**01 – Docker Install & Unix Command Line Basics**

**Activities**

COMP190 – Tools and Techniques for Software Development

Dickinson College

Fall 2022

**Name:**

**Screen Shot Tool:**

Many of the exercises in this course you will ask you to take screenshots and provide them as your answers. This will allow me to see the windows, commands or outputs that you have generated in Linux. If you have a favorite tool for capturing regions of the screen you should feel free to use it. If you do have one you use regularly then I recommend the following:

*For Mac:*

On Mac OS, pressing CTRL + Shift + CMD + 4 will change the mouse pointer to a crosshair (⌖). Whatever you select will then be copied to the clipboard. You can also read about other ways to screenshot on a Mac here if you’d like:

* <https://www.cnet.com/tech/computing/how-to-take-a-screenshot-on-your-mac-4-ways-to-capture-your-screen/>

*For Windows:*

On Windows 10, pressing WIN + Shift + S will open a screenshot tool. You can probably figure out how to use it, but if you’d like you can read more about how to use it here:

* <https://support.microsoft.com/en-us/windows/use-snipping-tool-to-capture-screenshots-00246869-1843-655f-f220-97299b865f6b>

1. Experiment with your screenshot tool to figure out how to capture a selected portion of the screen and how to paste the captured image into a document. Capture a small rectangular portion of the screen. Do not capture the entire screen. It doesn’t matter what you capture, this is just to be sure you can do it. Paste the image that you captured inside the answer box below.

**Linux via Docker:**

As described in class, we will be using *Debian Linux* with the *XFCE4* desktop manager in this course. Also as described, we will be running this Linux machine within a *Docker container* and accessing it via the *Tiger VNC Viewer*. This will allow you to use and learn about Linux either on one of the lab machines or on your own Mac or Windows machine. The subsections below guide you through the process of getting our Linux machine up and running in a Docker container.

*Docker Desktop:*

The *Docker Desktop application* will allow you to get, create, run and interact with Docker containers, including the ones we will be using in this course. The Docker Desktop application is pre-installed on all the lab machines, so you can use it there without any additional installation. But, the Docker Desktop application (and all of the other software we will use) is free, so if your machine can support it, you will be able to install and run it on your own machine.

**If you plan to use a lab machine for your work, skip to the *Testing Docker* section below. Otherwise, continue with this section to ensure that your machine can run Docker Desktop and if so, to install it.**

2. If you prefer to work on your own machine you will need to install the Docker Desktop software. Before doing so, you should ensure that your machine has sufficient processor, memory and disk space to run Docker effectively. Record the following specifications from your machine here.

Operating System and Version:

Processor Type and Version:

RAM:

Free Disk Space:

3. Follow the appropriate link for your operating system to find directions for installing Docker Desktop on your machine:

* Windows: <https://docs.docker.com/desktop/install/windows-install/>
  + Use the *WSL 2 backend* (not the Hyper-V backend)
* Mac: <https://docs.docker.com/desktop/install/mac-install/>
  + Be sure to choose the “Mac with intel chip” or “Mac with Apple Silicon” section as appropriate for your machine.
* Linux: <https://docs.docker.com/desktop/install/linux-install/>

What minimum OS and hardware specifications does the install documentation give for installing Docker Desktop on your machine?

4. In addition to the specifications you found in #3, you will need a minimum of 15 GB of free disk space for this course. Does your machine meet the minimum specifications, including the free disk space?

If you answered no to question #4, I strongly recommend that you use one of the lab machines for your work in this course. Log into one of the lab machines and skip to the *Testing Docker* section.

5. If you answered yes to question #4, follow the *Install Interactively* directions to install Docker Desktop onto your machine.

Nothing required here.

*Testing Docker Desktop:*

6. Find the Docker Desktop application. Where is it located on the machine you are using?

7. Run the Docker Desktop application. Notes:

* If you are running Docker Desktop on your machine, the first time you launch it you will be asked for your password.
* The first time you run Docker Desktop you will be asked to accept the license agreement.
* It takes a little while for Docker Desktop to startup and initialize itself. Once it does it will appear in a window.

Nothing required here, but you must have Docker Desktop running to continue.

8. Use a *terminal* to run the following command to test that Docker is working correctly:

docker run hello-world

If this command works you should see some output in your terminal window that includes the text: “Hello from Docker!” That message confirms that your Docker installation is working correctly.

Type your name into the terminal and then take a screen shot of the terminal output including the “Hello from Docker!” line and your name and paste it here.

*Our Linux Container:*

Now that we know you are able to run Docker containers, you’ll create and run the docker container that we’ll be using for this course.

*Pulling the Image:*

8. Use a terminal to run the following command that *pulls a copy of the image* for our container from DockerHub onto your machine. Note: this may take a few minutes because the image can be large.

docker pull braughtg/190-unix-intro:f22

When this command completes, go to the panel for “Images” in the Docker Desktop application. You should see an entry there that indicates that you now have a copy of the image that you just pulled. If not, check the output in your terminal to see if there are any errors that you can correct. You will need to have this image in order to continue to the next question.

Take a screenshot of the entry for the image in Docker Desktop and paste it below.

*Creating the Container from the Image:*

9. Use a terminal to run the following command that *creates a container from the image*. Note that the command is slightly different on Mac or Linux than it is on Windows:

For Mac/Linux:

docker create --name comp190 --publish 5901:5901 --publish 6901:6901 --mount type=bind,source=/var/run/docker.sock,target=/var/run/docker.sock braughtg/190-unix-intro:f22

For Windows:

docker create --name comp190 --publish 5901:5901 --publish 6901:6901 --mount type=bind,source=//var/run/docker.sock,target=/var/run/docker.sock braughtg/190-unix-intro:f22

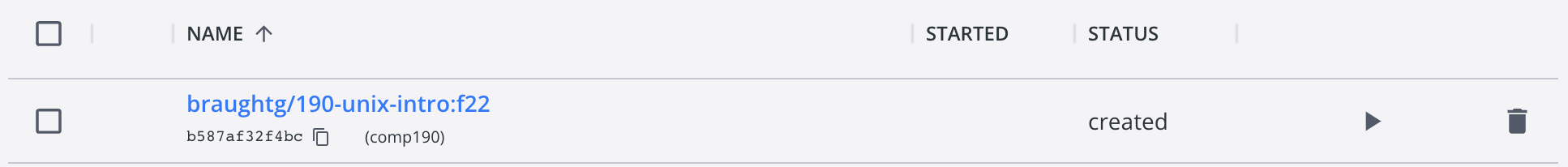
When this command completes, go to the panel for “Containers” in the Docker Desktop application. You should see an entry there that indicates that you have created a container from the image. If not, check the output in your terminal to see if there are any errors that you can correct. You will need to have this image in order to continue to the next question.

Take a screenshot of the entry for the container in Docker Desktop and paste it below.

10. What does Docker Desktop report as the status of the container?

*Starting a Container:*

11. Once a container has been created it can be *started* and *stopped* using Docker Desktop. Use the “play button” (circled below) in Docker Desktop to start the comp190 container.



When this command completes, the status of the container should now indicate that it is running. If not, check for any errors that you can correct. You will need to have the image running in order to continue.

Take a screenshot of the entry for the container in Docker Desktop now and paste it below.

*Stopping and Restarting a Container:*

When a container is running, Docker Desktop displays a number of additional buttons in its entry on the “Containers” pane. For our purposes we will usually only need the stop button.

12. Stop the container that is running. It may take up to 10-20 seconds for the container to stop. Once it does take a screenshot of the entry for the container in Docker Desktop now and paste it below.

13. What three *container status values* have you now seen? What does each status indicate?

It is worth noting here that you will only want to create our container once. After that you will just want to start it each time you want to work and stop it each time you are done. That way any work that you do between work sessions will be saved. If you instead create a new container from the image, any work you have done in the container will be lost.

*The Tiger VNC Viewer:*

Now that we know that Docker container with our Linux machine is running we’ll need a way to connect to and interact with it. For this we’ll use a *Virtual Network Computing (VNC)* client. More specifically, we’ll use the *Tiger VNC Viewer* as our client.

14. Download and install the Tiger VNC Viewer using the link for your operating system:

* Windows: <https://sourceforge.net/projects/tigervnc/files/stable/1.12.0/vncviewer64-1.12.0.exe/download>
* Mac: <https://sourceforge.net/projects/tigervnc/files/stable/1.12.0/TigerVNC-1.12.0.dmg/download>

Nothing is required here.

15. Run the Tiger VNC Viewer application. When the application starts you will see a dialog box similar to the one shown below:

Graphical user interface, website

Description automatically generated

Nothing is required here, but you will need to see this dialog box before you can continue.

16. The “VNC server” is the Linux machine that is running in our Docker Container. Enter the following into the “VNC Server” box in the dialog to connect to the Linux machine:

localhost:5901

When you have successfully connected you will see a “VNC authentication” dialog requesting a password, similar to the one below:

Graphical user interface, text, application, email

Description automatically generated

If you do not see this dialog requesting a password and Tiger VNC Viewer simply quits when you attempt to connect, check in Docker Desktop to be sure that the container is running. If it is not running use the “play button” to start it.

Nothing is required here, but you will need to see this dialog box before you can continue.

17. The password for our Linux machine is comp190. When you enter this password, you should see the desktop for our Linux machine appear in a window like the one shown below:

Graphical user interface, application

Description automatically generated

Nothing is required here, but you will need to see this dialog box before you can continue.

Congratulations! You have successfully started our Linux machine in a Docker Container and connected to it using the Tiger VNC Viewer!!!

**Unix Command Line Basics:**

Now that you have successfully started and connected to our Linux machine, this section contains a few activities that will get you started interacting with Linux through the command line. We’ll be adding to this experience over the next several weeks.

*Linux via the XFCE4 Desktop Environment:*

Like other operating systems (e.g. Mac/Windows) Linux allows you to run programs in windows within a *Graphical User Interface (GUI)* by using a *Desktop Environment*. Our Linux machine is configured to use the *XFCE4 desktop environment*. It is what you see when you connect to the machine using the Tiger VNC Viewer. The activities in this section will give you a little familiarity with the XFCE4 desktop environment, while preparing to learn about the Command Line Interface (CLI) shortly.

18. The *File Manager* application allows you to examine the files that are saved on the machine, much like the Finder on Mac or the File Explorer on Windows. Open the File Manager by clicking the “filing cabinet” icon at the bottom of the screen.

The location that the File Manager is displaying is called your *home directory*.

* What is the name of your home directory?
* What files or *directories* (i.e. *folders*) currently exist in your home directory?

19. The *Mousepad* application is a simple text editor provided with the XFCE4 desktop. It can be used to create and edit text files. You can open the Mousepad application from the “*Accessories*” section of the “*Applications*” menu at the left of the top menu bar in XFCE4. Use the Mousepad application to the following three files and save them into your home directory.

* *<name>*-places.txt – a list of at least 5 of your favorite places in the world.
* *<name>*-foods.txt – a list of at least 5 of your favorite foods.
* <name>-movies.txt – a list of at least 5 of your favorite movies.

< > are often used to denote text that is to be replaced. So here you will replace the *<name>* part of the filename with your actual name. So if your name is Nguyen, then your first file will be named Nguyen-places.txt. Notice that the < > are not included.

Use the File Manager to check that you have saved these files into your home directory. If not, either move them there or try again.

Give a screenshot of the File Manager showing these two files in your home directory.

20. Use the File Manager to accomplish the following tasks:

* Create a new directory (i.e. folder) named Notes in your home directory.
* Move one of the .txt files that you created into your Notes directory.

Give a screenshot of the File Manager showing the contents of your Notes directory.

*Linux via the Command Line Interface:*

The Linux *Command Line Interface (CLI)* provides a text-based way to interact with the operating system. Note that both Mac and Windows also have CLIs. In general, everything that can be done with the GUI can also be done with the CLI, and *mostly* vice versa. The exercises in this section will help to demonstrate that the GUI and CLI are just different ways of interacting with the same content.

20. The Linux CLI is usually accessed through the *Terminal* application. When the Terminal application is opened it runs a program called a *shell* which displays a *command prompt* at which you will be able to type Linux commands.

Open a *Terminal Window* by clicking on the icon shown below at the bottom of the desktop.



Give a screenshot showing the complete command prompt that is displayed in the Terminal?

21. Different shells will display different command prompts, and they can often be customized by advanced users. The command prompt that you see will give you several pieces of information. It begins with the *username* of the current user followed by an @ symbol. It then contains the name of the Linux machine on which the shell is running followed by a colon (:). Finally, it displays the name of the *working directory* followed by a $.

Use the command prompt you see to identify the following pieces of information:

Username:

Machine name:

Working directory:

Note: The machine name is an odd string of seemingly random letters and numbers because we are running Linux inside of a Docker container. If we were connected to a server somewhere you would see a more normal name for the machine.

22. You *execute (i.e. run)* commands in the CLI by typing a command at the command prompt and then pressing the *Enter or Return key* on the keyboard. The following commands will illustrate some of the ways in which the CLI can be used to accomplish the same things that you did in the File Manager.

a. The ls command lists the files and directories in that are in the working directory. Execute the ls command. Give a screenshot showing the command prompt, your ls command and the output that was generated. Just like the File Manager did, the output of the ls command should show your Desktop folder, your Notes folder and one of your .txt files.

b. The cd <dir> command changes the working directory to the directory specified by <dir>. So, you can display the contents of a directory by first using cd to make it the working directory and then by using the ls command to display the contents.

Use the cd and the ls commands to display the contents of your Notes directory. Give a screenshot that shows the commands that you used and the output that was generated.

c. The File Manager provides the  button to move out of a directory (e.g. from your Notes directory back into your home directory). We can do the same thing in the CLI using the cd .. command. Use this command to change the working directory from Notes back to your home directory. Give a screenshot that shows the command that you used and the new command prompt that is given (note: The working directory displayed in the command prompt will have changed!)

d. Just like you can make a directory using the File Manager you can also make a directory using the CLI. The mkdir <dir> command creates a new directory with the name specified by <dir>.

Use the CLI to create a new directory named Notes2 in your home directory. Then use ls to show the contents of your home directory to confirm that the directory was created. Give a screenshot that shows the commands that you used and the output that was generated.

e. You were able to move one of your text files from your home directory into your Notes directory using the File Manager. The mv <file> <dir> command moves the file specified by <file> into the directory specified by <dir>.

Use the CLI to move the .txt file from your home directory into your Notes2 directory. Then display the contents of your Notes2 directory to confirm that the file has been moved. Give a screenshot that shows the commands that you used and the output that was generated.

23. Complete the table below by filling in the Linux command that corresponds to each task.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **Command** | **Task** |  |
|  |  | Change the working directory. |  |
|  |  | Display the files in the working directory. |  |
|  |  | Create a new directory. |  |
|  |  | Move a file into a directory. |  |
|  |  |  |  |

24. Optional Bonus Question: Move the .txt file from your Notes2 directory back into your home directory. Display the contents of your home directory to confirm that the file was moved. Hint: See #22c. Give a screenshot that shows the commands that you used and the output that was generated.

25. Optional Bonus Question: What do you think the command cd ~ does? Explain why? Hint: Try experimenting with using this command from different working directories and consider your answer to #21.

**Optional:** To help us improve and scope these activities for future semesters please consider providing the following feedback.

a. Approximately how much time did you spend on this activity outside of class time?

b. Please comment on any particular challenges you faced in completing this activity.