SimCenter NHERI
Center for Computational Modeling and Simulation

2020 Programming Bootcamp

Git

A Distributed Version Control System

Frank McKenna
University of California at Berkeley

NSF award: CMMI 1612843

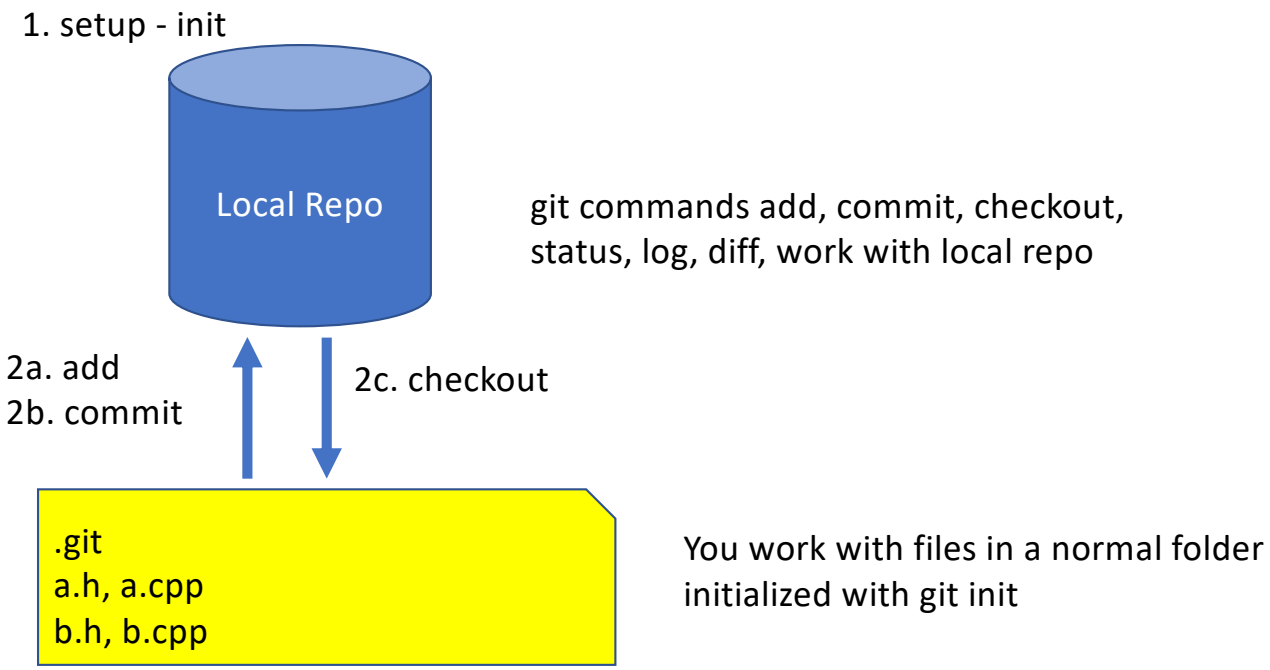
Version control systems

- Version control (or revision control, or source control) is all about managing multiple versions of documents, programs, web sites, etc.
  - Almost all "real" projects use some kind of version control
  - Essential for team projects, but also very useful for individual projects
- Some well-known version control systems are CVS, Subversion, Mercurial, and Git
  - CVS and Subversion use a "central" repository; users "check out" files, work on them, and "check them in"
  - Mercurial and Git treat all repositories as equal
- Distributed systems like Mercurial and Git are newer and are gradually replacing centralized systems like CVS and Subversion

Why version control?

- For working by yourself:
  - Gives you a “time machine” for going back to earlier versions
  - Gives you great support for different versions of the same project.
- For working with others:
  - Greatly simplifies concurrent work, merging changes

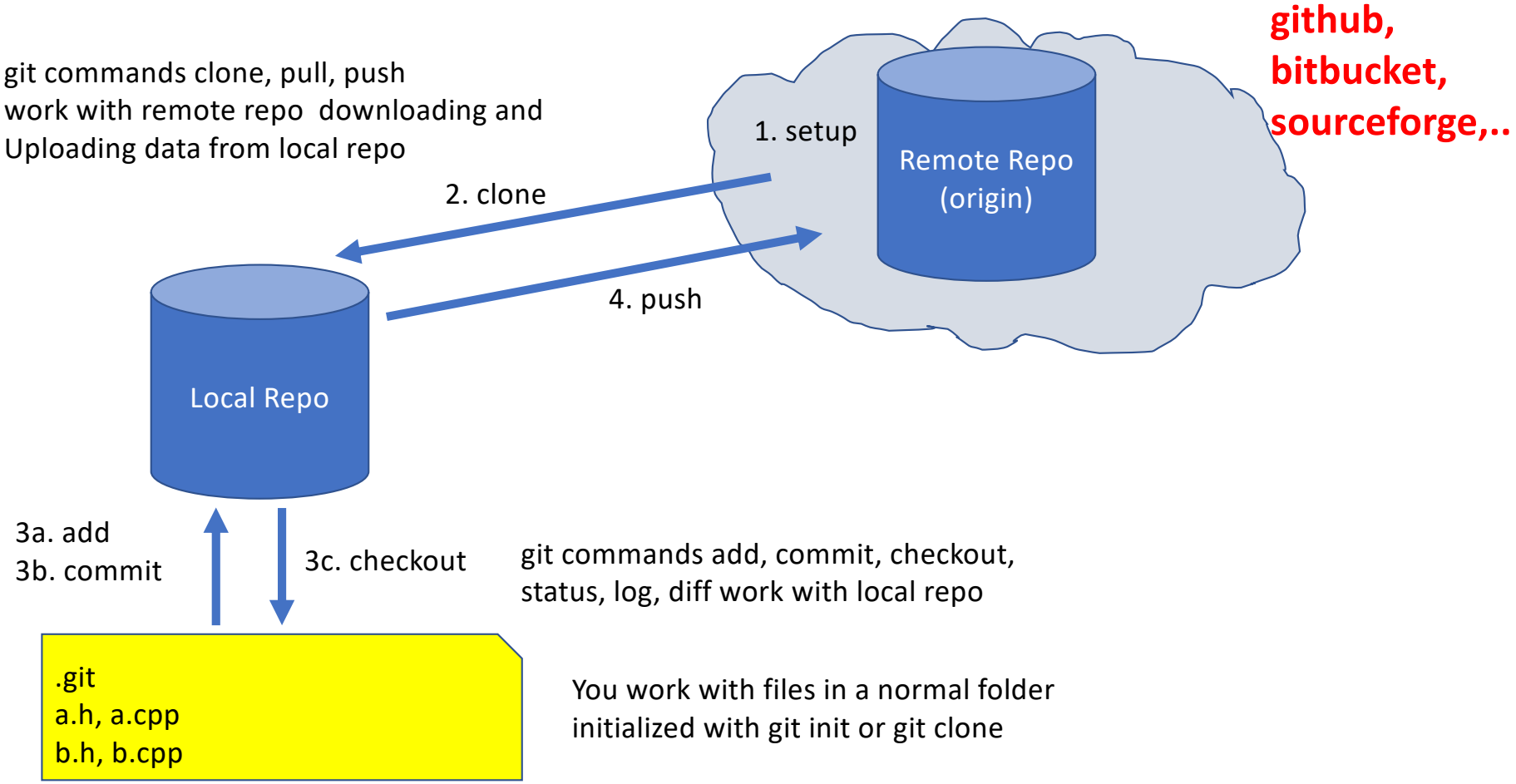
Local Repo on your local desktop



Demo

1. Create some files in local directory
2. Create local rep
   git init
3. Add files
   git add .
4. Commit file
   git commit –m “fmk – initial commit”
5. Edit files
6. Use checkout to get last committed version
   git checkout file
7. Add file
   git add file
8. Commit
   git commit –m “fmk – next commit”
9. Tag
   git tag –a v1.0 0 -m "Version 1.0"

Personnal Online Repo



Demo

1. Create repo at github.com: fmckenna/test in browser
2. Point local test to it

   git remote –v

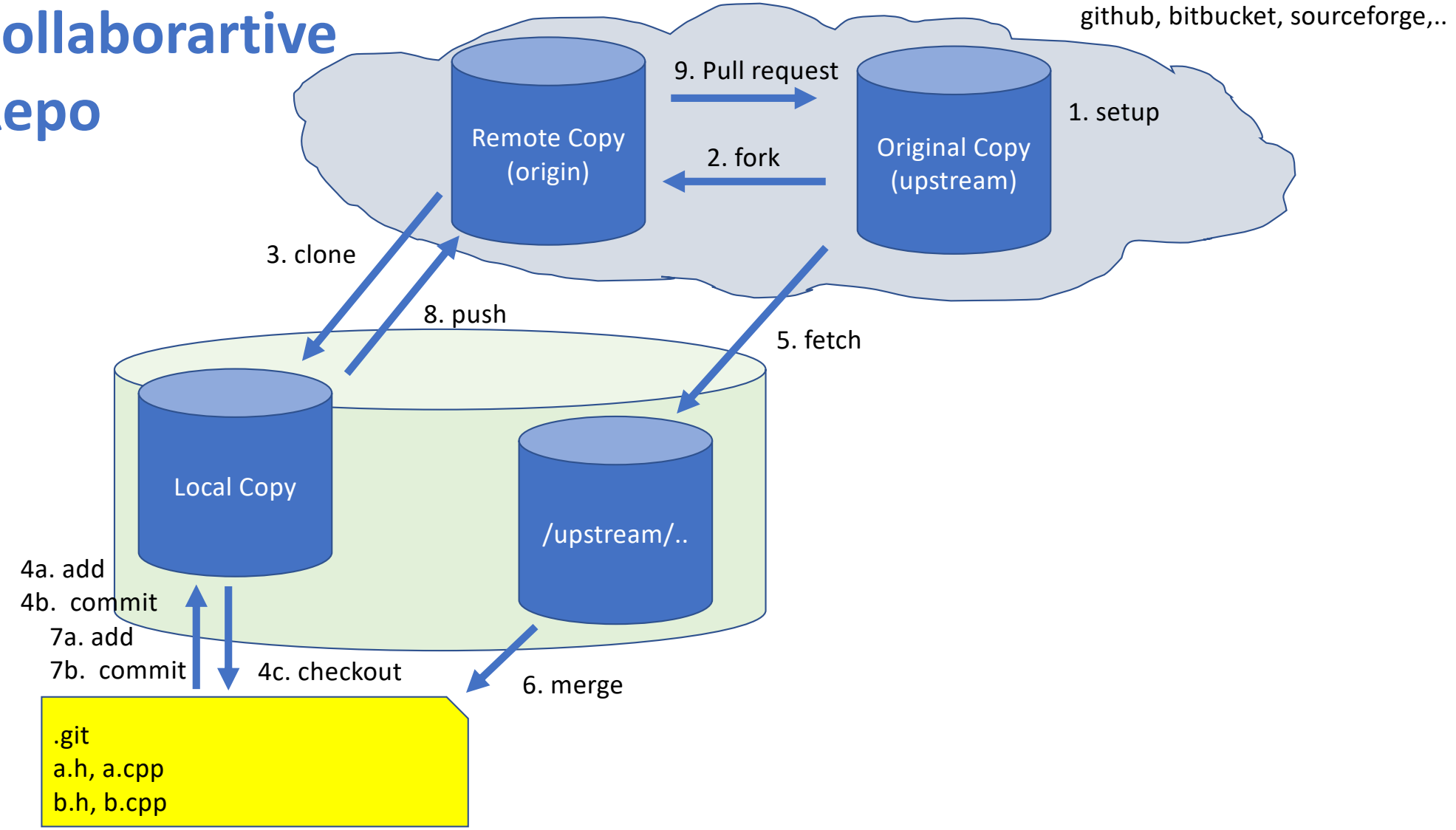
   git remote add origin https://github.com/fmckenna/test.git

   git add .
3. Push to remote

   git push
4. Look at files in browser

   git add file
5. (If time clone that one)

Collaborartive Repo



Exercise – Lets Synch your FORK with SimCenter

1. Open terminal window/ Powershell
2. cd to SimCenterBootcamp2020 folder
3. Lets add and commit any changes (or checkout original) .. Type: **git status**
   git add .
   git commit –m "my great changes"
4. Lets look at remotes repos
   git remote -v
5. Add NHERI-SimCenter/SimCenterBootcamp-2020 as upstream
   git remote add upstream https://github.com/NHERI-SimCenter/SimCenterBootcamp2020.git
6. Look at remotes again
   git remote -v
7. Lets fetch upstream
   git fetch upstream
8. Use checkout to get last committed version
   git merge upstream/master
9. Git add, commit and push to our own remote repo (origin)
   git add .
   git commit –m "fmk – updated code from merge"
   git push origin master
10. Go to your browser, your remote repo (origin) should contain latest code