Intro to Containerization

Server Setup

Using Oracle VM VirtualBox, build a minimal installation of Ubuntu server 18.04. This means that you should not be using a Graphical User Interface (GUI), everything MUST be done through command line only. If prompted, do not install Docker from Featured Server Snaps during installation.

Using the "NAT" network interface is probably the best way to ensure that your project will work regardless of what network it is connected to (I.e at home vs on campus.) Even though you may be given a DHCP IP address initially, you **MUST CONFIGURE A STATIC, PRIVATE IP ADDRESS AFTER INSTALLATION.**

Immediately after installation is complete, set up a basic IPtables firewall to allow SSH, loopback, and HTTP connections. Ensure your rules persist after a reboot.

You will need to use PuTTy (or your preferred SSH client) to configure your server. The VirtualBox console interface should be used only for the initial installation, networking, and SSH setup, so you should be exclusively using the SSH window from this point onward.

Create three groups titled “grad”, “ugrad”, and “staff”

Grant your staff group Super User (sudo) privileges.

Create three users and assign the first to have “grad” as the primary group, the second to have “ugrad” as the primary group, and the third to have “staff” as the primary group.

Create three directories: /grad, /ugrad, and /everyone

You will need to configure permissions correctly so that users in the grad and ugrad groups are the only ones who can write to their respective directories. Anyone should be able to write to the /everyone directory, but they should not be able to delete files belonging to other users.

Docker Setup

Install the latest stable version of Docker (Community Edition) from Docker’s official repository. You will need to add the repository to the apt package manager before installation else you will install a relatively old version of the software.

It is highly recommended that you familiarize yourself with the basic concepts, functionality, and commands of Docker before continuing.

You are going to create a simple, containerized website using Apache, PHP, and Postgresql using the following images from Docker's registry:

- php:apache
- postgres:latest

Your setup must meet the following criteria:

- The files for your website and database must be located on persistent volumes mounted to their respective containers
- The php container will likely need additional configuration to work with the specified database. Use a Dockerfile to make these changes to the image
- Containers must be on a user-defined bridge network
- Your containers must restart when your virtual machine restarts
- Your website must interact with your database in some way (read information, add information, etc). Be prepared to show this, either by displaying such functionality on the website or by opening a shell in the database container and viewing the appropriate table(s)
- You must be able to navigate to your website from a browser on your host machine

Bonus

Use docker-compose to recreate "Docker Setup" in one command. **DO NOT** skip other parts of the project in favor of this bonus.

Additional Notes

Be able to explain the steps you took to accomplish these tasks and what you learned from this project. You will also need to complete this project in such a way that you can show us the results in person, such as on a laptop or on a desktop computer that can be remotely accessed.