Lab *02 – Versioning Your Work*

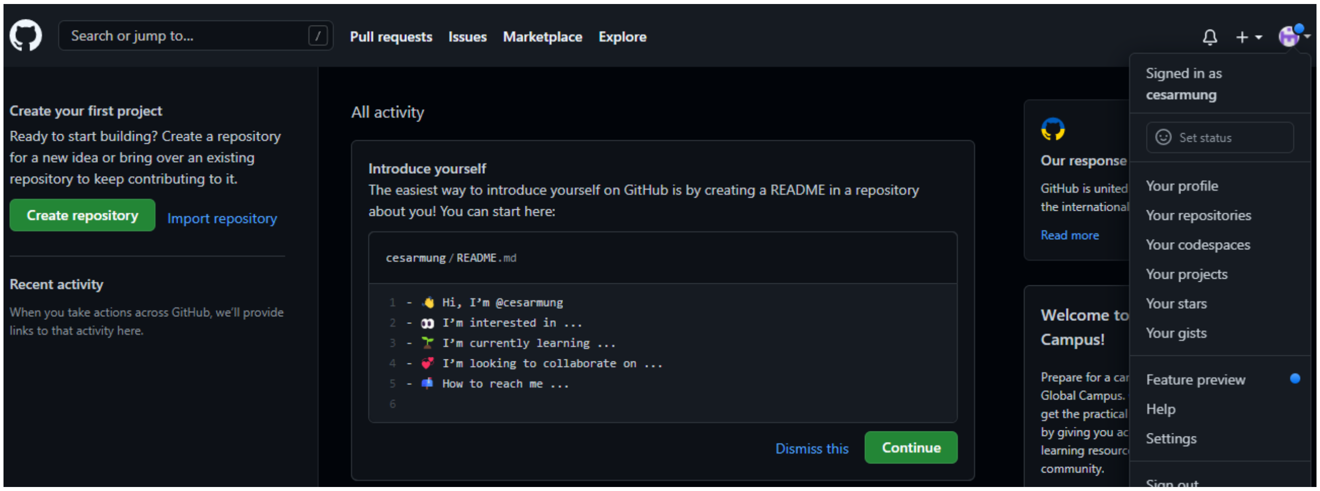
Author: Cesar Munguia
Course Section: IS-1003-002
Date: 03/12/2022

Introduction

The objectives of this lab are to set up a GitHub account, set up git on a local machine, and sync a local git repository with my online GitHub account. I should be able to experiment with the GitHub environment and experiment further. This will allow me to have a broader knowledge when working with virtual local machines.

Process

1. In phase 1 I created a GitHub account with my information and have not really started anything. My screen looks like the picture below. Something to mention is that I tried to get the student access version but I may need to wait up to 12 days for them (GitHub) to confirm my information about the school. So, for now I've got the free version in which I've used both my personal and school email.



Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

All activity

Introduce yourself

The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

```
cesarmung / README.md

- 👋 Hi, I’m @cesarmung
- 👀 I’m interested in ...
- 🌱 I’m currently learning ...
- 💞️ I’m looking to collaborate on ...
- 📫 How to reach me ...
```

Our response

GitHub is united
the international

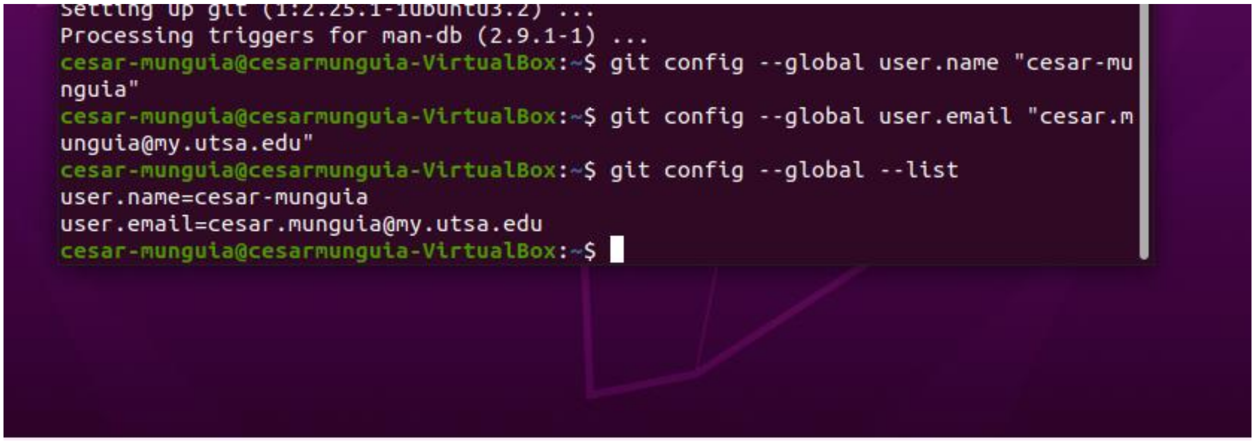
Welcome to
Campus!

Prepare for a car
Global Campus.
get the practical
by giving you ac
learning resourc
community.

2. In phase 2 I will be defining the six most common used git commands in my own words.
   - Push – it allows you to move a branch to another repository, opposite of fetching
   - Pull – downloads a branch from a remote repository and then it places it into the current branch
   - Fork –
   - Commit – takes the snapshot and commits it to the project history
   - Clone – what the word literally means, to make a copy of an existing repository
   - Branch – it allows you to make isolated development environment within a repository

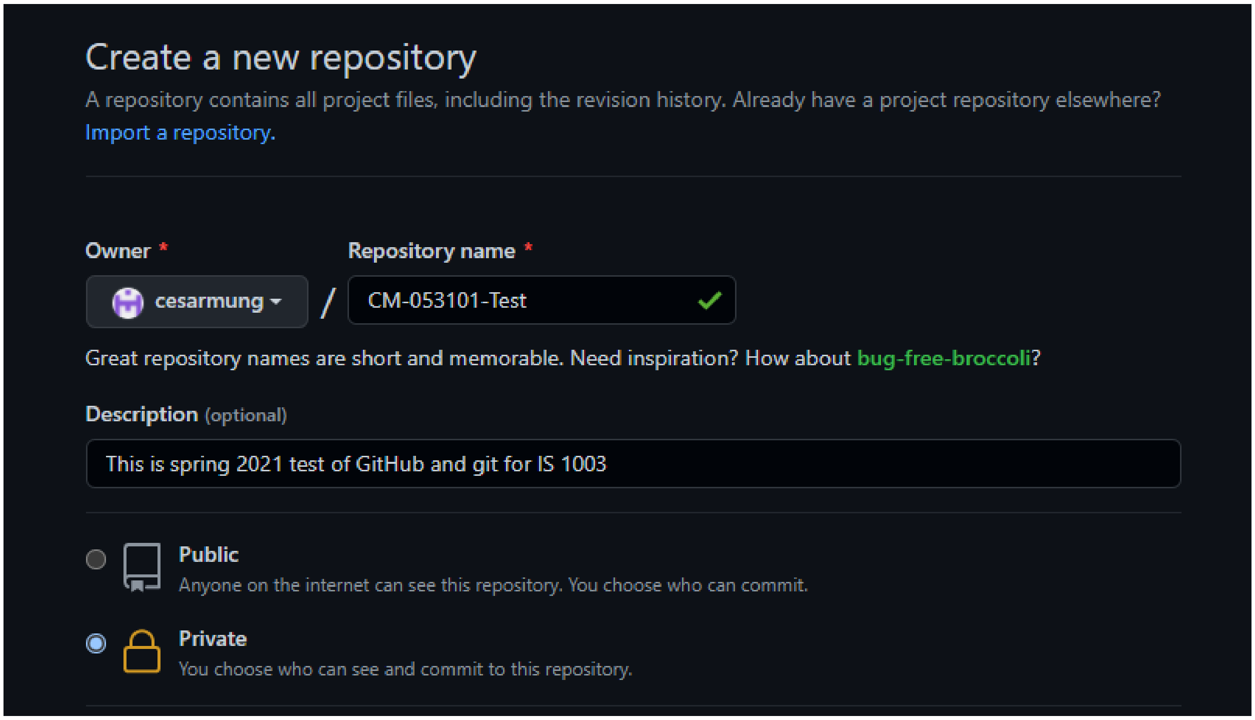
   I learned these definitions by looking them up on search machine with the website being https://www.atlassian.com/git/glossary

3. In phase 3 I started the download and installation of git on the command line of my virtual machine. I followed the steps from the lab instructions and successfully created a username and inserted my school email. Here is a proof of what I've just stated.



```
Setting up git (1:2.25.1-1ubuntu3.2) ...
Processing triggers for man-db (2.9.1-1) ...
cesar-munguia@cesarmunguia-VirtualBox:~$ git config --global user.name "cesar-mu
nguia"
cesar-munguia@cesarmunguia-VirtualBox:~$ git config --global user.email "cesar.m
unguia@my.utsa.edu"
cesar-munguia@cesarmunguia-VirtualBox:~$ git config --global --list
user.name=cesar-munguia
user.email=cesar.munguia@my.utsa.edu
cesar-munguia@cesarmunguia-VirtualBox:~$ 
```

4. In phase 4, I created a new repository for this assignment and since our professor stated not to name it like hers, I put mine as "CM-053101-Test" which is my two initials and the date I was born. I put the same description as her and made the repository private.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner * cesarmung / Repository name * CM-053101-Test ✓

Great repository names are short and memorable. Need inspiration? How about bug-free-broccoli?

Description (optional)

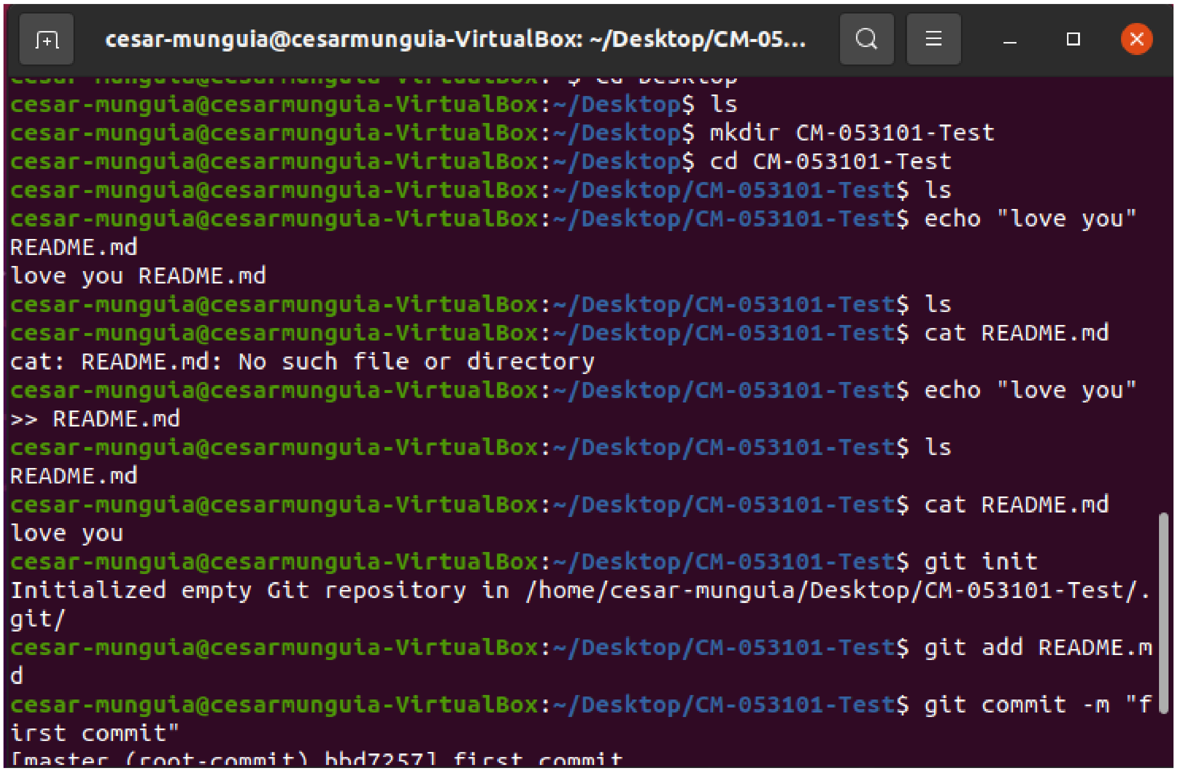
This is spring 2021 test of GitHub and git for IS 1003

○ Public
Anyone on the internet can see this repository. You choose who can commit.

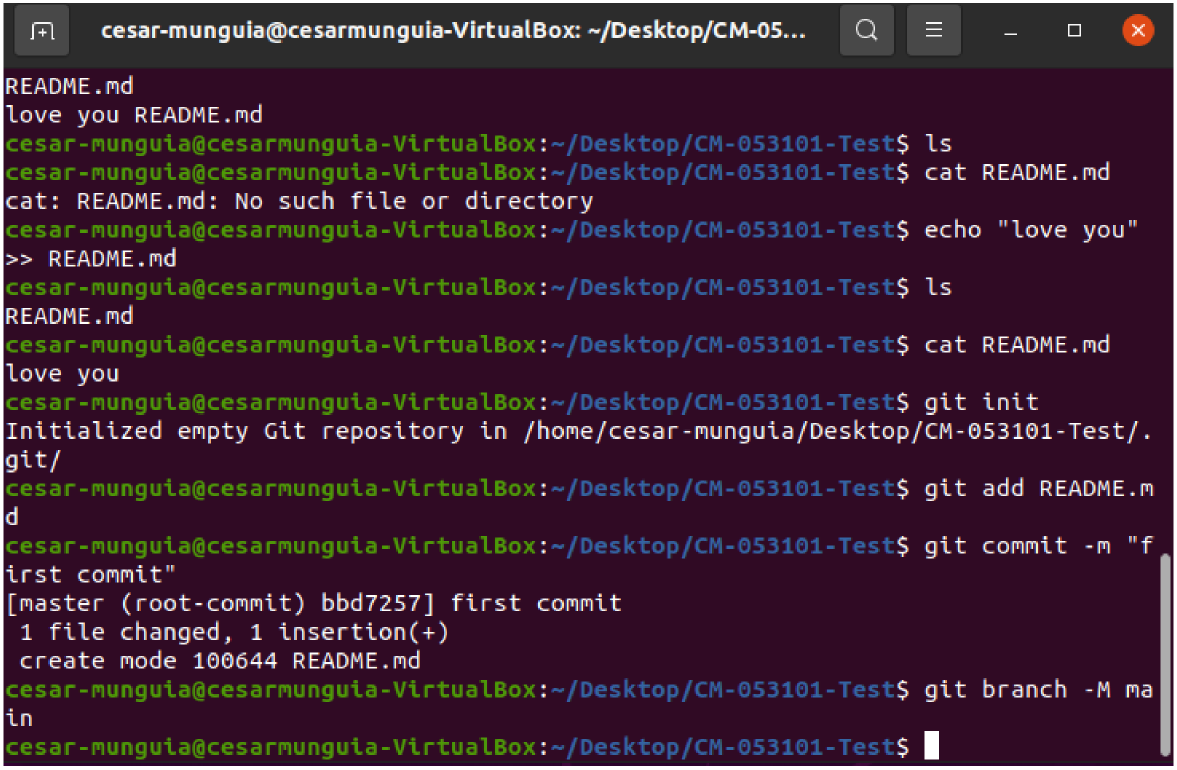
● Private
You choose who can see and commit to this repository.

5. In phase 5 I committed my first git repo. The way I have done it is by using several command lines that I'm already used to and git commands. First, I created a new directory with the command mkdir and named it CM-053101-Test, the same name as my new repo on GitHub. Then, I "entered" the directory by using command line cd CM-05… and created a README.md file in the folder. Then I redirected my echo message "love you" to the file README.md. After that, I initialized the local git repo I have just created by using command git init. Then I added the file using git add README.md. Finally, I committed all the files in the staging area to the repo

to create an actual version of the repo using command line git commit -m “first commit”. I named the repo’s main branch as “main” using command line git branch -M main.



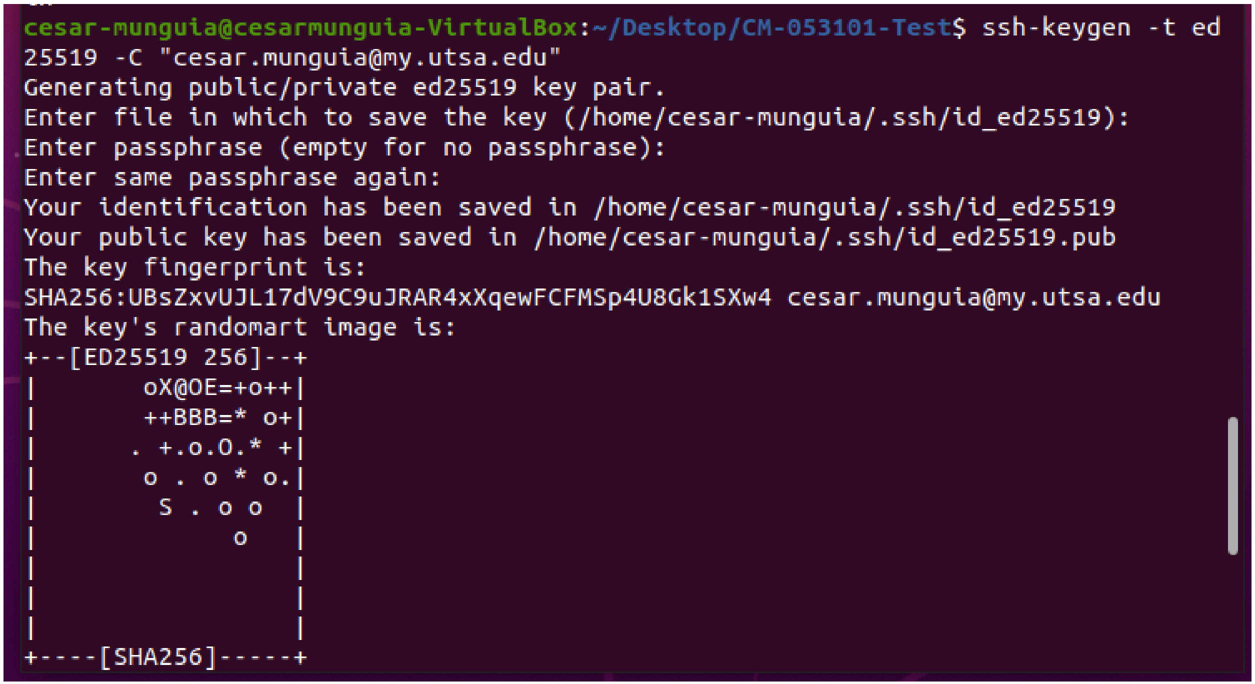
```
cesar-munguia@cesarmunguia-VirtualBox:~$ cd Desktop
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop$ ls
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop$ mkdir CM-053101-Test
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop$ cd CM-053101-Test
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ ls
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ echo "love you"
README.md
love you README.md
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ ls
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ cat README.md
cat: README.md: No such file or directory
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ echo "love you"
>> README.md
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ ls
README.md
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ cat README.md
love you
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git init
Initialized empty Git repository in /home/cesar-munguia/Desktop/CM-053101-Test/.
git/
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git add README.m
d
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git commit -m "f
irst commit"
[master (root-commit) bbd7257] first commit
```



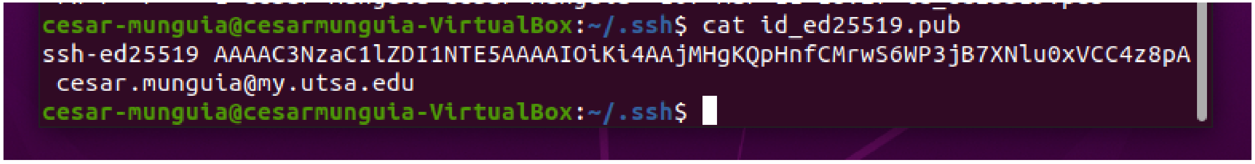
```
README.md
love you README.md
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ ls
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ cat README.md
cat: README.md: No such file or directory
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ echo "love you"
>> README.md
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ ls
README.md
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ cat README.md
love you
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git init
Initialized empty Git repository in /home/cesar-munguia/Desktop/CM-053101-Test/.
git/
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git add README.m
d
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git commit -m "f
irst commit"
[master (root-commit) bbd7257] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git branch -M ma
in
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ 
```

6. In phase 6 I had to establish a secure connection between this local repo and my remote repo on GitHub. To do this I decided to use the SSH key method because that’s the one shown to me in the lab instructions and thought it was the simplest one. I initialized the connection by using

command line ssh-keygen -t ed25519 -C “cesar.munguia@my.utsa.edu”. Then I copied the output given to me by the terminal and pasted in my GitHub account.

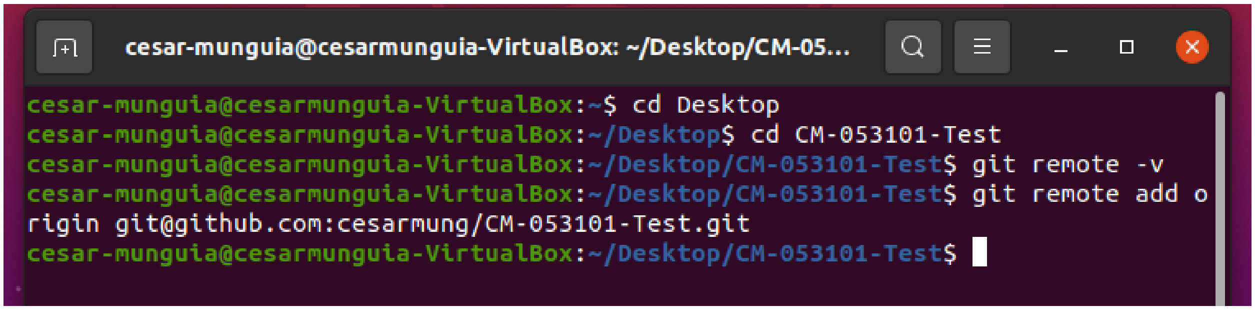


```
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ ssh-keygen -t ed
25519 -C "cesar.munguia@my.utsa.edu"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/home/cesar-munguia/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cesar-munguia/.ssh/id_ed25519
Your public key has been saved in /home/cesar-munguia/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:UBsZxvUJL17dV9C9uJRAR4xXqewFCFMSp4U8Gk1SXw4 cesar.munguia@my.utsa.edu
The key's randomart image is:
+--[ED25519 256]--+
|       oX@OE=+o++|
|       ++BBB=* o+|
|      . +.o.O.* +|
|       o . o * o.|
|        S . o o  |
|             o   |
|                 |
|                 |
|                 |
+----[SHA256]-----+
```



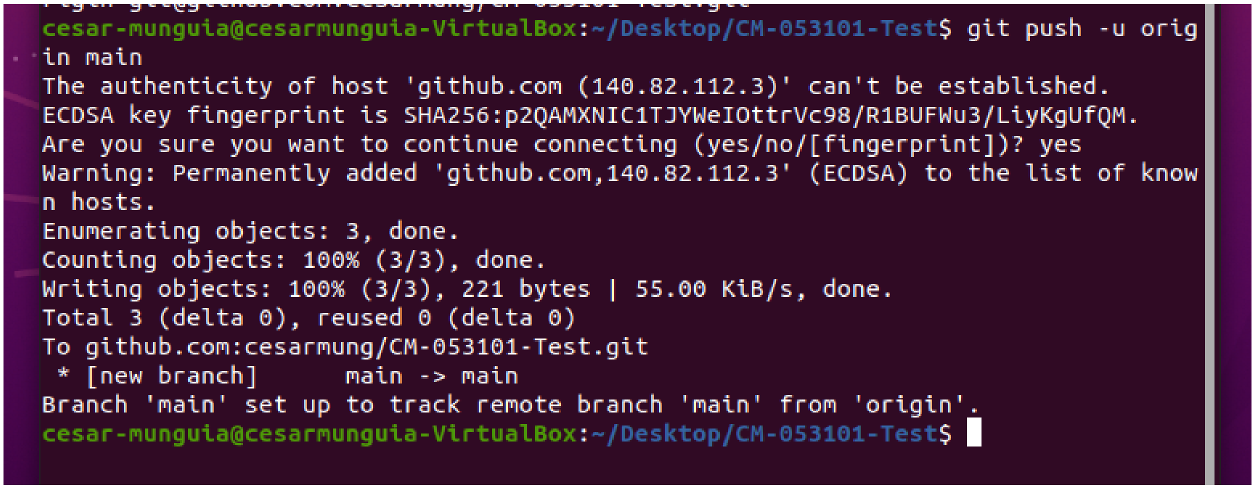
```
cesar-munguia@cesarmunguia-VirtualBox:~/.ssh$ cat id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOiKi4AAjMHgKQpHnfCMrwS6WP3jB7XNlu0xVCC4z8pA
 cesar.munguia@my.utsa.edu
cesar-munguia@cesarmunguia-VirtualBox:~/.ssh$
```

7. Here I used the cd command to get to the CM-053101-Test folder and synced my repos. I pasted the URL that was given to me on my account into the terminal. Below is a picture that describes what I’m explaining.



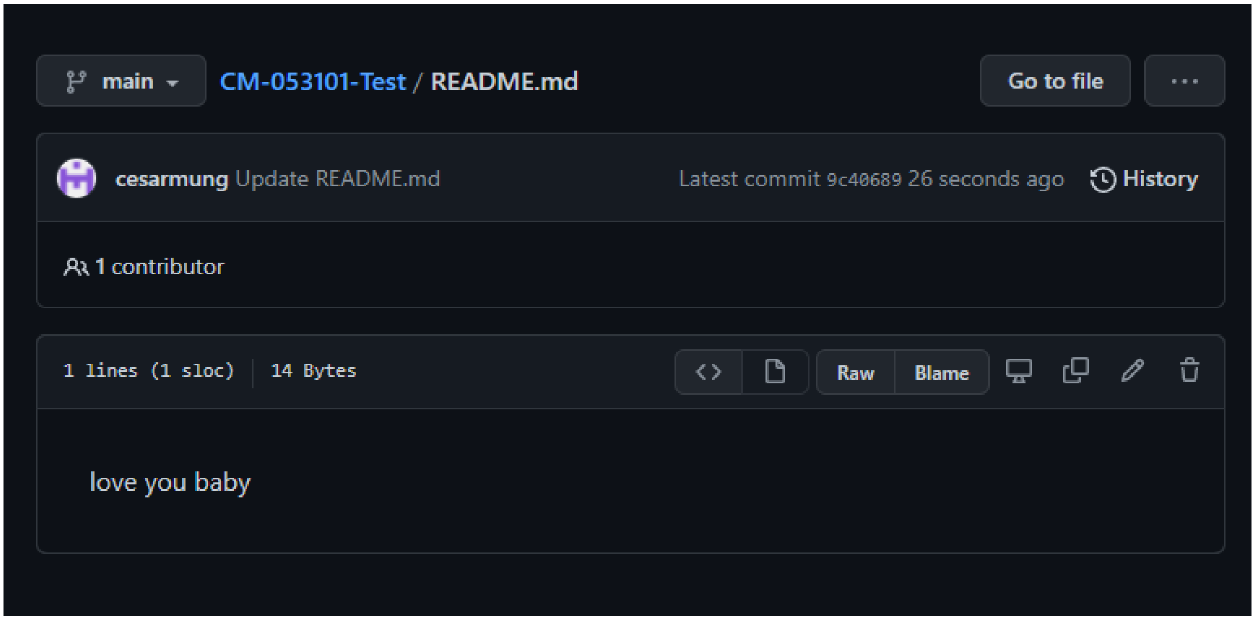
```
cesar-munguia@cesarmunguia-VirtualBox:~$ cd Desktop
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop$ cd CM-053101-Test
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git remote -v
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git remote add o
rigin git@github.com:cesarmung/CM-053101-Test.git
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$
```

8. Here I pushed changes on my local repo to my remote repo. I had a uncommon message shows to me that was telling me that “the authenticity of host ‘github.com … can’t be established” but I had the option of continuing with the connection and so I did so and it successfully worked. Here’s a picture that describes what I just stated.



```
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git push -u orig
in main
The authenticity of host 'github.com (140.82.112.3)' can't be established.
ECDSA key fingerprint is SHA256:p2QAMXNIC1TJYWeIOttrVc98/R1BUFWu3/LiyKgUfQM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com,140.82.112.3' (ECDSA) to the list of know
n hosts.
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 221 bytes | 55.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:cesarmung/CM-053101-Test.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ 
```

9. In this phase I made an edit on my remote repo (GitHub). The only thing that I added on the text was “baby”. I initially had “love you” before and now it’s “love you baby”.



CM-053101-Test / README.md

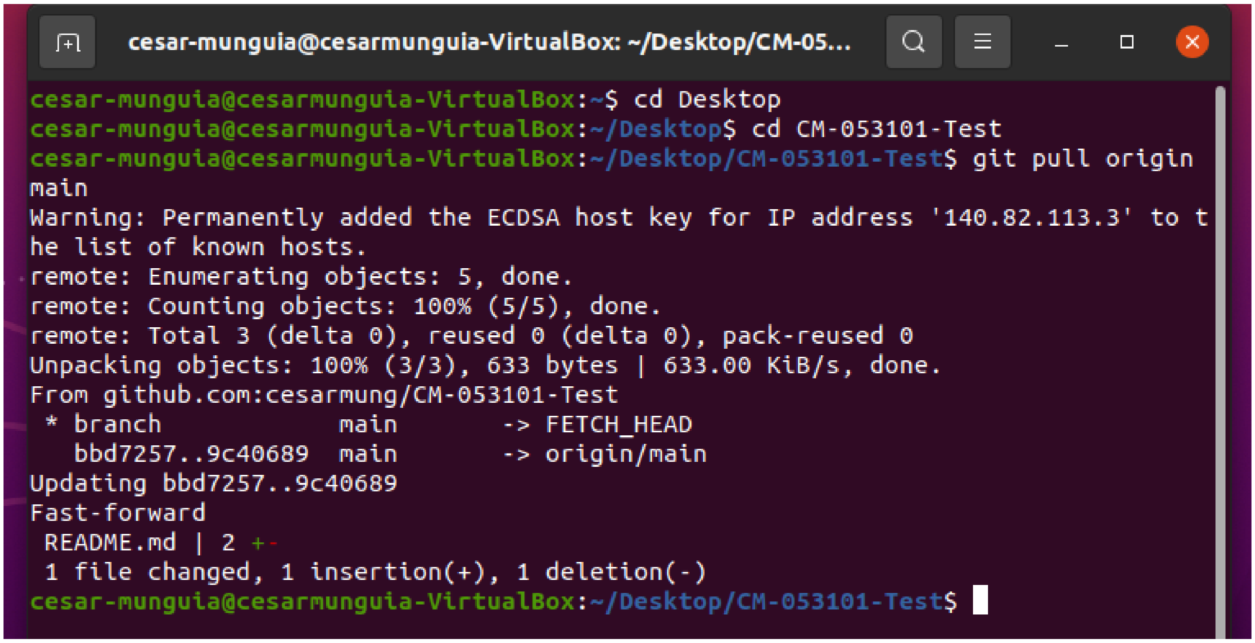
cesarmung Update README.md   Latest commit 9c40689 26 seconds ago

1 contributor

1 lines (1 sloc) 14 Bytes

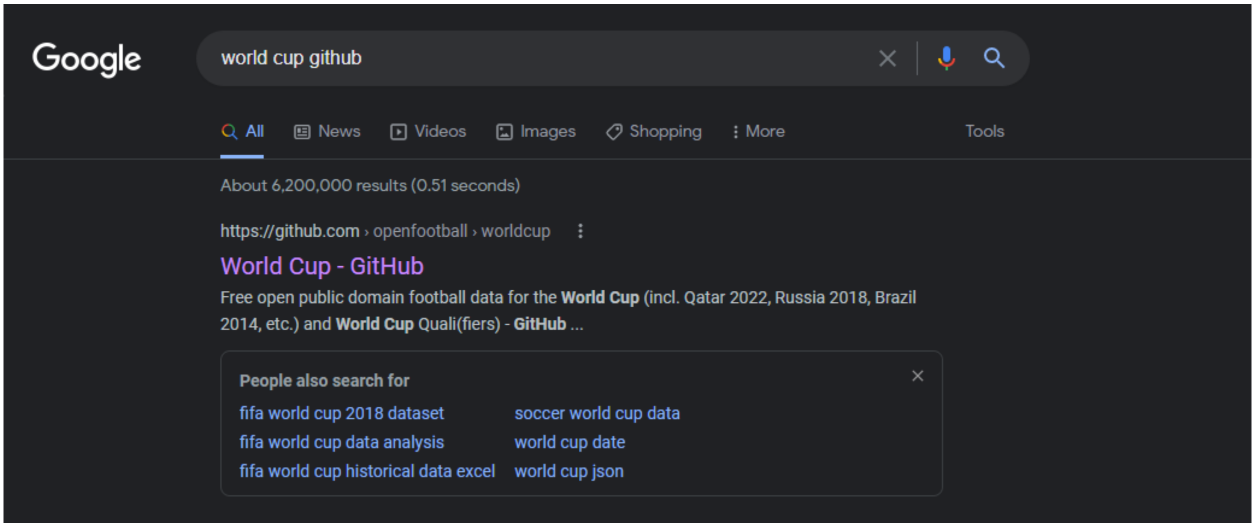
love you baby

10. In phase 10 I pulled changes from my remote repo to my local repo (GitHub). To do that I used the command line git pull origin main which successfully “pulled” the changes made.



```
cesar-munguia@cesarmunguia-VirtualBox:~$ cd Desktop
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop$ cd CM-053101-Test
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git pull origin
main
Warning: Permanently added the ECDSA host key for IP address '140.82.113.3' to t
he list of known hosts.
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 633 bytes | 633.00 KiB/s, done.
From github.com:cesarmung/CM-053101-Test
 * branch            main       -> FETCH_HEAD
   bbd7257..9c40689  main       -> origin/main
Updating bbd7257..9c40689
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ 
```

11. In phase 11 I was asked to choose a hobby-related topic to copy a repo but not propose changes. I chose the world cup topic because I am really passionate about soccer. I've played soccer since I was little and I still play but not at the professional level. The first picture below is an image of the google search I did.



About 6,200,000 results (0.51 seconds)

https://github.com › openfootball › worldcup

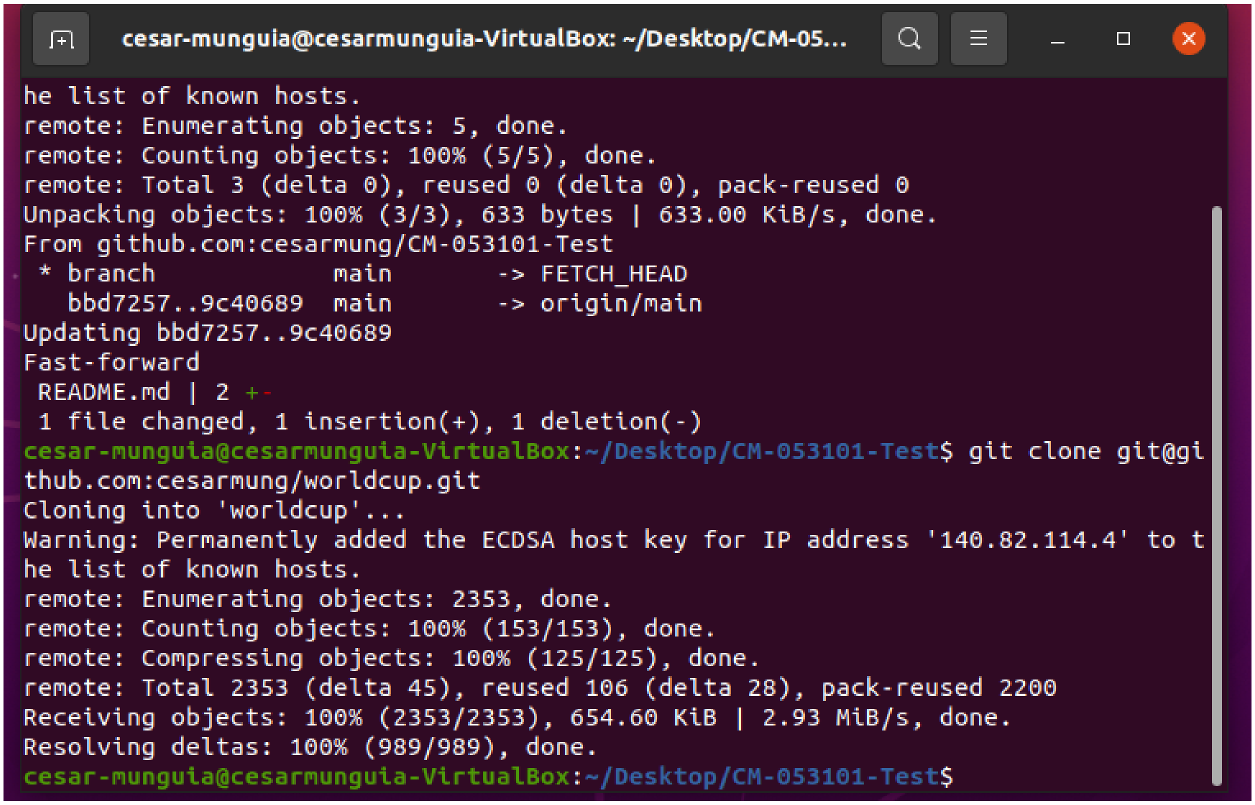
World Cup - GitHub

Free open public domain football data for the **World Cup** (incl. Qatar 2022, Russia 2018, Brazil 2014, etc.) and **World Cup** Quali(fiers) - **GitHub** ...

People also search for

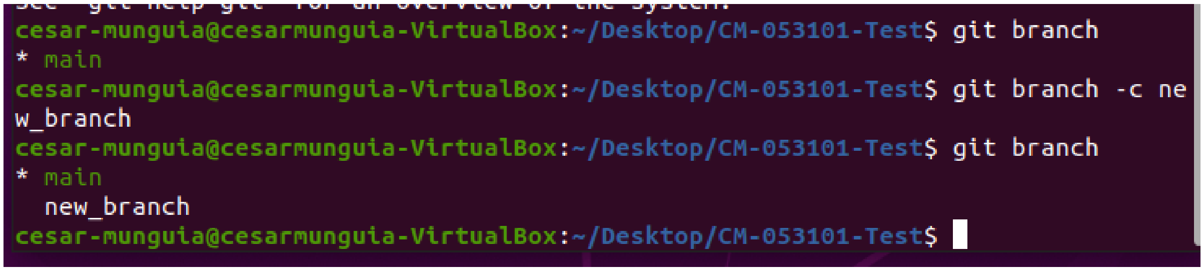
fifa world cup 2018 dataset | soccer world cup data
fifa world cup data analysis | world cup date
fifa world cup historical data excel | world cup json

Next, I cloned my new world cup repo to a local repository. I did such by typing the command line git clone git@github.com:cesarmung/worldcup.git. I was able to do it successfully and below is a picture that demonstrates so.



```
he list of known hosts.
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 633 bytes | 633.00 KiB/s, done.
From github.com:cesarmung/CM-053101-Test
 * branch            main       -> FETCH_HEAD
   bbd7257..9c40689  main       -> origin/main
Updating bbd7257..9c40689
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git clone git@gi
thub.com:cesarmung/worldcup.git
Cloning into 'worldcup'...
Warning: Permanently added the ECDSA host key for IP address '140.82.114.4' to t
he list of known hosts.
remote: Enumerating objects: 2353, done.
remote: Counting objects: 100% (153/153), done.
remote: Compressing objects: 100% (125/125), done.
remote: Total 2353 (delta 45), reused 106 (delta 28), pack-reused 2200
Receiving objects: 100% (2353/2353), 654.60 KiB | 2.93 MiB/s, done.
Resolving deltas: 100% (989/989), done.
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$
```

12. In this last phase of the lab, I had to create a new branch off from my repo that I've been working here. This was quite simple because I just had to type down a couple of commands and the task was done. I had to look up help on YouTube in which I searched for "how to create a branch in git". This video got my attention and helped complete the task https://www.youtube.com/watch?v=Lf3DYRvCPFo. The only thing I had to type down was git branch first to show that there were no current branches and only main was available. Then I typed git branch -c <branch.name>, doing such created a new branch for me. Finally, I again typed git branch to see the current branches and my new branch named "new_branch" was created. Below is a picture that shows what I explained.



```
See 'git help git' for an overview of the system.
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git branch
* main
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git branch -c ne
w_branch
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ git branch
* main
  new_branch
cesar-munguia@cesarmunguia-VirtualBox:~/Desktop/CM-053101-Test$ 
```

Limitations/Conclusion

I really had no limitations regarding this lab. I feel like I was able to complete all phases/tasks with no problem at all. The only thing that bothered me a little was the fact that my virtual machine was working very slow. I'm assuming it's because my computer doesn't hold enough RAM. This made time working on the lab 5x longer, it took me about 4-5 hours to complete this lab which I think made me a little frustrated at the time I was doing it. Some of the goals I think that were intended in the completion of this lab were to be able to feel familiar with all the terminal commands and in navigating it correctly and efficiently. I think these are essential and crucial topics and concepts that, for example, an IT professional needs to know. I feel like these techniques were very effective because they are simple and easy to execute. You just need to know the logic between everything. Other aspects I would like to explore is getting into the security aspect of the course, not just purely code.

References

Atlassian. (n.d.). *Basic git commands: Atlassian Git Tutorial*. Atlassian. Retrieved March 11, 2022, from https://www.atlassian.com/git/glossary

How to create a new branch on GitHub // commit ... - youtube. (n.d.). Retrieved March 12, 2022, from https://www.youtube.com/watch?v=Lf3DYRvCPFo

Collaboration

No collaboration.