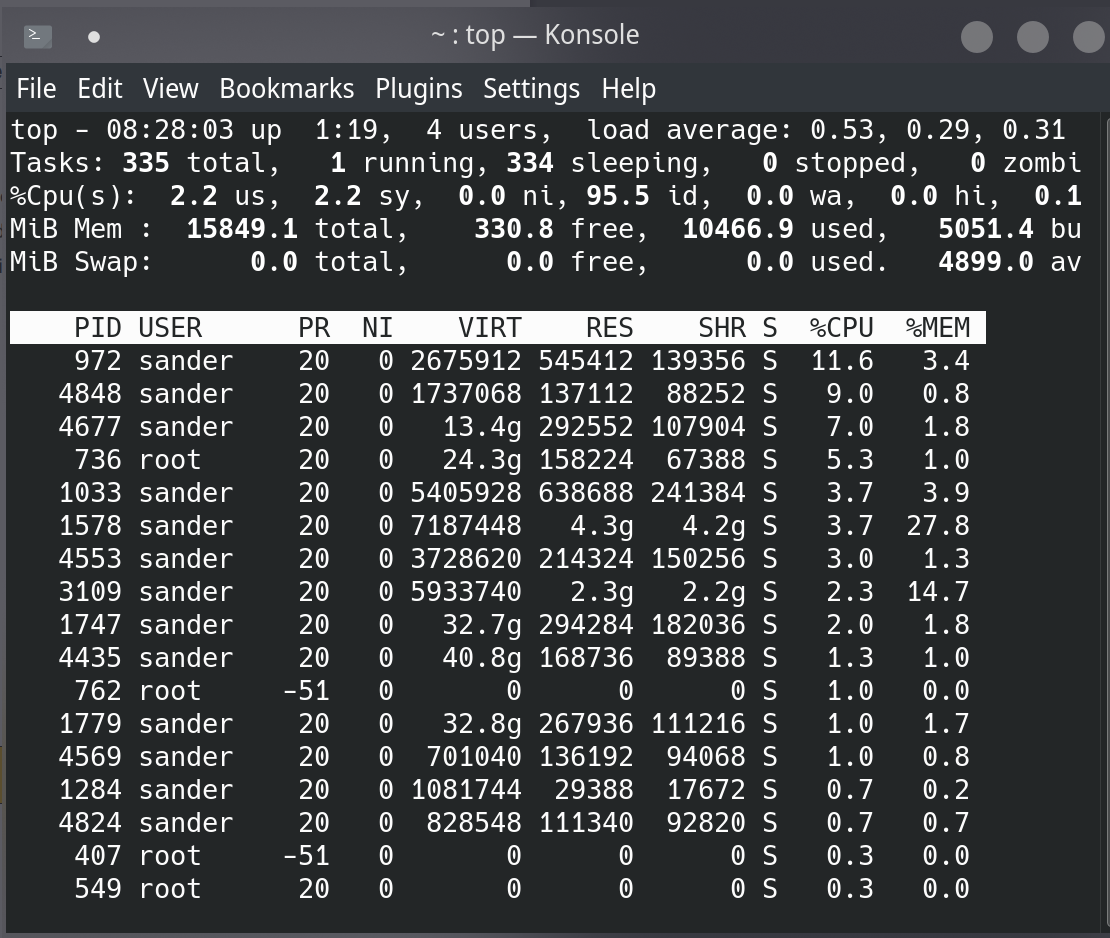
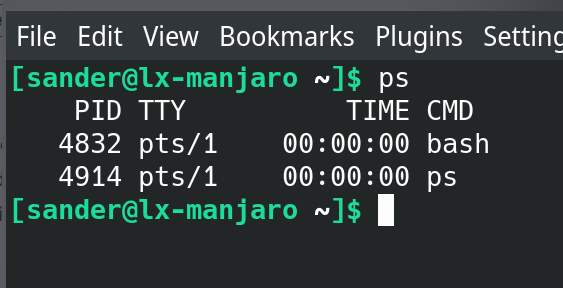
**Question 1:**

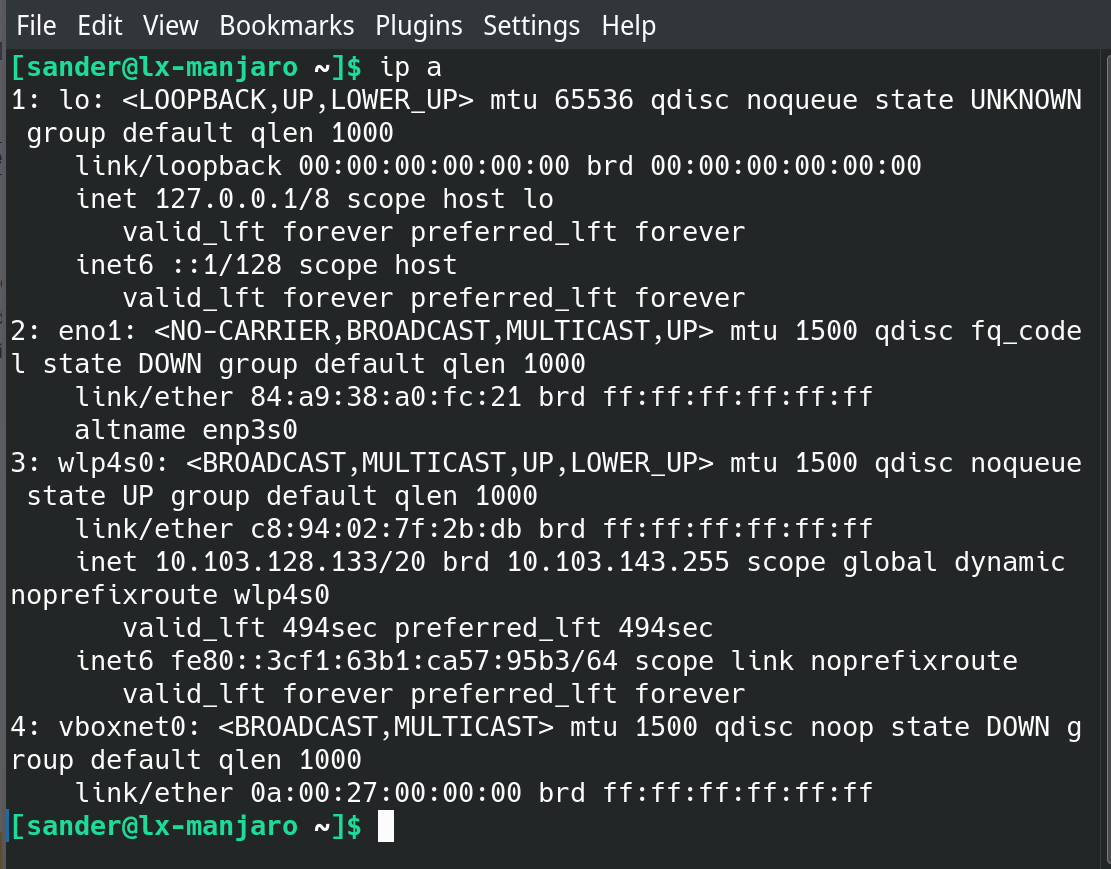
**Top**: is a command used to monitor the systems memory. It provides a list of all running processes and updates this periodically. By default it shows: Process Identification Number, User, Priority, CPU percentage, and Memory percentage. It can also identify the current up-time, amount of users and the average load.



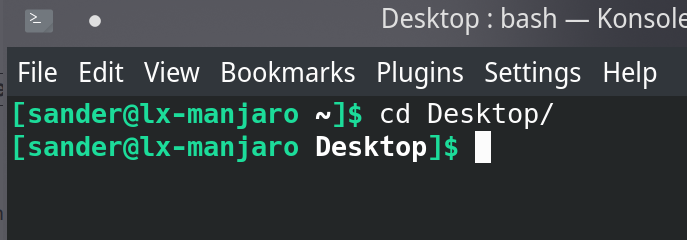
**PS**: is a utility used for viewing information about the current running processes on the system. It shows the running processes along with the up-time and Process Identification Number.



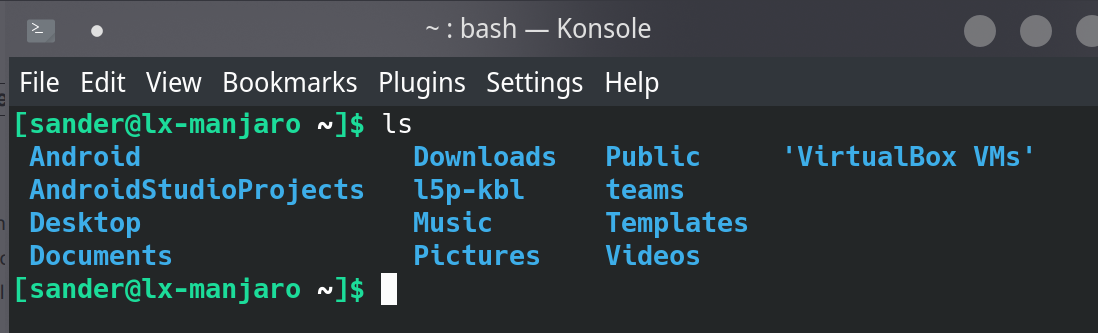
**IP A**: is a command that opens the Linux network tool, showing every interface on the system. It also shows extra information such as the interface’ IP-address, Sub net-mask, Up-link, and description/name.



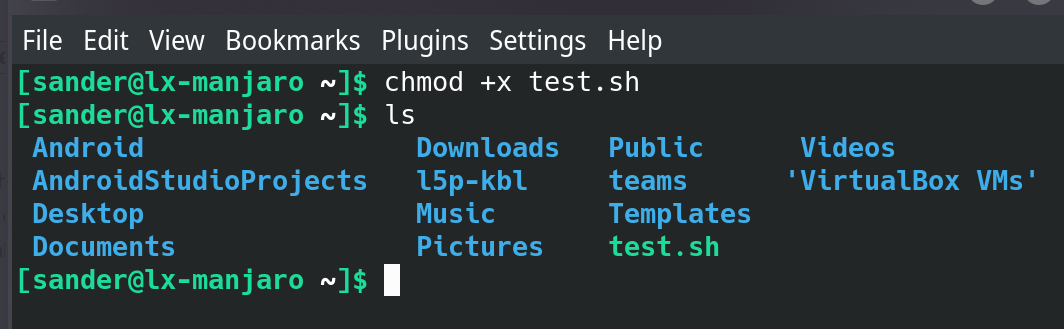
**CD**: is the command to change the current directory in Linux. It can be used to navigate around the file system while using the command-line interface. In the screen-shot below I use the command to navigate to my Desktop folder.



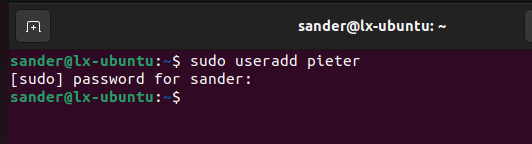
**LS**: is the command used to list all the files or folders in the current directory. I used this command inside of my home folder, which is why all my documents of my local machine are shown in the screen-shot below.



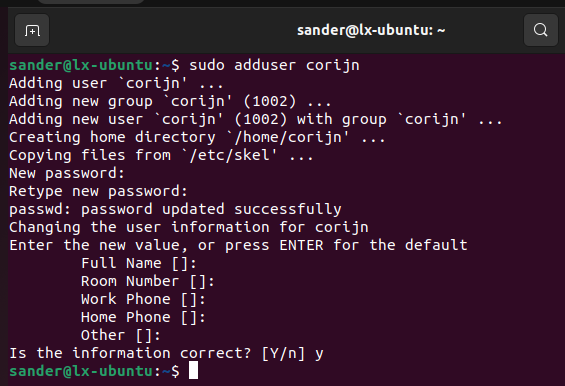
**CHMOD**: is a tool that users can use to change the permissions of a file or folder. Users can alter the permissions to set a file or folder to write, read or execute. In the demonstration below, I created a file, which I then permitted with CHMOD to be able to be executed.



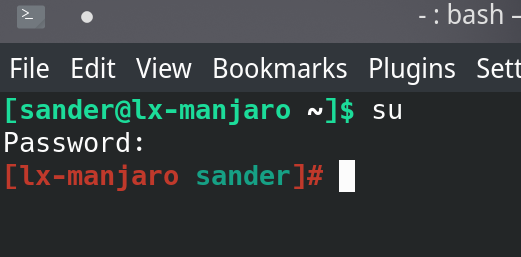
**USERADD**: creates a new user using only values specified on the command-line interface. When used with no extra options, it will create a user only. Home directory, default group and personal information must be set up manually using provided options in the help menu.



**ADDUSER**: this command essentially does the same as useradd. However, it automatically asks the creator of the new user to set up the password and personal information. In addition, it will also set the home directory and other user settings.



**SU**: this command is used to switch to another user. However, using this command without a user name you will automatically log-in as the superuser.



**Question 2:**

**1). Discuss the advantages and disadvantages of using a Version Control Too**

A version control system keeps track of every modification made to a file or group of files. By keeping track of all code alterations, version control has the advantage of helping a software development team manage source code. It shields the source code from accidental human mistakes or their effects. Furthermore, the development staff regularly writes new code. These modifications are noted for future use if it becomes necessary to revisit them to identify the root of a problem.

One drawback of utilizing a version control system is that it will merge any locally uploaded changes into the server's working copy even if they are not yet complete. Conflicts will result, but you risk losing whatever modifications you have made or brand-new code you created. Within a development team, miscommunication and errors are related to merging.

**2). List and compare at least three Open-Source Version Control Tools. Discuss the pros and cons of each of the Version Control Tools.**

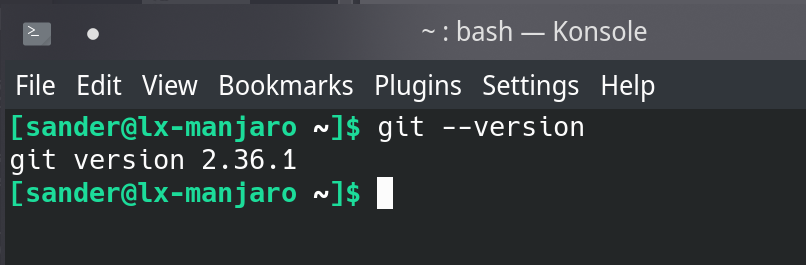
Git is one of the most effective and widely used version control systems available today. Small and big businesses, as well as individuals, utilize it. It is a rapid and effective tool that performs superbly. The cross-platform tool allows for rapid tracking of code modifications. However, it will be challenging to interpret your history log as it grows more extensive and complicated. Additionally, it does not enable time-stamp retention or keyword expansion.

Another tremendous open-source version control technology that has been around for a while is CVS. Like Git, it provides exceptional cross-platform support and is loaded with robust command-line scripting possibilities. Because of the tool's longevity, it has received support from a sizable CVS community.

Monotone is a less common alternative. It serves as a distributed revision control system and is developed in C++. This revision control system requires little upkeep, has clear documentation, and is simple to use. However, for some actions, like the first pull, the system does not experience performance concerns. Due to the non-HTTP protocol, another drawback is Monotone's inability to commit or checkout behind a proxy.

**3). Explore and Install Git Version Control Tool.**

Because I am using my device to demonstrate Git, the package had already been installed on my Linux Manjaro. In the screenshot below is the version of Git I am using.

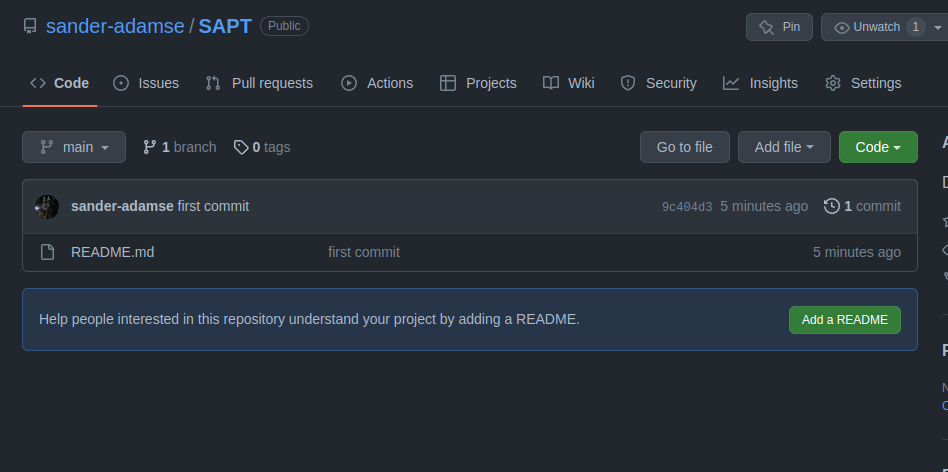
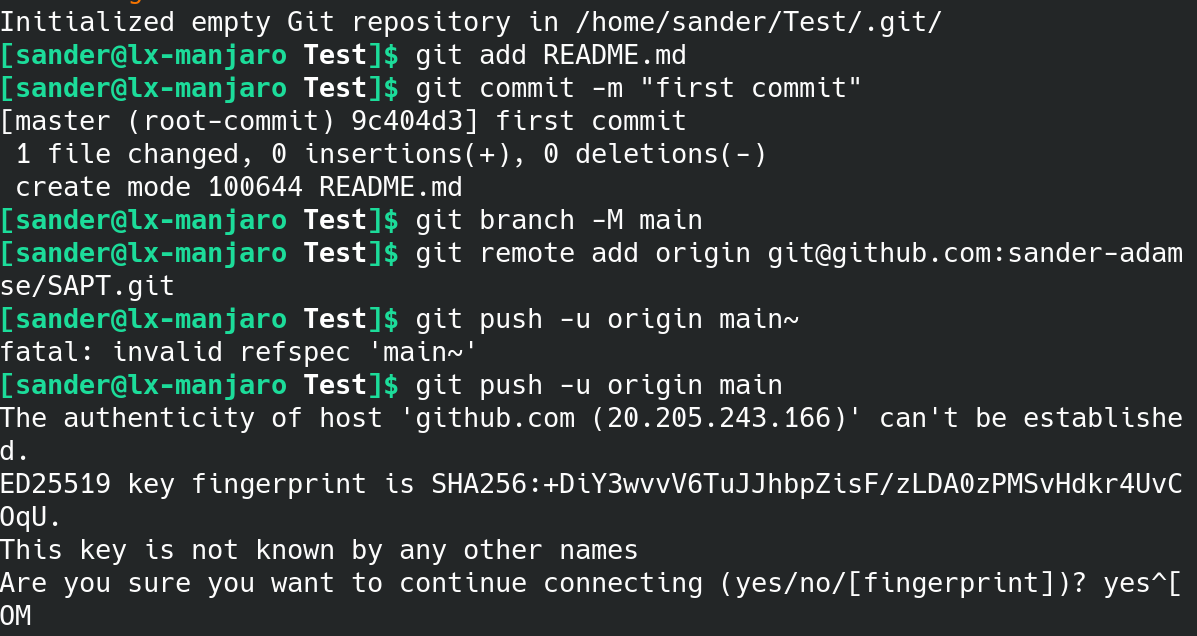


I have initialized a new Git repository in the new folder I created. Because I am using my public SSH key on GitHub, I do not need to keep verifying every action I perform. It is because GitHub has recently moved on from using personal credentials such as user-name and password.

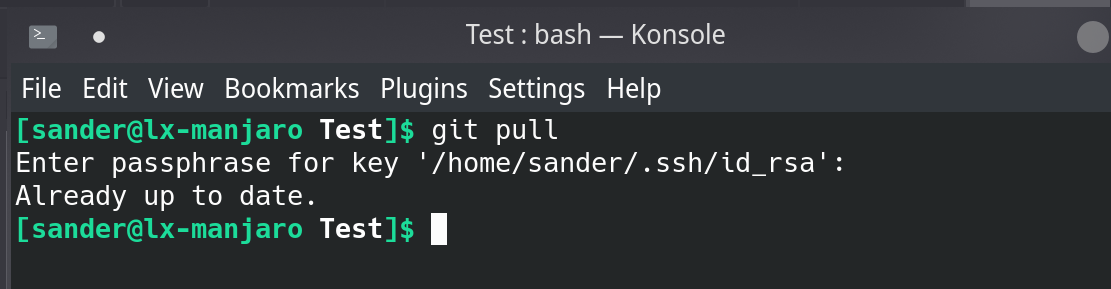
In the screen-shot below, I have set up a folder along with a repository on my personal GitHub. Then, I created a file, added this to the list of files ready to commit to GitHub, and then committed all to my GitHub repository named "first commit". Furthermore, I created the main branch where I will commit and push all files for this demonstration. All that is left to do is make Git push my changes to the main branch on GitHub. You can visit my GitHub page by going to <http://github.adamse.co/>

Setting up the repository with GitHub can be done in the following steps:

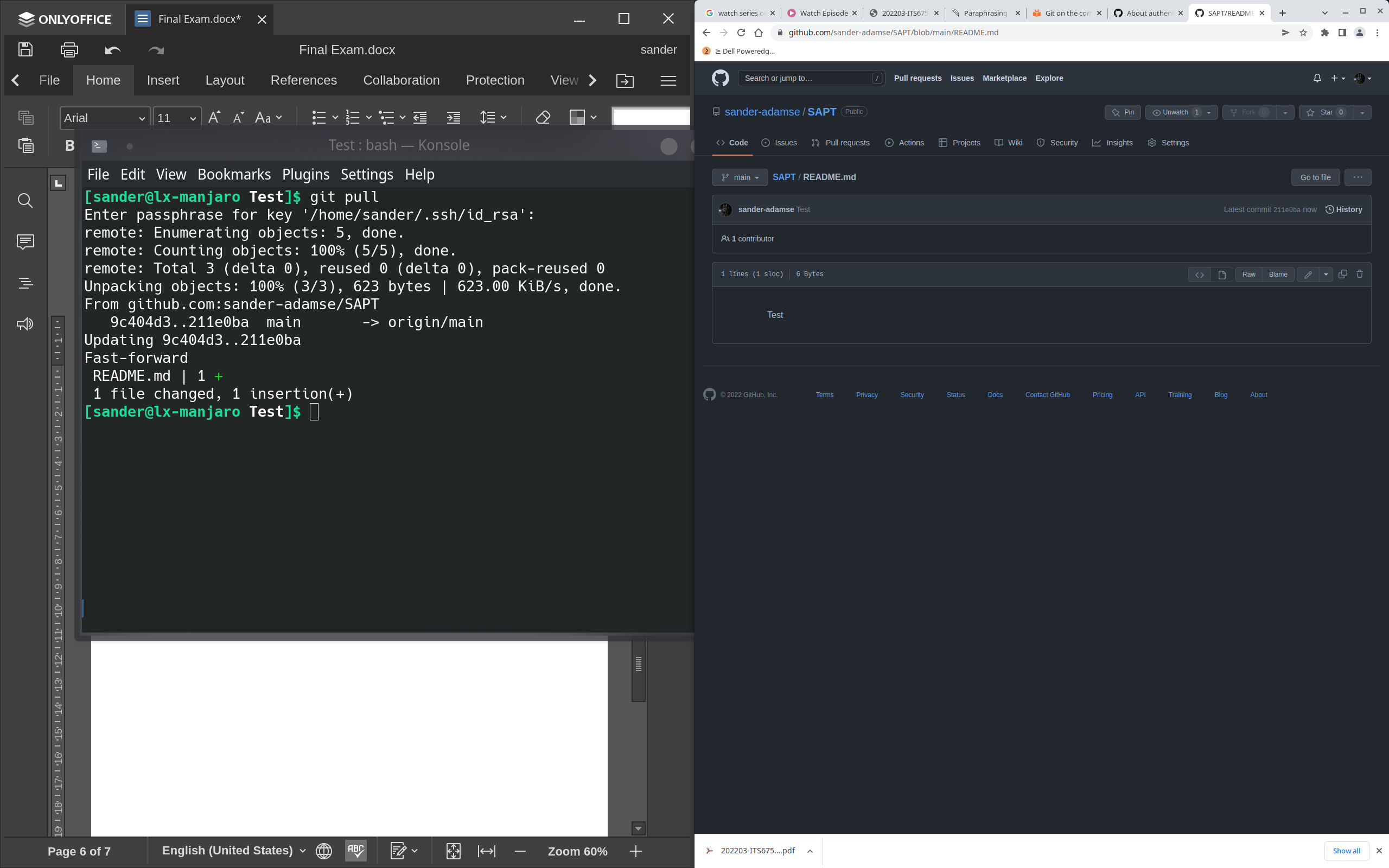
1. Create a folder on the local machine (mkdir test).
2. Change directory to that folder (cd test).
3. Create a file for the first commit (touch test.txt)
4. Initialize Git (git init).
5. Add the newly created file to the list of commits (git add test.txt).
6. Commit the files to be ready to push to the repository (git commit -m "description").
7. Create a branch (git branch -M main).
8. Create the remote origin using SSH (git remote add origin git@github.com:insertuser/test.git).
9. Push your files to the repository (git push -u origin main).



To show one more core functionality of Git is the “git pull” command. This command will pull any changes made to the repository from the repository to your local machine. The screen-shot below shows that it is currently up to date when I run the command because nobody has changed the file since I originally uploaded it to GitHub.



Now I have edited the file within the GitHub text editor on GitHub itself. When I run the command again, you can see it has pulled the changes to my local machine.



The difference between Git and GitHub can be easily explained. Git is a version control system/tool that allows end-users to manage and monitor the history of their source code. GitHub is a cloud-based hosting service where the end-user can manage these Git repositories. It was created to give the end-user an easier experience managing all their open-source projects.