Git Basics and Practical IT Use

Aaron Burgemeister
Identity / Security / Linux Consultant
aburgemeister@gmail.com

Version Control 101

- Keep Files Safe
  - Losing files, especially tools/code, is lame, discouraging
- Share With Others
  - Leads to efficiency in large groups, to thousands of peers
  - Find the bugs; many eyes make bugs shallow
- See Progress Over Time
  - Personal growth, career growth

Git Basics

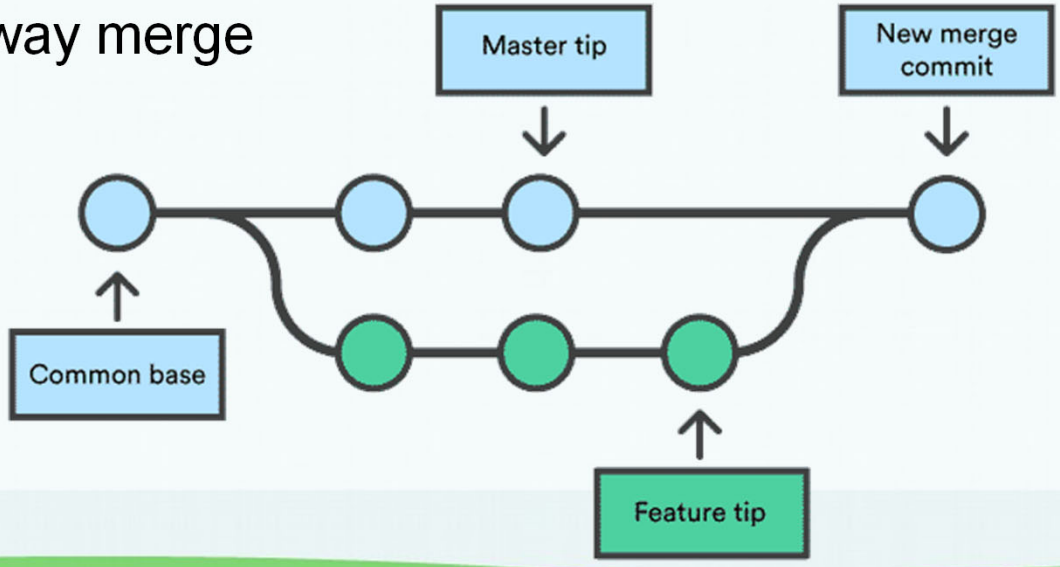
- Repository – A collection of files under version control
  - Local – the one you work in
  - Remote – Optional, and the one somewhere else
- Clone – Copying a COMPLETE repository locally
- Working/Staging Area – Where actual work (coding) happens
  - Tracked vs. untracked files
  - Staged vs. modified/dirty files
- Commit – Snapshot of ALL code at a specific point
  - Includes all commits leading up to that specific commit

Git Basics

- Sharing with Others
  - Push – Send local repository commits to a remote
  - Pull – Get commits from a remote to the local repository and merge
  - Fetch – Get commits from a remote to the local repository
- Branch
  - A movable pointer to track a sequence of commits with a common purpose
- Tag
  - A permanent pointer to a commit, often used to track a release

Git Basics

- Merging
  - Joining multiple commits together
  - Necessary when working with others, or even on our own
  - Three-way merge

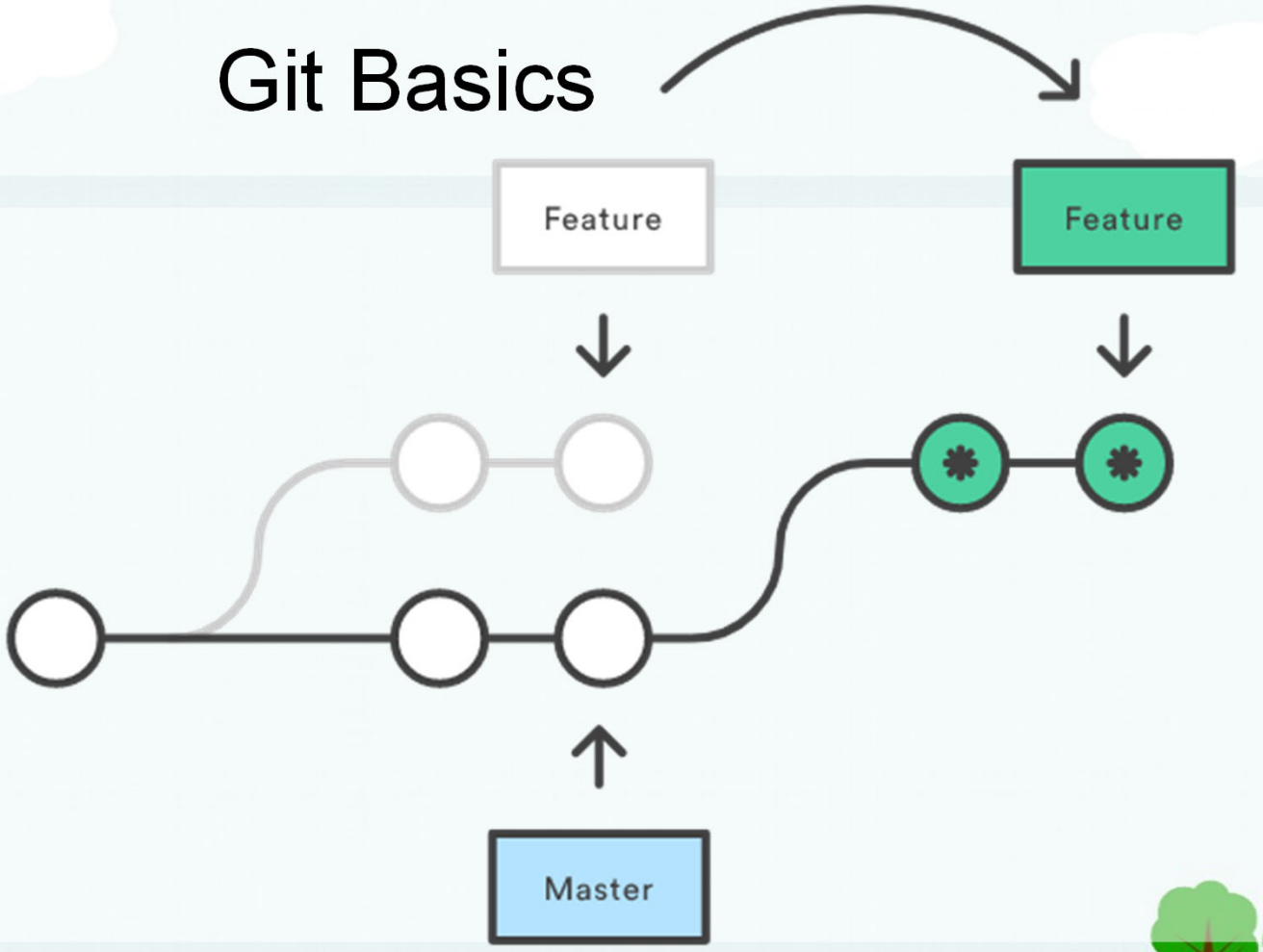


Git Basics

- Fast-forward
  - Possible only when a direct linear path from current commit to the new commit exists (no merges have happened in between)
  - Leaves a clean, linear, commit history
- Rebasing
  - Joining multiple commits together by rewriting history
  - Creates a linear history (fast-forward), so easier to find bugs
  - Uses three-way merge, but changes new commits along the way
  - Should not be used with new commits that are public

Git Basics

- Rebasing



Git Basics

- Log
  - Git command to show commits
    - Related to the current branch
    - Related to a specific file
    - Related to strings in the history (pickaxe)

Use in Information Technology (IT)

- Requirements, Policies, Rules
  - Track changes to requirements over time, including who made them
- Scripts
  - Tools you should already be creating to do your job better, faster
- Test cases
  - If you have code, you should have test cases; track them together
- Code
  - Of course, any other code
- Documentation, Daily Work Notes

Warnings / Cautions / Hazards

- Do not store secrets in a repository
  - They might exist in the directory structure, but are never committed
  - Use the .gitignore file to explicitly ignore them forever from the start
  - Examples:
    - SSH keys
    - Private certificate information
    - Password files
    - Scripts with hard-coded passwords

Get More Git Information

- An interactive, cross-platform Git tutorial
  - https://github.com/jlord/git-it-electron#what-to-install
- A great walk-though from Atlassian
  - https://www.atlassian.com/git/tutorials
- Git command cheat sheet
  - https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet
- Linux Kernel Repository Clone:
  - git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git linux-git