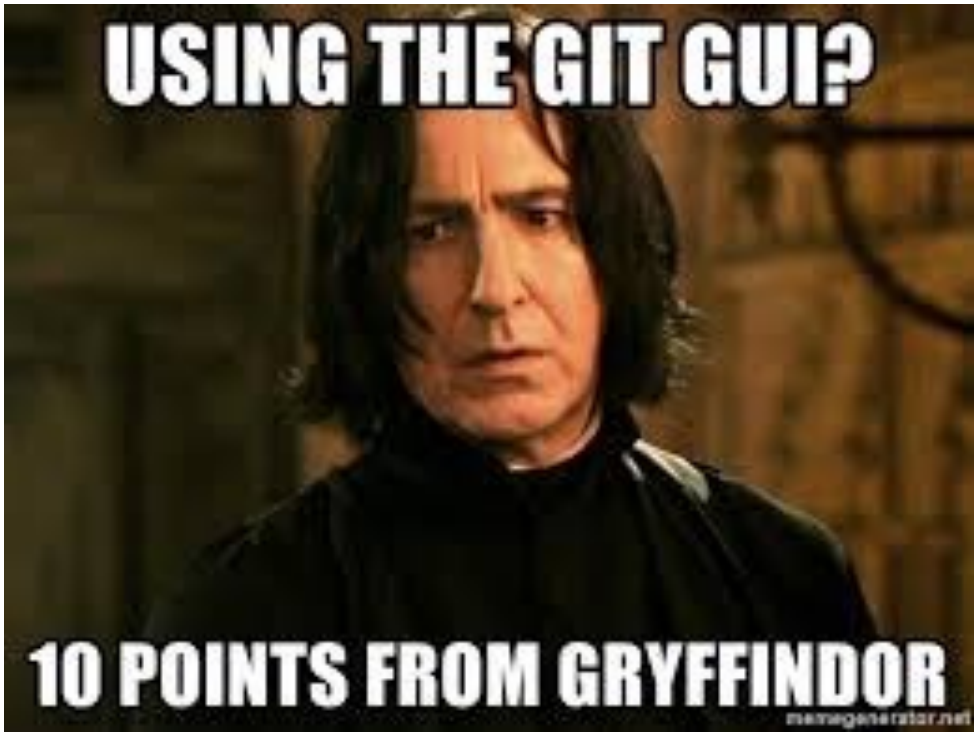
UNIVERSITY OF NOTRE DAME

MENDOZA COLLEGE OF BUSINESS

Software Engineering
ITAO 30210

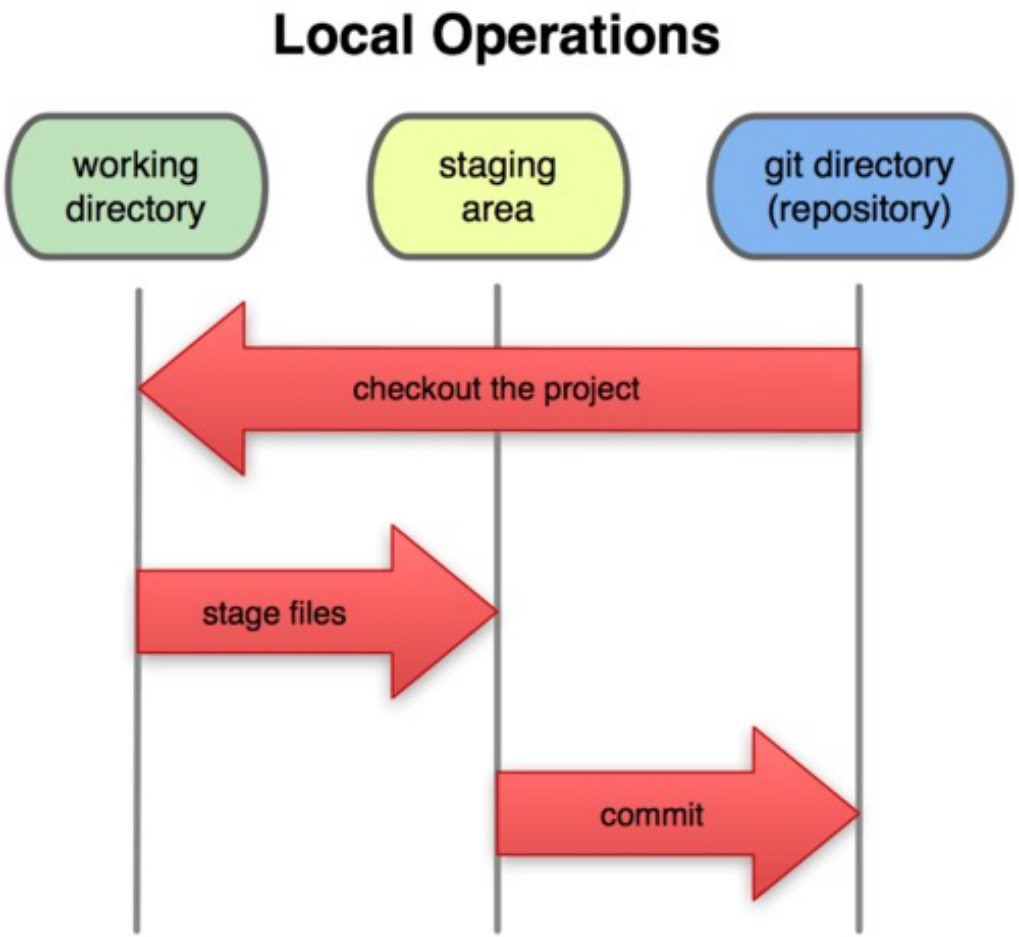
Source Control with Git


USING THE GIT GUI?
10 POINTS FROM GRYFFINDOR
memegenerator.net


git remote add origin
git remote origin add
git add remote origin
git add origin remote
JAKE-CLARK.TUMBLR

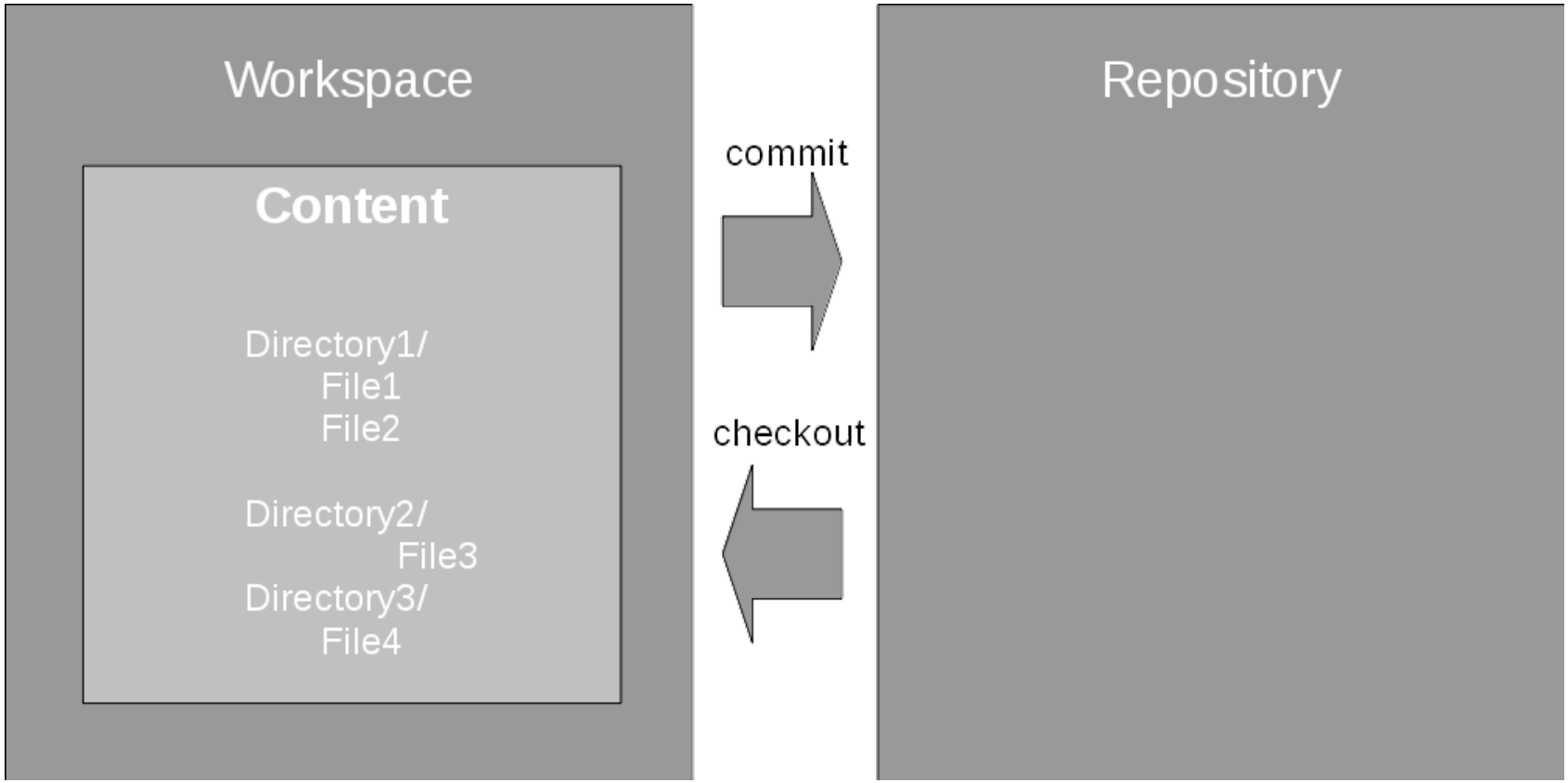
Why Source Control?

- Track every change you make to a project
- Track every change everyone else makes to a project
- Work on new features without breaking the current code base
- Return to any point in time, at any time
- Know exactly what code is running in your production environment



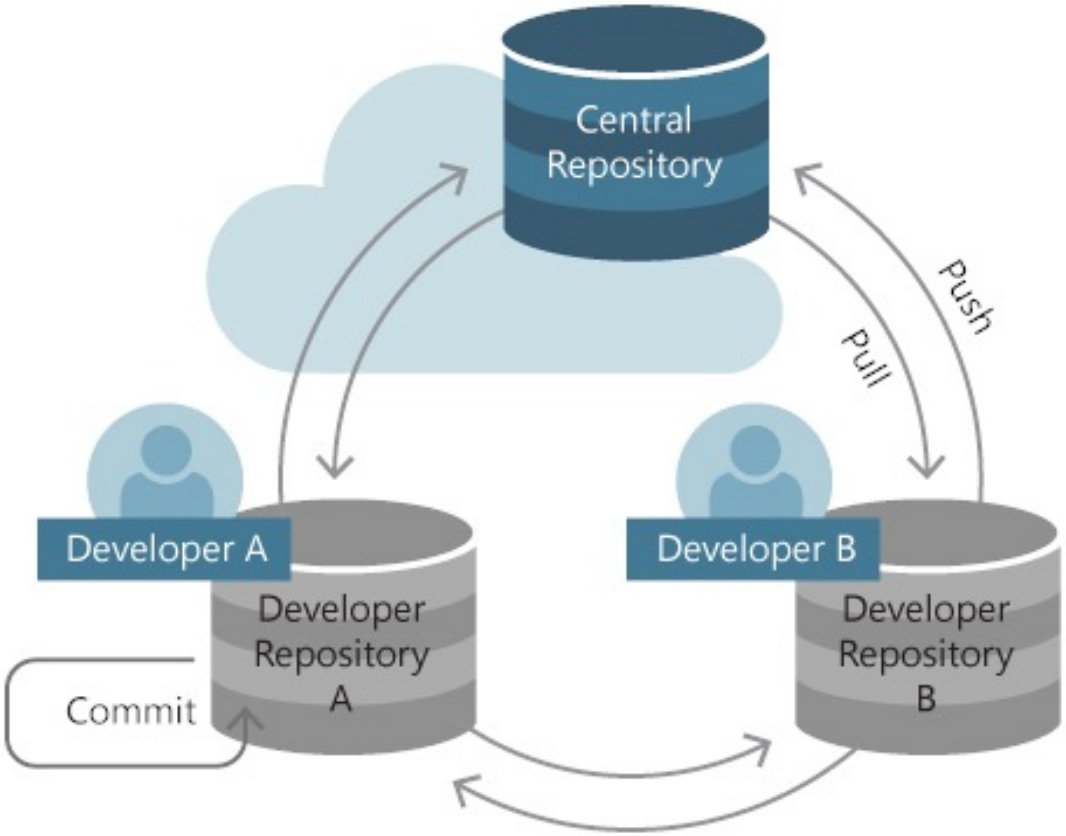
https://juristr.com/blog/2013/04/git-explained/

Git vs. Classical Source Control



https://developer.ibm.com/technologies/web-development/tutorials/d-learn-workings-git/

Git vs. Classical Source Control



https://docs.microsoft.com/en-us/devops/develop/git/set-up-a-git-repository

Git Terminology and Workflow

- Git
  - A CLI source control system.
  - Can be installed on your computer or on a remote server for distributed work
- GitHub
  - Third party service using git but with it's own set of services and features
- BitBucket
  - Similar to GitHub, but a different service

Git Terminology and Workflow

- Repository (repo)
  - The code base and all the changes that have occurred
  - The **`remote`** repo is the code base that all developers share and interact with (GitHub, BitBucket)
- Master/Main/Production
  - The branch that is considered the current working version of the code base. Name can very but Master has been the convention

Git Terminology and Workflow

- `git init`
- `git clone`
  - Create a new repo or retrieve a repo from remote location

Git Terminology and Workflow

- `git status`
  - Lets you know if you
    - have files to be added (untracked files)
    - files not yet committed (not staged)
    - Files updated locally but not on remote (ahead of…)

Git Terminology and Workflow

- `git add`
  - When you create new files, you must tell git that there are new files

Git Terminology and Workflow

- `git commit`
  - When you add or update files, you must tell git about the changes.
  - When you commit, you must add a message.
  - Messages should explain the nature of the change and are very helpful to your future self and other developers

Git Terminology and Workflow

- `git push`
  - When you add files and commit changes, it is all done locally on your workstation.
  - Once you are happy with the state of the code, you push your changes to the remote repo
  - Once you do this, anyone else working on the code will be able to incorporate your changes

Git Terminology and Workflow

- `git push`

```
(virtualenv) (base) jarp@jarp-mbp16 Python-Examples % g po main
To github.com:ITA0-40540/Python-Examples.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'git@github.com:ITA0-40540/Python-Examples.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

Git Terminology and Workflow

- `git pull`
  - Once a developer has pushed their changes to the remote repo, you still will not see the changes
  - To incorporate those changes, you pull them from the remote repo to your local repo

Git Terminology and Workflow

- `git branch`
  - Most of the time when you are working on a new feature (or even a bug fix), you will want to create a branch off the current code base
  - You will name the branch something that explains what you are working on
    - image-uploads
    - duplicate-username-bug
  - Branches live in their own universe and all the previous steps (add, commit, push, pull) will be used on the branch

Git Terminology and Workflow

- `git merge`
  - Once the work is done on the branch, you merge the changes there into the main branch
  - Post-merge, you are free to delete the branch
  - Post-merge, you push the updated main branch
  - Others then can pull your changes

Git Terminology and Workflow

- Conflicts!!!!!
  - If two people have worked on the same file (usually the same lines of code) sometimes git cannot merge the changes and will notify you of the conflict
  - In cases of conflicts, git will show you the differences and you must resolve the conflict.
  - Then you commit and push

Git Terminology and Workflow

- `git tag`
  - Along with branches, there are also tags
  - Tags are labels given to the code base at a specific place in time
  - Unlike branches tags generally should never change
  - Often used to label the code base at time of deploy but can be used for any reason
  - Once you create a tag, you push it to the remote repo

Git Terminology and Workflow

- Repeat for the life of the project

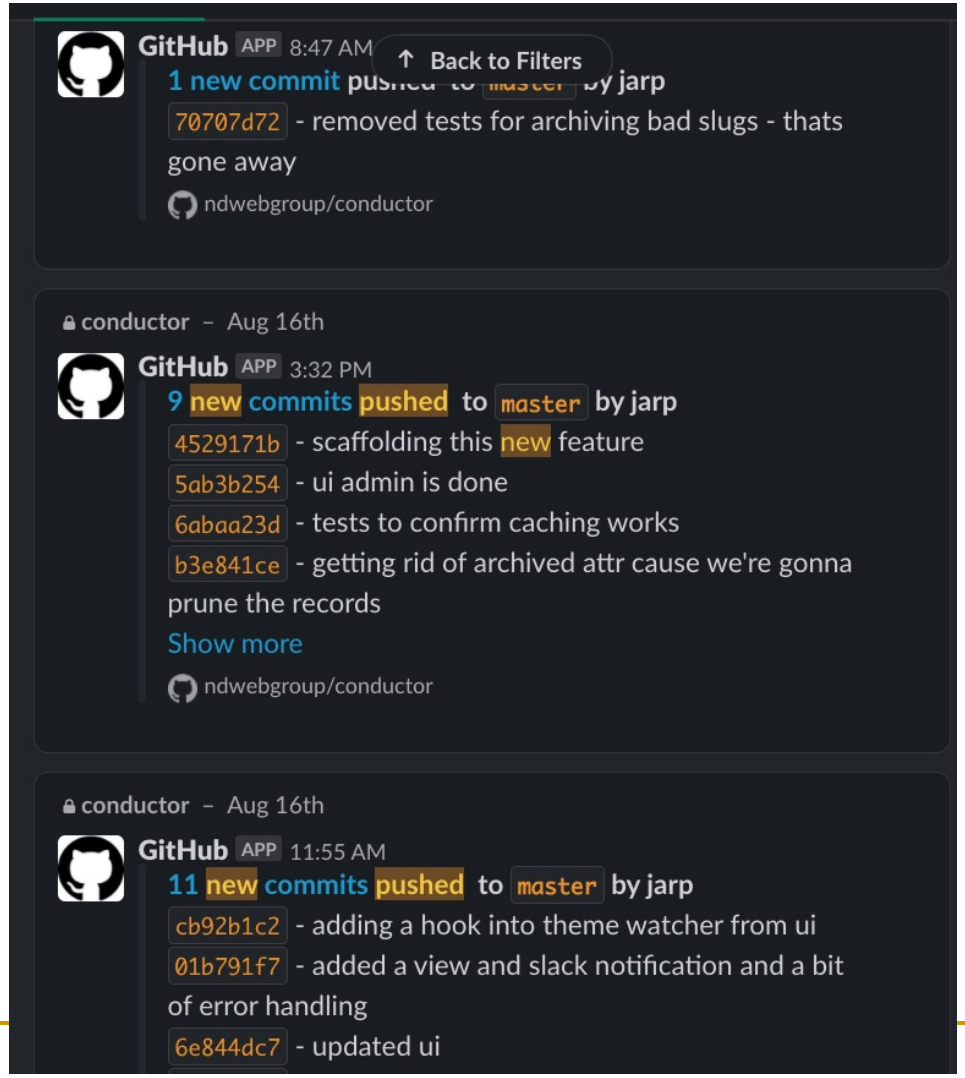
Git Recommendations and Best Practices

- Commit all the time
  - Like, every change… go ahead and commit it
  - Seriously, you can't commit too much
- Take the time to enter useful messages
  - There is a good chance you will go back to a commit and try to understand what you were thinking last year
  - So tell your future self with a good commit message
  - Same applys to co-workers

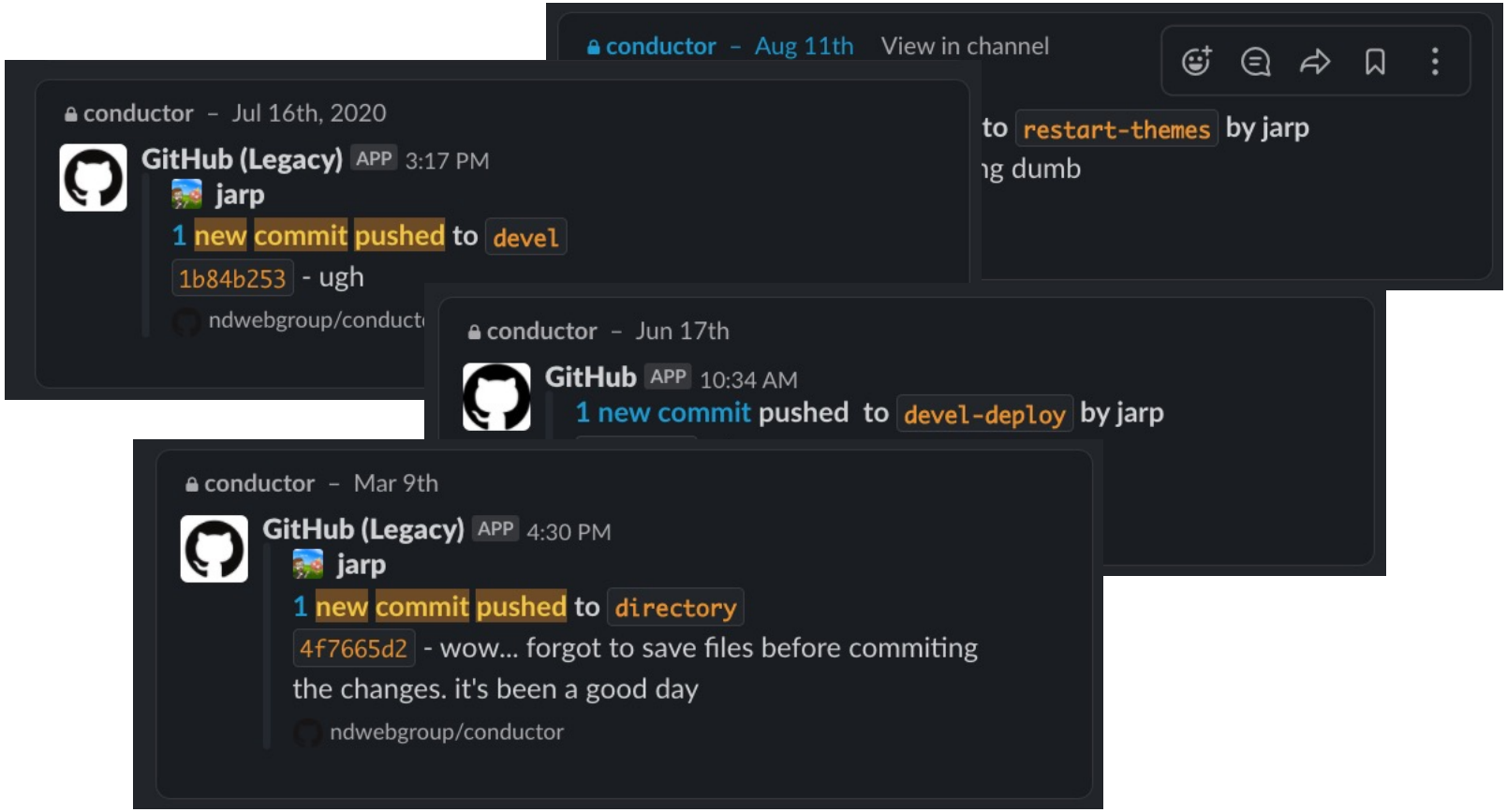
Git Recommendations and Best Practices

- Make use of branching for most of your development
  - This keeps your working code base clean and bug free
  - You may be working on a new feature and a bug (easy to fix) gets reported. If you are on a branch, you can easily switch over to master, make a new branch, and fix the bug
  - master should always be deployable

Useful Commit Messages



Useful Commit Messages



Git Recommendations and Best Practices

- Pull Requests
  - Not part of git the system, but introduced by GitHub and has become a standard feature of most git based systems
  - Allows you to:
    - push all the changes in a branch
    - Comment on the changes
    - Create a workflow for feedback and approvals

Git Recommendations and Best Practices

- .gitignore
  - Tells git which files should not be added to repo
    - Workstation files like .DS_Store on macs
    - Files with sensitive information and passwords
  - Uses patterns
    - Exclude specific files
    - Exclude directories
    - Exclude file extensions
  - There are templates available

SSH Commands

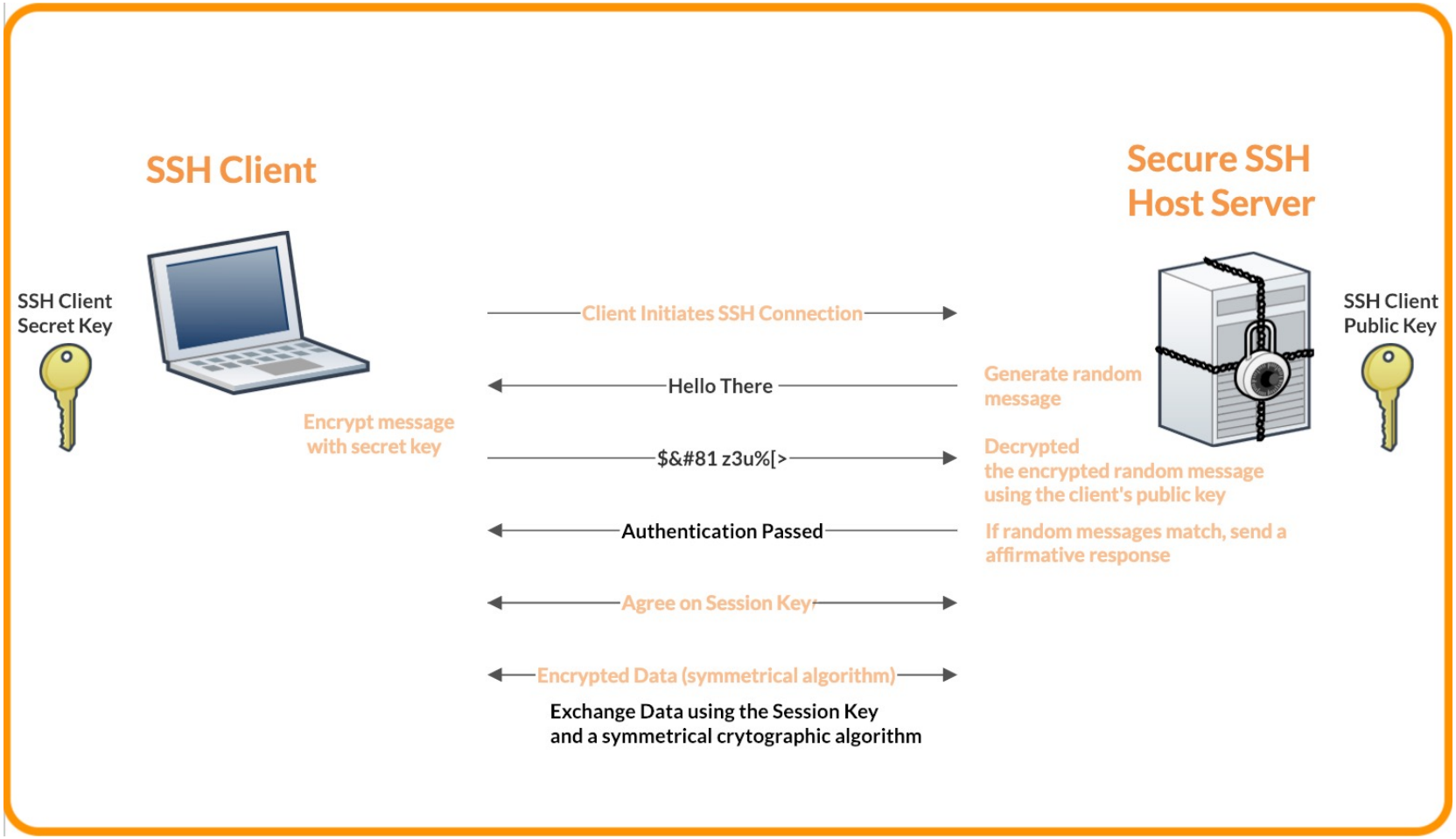
- Secure Shell is a network communication protocol that enables two computers to communicate. Similar to HTTP (web browsers)

- Common uses of SSH
  - Server Administration
  - Code deployments to servers
  - GitHub and other Server based services
- Syntax:
  - `ssh user@ip-address` (`ssh deploy@192.345.45.8`)
  - `git@github.com:jarp/Farkle-Cli-40540.git`

SSH Commands

- Communication between servers can be “key based” or login based
- Create a Key and Secret pair
- Let the target server know your “public key”
- Many Uses:
  - SSH into servers
  - Connect to git remote servers (GitHub)

SSH Commands



https://www.foxpass.com/blog/learn-ssh-keys-in-minutes/

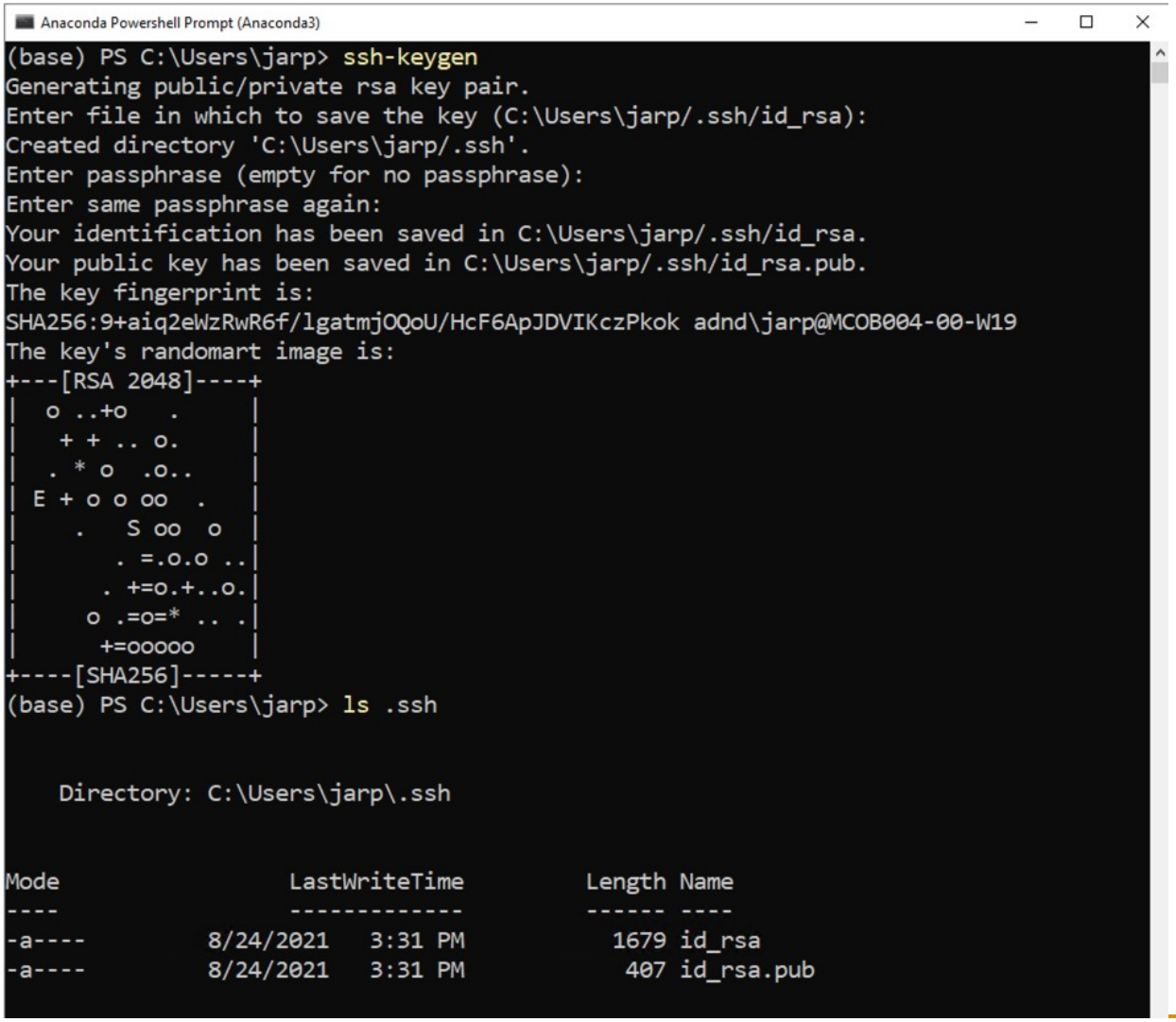
Setting up SSH with Key Gen

- Enter:

```
ssh-keygen
```

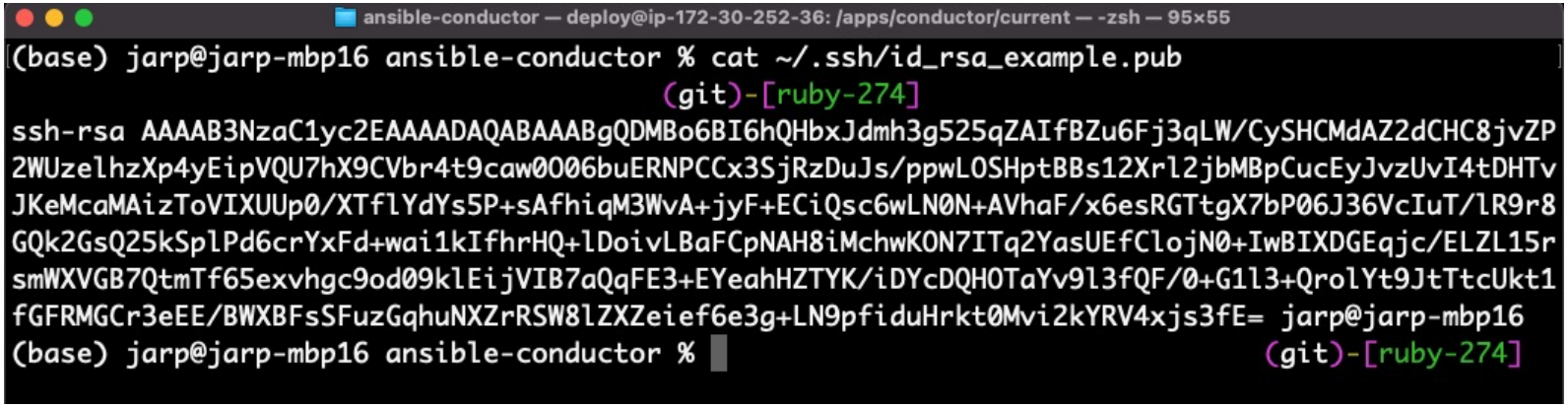
- Leave password blank (maybe not?)
- Creates two files in a ~/.ssh directory:
  - ~/.ssh/id_rsa
  - ~/.ssh/id_rsa.pub

Setting up SSH with Key Gen

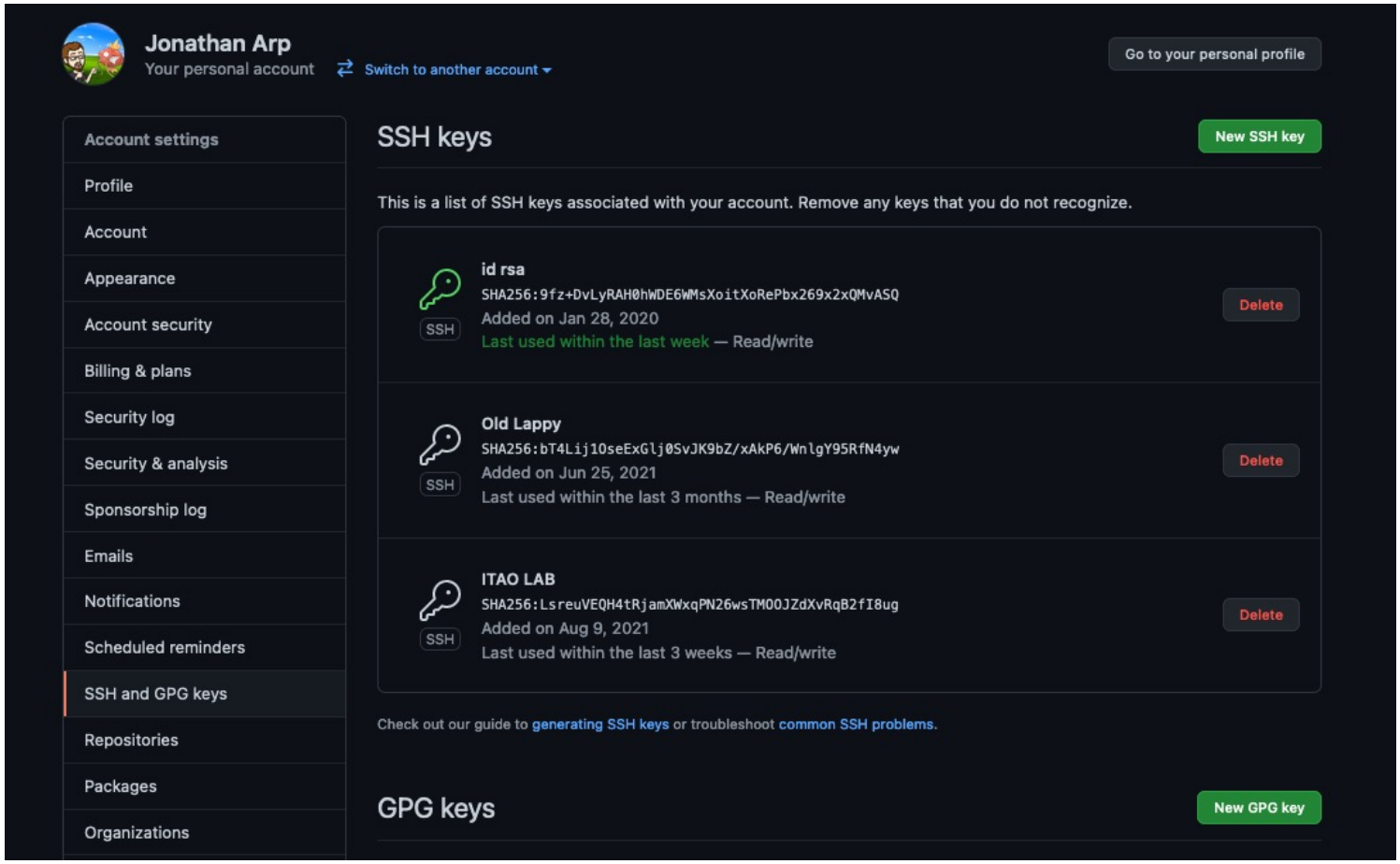


Setting up SSH with Key Gen

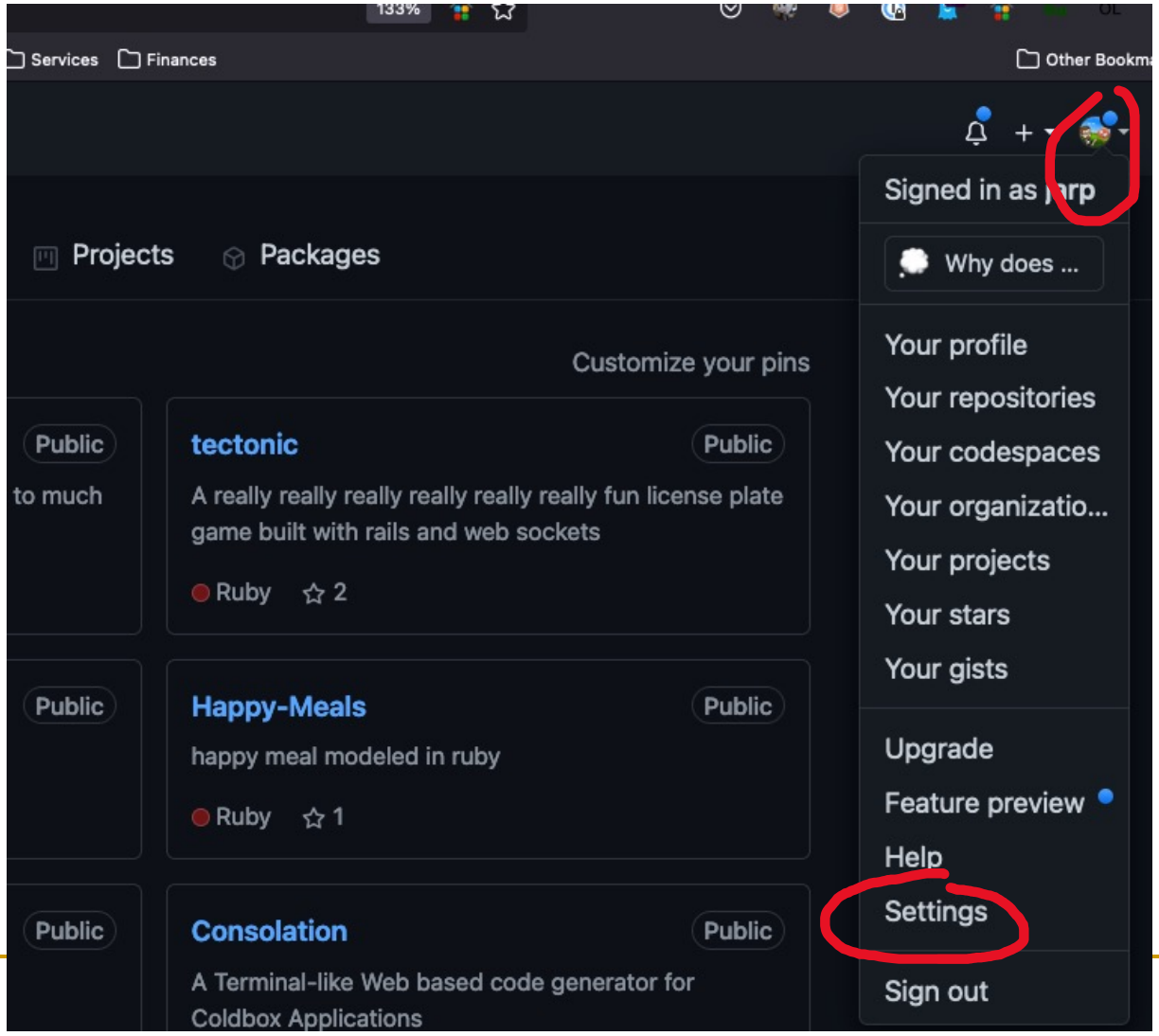
```
cat ~/.ssh/id_rsa_example.pub
```



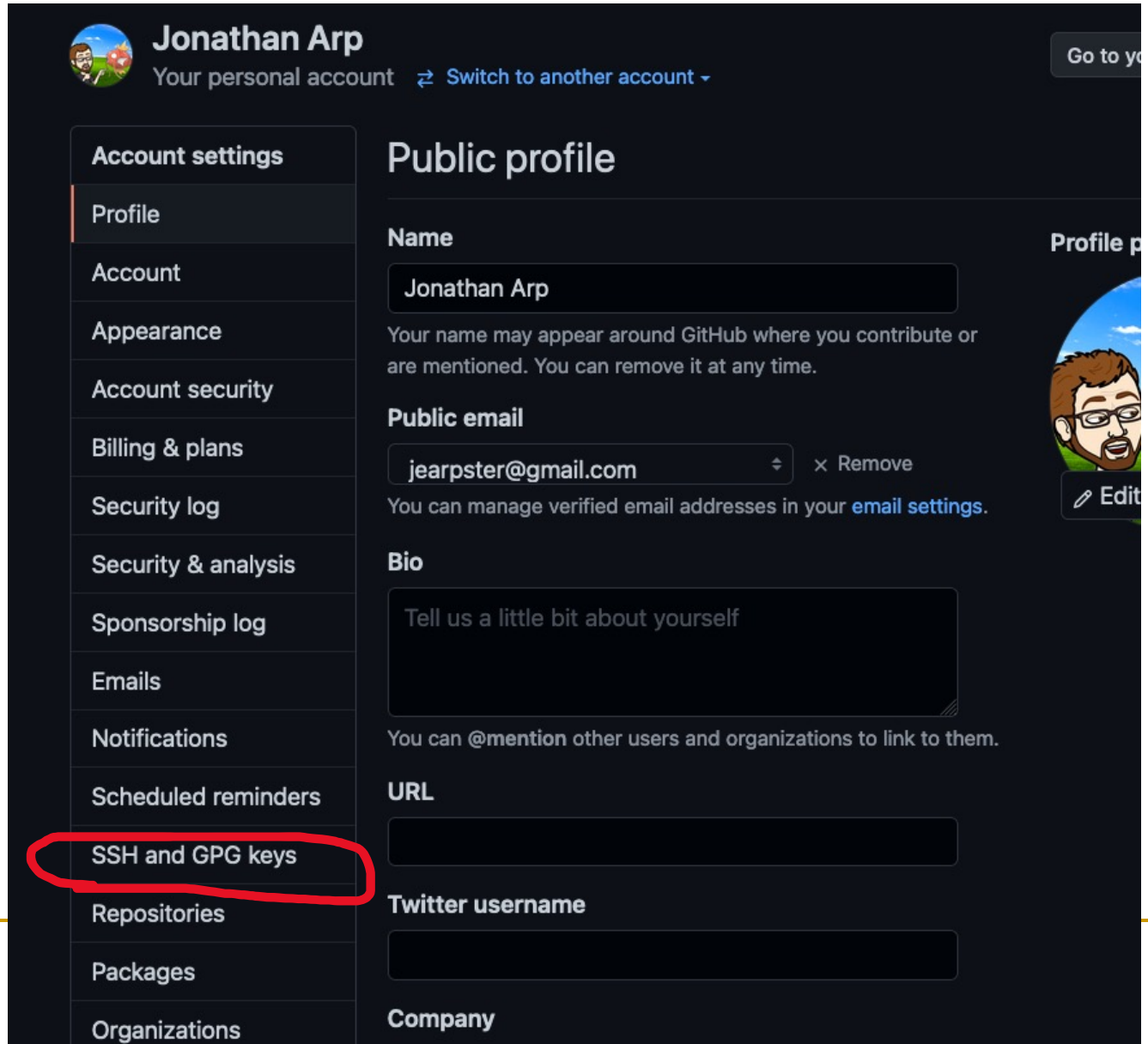
Using Key Pairs with GitHub



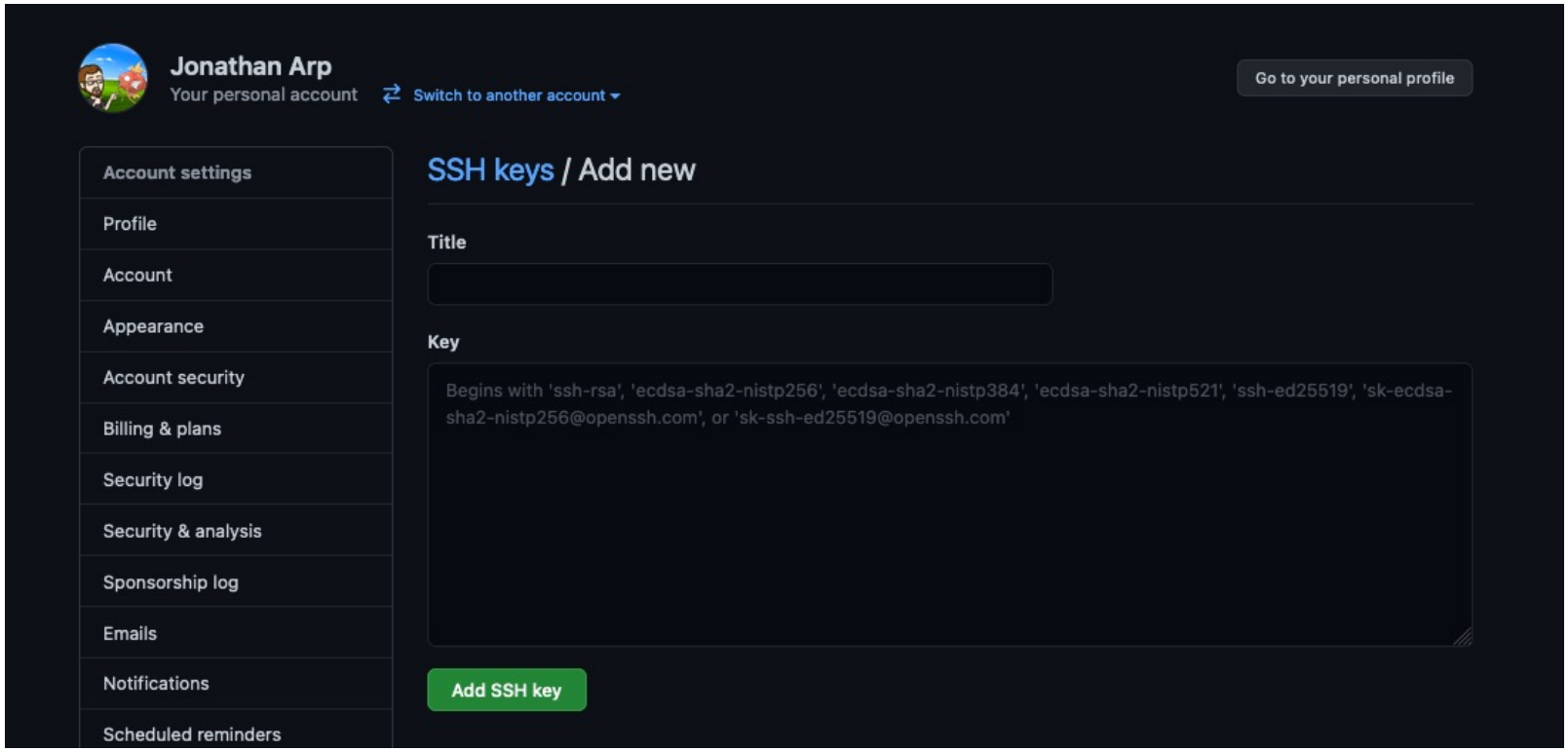
Using Key Pairs with GitHub



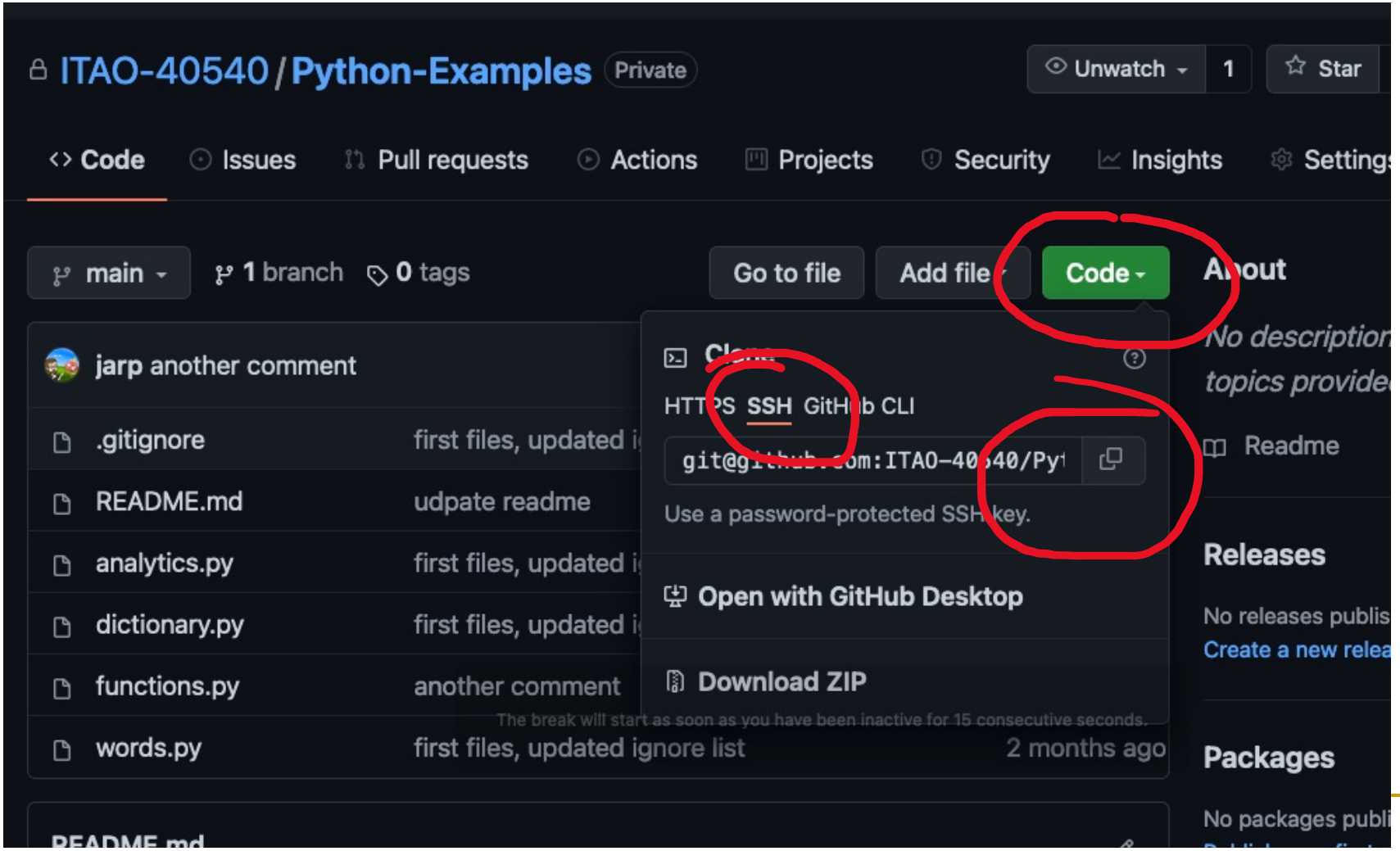
Using Key Pairs with GitHub



Using Key Pairs with GitHub



Cloning a Repo from GitHub



Getting Started

- `cd ~/`
- `mkdir software_engineering`
- `cd software_engineering`
- `git clone git@github.com:ITAO-40540/Python-Examples.git`
- `cd Python-Examples`

https://github.com/ITAO-40540/Python-Examples

Coding Prompt: Create Coding Journal Repo

- Brush up on your basic python skills
  - Lists, looping, if/else, dictionaries
- Do it in context of git and source control
  - Add files
  - Commit files
  - Create a branch
    - Add and commit files
  - Merge branch
  - Push to remote