## Homework: GitHub Fundamentals

### Background

To understand GitHub, you need to know the basics of \*\*version control\*\*.

- Version control is a system that allows users to save all versions of a file while working on it. It's like adding an undo function to any document or file. You create save points as you work, which you can revert to at any time.

- \*\*Git\*\* is the most popular software used for version control. It runs on your local computer and allows you create save points (known as \*\*commits\*\*) for your documents.

- You can use Git to manage any directory and track every item inside that directory. At any point, you can revert the Git directory (known as a \*\*repository\*\*) to a previous commit.

GitHub is a website that allows you to sync your local Git repository with a repository in the cloud. This allows you to save your work to the cloud, share your work with others, and easily collaborate on a project.

- Other users can access your online GitHub repository and sync their own changes. They can also make a copy of your repository to create an entirely new project based on your original project. This is known as \*\*forking\*\*.

In this activity, you will:

- Create a new, empty Github repository and sync it to your local machine.

- Once your repository is up, you will add all of your Ansible scripts, Bash scripts, and network diagrams to the repository, and sync it again with the cloud.

- When everything is synced, you will update the GitHub README file, which will explain each of the items in the repo and display a network diagram. You will then have a GitHub repository to present to future employers.

You will also use your GitHub account for other activities in the course.

### Required Files

- Ansible YAML scripts from the week on cloud security.

- Gather all of your Ansible YAML scripts from your Ansible container on your jump box.

- Copy and paste these into new documents on your local machine.

- Bash scripts from the week on Linux.

- Gather all of your system configuration scripts you created during the weeks on Linux.

- Network diagrams.

- Gather all of the network diagrams you created during the weeks on cloud security and networking.

### Your Goals

1. Create a GitHub repository for all of your files.

2. Copy all of your files into the repository and create a README explaining the repository.

### Topics Covered in this Assignment

The topics covered in this homework assignment are:

- Creating a new GitHub repository.

- Syncing a local repository.

- Creating a README file.

- [Markdown](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet)

- The following commands:

- `git pull`

- `git add`

- `git commit -m`

- `git push origin --set-upstream <-branch->`

### Instructions

1. Create your GitHub repository.

- Go to [github.com](https://github.com/) and sign up for GitHub.

- Confirm your email address.

- Click \*\*Create a Repository\*\*.

- Name your repository and give it a short description.

- Check the box for \*\*Initialize this repository with a README\*\*.

- Click \*\*Create Repository\*\*.

2. Download your repository.

- Click the green \*\*Clone or Download\*\* button on the right side.

- Copy the link.

- Go to your command line and run the command: `git clone https://github.com/your-username/yourlink.git`

- Enter your GitHub username and password to complete the download.

3. Once you have the repository downloaded, copy your scripts and diagrams into it.

- Create folders for `Linux`, `Ansible` and `Diagrams`.

- Copy your scripts and diagrams to the appropriate folder.

4. Sync your local and remote repositories.

- In your terminal, make sure you're located in the top directory of your repo.

- Run `git add .` to specify that you want to sync \_all\_ the items and directories that you just added to your repo. This command stages your files for a commit.

- Run the command `git commit -m "First commit"` to confirm the commit and add a note describing it ("First commit").

- Run `git push` to finalize the sync.

- Go to github.com and confirm your content is there.

5. Add the README file you created during Day 3 of the project week.

- Click on the `README.md` file in your GitHub repo.

- Click on the small pencil that reads \*\*Edit this file\*\* on hover.

- Copy and paste the `README.md` file you wrote during class.

- Make any desired changes and click \*\*Commit Changes\*\* at the bottom of the screen.

\*\*Note:\*\* READMEs are written in Markdown. This [Markdown Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet) has more information about writing in Markdown.

Check your repo for any errors or typos. You now have a GitHub repository that is ready to present and share with the world.

---

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.