Getting Started with Git

Introduction to version control

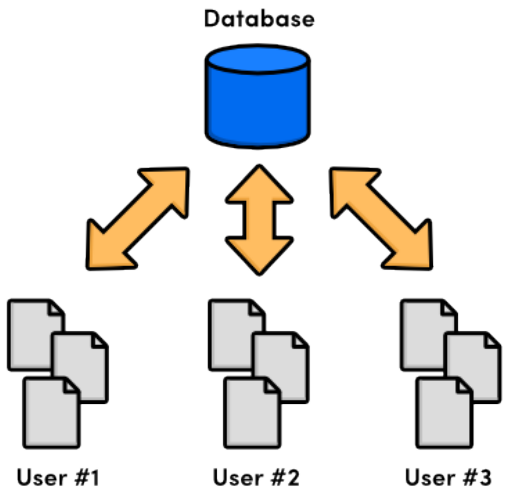
Benefits of using Git

Basic commands

Workflow

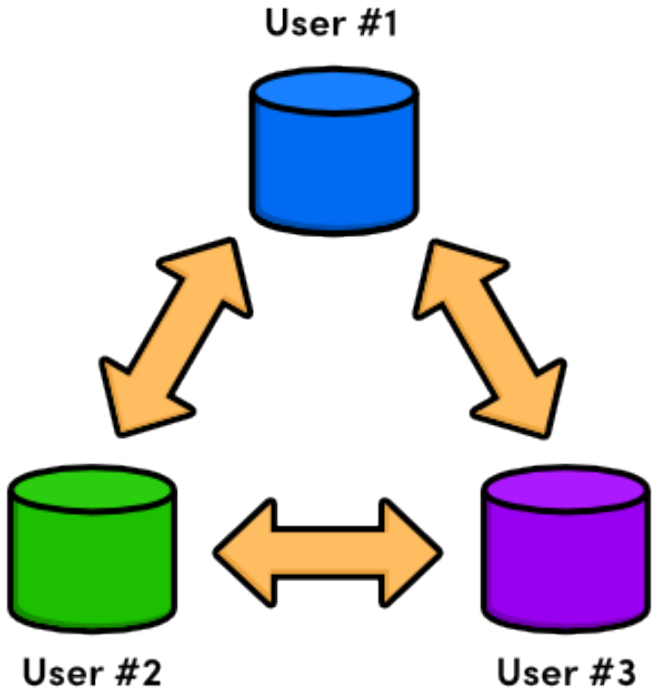
Version Control Systems

- Goal of a Version Control System
  - Software that manages changes that you make to your files (source code).
  - Track versions of each file (more accurately, versions of *sets of changes* to your files).
  - Handles concurrent changes from multiple sources (e.g. different developers working on the same code base).
  - Typically some central repository that stores every version of every file.
- Popular VCS'
  - SourceSafe (local/file based)
  - CVS, Subversion, Perforce (centralized)



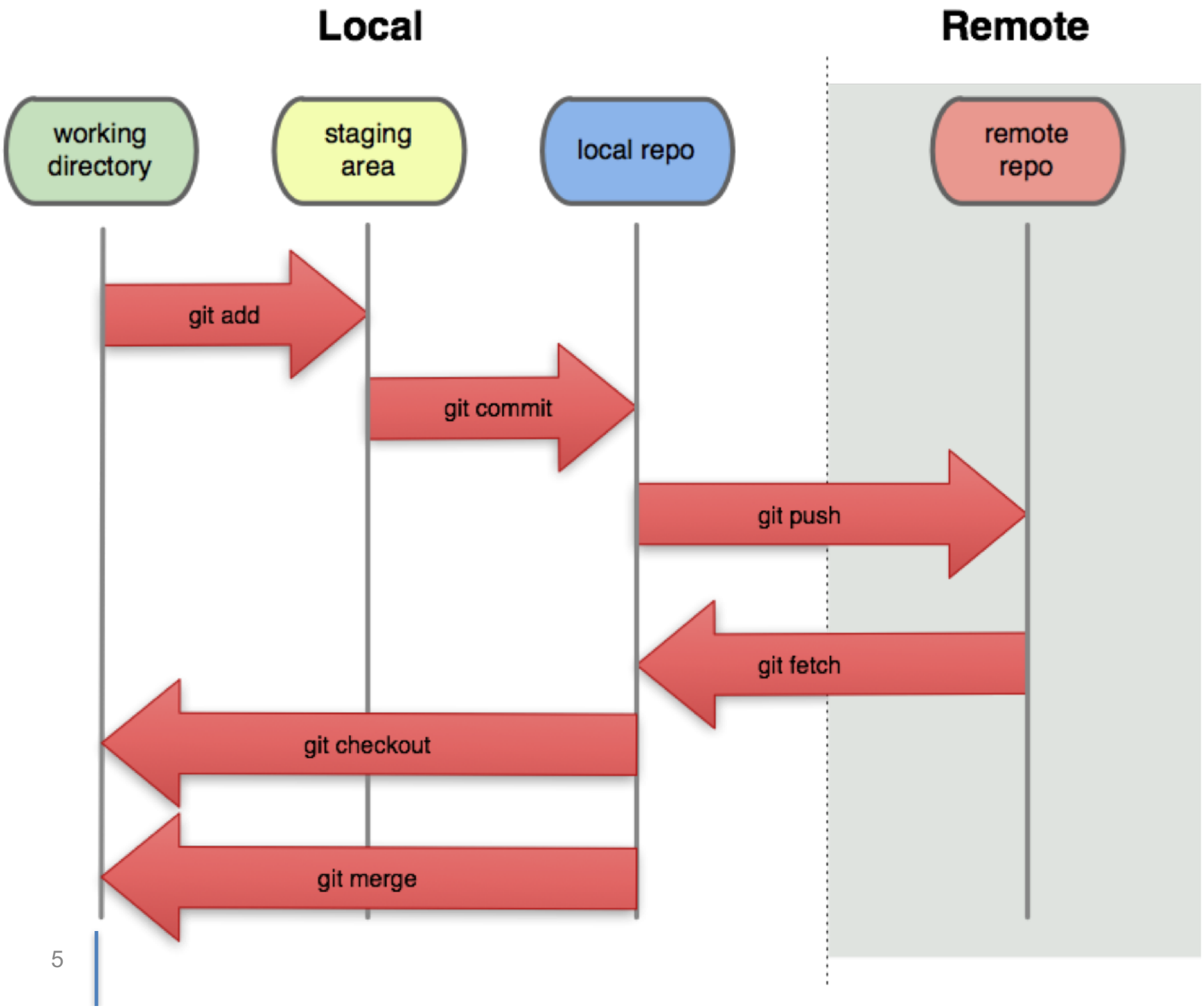
Distributed Version Control

- Distributed version control systems (DVCS)
  - No central server required!
  - Every user has a copy of every file.
- We use Git, a very popular DVCS.
  - Developed in 2005, to manage development for the Linux kernel (Bitkeeper -> Git)
  - Very specific design goals
    - Large-scale development
    - Distributed
- Git doesn't require a server, but it's common to use one for coordination
  - e.g. Github



Concepts

- Working directory
  - your local copy of the files that you're working with.
- Staging area
  - a "place" where you tell Git to hold a set of changes, temporarily.
- Repository
  - a place where Git stores copies of your files and their history.
    - Local repository: on your working machine
    - Remote repository: a server (e.g. GitHub)


Local
Remote
working directory
staging area
local repo
remote repo
git add
git commit
git push
git fetch
git checkout
git merge

Commands

- You perform these operations using a Git client (command-line or GUI, the work the same).
- Commands typically move files between working directory, staging area and local or remote repository.

Local Commands	Description
git add	Add a file from working directory to staging area
git commit	Commit changes from staging area to repo
git checkout	Get files from repo to working directory

Remote Commands	Description
git clone	Make a copy from remote repo to working dir
git pull	Pull (merge) changes from repo to working dir
git push	Push (merge) changes from staging area to repo

Installing Git

Before anything else, you need to install and configure Git.

1. Install a Git client (command-line or GUI)

   https://git-scm.com/downloads

2. Setup your email address for Git:

```
git config --global user.email
"your_email@example.com"
```

Working on A0

1. Get a copy of the repository from the server
   - Use '`git clone`' to get a copy of starting code from remote repository to your working directory.

```
git clone
https://<questid>@git.uwaterloo.ca/cs349-spring2016/<questid>.git
```

2. Work on the assignment
3. Save and commit your changes to git
4. Push changes to the server

Working on A0

1. Get a copy of the repository from the server
2. Work on the assignment
   - copy the `Check.java` file into your working directory.
   - '`javac Check.java`' to compile
   - '`java Check`' to run and produce `results.txt`
3. Save and commit your changes to git
4. Push changes to the server

Working on A0

1. Get a copy of the repository from the server
2. Work on the assignment
3. Save and commit your changes to git
   - '`git add Check.java results.txt`' to add files to your staging area
   - '`git commit`' to save the changes to the local repository.
4. Push changes to the server

Working on A0

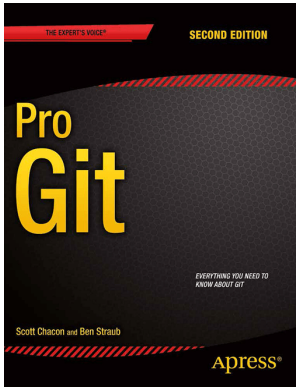
1. Get a copy of the repository from the server
2. Work on the assignment
3. Save and commit your changes to git
4. Push changes to the server
   - ‘`git push`’ to push to the remote repository.

"Good Ideas"

1. Edit `.gitignore` file in your working copy
2. Consider cloning via SSH or HTTPS.
   - For SSH, generate public/private keys.
     - https://git.uwaterloo.ca/help/ssh/README
   - If you use HTTPS, you might want to cache credentials
     - https://git-scm.com/docs/git-credential-store
3. Learn how to use tags or branches
   - See documentation

Resources

- Git Home
  - documentation, binaries
  - https://git-scm.com
- Git Reference
  - cheat-sheet of commands
  - http://gitref.org
- Book: Pro Git
  - Scott Chacon and Ben Straub
  - extensive manual
- Ry's Git Tutorial
  - http://rypress.com/tutorials/git/index



RyPress