Command Line & Git

Linux Philosophy

- Do one thing and do it well. → Single-tasking is your friend!
- Succeed silently, fail noisily. → This requires **trust** in the system.

Command Line

Command Line Building Blocks

- commands
- flags
- parameters
- permissions
- examples:
    - cp -r dir1 dir2
    - ls -la
    - heroku run console --app myproject

Core Commands

- cd
- ls
    - What's shown by default? What's hidden?
    - ls -la → What information can we glean?
- pwd
- cp, cp -r
- mv
- cat
- less
- man
- sudo
- | (pipe = send output from one command as input to another command)
- > (output)
- >> (append)

What To Do at Vim-like Prompts

- esc (escape key kicks you out of edit mode)
- : (colon key opens command mode)
- w (write = save)
- q (quit)

Stuff We're Not Going to Do Today

- vi / vim / emacs / nano
- chmod
- chown
- shell scripts

Search and Ye Shall Find (Maybe)

- grep
- find
    - 25 simple examples of Linux find

Git

Git Guts

- Commit SHAs
    - Secure Hash Algorithm, aka "hash"
    - Unique identifier
    - Useful argument to many git commands
    - Changes if you modify a commit in any way (amend, rebase)
- Diffs
    - What are the differences?
- Deltas
    - What's a "delta"?
    - Git's smallest building block

What happens when you branch?

- Git calculates and applies deltas
- The filesystem is changed in place
- Local branches vs. remote branches

Git Commands

- git status
- git add, git add -i
- git revert
- git commit
- git push
- git pull (= git fetch + git merge)
- git fetch
- git checkout
- git branch, git branch -a, git branch -v, git branch -vv, git branch -avv
- git log, git log --graph
- git diff

- git show [HEAD|<SHA>]
    - author vs. committer
- git merge
- git cherry-pick
- git stash
- git bisect
- More stuff we won't do today:
    - git reset
    - git reset --hard
    - git rebase, git rebase -i
    - git push --force

A Word About Git Rebase

- Treat it with respect
- Don't freak out about it
- Do it the right way, at the right time, and understand the risks
- Respect your team's Git workflow
- What exactly is "good branch hygiene"?

Git Locations

- HEAD
- HEAD~n
- <SHA>

The Secret Git Folder

- cat .git/HEAD
- cat .git/config
- ls .git/heads/refs

Merge Conflicts

- How do they happen? (Deltas are the key!)
- What do they look like?
- How do we resolve them?
- How to abort a bad merge
- Pull often, push thoughtfully