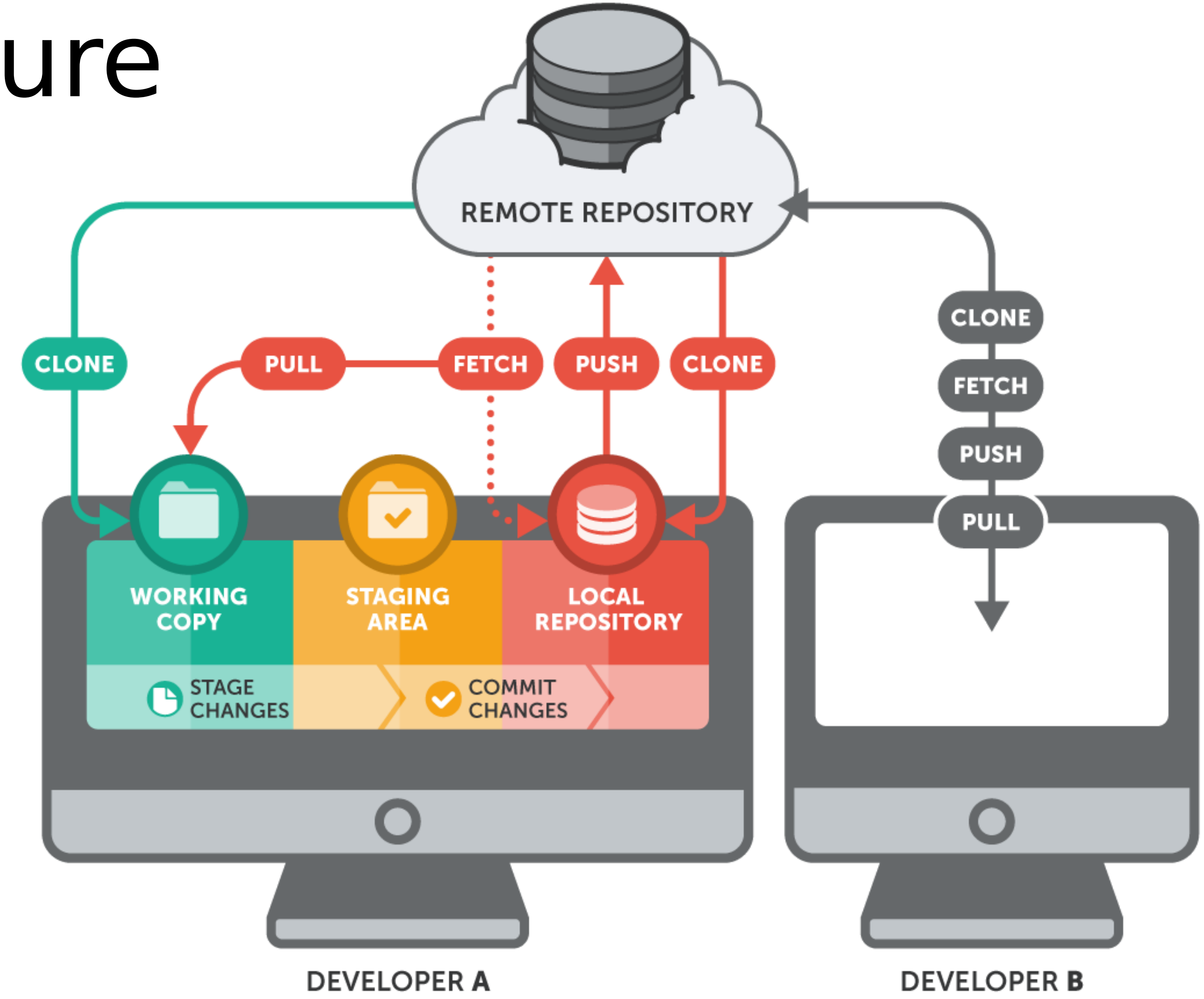
GIT Source Control

Purpose

- Helps maintain record of code development
  - A detailed backup
  - Allows you to revert to previous versions of code
- Aids in collaboration among coders
  - Allows coders to work edit copies of the code on local machines and automatically merge versions together
- GIT is a suggested tool for this project but is NOT required

Structure



http://gdibtv.github.io/gdi-core-git-github/#/

Setup

- Git comes with most Linux distributions
    - Available on linprog
- Can be installed on other OS’s as well
- Online repositories:
    - Github
    - Bitbucket (free public and private repositories)
    - Gitlab
- Your repository for this project MUST be private and inaccessible to anyone outside of your team

Creating a New Repository

- From scratch:
  - git init
  - git remote add origin <location>

- From an existing remote repository:
  - git clone

Specify which files to track

- git add <file> to begin tracking file through git
- Ignore files using .gitignore

Commits

- git commit –am “message”
  - -a to commit all changes
  - -m to add message
- Essentially confirms you want to record the changes made since the last commit
- Commits do NOT change the remote repository, they only commit changes to your local repository

Accessing the remote repository

- git push origin <repo>
  - pushes your changes to the remote repo
- git pull
  - pulls changes made to the remote repo (by others)

Branches

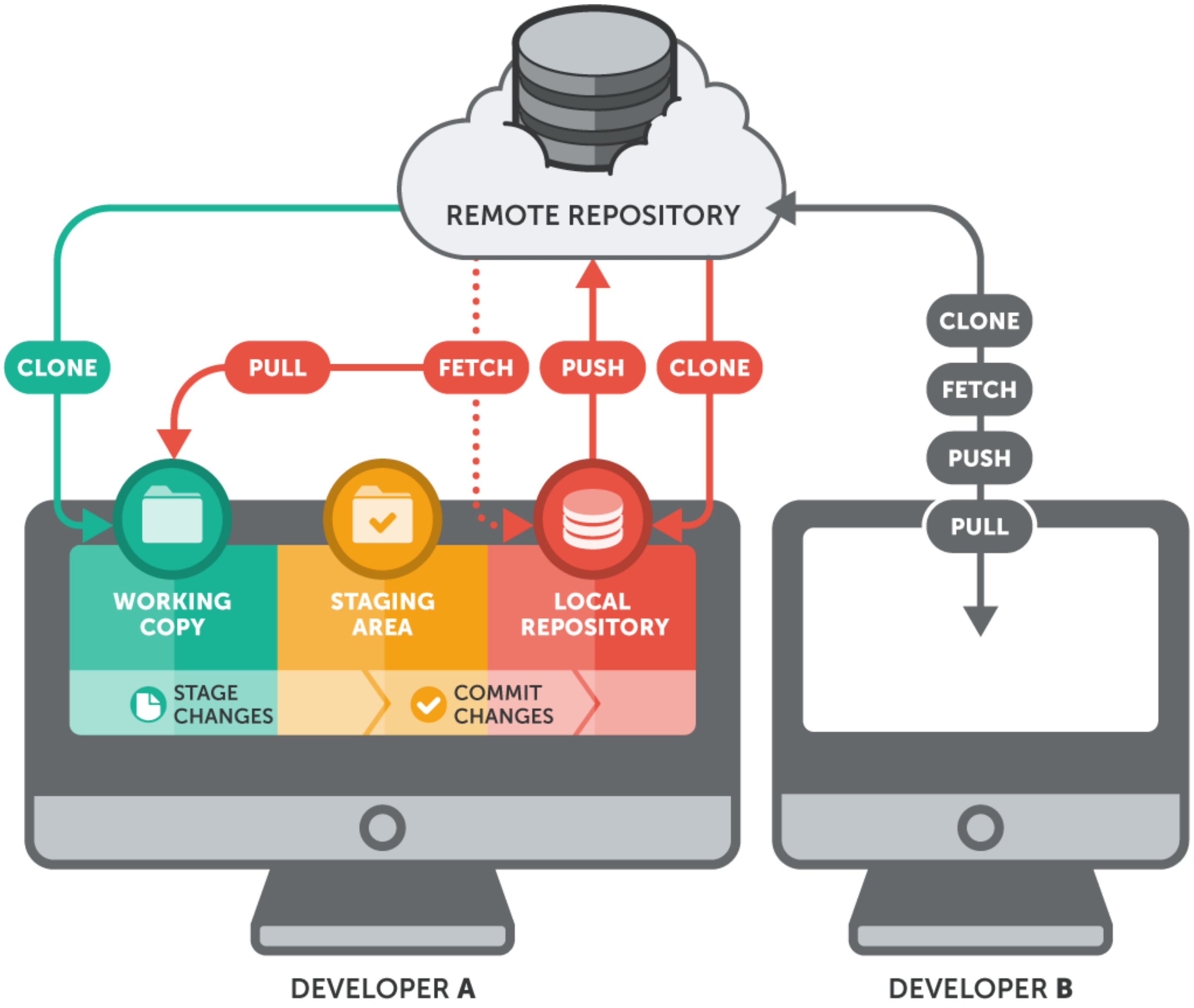
- Main branch usually called master branch
- Good practice to create new branch for each new feature
- git checkout –b <branch_name>
    - Creates new branch and switches you to that branch
- git checkout <branch_name>
    - Switches you to already existing branch named branch_name
- git merge <branch_name>
    - Merges changes from <branch_name> INTO current working branch
    - Ie) if I am in the master branch and run git merge leah, it will pull all the changes I have committed in branch leah to the master branch

Oops

- If you REALLY messed up your local repo and want to erase your local changes and remake your local repo as a copy of the remote repo, you can run:
  - git fetch origin
  - git reset -- hard origin
- Warning: be sure your remote repo contains everything you want because your local changes will be lost

Other tips

- Branching makes things a bit more complicated. Although it is good practice to create a new branch for each feature, for this project, it is probably fine for everyone to work off the master branch.
- If you are using branches, pay attention to where you are executing each command from!
- git status and git log are additional commands which help see the status of your git branches/activity
- Simple guide: http://rogerdudler.github.io/git-guide/



http://gdibtv.github.io/gdi-core-git-github/#/