CMPSC 100
Computational Expression
Fall 2019
Janyl Jumadinova

Practical 1
30 August 2019
Due by midnight of the day of your practical
"Checkmark" grade

Summary

To learn how to set up a container-based platform, called Docker, for use in all class activities, labs and practicals. To set up Github and Github Classroom for use in the course. You will also continue to practice using Slack to support communication with the technical leaders and the course instructor.

Configuring Git and GitHub

During the subsequent practical and laboratory assignments, we will securely communicate with the GitHub servers that will host all of the project templates and your submitted deliverables. In this assignment, you will perform all of the steps to configure your account on GitHub, so that you are ready to start your first lab assignment using GitHub Classroom next week. You can also learn more about GitHub Classroom by visiting `https://classroom.github.com/`. As you will be required to use Git, an industry standard tool, in all of the laboratory and remaining practical assignments and during the class sessions, you should keep a record of all of the steps that you complete and the challenges that you face. You may see the course instructor or one of the teaching assistants if you are not able to complete a certain step or if you are not sure how to proceed.

1. If you do not already have a GitHub account, then please go to the GitHub website (`https://github.com/`) and create one, making sure that you use your "allegheny.edu" email address so that you can join GitHub as a student at an accredited educational institution. You are also encouraged to sign up for GitHub's "Student Developer Pack" at `https://education.github.com/pack`, qualifying you to receive free software development tools. Additionally, please add a description of yourself and an appropriate professional photograph to your GitHub profile. Unless your username is taken, you should also pick your GitHub username to be the same as Allegheny's Google-based email account. Now, in the #practicals channel of our Slack team, please type on one line your full name, "allegheny.edu" email address, and your new GitHub username.

2. Download `git` for the operating system of your laptop: `https://git-scm.com/downloads`

3. If you have never done so before, you must use the "ssh−keygen" program to create secure-shell keys that you can use to support your communication with GitHub. But, to start, this task requires you to type commands in a terminal. Open the terminal as you have done in the previous step. Alternatively, you can search for it by starting to type the word "terminal", and then select that program. Another way to open a terminal involves typing the key combination "<Ctrl>−<Alt>−t".

4. Now that you have started the terminal, you will now need to type the "ssh−keygen" command in it. Follow the prompts to create your keys and save them in the default directory. That is, you should press "Enter" after you are prompted to "Enter file in which to save the key ... :" and then type your selected passphrase whenever you are prompted to do so. Please note that a "passphrase" is like a password that you will type when you need to prove your identify to GitHub. What files does "ssh−keygen" produce? Where does this program store these files by default? Do you have any questions about completing this step?

5. Once you have created your ssh keys, you can raise your hand to invite either a technical leader or the course instructor to help you with the next steps as needed. First, you must log into GitHub and look in the right corner for an account avatar with a down arrow. Click on this link and then select the "Settings" option. Now, scroll down until you find the "SSH and GPG keys" label on the left, click to create a "New SSH key", and then upload your ssh key to GitHub. You can copy your SSH key to the clipboard by going to the terminal and typing "`cat ~/.ssh/id_rsa.pub`" command and then highlighting this output. When you are completing this step in your terminal window, please make sure that you only highlight the letters and numbers in your key—if you highlight any extra symbols or spaces then this step may not work correctly. Then, paste this into the GitHub text field in your web browser.

6. Again, when you are completing these steps, please make sure that you take careful notes about the inputs, outputs, and behavior of each command. If there is something that you do not understand, then please ask the course instructor or the teaching assistant about it.

Using a Container-based Platform, Docker

Docker is a platform for developers and system administrators to develop, deploy, and run software applications with containers. [TODO: add more flowery language about Docker]

The instructor in this course will deploy Docker containers with all the necessary software to run all class exercises, lab and practical assignments. Before using Docker, students must first complete the installations relevant to their operating system.

1. As the first step of the Docker set up process, please check and make a note of the version of your operating system. If your operating system meets the requirements outlined below, then proceed to the next step. For older MacOS and Windows Operating Systems, see the next section.

2. Then, go to `https://docs.docker.com/install/` and from the menu on the left-hand side select and follow the installation tutorial for your operating system (Linux, MacOS, or Windows). Please see additional notes below before proceeding.

! Docker Desktop for Mac and Windows requires you to create an account to download. To bypass this, use the following download links:

- Windows: `https://download.docker.com/win/stable/Docker%20for%20Windows%20Installer.exe`
- Mac: `https://download.docker.com/mac/stable/Docker.dmg`

!! You will need to use a command-line interface, also known as a terminal, in this course, which allows the user to interact with the computer by typing in commands instead of clicking on items. Operating systems such as MacOS and Linux have terminals that are built in by default. For Windows OS, we recommend that you use Git for Windows. [TODO: should delete: this will be installed when they install Git in the previous step].

Older Mac and Windows systems

Older Mac and Windows operating systems that do not meet the requirements of Docker Desktop for Mac and Docker Desktop for Windows can use Docker Toolbox. You can follow the tutorial on `docs.docker.com/toolbox/toolbox_install_windows/`, while keeping the following steps in mind.

1. Make sure your operating system version satisfies the requirements for the Docker Desktop.
2. Check that virtualization is running using Specci. If not enabled, then the BIOS settings must be changed. For example, to change this setting on an HP Windows 10 Home, you should first shut down your computer. Secondly, you turn on your machine, while holding the `esc` button and wait for the BIOS menu to appear. Then, select the `F10` key for the BIOS configuration. Now you should be in BIOS settings, where you can use the arrow keys to navigate to the "System configuration" tab and select "Virtual Technology" entry. Push `enter` and select "Enabled." Finally, use `F10` key to "Save and Exit" and reboot your computer. Please note that the specific keys and operations maybe different on your machine.
3. During the toolbox installation, there are several different items to select for installation. Make sure that the (Oracle) VirtualBox option is checked.

Since this is your first assignment and you are still learning how to use the appropriate software, don't become frustrated if you make a mistake. Instead, use your mistakes as an opportunity for learning both about the necessary technology and the background and expertise of the other students in the class, the teaching assistants, and the course instructor. Remember, you can use Slack to talk with the instructor by typing "@jjumadinova" in a channel.

General Guidelines for Practical Sessions

- **Experiment!** Practical sessions are for learning by doing without the pressure of grades or "right/wrong" answers. So try things! The best way to learn is by trying things out.
- **Complete *Something*.** Your grade for this assignment is a "checkmark" indicating whether you did or did not complete the work.
- **Practice Key Laboratory Skills.** As you are completing this assignment, practice using the Ubuntu terminal until you can easily use its most important features. Additionally, ask a teaching assistant or a course instructor to teach you some of the advanced features of the terminal, thereby helping you to work more effectively.

- **Try to Finish During the Class Session.** Practical exercises are not intended to be the equal of the laboratory assignments. If you are simply a slow typist, I've given you until the end of the day, but ideally you should complete the assignment by the end of the class period.
- **Help One Another!** If your neighbor is struggling and you know what to do, offer your help. Don't "do the work" for them, but advise them on what to type or how to handle things. If you are stuck on a part of this practical session and you could not find any insights in either your textbook or online sources, formulate an intelligent question to ask your neighbor, a teaching assistant, or a course instructor. Try to strike the right balance between asking for help when you cannot solve a problem and working independently to find a solution.