

HUAWEI DEVELOPERS

GIT Basics



Table of Contents

- VCS (Version Control System)
- Need of VCS
- What is GIT?
- GIT Branching
- GIT most common commands
- An example scenario

Version Control System (VCS)

Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done to the code.



Need of VCS



Easy Modification
of the codebase



Reverting errors



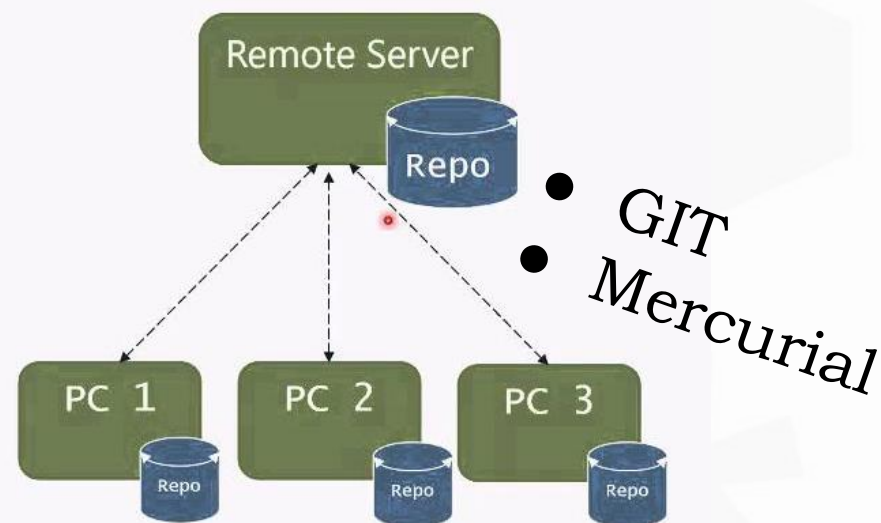
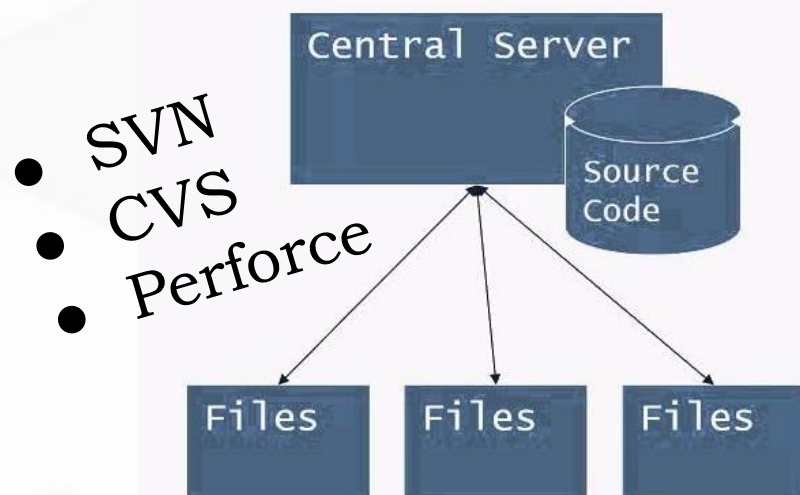
Work as a team



Backup

Types of VCS

Centralized vs. Distributed



GIT – A distributed VCS

- ❖ GIT is free and open source distributed system with the emphasis on speed and data integrity.
- ❖ No centralized connectivity is needed.

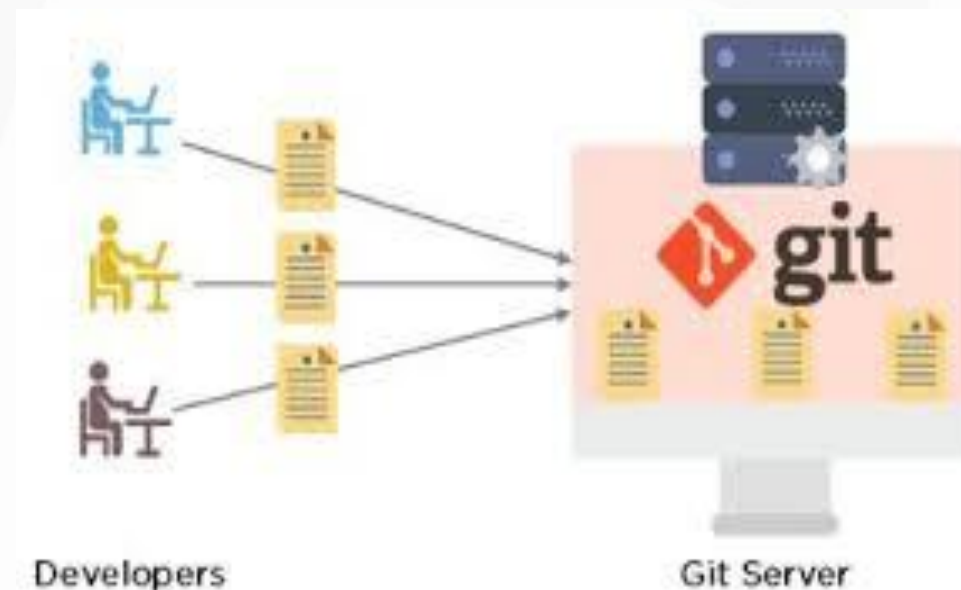
GIT - the stupid content tracker

"git" can mean anything, depending on your mood.

- random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
- stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
- "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
- "goddamn idiotic truckload of sh*t": when it breaks

This is a stupid (but extremely fast) directory content manager. It doesn't do a whole lot, but what it does do is track directory contents efficiently.

- ❖ Powerful and cheap branching with easy to merge.
- ❖ Losing work in your project is very very hard.



Directory structure

- .git
- HEAD/ (A pointer to your current branch)
- config/ (contains all configuration preferences)
- description/(description of your project)
- Index/ (is used as staging area between working directory and repo)
- logs/ (keeps records to changes that are made in ref) objects/ (all data are stored here: commits, trees and tags)
- hooks/ (shell scripts that are invoked after executing a command)
- refs/ (holds your local branch remote branch and tags)

GIT

usage: git [OPTIONS] COMMAND [ARGS]

The most commonly used git commands are:

- add: Add file contents to the index
- commit: Record changes to the repository
- diff: Show changes between commits, commit

git help <command>

<https://git-scm.com/downloads>

Configure GIT

git config --global user.name "Abhishek Parihar"

git config --global user.email "abhishek.Parihar@huwaei.com"

Git can be used locally

- First step towards a shared project.
- Backup.
- To keep the memory of your work.



Steps

To add GIT to your
project



"Git init"

- Creates an empty git repository.
- Creates the git directory: .git/

**working
directory**

**staging
area**

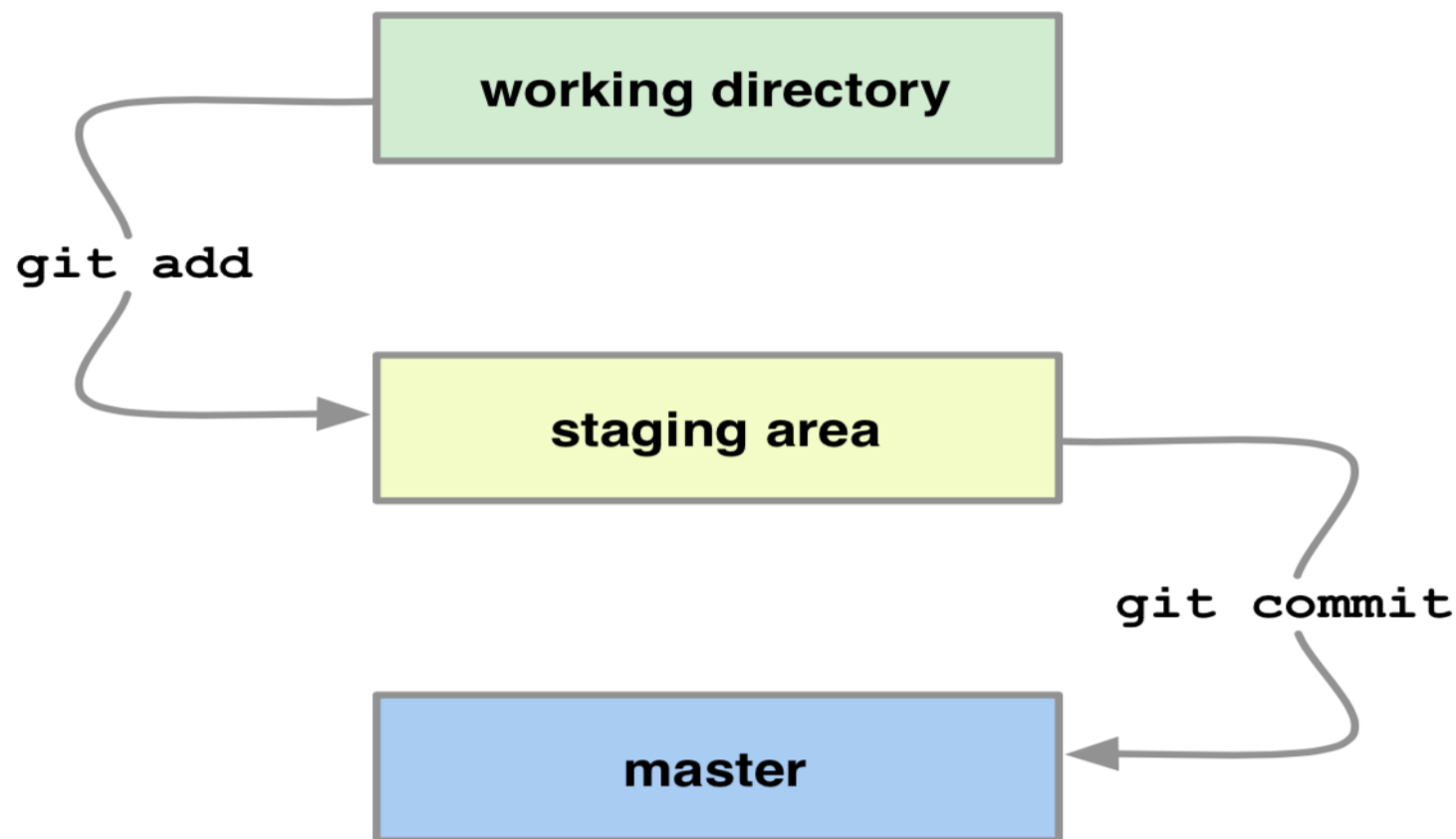
master

****Local commands. Only for single user**

Note: it is **safe. It does not change your pre-existing files.**

“git add” and “git commit”

git add <filename>



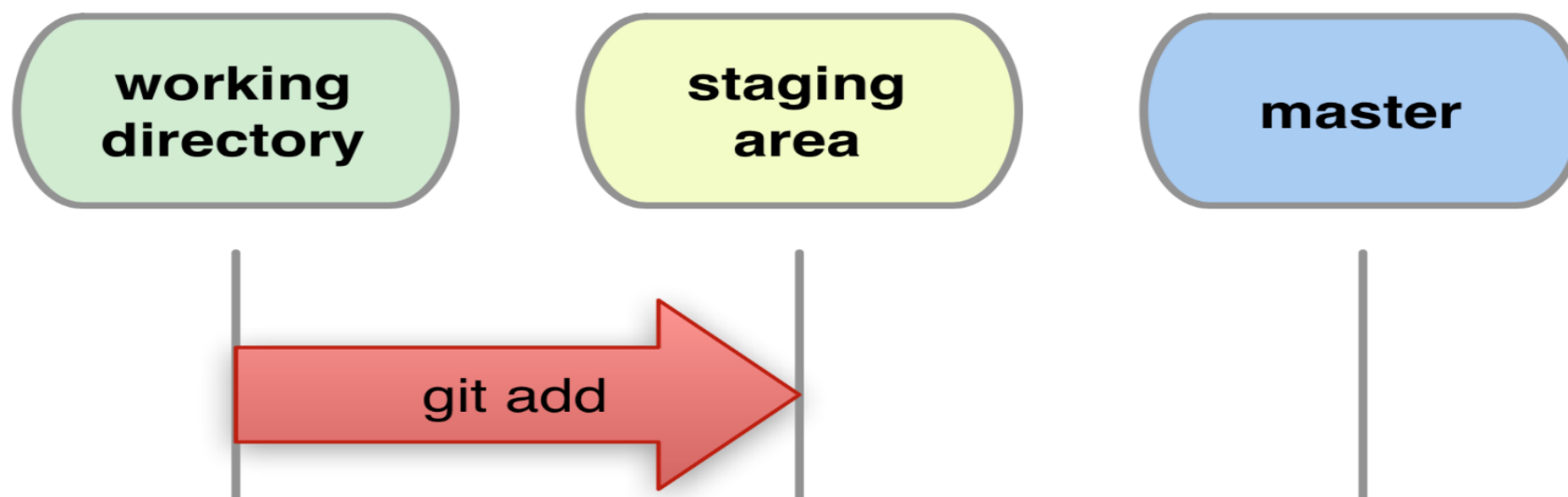
“A staging area is a location where organisms, people, vehicles, equipment or material are assembled before use”.

git commit -m "Let us begin."

“git add”

git add file1 [file2 ...]

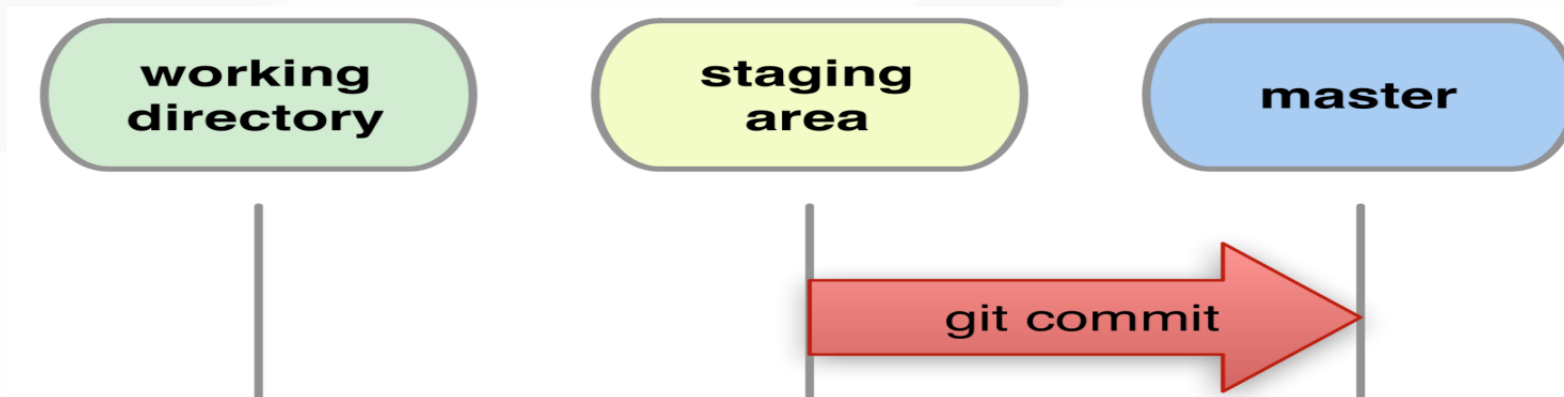
- Adds new files for next commit.
- Adds content from working dir to the staging area (index) for next commit.



“git commit”

`git commit -m [“Commit message.”]`

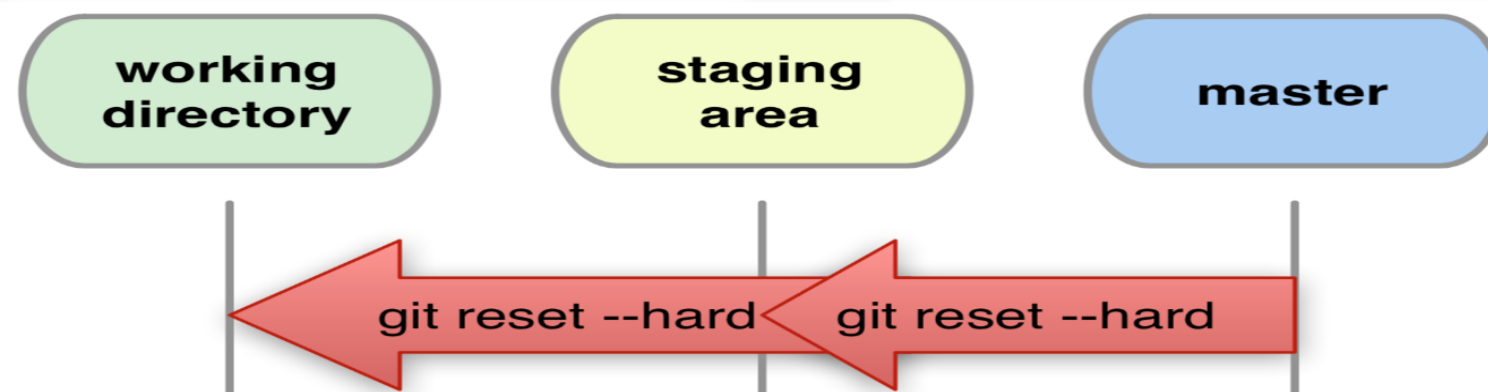
- Records changes from the staging area to master.



- Every commit is a git-object.
- The history of a project is a graph of objects referenced by a 40-digit git-name: SHA1(object).
- Ex : d921970aadf03b3cf0e71becdaab3147ba71cdef

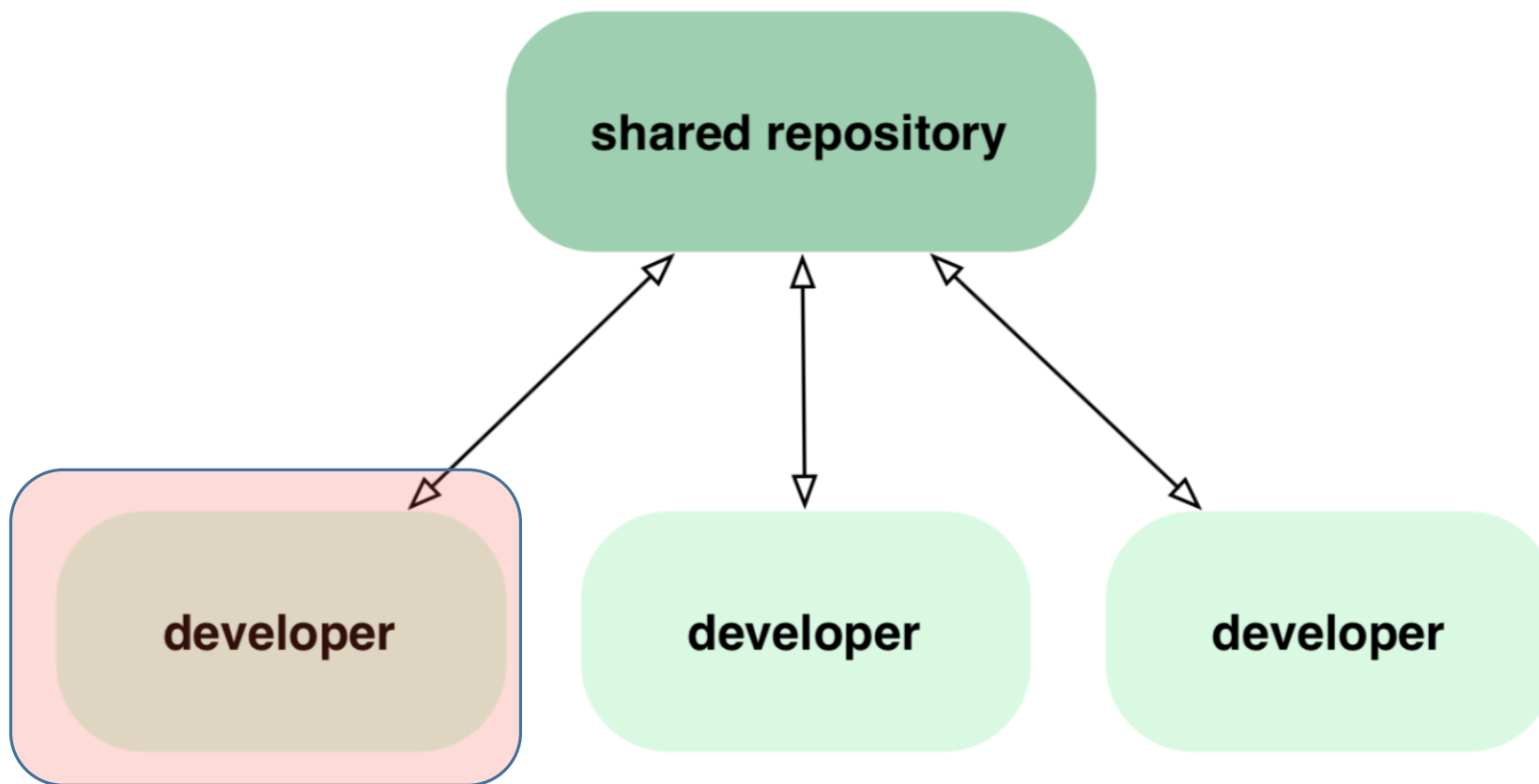
Few other common GIT commands

- **git log** → Shows details of the commits.
- **git status** → displays the state of the working directory and the staging area
- **git reset --hard HEAD** → Restore all files as in the last commit.
 - If a lot of mess and need to clean, then this is THE COMMAND
 - Will make working directory same as the last commit

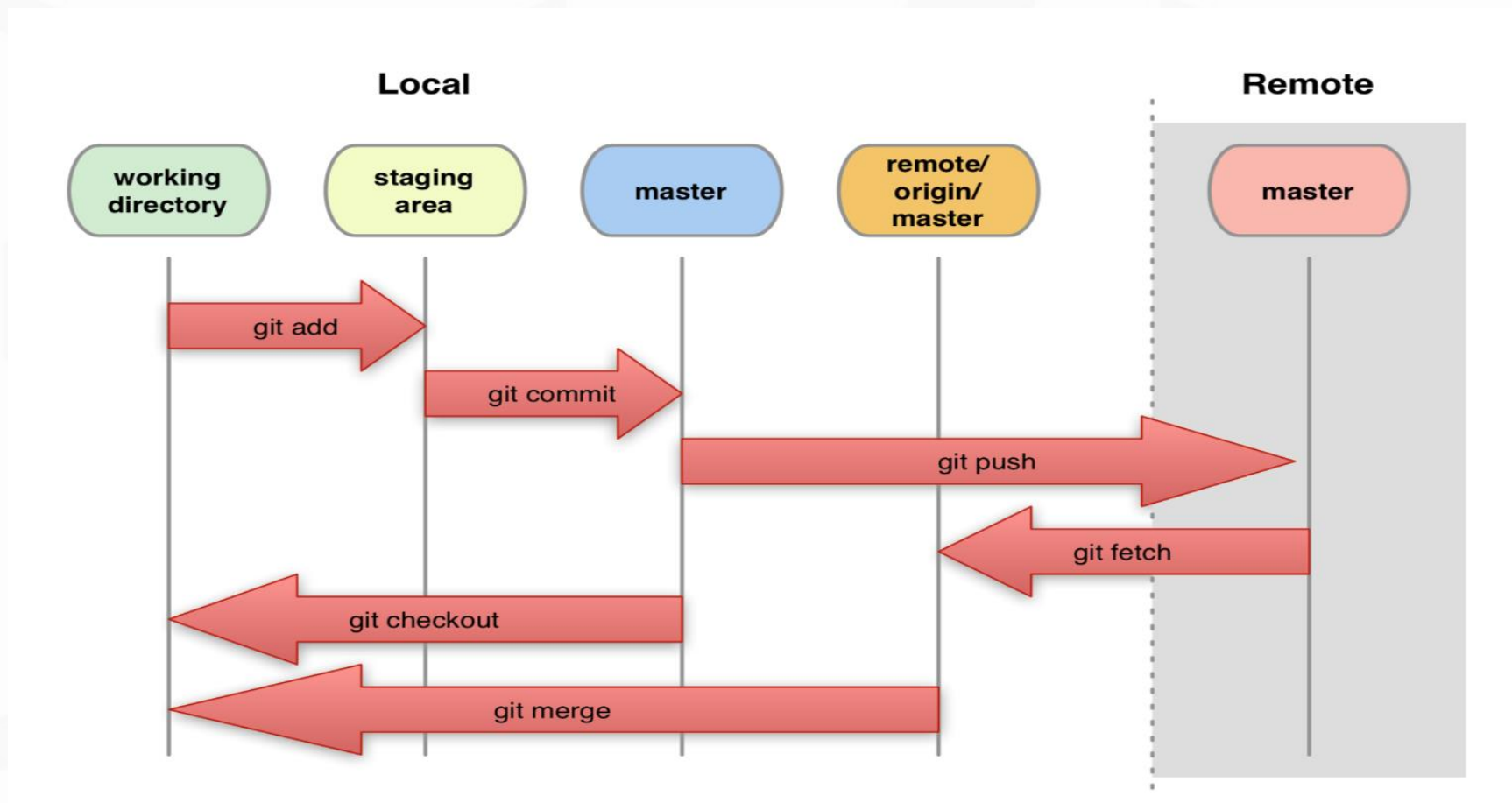


****Warning: reset can destroy history!**

GIT remote repository



From Local to reote



GIT clone

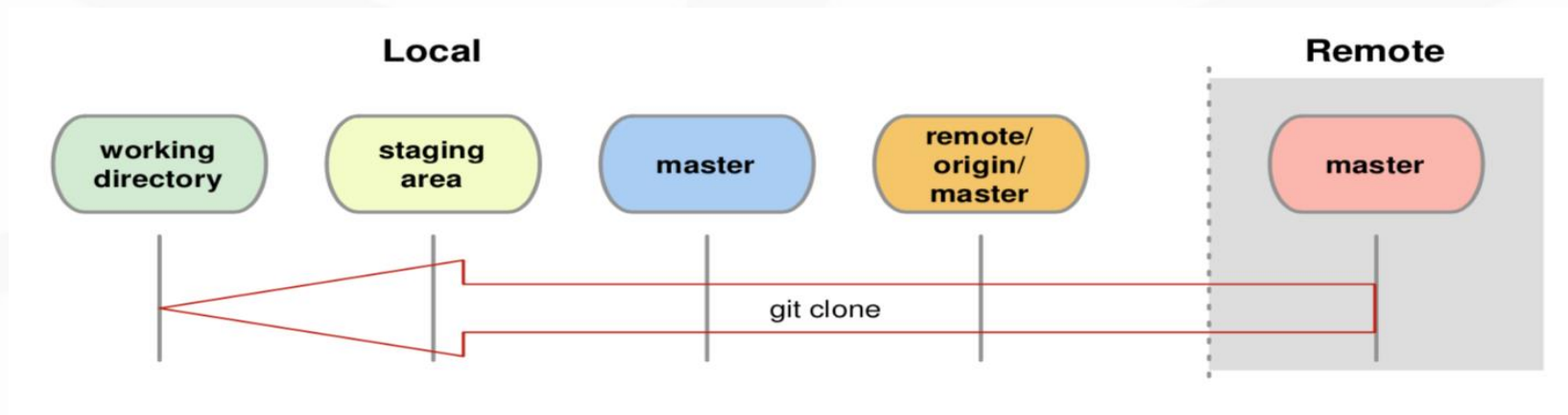
`git clone <url>`

Creates two local copies of the whole remote repository.

Ex.: `git clone https://github.com/abhi/demo.git`

`git remote -v`

shows name and URL of the remote repository.



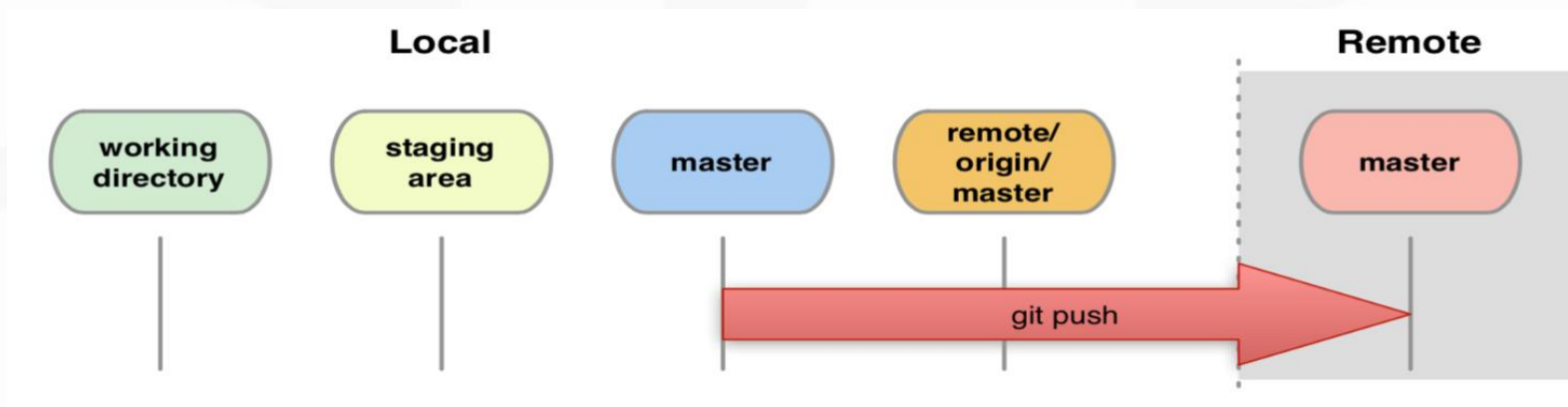
GIT

- git pull
 - Fetch changes from a remote repository into the current branch.

git pull origin <branch name>

- git push
 - Updates remote masters (both Local and Remote).
 - Requires fetch+merge first (i.e pull).

git push origin <branch name>

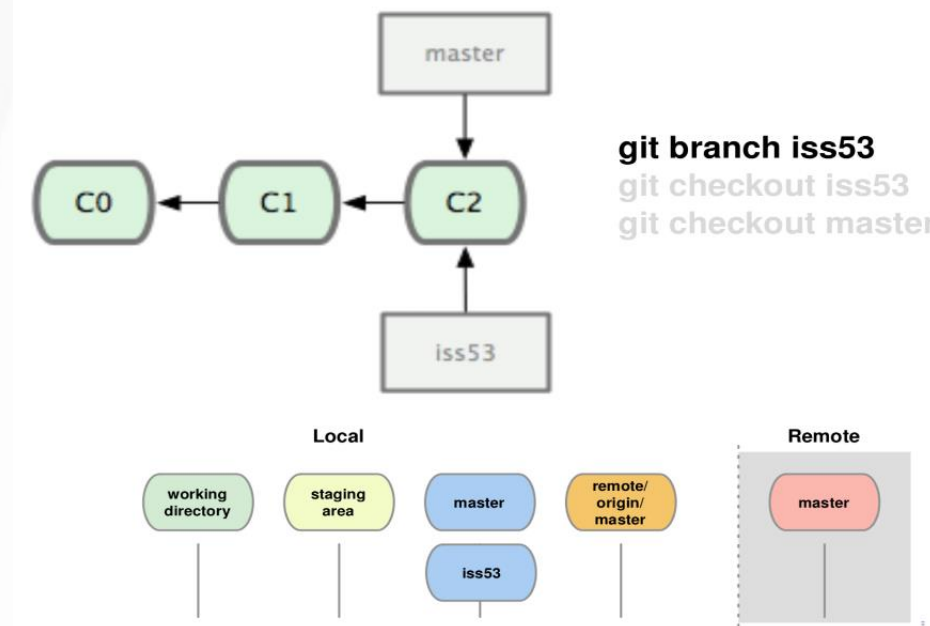
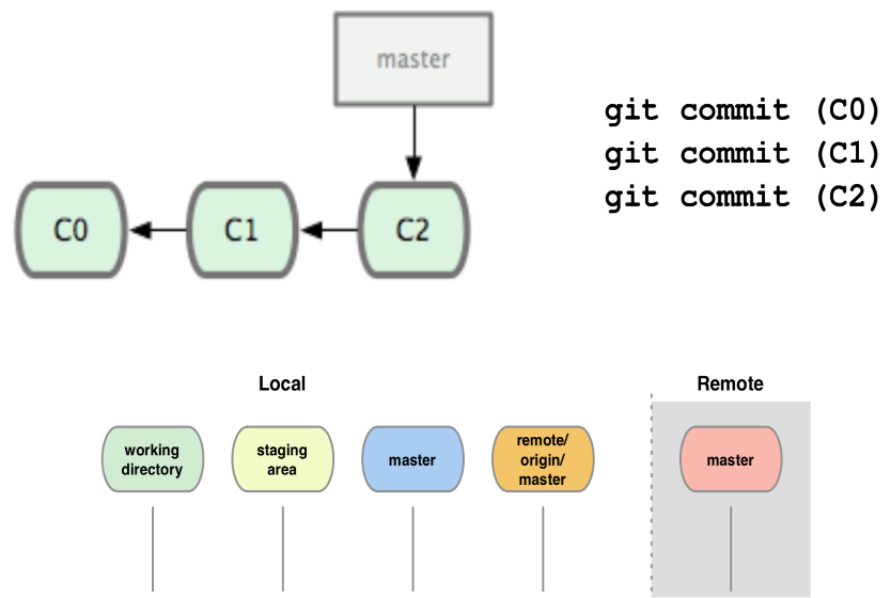


Conflicts

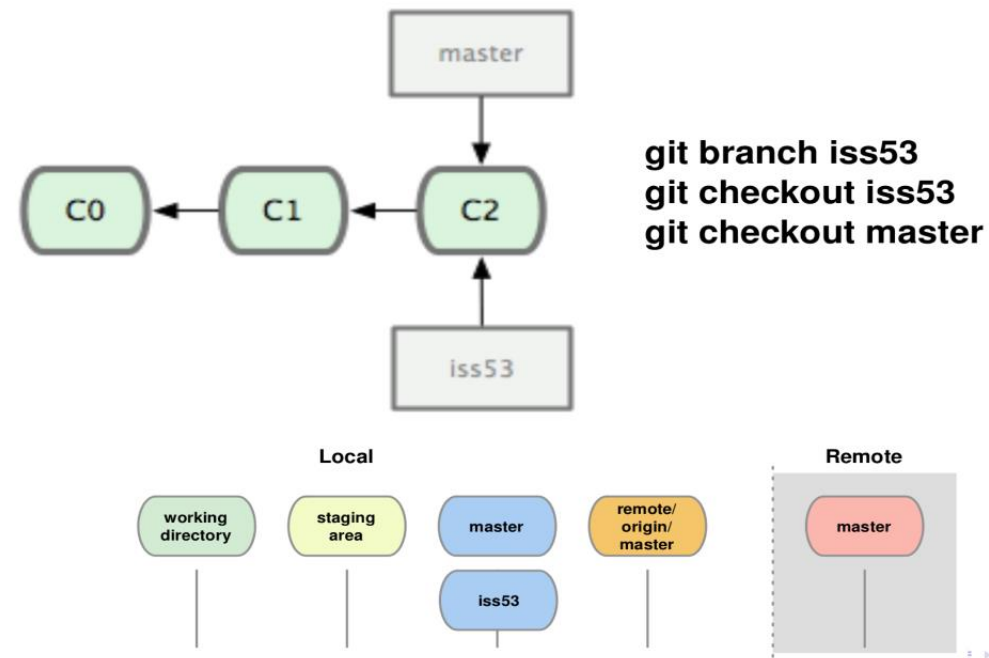
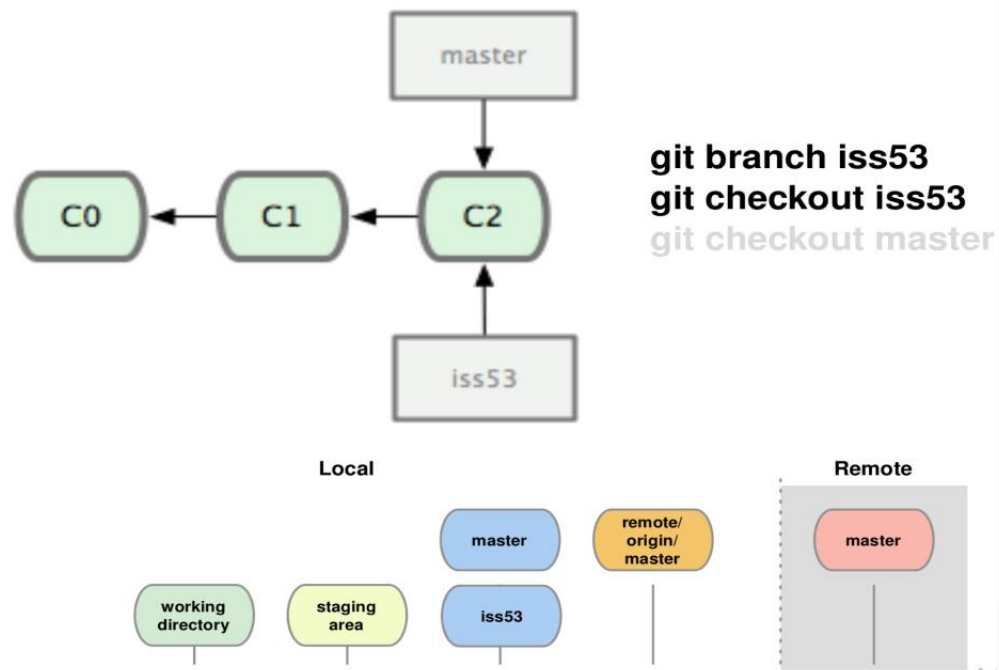
- Conflicts occurs when same file changed by multiple users
- Or same file changed in a contradictory ways.
- It's a rare scenario when GIT cannot figure out which changes needs to be kept.

```
...
<<<<<< yours:sample.txt
Conflict resolution is hard;
let's go shopping.
=====
Git makes conflict resolution easy.
>>>>>> theirs:sample.txt
...
```

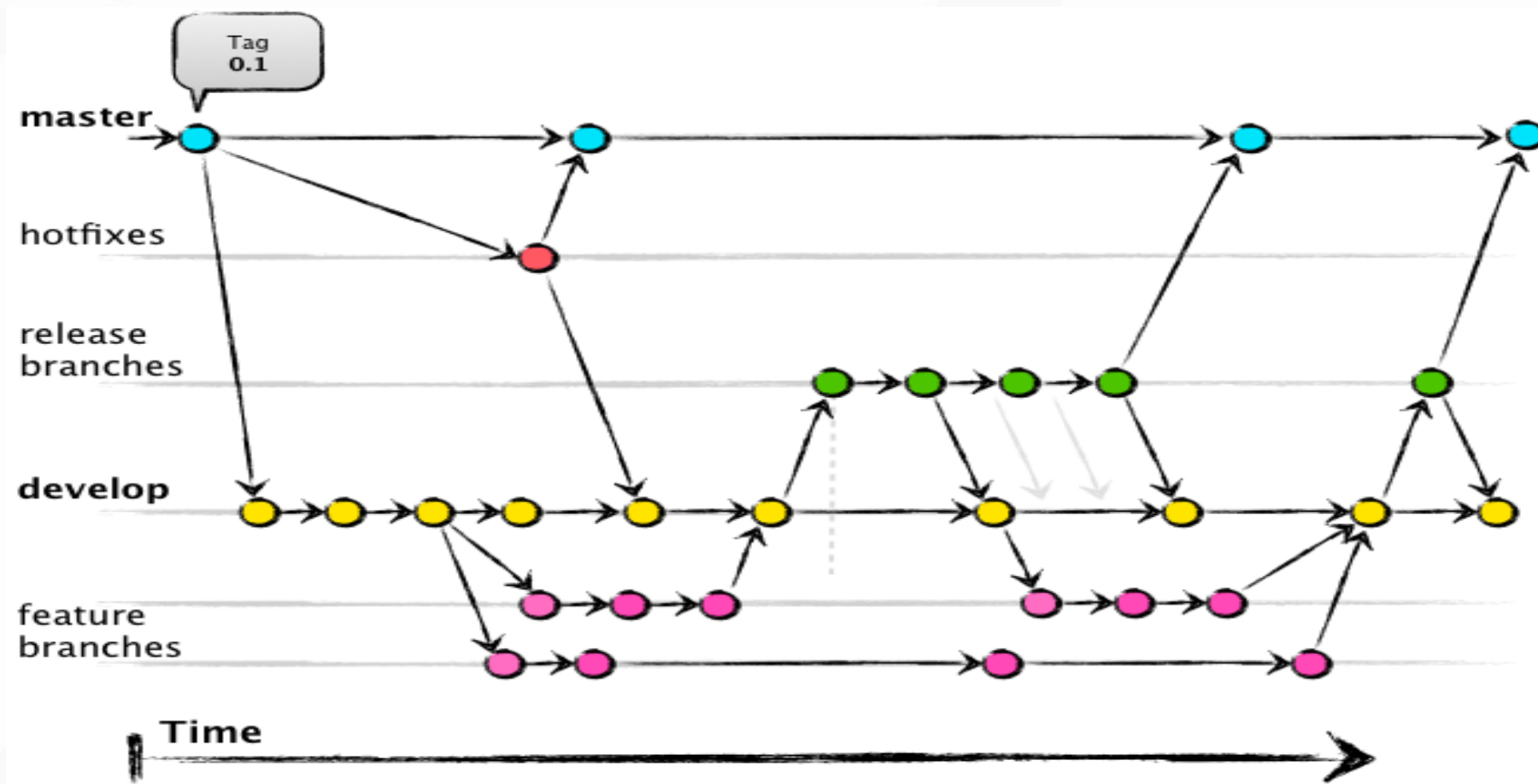
Branching basics



Branching basics



Branching Strategy



.gitignore

- A gitignore file specifies intentionally untracked files that Git should ignore.
- files and directories both can be added.
- Files already tracked by Git are not affected.
- It reads file in the same directory as the path
- A project normally includes such .gitignore files in its repository, containing patterns for files generated as part of the project build.

<https://github.com/github/gitignore>

Popular GIT servers



AWS CodeCommit



Bitbucket

Bitbucket



GitLab



GitHub

Assignment

- ☐ Install git on your machines
- ☐ *Create repo in one of the folder*
- ☐ *Add at least 3 files and then commit separately*