Project 0: Install VM, learn git

Overview

In this project, you will need to setup VM environment for later projects and learn the fundamentals of git.

You are required to use `git` to track the progress of your work. **The project can receive a full grade only if the submission includes git history with at least one commit.**

You are encouraged to host your code in private repositories on GitHub, GitLab, Bitbucket, etc. At the same time, you are PROHIBITED to make your code for the class project public during the class or any time after the class. If you do so, you will be violating academic honesty policy that you have signed, as well as the student code of conduct and be subject to serious sanctions.

Task Description

The project contains two parts: basic git tutorial and VM setup.

Set Up Git Repo

Git is a distributed version control tool. In this project, you are required to learn several basic git commands, for example, duplicate the given project repo, make some changes and commit your local changes to remote repo. In terms of advanced git commands, please refer to git tutorials for further information.

Note that all example commands are executed on the host machine (your laptop), e.g., in `Terminal.app` (or `iTerm2.app`) on macOS, `cmd` in Windows, and `console` or `xterm` on Linux. After the last step (`vagrant ssh`) you will get inside the virtual machine and can compile your code there.

- Install git on your local machine

- Project skeleton is hosted in public repo `https://github.com/CS118W19/winter19-project0.git`. You need to duplicate it to your personal private repo (You are NOT fork it since github only support public fork now). Note github provides unlimited private repo for student accounts.
- Clone project template
    - `git clone [private git repo path] ~/cs118-proj0`
    - For example, `git clone https://github.com/CS118W19/winter19-project0.git ~/cs118-proj0`
- Make some changes to README.md.
- Commit updates.
    - `git add README.md`
    - `git commit -m 'update README.md'`
    - `git push`

Set Up Vagrant and Create VM Instance

Vagrant is a tool for building and managing virtual machine environments. In this project, you are required to setup your VM environment with provided vagrantfile. Please refer to vagrant tutorials for further information.

- Download and install your favourite virtualization engine, e.g., VirtualBox
- Download and install Vagrant tools for your platform
- Set up project and VM instance
    - Open project directory
    - `cd ~/cs118-proj0`
    - Initialize VM with provided vagrantfile. More info on vagrantfile.
    - `vagrant up`
    - Do not start VM instance manually from VirtualBox GUI, otherwise you may have various problems (connection error, connection timeout, missing packages, etc.)
    - To establish an SSH session to the created VM, run
    - `vagrant ssh`

- If you are using Putty on Windows platform, `vagrant ssh` will return information regarding the IP address and the port to connect to your virtual machine.
- Work on your project
    - All files in `~/cs118-proj0` folder on the host machine will be automatically synchronized with `/vagrant` folder on the virtual machine. For example, to view your codes, you can run the following commands:
    - `$ vagrant ssh`
    - `vagrant@vagrant:~$ cd /vagrant`
    - `vagrant@vagrant:~$ ls -a`

Notes

- If you want to open another SSH session, just open another terminal and run `vagrant ssh` (or create a new Putty session).
- If you are using Windows, read this article to help yourself set up the environment. You can also use Windows shell babun.

Demo Requirements

In discussion section, you need to show the TA:

1. Run following commands
   `$ vagrant ssh`
   `vagrant@vagrant:~$ cd /vagrant`
   `vagrant@vagrant:~$ git log`

Grading

You will receive full grades if your demo meet following grading criterias. Note you are free to use any git platform (GitHub, GitLab, Bitbucket, etc.) to hold your repo.

Grading Criteria

1. (50 pts) VM setup ready
2. (50 pts) At least one git commit