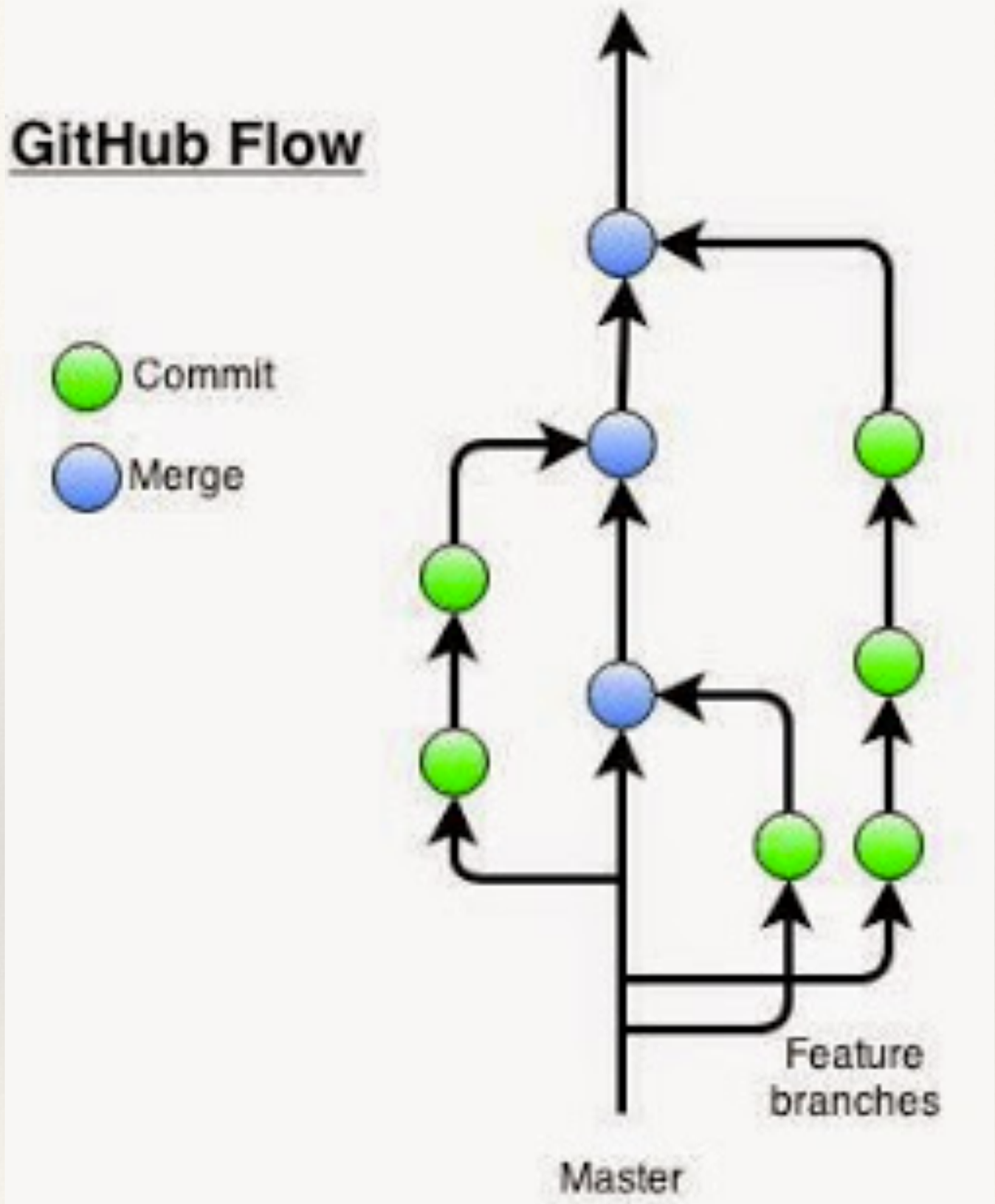
# Why use version control?

- Keep track of your own change to code
- Efficient updating, error tracking
- Multiple people working on a project
- User A makes changes to a particular part of the project
- User B also makes changes to the same part of the project
- Git allows both user A and user B to upload their revisions without them overwriting one another
- Both revisions can be merged together without losing work from either

# If you "muck" things up and want to go back

- before you commit: git revert (in Rstudio) or
- after you commit
  - find where you want to go back: in shell
    - git log
    - git log —reverse
    - notice the number
    - git reset #*commit* # leaves changes as "staged" but not 'committed'
    - git reset #*commit* —hard #gets rid of all changes

# If you "muck" things up and want to go back

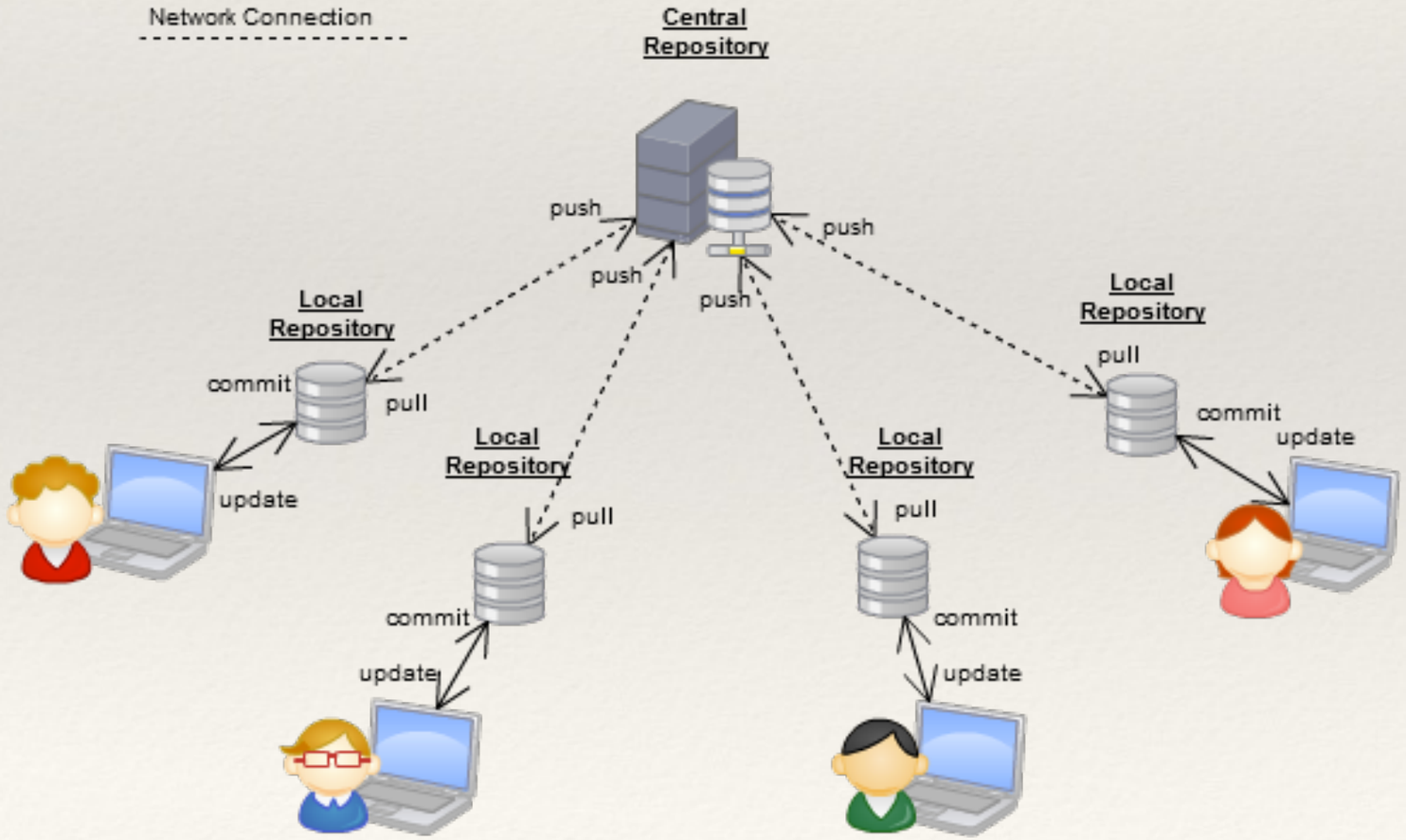
- note that if you don't use —hard on reset, you may have to deal with any conflicts
- a conflict occurs with git doesn't know what to do, it will generally outline the options for you by marking the difference using "*****" or "#####"

# If you “muck” things up

- Order of ‘fixing’

1. go back and look at the old version - see what is different
   - in shell - use git log to find the version number,
     - git log to find the version number
     - git diff version# to see what has changed
2. git revert
   - and resolve any conflict
3. git revert —hard
   - you really messed up and just want to go back

# git and github work together

# Linking Git to GitHub

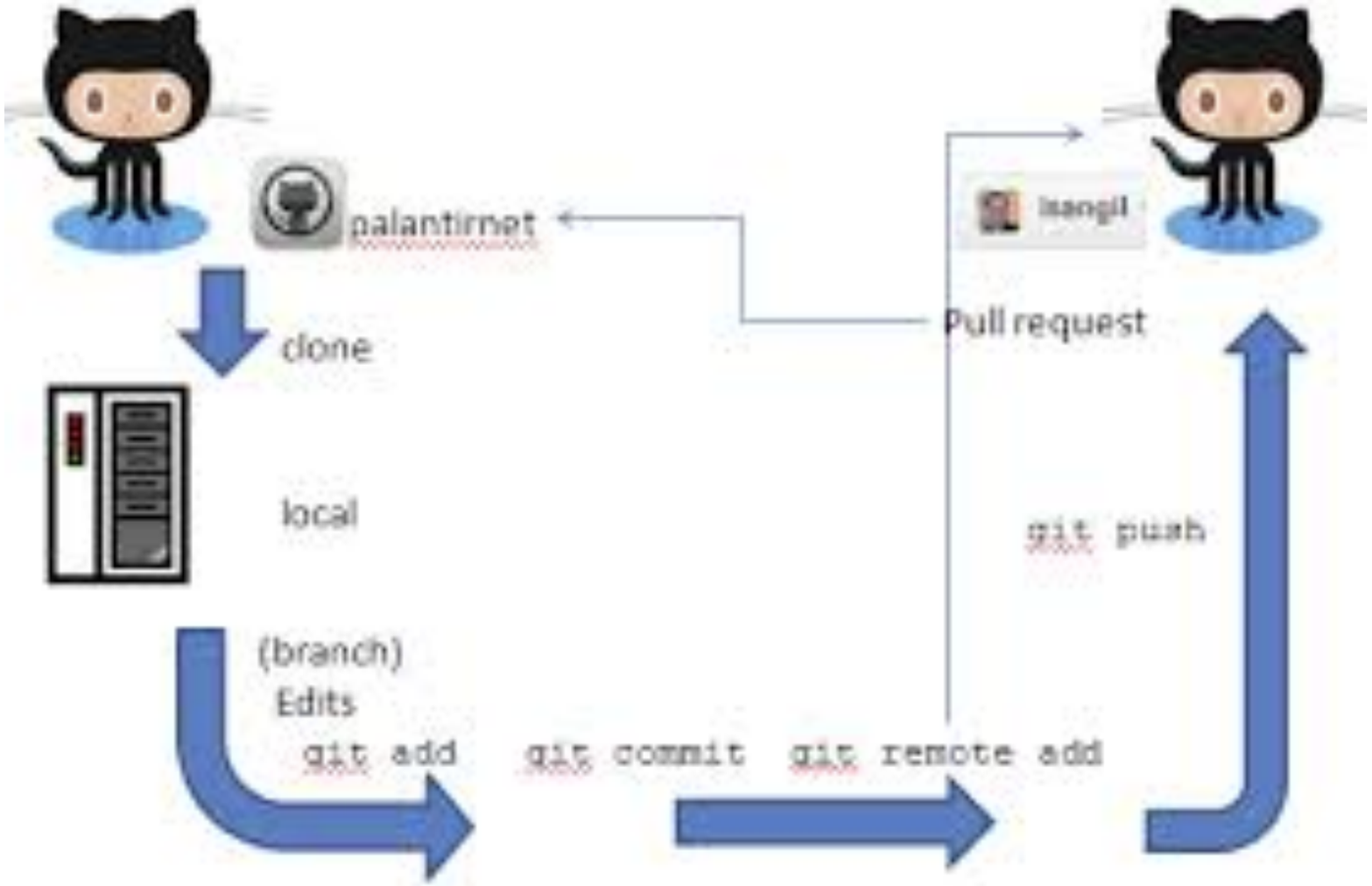
- LOCAL
- Design/Revise your branch
- Test
- Commit to your branch
- Merge your branch with master (or other main branch)
- LINK TO SHARED GIT REPOSITORY
    - Push - add your updates to remote repository
    - Pull - gets other peoples updates to your local repository

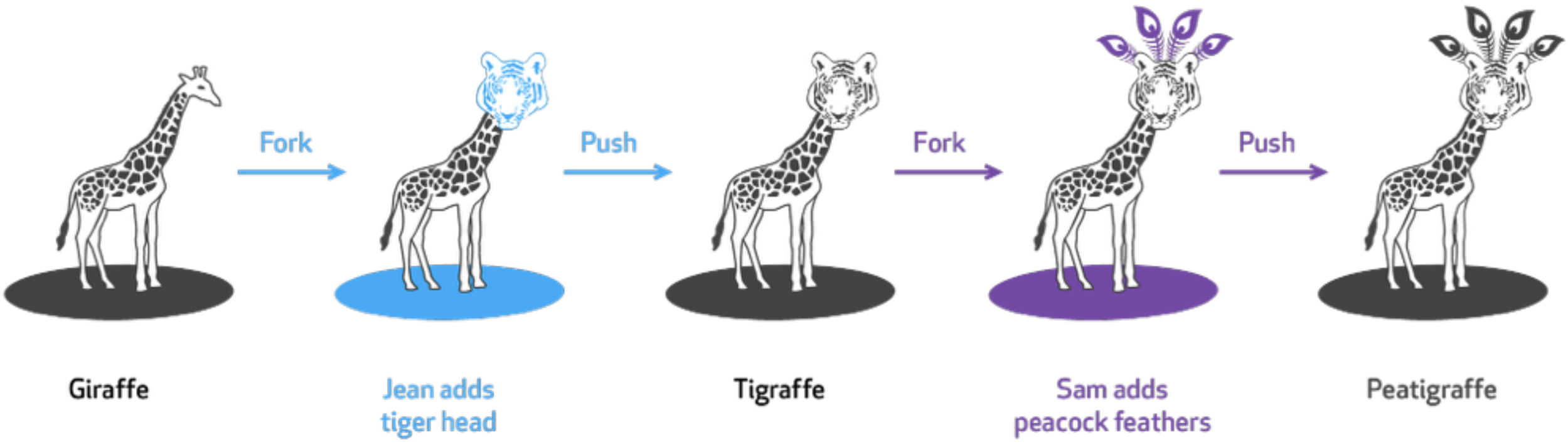
# Linking Git to GitHub

- When you initially create your project in Rstudio, include the link to the github repo
  - (http://github.com/…) (you can always find this on the github website)
- If you've been working with a local repository and now want to link your project to a github repository
  - In shell on Rstudio
    - git remote add origin *git@github.com:username/reponame.git*
    - git push -u origin *branchname*
    - usually easiest to copy and past the name from github website
- you may also need to set your ssh key if you have not done this
    - in Rstudio, click "preferences" and "view public key" (or create if needed)
    - in Github, "edit user profile" and "SSH keys" and add key

# Linking Git to GitHub

- Use git push origin *branchname* to send your changes to the repository
- Use git pull to bring down changes from the repository
- shell commands push and pull are also available in the git window


palantirnet
isangil
clone
Pull request
local
git push
(branch)
Edits
git add
git commit
git remote add


Fork
Push
Fork
Push
Giraffe
Jean adds
tiger head
Tigraffe
Sam adds
peacock feathers
Peatigraffe

# Linking Git to GitHub

- If you want to work on a project that is already under github (e.g. start working on a project created by someone else)
- When you open a new project,
  - select “Version Control” option
  - chose Git and add the Repository URL: (from the github website), and use the name of the R project from the existing repo