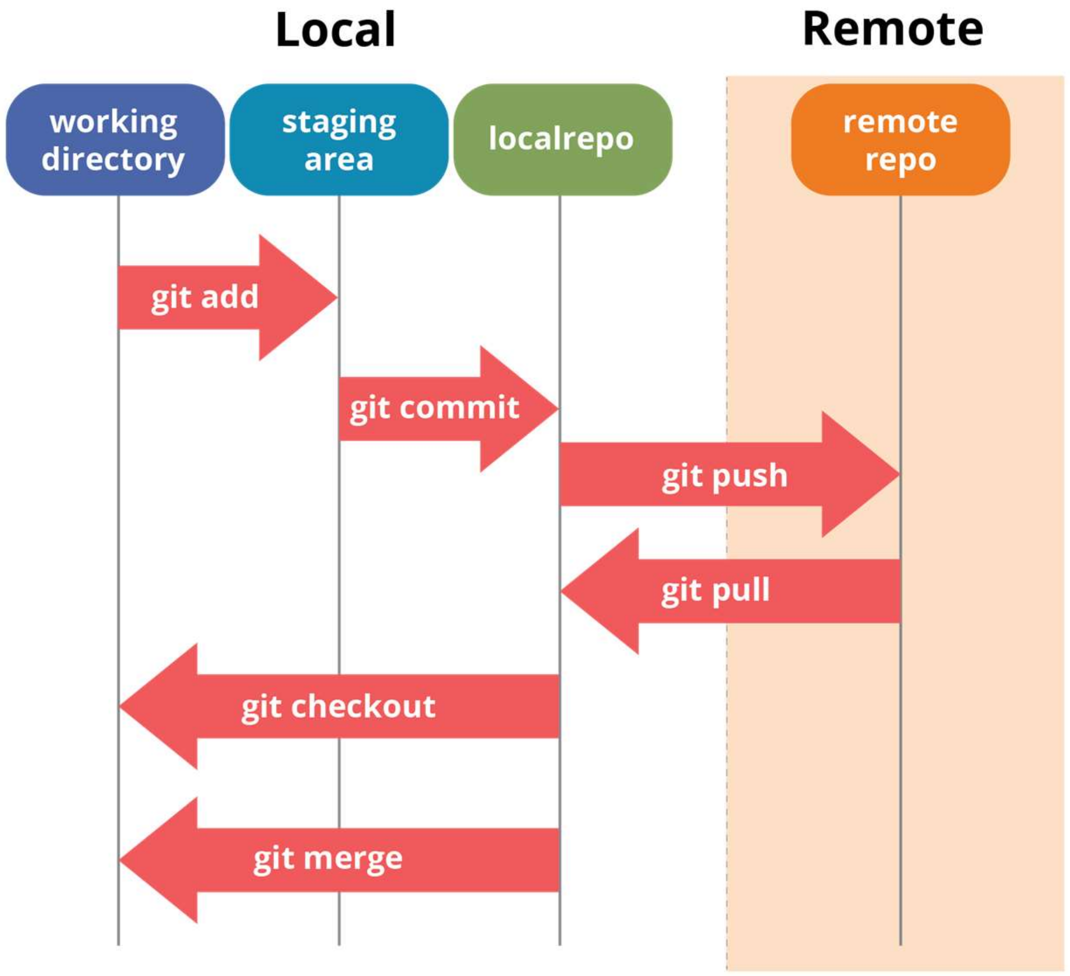
- Save pending additions to local repository
  - git commit –m “<commit message>”
- To view the statistics and about last commit:
  - git show



THE REMOTE WORKFLOW

- Working with remote repositories is one of the primary features of Git

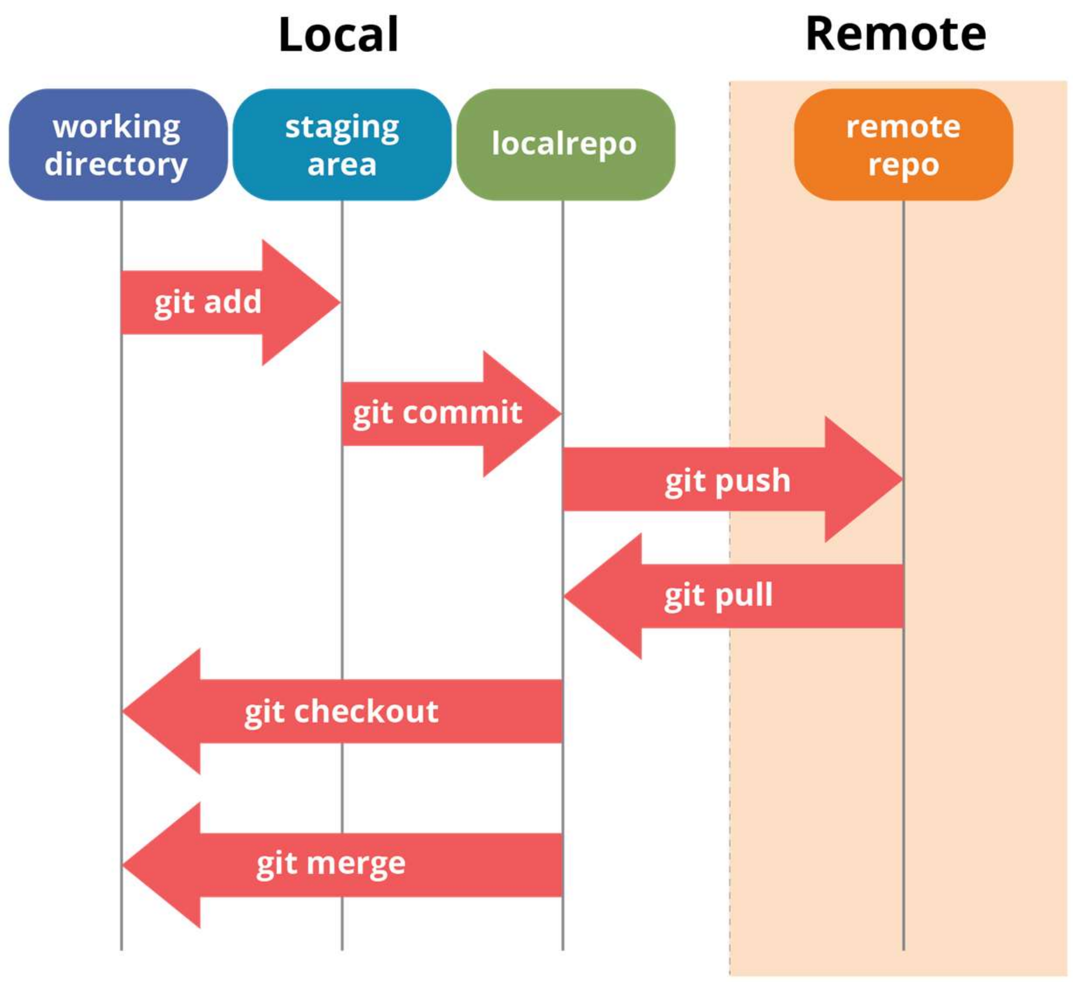
- REMOTES
  - A remote called origin is automatically created if we cloned a remote repository.
  - The full address of that remote can be viewed with:
    - git remote -v
  - To add a new remote name:
    - git remote add <remote name> <remote address>



PUSH / PULL

- To put changes from local repo in the remote repo
  - git push origin master

- From remote repo to get most recent changes
  - git pull <remote name> <branch name>



GitHub.com

- For online storage of Git repositories
  - Can create a remote repo there and push code to it
  - Free space for open source projects
- Its not mandatory to use Github to use Git.
  - We can use Git locally for our own purposes.
  - We can also set up a git server locally

Thanks