How to be a 1337 h4ck3r: Git + Linux

An introduction to Git, version control, and Linux

by Samsara Counts, Pat Cody, and Joseph Schiarizzi

Slides adapted from Neel Shah and Phil Lopreiato

DISCLAIMER: GW ACM does not promote illegal and/or unethical activity (hacking or otherwise), and that's not the focus of this workshop!

Most importantly, computer scientists come in ALL forms and that's why our field is so great!

To quote Melinda Gates:

"Not every good idea comes wrapped in a hoodie."

"It's time the world starts recognizing that the next Bill Gates might not look anything like the last one."

1337 h4ck3rz??

- Why 1337 h4ck1ng?

...well, you've seen the *hacker* stereotype, that's not any sort of true. (smh)

The cool thing about hackers is that **they're powerful af**. And you know what?

So are you, when you learn how to use Git and the command line!

1337 h4ck3rz??, cont.

git

- Why Git?
- Why Linux?

git and linux are the essential tools of good programming: streamlining your workflow and keep track of different versions of your code

Linux

Goals of this workshop:

1. you know the basic tools to be a good CS and software developer
2. You have a jumping off point to **build up a digital portfolio** and dive into software projects

This workshop

PART I: Linux and UNIX-like environments

A. Linux background, the command line, + commands
B. Command line hacking: make a directory!
C. More background!
D. More Command line hacking: edit a file in vim!

PART II: Git and version control

A. git background
B. get git; try out github
C. github background
D. Clone our git repo + add a file; make your first pull request

Your Workshop TODO:

1. Put away your laptop (until we say so) and turn off your phone

2. Follow along!

3. Ask lots of questions

4. When it's time to try things out, READ OUR DOCS.

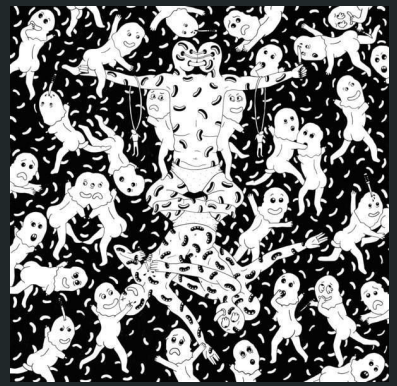
Resources + Documents for this workshop

1. Unix/Git Workflow Cheatsheet
2. Git Cheatsheet
3. Workshop Repository + README
4. Presentation Slides

PART I: Linux and UNIX-like environments

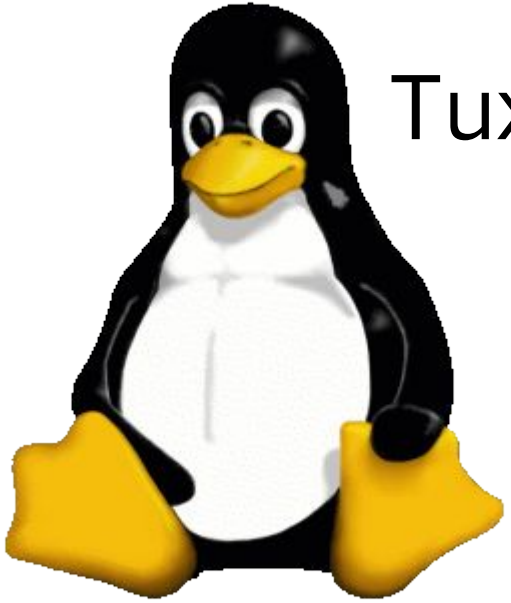
*“In thin socks, I'm butthurt...**crashing Linux server**”*

- Busdriver, “Worlds To Run (ft. Anderson.Paak & milo)”



Wat is a Linux.

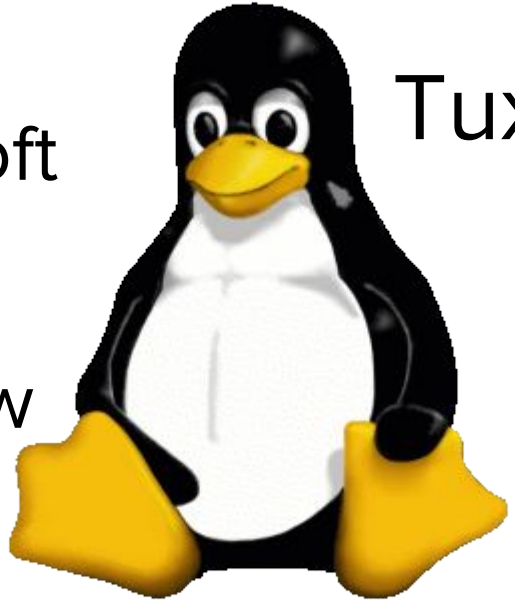
- An Open Source Operating System modelled on UNIX.
- Always FREE and community built & maintained.
- All supercomputers use linux and most servers.



Tux

History of Linux.

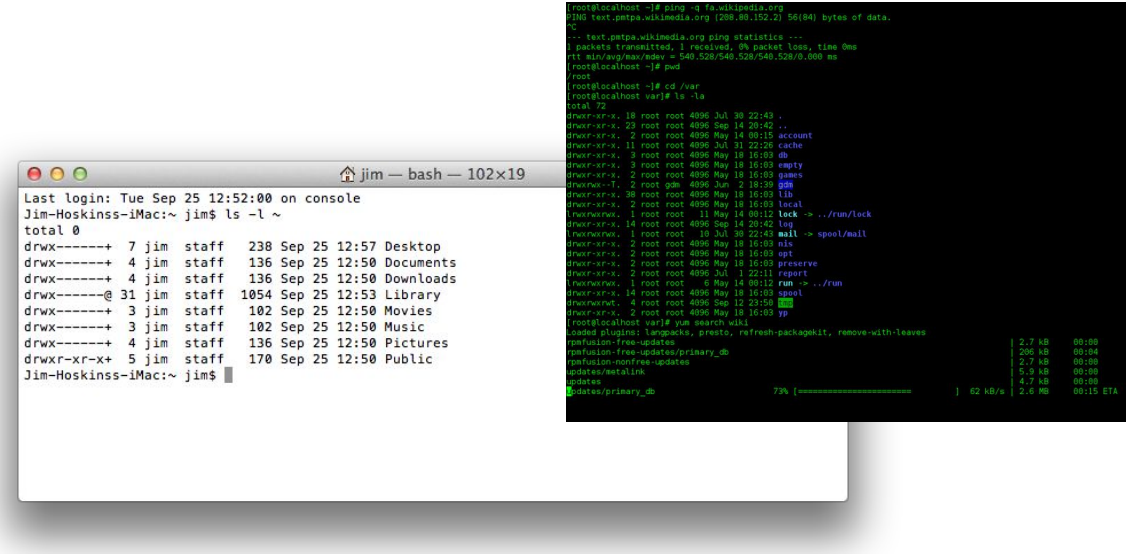
- Released on September 17th, 1991 by Linus Torvald
- Took off fast to escape Microsoft monopoly
- Many different distributions now available



Tux

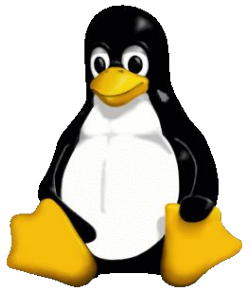
Wat is a command line interface

- a text-based application for viewing, handling, and manipulating files directly on your computer
- What 1337 h4ck3rz/ Comp Scis use



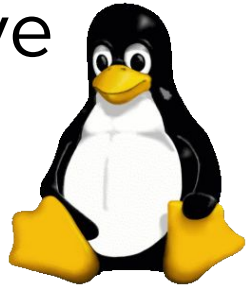
Basic Linux commands

ls	**-**	**list files**
cd	**-**	**change directory**
mv	**-**	**move a file**
mkdir	**-**	**make directory**
rm	**-**	**delete a file**



basic Linux names/commands, cont.

> [**filename**]	-	make a new file called **filename**
~	-	reference to the home directory
.	-	reference to this (current) directory
..	-	reference to directory above current directory



It’s hacking time! (part I)

Do now (on your laptop):

1. open https://github.com/gw-acm/git-linux-2017 for UNIX cheat sheet, basic instructions, + links
2. https://acm.seas.gwu.edu/ws/git-linux-17/index.html
3. **Then, - Mac/Unix: open** Terminal (command line)
   - **Windows:** download **git bash** and **open it**

Do now, cont: (on the command line):

1. list visible directories from current position
2. Make a directory
3. Navigate into that directory

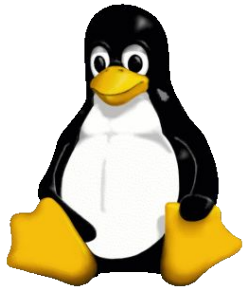
---

4. make a file called ‘new.txt’
5. Copy a file from another location and put it in this folder (.)
6. Remove that file

DEMO I!

More complex Linux commands

pwd	-	print working directory
head [file]	-	show first 10 lines of file
tail [file]	-	show last 10 lines of file
clear	-	clear screen



Intro to vim

- **vim** is a multipurpose text editor for the command line
- The name vim comes from “vi improved,” where vi is an older, less cool command line text editor

Vim commands

vim [filename]	-	open [filename] with vim
esc	-	go back to normal mode
i	-	go to insert
:wq	-	write (save) and quit
:w	-	write
:q!	-	force quit (doesn’t save)

It’s hacking time! (part II)

Do now (on the command line):

1. enter ‘cd’
2. Navigate back to directory
3. enter **vim** and edit ‘new.txt’
4. In vim, edit that file
5. Write and exit vim
6. Look at the first lines of that file
7. Remove that file

DEMO II!

PART II: GIT and version control

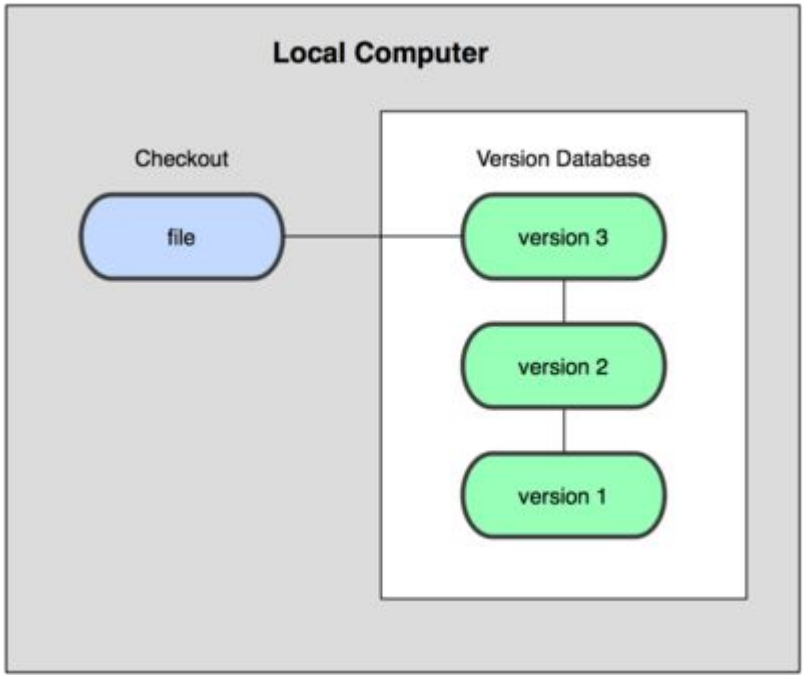
Git your life together

Slides adapted from Neel Shah and Phil Lopreiato

What is git?

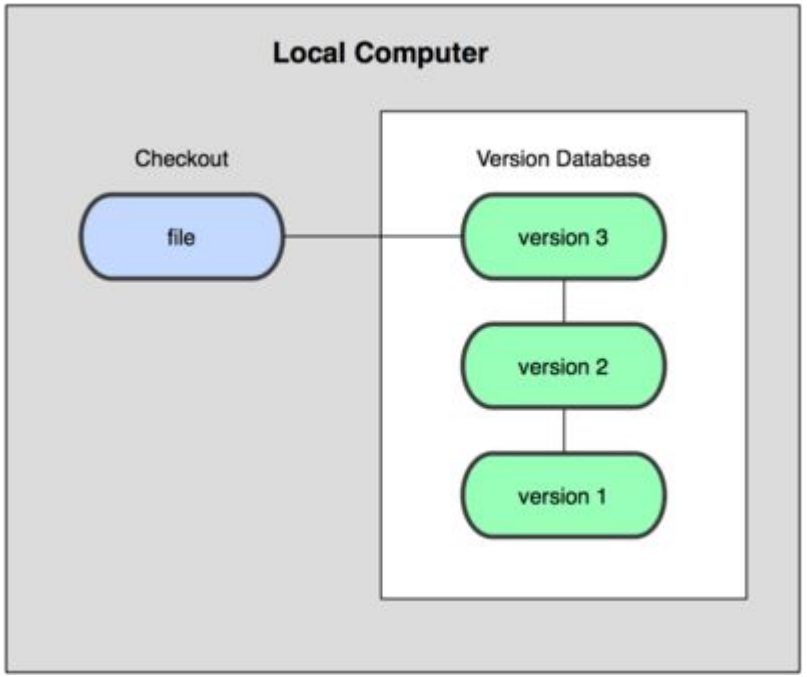
- Git is a type of *version control software*
  - It's a way to track changes to files
- How would you do that?
  - Many zip files
  - Timestamped directories
  - A specially ordered collection of stones
  - Magic?

- Somebody smart decided to synchronize their files with a local database - this is **local version control**



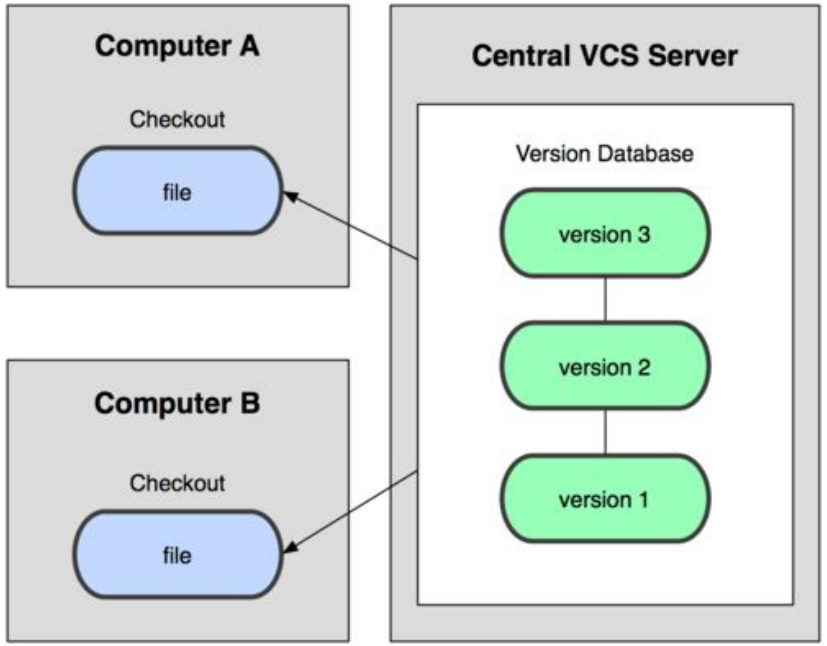
How is git structured?

- You keep projects in **repositories** (repos) that are backed up on a server
- you work in local copies of repos, then push changes to server



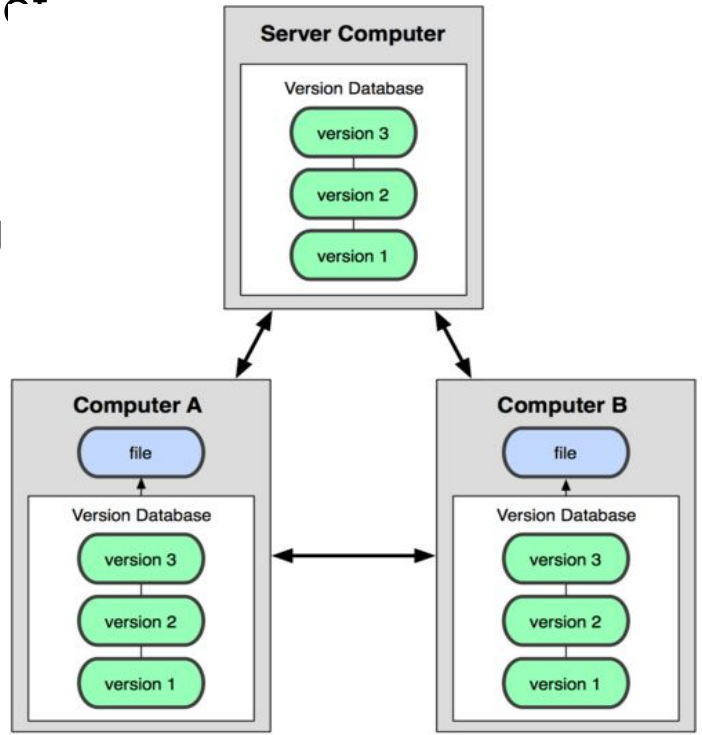
What Is It?

- But what if you want to share your changes with someone else?
  - Easy - just send the database to a server!
- This is called **centralized version control**
- But what happens if the central server goes down?



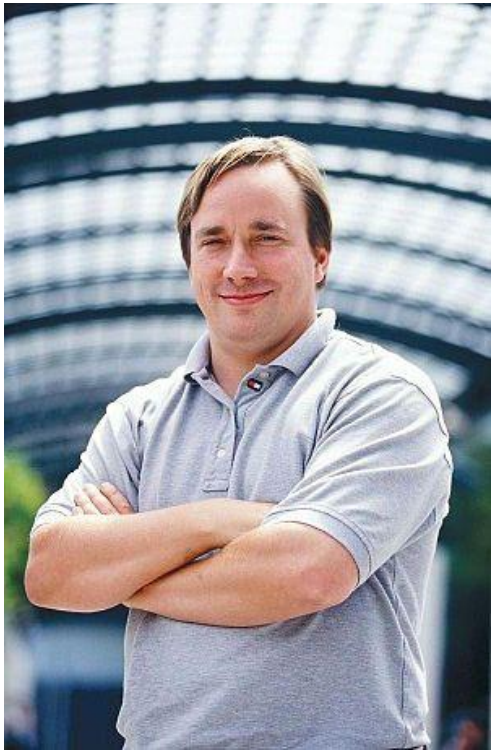
What Is It?

- Just make clients check out the full contents of the repository then!
  - Now, everybody who works on a project full copy of the history in case something happens
- This is called **distributed version control**
  - Many common *VCS* systems use this (git or Mercurial, for example)
- **GitHub** (github.com) is a popular web-based Git repository hosting service



Git Ancient History

- In 2005, the Linux Kernel project needed a new source control system
- Linus Torvalds set out to write his own
  - Popular version control software at the time was not “good enough” for him
  - Needed to be distributed and protect against corruption
- Development began April 3, the project was announced April 6, became self-hosting on April 7, and used in the kernel by June



How does it work?

- Git is a **Directed Acyclic Graph** of repository “snapshots”
- Every change is initially done locally
- Every change has verified integrity
  - The repository is “checksummed” after every change
  - That checksum is used to refer to each **commit**

“In many ways you can just see Git as a filesystem — it is content-addressable, and it has a notion of versioning, but I really really designed it coming at the problem from the viewpoint of a filesystem person (hey, kernels is what I do), and I actually have absolutely zero interest in creating a traditional SCM system.” - Linus

It’s (git) hacking time! (part III)

How can I “Git” Git?

- http://git-scm.com/downloads
- You can now use git in a terminal or “Git Bash” on Windows

Configuration...

```
$ git config --global user.name "YOUR NAME"
```

```
$ git config --global user.email "YOUR EMAIL ADDRESS"
```

Do now: make an account!

- Go to https://github.com/join

and create an account

(or log in)

- Get familiar with git!
  - https://try.github.io/

(Demo unnecessary)

What's this GitHub?

- GitHub makes coding "social" by providing a Git repository hosting service that maintains all of the "distributed" features of Git and adds a social aspect
  - Issue Trackers
  - Code Releases
  - Project Websites
  - Anyone can contribute!

GitHub

- Example GitHub profiles:
  - https://github.com/aaroncoplan
  - https://github.com/samsaranc
- Example Repositories:
  - Linux
  - The Blue Alliance
- Example Organization:
  - GWCloudLab
- Explore other repos and projects:
  - https://github.com/explore

basic git workflow commands

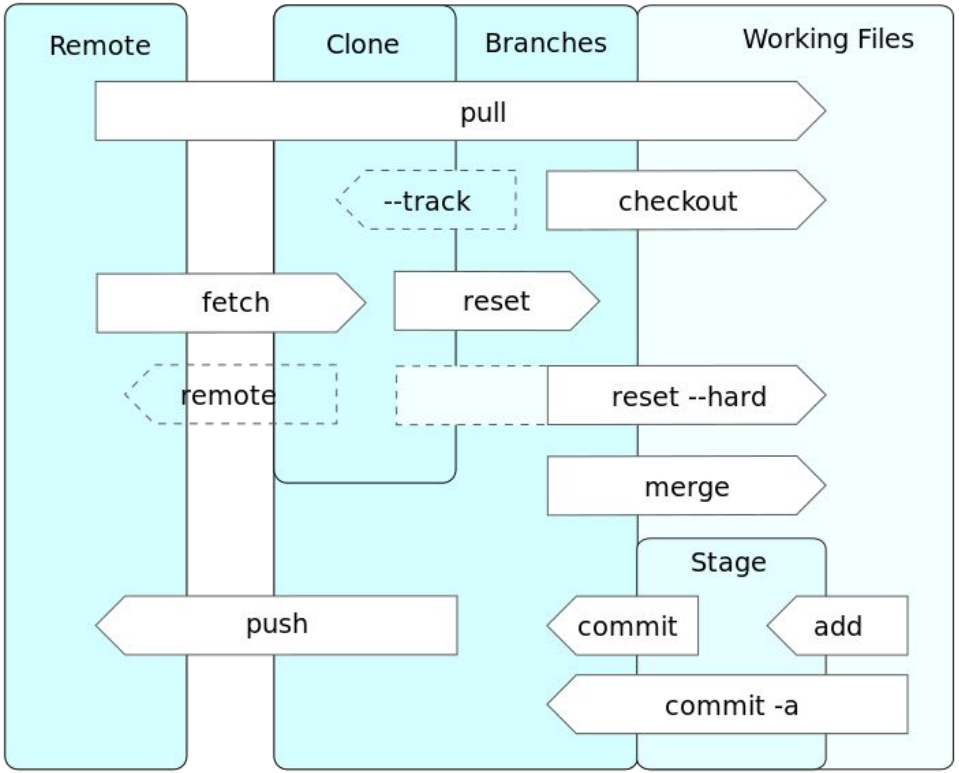
git clone “repo url” - create local copy of repo on your computer

git add * - add all files to commit

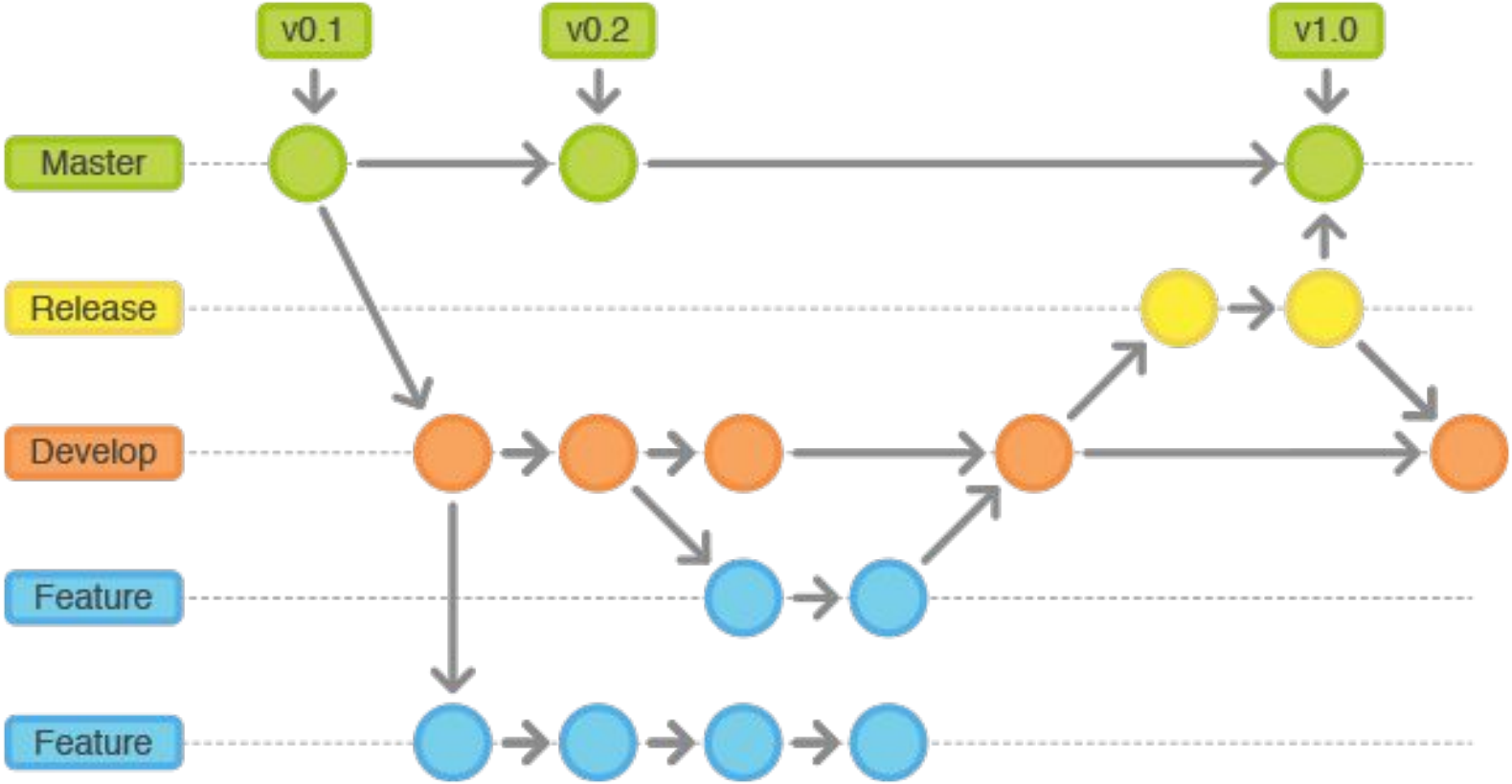
git commit -m “init” - save all your edits with message “init”

git push - push your changes to origin branch of repo you’re working on

Distributed versions, visualized



Cloning, forking, and branches



It’s (github) hacking time! (part IV)

Do now (on github + command line):

1. Full instructions in the README: https://github.com/gw-acm/git-linux-2017
2. fork the above repository on your account
3. On the command line, navigate into the directory you made
4. Clone your fork of the repository on your computer
5. Navigate into that repository
6. Use vim to make a .txt file with your name on it (ex: samsara.txt)

Do now (on github + command line):

7. Tell git you made changes with add
8. Commit those changes with a nice message
9. Push them to your forked repo
10. On github, submit a pull request to our original repository

DEMO IV!

Links + Further reading

- Git documentation
  - https://git-scm.com/docs
  - http://wildlyinaccurate.com/a-hackers-guide-to-git/
- Git + GitHub workflow
  - https://guides.github.com/activities/hello-world/
  - http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1
  - http://blog.udacity.com/2015/06/a-beginners-git-github-tutorial.html
- Some other SCMs
  - Mercurial
  - SubVersion
- Some other Git hosts
  - Phabricator
  - Bitbucket
  - GitLab

Fin.

Happy h4ck1ng!