Objective

Assignment 1 is intended to get everyone to the same baseline development platform, verify that you have a git repo, write a shell script, and stretch your skills at building/configuring software.

Step 1: GitHub Account

Setup an account at github, if you don't have one already, and create a new public repository for your CIS322 work. You will use this repository for all of your assignments this term in CIS322. Version control allows you to keep snapshots of your code in a way that makes them easy to compare. We strongly suggest that you commit your work as you make progress (e.g. after implementing a new function or updating documentation).

You may, during the term, get to a point where you would like to delete the whole repository and start over... don't do that. At the end of the term we will be looking at the history of commits git keeps during grading; starting a fresh repository part way through the term will cause problems. Instead, come to office hours and we can figure out the git commands needed to recover the repository to a good state. Recovering a messed up repository is an extremely useful skill.

Git also has a functionality referred to as "rebasing". Please connect with the instructor or GTF before rebasing a branch being used for grading. Rebasing destroys history, which can be a problem for grading, and can cause trouble when we pull your repository.

Step 2: Baseline platform

Download the OSNAP base virtual machine image and connect to it. The OSNAP base image is a fairly minimal installation of the Arch Linux distribution.

There are two users on the base image:

- osnapadmin - An administrative account that can use sudo to perform privileged operations
- osnapdev - A regular user account for doing development work

The virtual machine does not have a GUI installed and when working with the machine directly you'll need to use the console. An SSH server has been installed on the image and configured to start when the virtual machine boots; you'll need to configure the hypervisor to support using ssh to connect to an instance of your virtual machine.

When your code is tested, it will be done on a clean copy of the OSNAP virtual machine image and all commands will be run using the osnapdev user. You are free to use the osnapadmin account to configure your virtual machine any way you like but none of your local configuration changes or software installs will be available on the testing/grading platform. "It works for me" or "Just use pacman to install..." will not be accepted when a test fails to run when grading.

Step 3: Shell script and building

Shell scripting is a useful tool for automating tasks. Automation can help with repetitive tasks and to standardize how a task is done across several machines. Basic shell scripting, like we will be doing, works well when the environment is the same on all hosts where the script will be run. One way to think of a shell script is as a file that gives the same commands you would type to complete the task interactively.

In general, building software is a complex task that must work in many different environments. Build systems, like autotools, are able to detect differences between environments and adapt the compilation and installation behavior for the target computer. While I do not expect we will be implementing compilation using a build system in this course, you should be familiar with how to use standard build scripts created by others. Your script will compile and install postgres and Apache httpd from source using the configuration scripts provided by those projects.

The programming part of your assignment is to write a bash shell script to:

1. clone the postgres source code from github
2. configure/make/install postgres 9.5.x
3. use curl to download Apache httpd-2.4.25
4. configure/make/install httpd

Your shell script will be named "install_daemons.sh" and will be committed at the top level of your git repository. The script will be invoked with an argument giving the install prefix to be used when running the configure script for postgres and httpd.

You should configure your installations of postgres and httpd so that the daemons can be run from the commandline as the osnapdev user. Httpd should be configured to listen on port 8080 so that the port forwarding rule for the image will allow you to connect from your desktop. Use psql to verify that your postgres installation is working from the commandline. Use curl to verify that your httpd installation is working from the commandline. When we test your project for grading, we will use our own configuration files for the postgres and httpd binaries your script builds.

Step 4: Verify commit and push

Git has a two step process for publishing changes. The first step is to "commit", which adds a change set to the local copy of the repository. The second step is to "push", which copies all of the committed changes to the remote repository. To verify that the current version of your work will be seen by the grader, login to git hub using your browser and verify that the most up to date version of "install_daemons.sh" is there.