Introduction to Git

Jeremy Espino MD MS
Outline

- What is Git?
- Why is everyone using Git?
- Git concepts
- Git commands
What is Git?

- A distributed revision control system
- Designed and developed by Linus Torvalds
- British English slang meaning "unpleasant person"
Why is everyone using Git?

Branching  Workflows
Local
Fast
Distributed
Small
Staging Area

Popularity - Github, Bitbucket
Git concepts
# Common Git Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>add</td>
<td>Add file contents to the index</td>
</tr>
<tr>
<td>bisect</td>
<td>Find by binary search the change that introduced a bug</td>
</tr>
<tr>
<td>branch</td>
<td>List, create, or delete branches</td>
</tr>
<tr>
<td>checkout</td>
<td>Checkout a branch or paths to the working tree</td>
</tr>
<tr>
<td>clone</td>
<td>Clone a repository into a new directory</td>
</tr>
<tr>
<td>commit</td>
<td>Record changes to the repository</td>
</tr>
<tr>
<td>diff</td>
<td>Show changes between commits, commit and working tree, etc</td>
</tr>
<tr>
<td>fetch</td>
<td>Download objects and refs from another repository</td>
</tr>
<tr>
<td>grep</td>
<td>Print lines matching a pattern</td>
</tr>
<tr>
<td>init</td>
<td>Create an empty Git repository or reinitialize an existing one</td>
</tr>
<tr>
<td>log</td>
<td>Show commit logs</td>
</tr>
<tr>
<td>merge</td>
<td>Join two or more development histories together</td>
</tr>
<tr>
<td>mv</td>
<td>Move or rename a file, a directory, or a symlink</td>
</tr>
<tr>
<td>pull</td>
<td>Fetch from and integrate with another repository or a local branch</td>
</tr>
<tr>
<td>push</td>
<td>Update remote refs along with associated objects</td>
</tr>
<tr>
<td>rebase</td>
<td>Forward-port local commits to the updated upstream head</td>
</tr>
<tr>
<td>reset</td>
<td>Reset current HEAD to the specified state</td>
</tr>
<tr>
<td>rm</td>
<td>Remove files from the working tree and from the index</td>
</tr>
<tr>
<td>show</td>
<td>Show various types of objects</td>
</tr>
<tr>
<td>status</td>
<td>Show the working tree status</td>
</tr>
<tr>
<td>tag</td>
<td>Create, list, delete or verify a tag object signed with GPG</td>
</tr>
</tbody>
</table>

hey why are you trying to read this small text?
Common Git Commands We’ll Cover

- add: Add file contents to the index
- branch: List, create, or delete branches
- checkout: Checkout a branch or paths to the working tree
- clone: Clone a repository into a new directory
- commit: Record changes to the repository
- fetch: Download objects and refs from another repository
- push: Update remote refs along with associated objects
- status: Show the working tree status
Hands on with Gitlab
Using Git in Agile development

- Tasks are defined units of work - a branch
- Peer Code Review - merge (pull) request
- Frequent Releases - tagging
Feature Branch Workflow

- Tom and Jerry are coding partners
- Tom implements new feature as a branch
- Tom makes a pull request to Jerry
- Jerry reviews Tom’s code
- After review, Jerry merges Tom’s code to master
Hands on exercise

- Grab the cheat sheet
- Obtain a copy of a repository
- Create a branch and check it out
- Do some programming on your branch
- Push the branch to the server
- Make a merge request
- Respond to merge requests
- Checkout the project master
Steps in the exercise conventions

command1 to type in <variable>
command2 to type in <variable>
command3 to type in <variable>

[username@olympus ~]$ command1 to type in <variable>
result1
[username@olympus ~]$ command2 to type in <variable>
result2
[username@olympus ~]$ command3 to type in <variable>
result3
Obtain a copy of the repository

```
git clone https://<git username>@git.isg.pitt.edu/mission/gitlab-tutorial.git

[jespino@olympus ~]$ git clone https://juest4@git.isg.pitt.edu/mission/gitlab-tutorial.git
Initialized empty Git repository in /home/jespino/gitlab-tutorial/.git/
Password:
remote: Counting objects: 9, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 9 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (9/9), done.
```
Create a branch

git branch <branch name>

[jespino@olympus ~]$ cd gitlab-tutorial/
[jespino@olympus gitlab-tutorial]$ git branch helloToJeremy
[jespino@olympus gitlab-tutorial]$ git checkout helloToJeremy
Switched to branch 'helloToJeremy'
Do some “programming”

echo "<your name>" > <your name>.txt

git add <your name>.txt

git commit -m "adding the <your name> file"

[jespino@olympus gitlab-tutorial]$ echo "Jeremy" > jeremy.txt
[jespino@olympus gitlab-tutorial]$ git add jeremy.txt
[jespino@olympus gitlab-tutorial]$ git commit -m "adding the jeremy file"

[helloToJeremy c769076] adding the jeremy file
1 files changed, 1 insertions(+), 0 deletions(-)
create mode 100644 jeremy.txt
Upload your branch to Gitlab

`git push origin <branch name>`

[jespino@olympus gitlab-tutorial]$ git push origin helloToJeremy
Password:
Counting objects: 4, done.
Delta compression using up to 64 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 315 bytes, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://juest4@git.isg.pitt.edu/mission/gitlab-tutorial.git
 * [new branch] helloToJeremy -> helloToJeremy
Make a merge request

- [https://git.isg.pitt.edu/mission/gitlab-tutorial/merge_requests](https://git.isg.pitt.edu/mission/gitlab-tutorial/merge_requests)
- Click on “+New Merge Request”
- In From field, select your branch
- In To field, select master
- Click Submit
- Assign the request to your co programmer
- Your co-programmer will get an email notification saying a merge request is waiting for them
Respond to merge requests

- [https://git.isg.pitt.edu/mission/gitlab-tutorial/merge_requests](https://git.isg.pitt.edu/mission/gitlab-tutorial/merge_requests)
- Browse the code and make comments
- Optionally fetch the branch, check it out and run it in your working directory
- Accept the merge when you are satisfied
Checkout the project master

git checkout master

git pull origin master

./run.sh

[jespino@olympus gitlab-tutorial]$ git checkout master
Switched to branch 'master'

[jespino@olympus gitlab-tutorial]$ ls
MISSION.txt README run.sh

[jespino@olympus gitlab-tutorial]$ git pull origin master
...

[jespino@olympus gitlab-tutorial]$ ls
jeremy.txt john.txt MISSION.txt README run.sh

[jespino@olympus gitlab-tutorial]$ ./run.sh
Hello, Jeremy!
Hello, John!
Hello, MISSION!
Other great Git resources

Tutorials
https://www.codeschool.com/courses/git-real
https://www.atlassian.com/git/tutorials/

Cheatsheet
If you have extra time...
Do work on someone else’s branch

#Fetch all the branches from remote
git fetch --all

#Show a list of all branches
git branch -a

#Create a local branch from remote
git branch --track <branchName> remotes/origin/<branchName>

#Check out the branch
git checkout <branchName>