CS 470 Module One Assignment Guide

Introduction
In this lesson, you will gain exposure to using Docker. In previous courses, you downloaded and installed MongoDB onto your own computer and then developed applications to use it. Before your application can run on another computer, the correct version of MongoDB must be installed there. Installing software also means that your computer's configuration and settings are altered, which may impact other software you have installed. Additionally, there is a risk that the other computer or its installation of MongoDB are not configured identically to your machine.

To overcome this common situation, you should run MongoDB inside a container. Moving the container to a new computer brings all the configuration and settings along as well. Practice using a pre-built Docker container with MongoDB already installed and configured.

Summary Steps

- Install Docker Desktop.
- Open Docker Dashboard.
- Ensure Docker is running.
- Open the PowerShell prompt.
- Create a default MongoDB container with the following commands:
    - docker pull mongo
    - docker run -d -p 27017-27019:27017-27019 --name mongodb mongo
    - docker exec -it mongodb bash
        - mongo
        - use admin
        - show dbs
        - **Ctrl+D** (exit mongo shell)
        - **Ctrl+D** (exit Docker remote bash shell)

Detailed Steps
Open Docker Dashboard by right-clicking on the Windows System Tray icon.
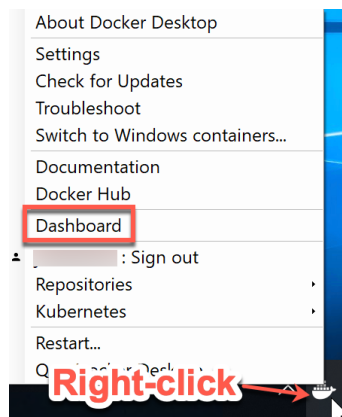
Ensure Docker is running. The icon to the lower left of the dashboard should be green.

Use the pull command to retrieve a pre-built and configured image from the Docker Registry:

> docker pull mongo

Note: The Docker Registry has official container images from many popular vendors and open source projects, including operating systems, databases, web servers, applications, and programming environments. Be careful, as you may find yourself up all night checking out the software available at your fingertips!

Use the Docker run command to start the container:

> docker run -d -p 27017-27019:27017-27019 --name mongodb mongo

-d means "detach" the console from the container so it runs in the background and lets you use additional commands in your window.
-p 27017-27019:27017-27019 means to map the range of ports on your computer to the range of ports listed after the colon inside the container.
-- name mongodb is the friendly name you wish to assign to the container.
mongo is the name of the image you just pulled down from the registry.

Note: The long string of characters displayed after the run command is the unique instance identifier of the container.

Use the Docker images command to view the installed images on your computer:

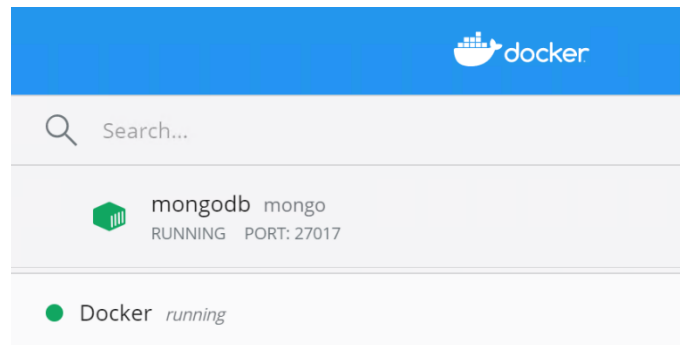> docker images

```
> docker images
REPOSITORY          TAG             IMAGE ID            CREATED           SIZE
mongo               latest          3f3daf863757        8 days ago        388MB

>
```

You will also see the new container in the Docker Dashboard.

Use the Docker exec command to execute a command within the running container. To open a bash shell session inside the container, use the following command:

> docker exec -it mongodb bash

-it means the command will be run interactively and the input/output will be reflected in your local console terminal (window).

mongodb is the friendly name of the container that you want the command to run in.

bash is the name of the command to execute.

Note: The "uname -a" command prints information about the machine. As you can see, even though the host computer is running Windows 10 and the blue window is a PowerShell command window, the remote container is running Linux! The "root@e1d5ffe938fb" on the shell prompt indicates the user and hostname of the machine running inside the container. Advanced tip: The hostname is the first several characters of the unique container ID shown earlier.

Until you exit the remote bash shell, anything you type will be run inside the container, as if your display and keyboard were connected directly to the Linux computer in there. Try the following commands:

> mongosh #launch the MongoDB shell
> use admin #use the admin database
> show dbs #list the databases
> exit #exit mongo shell
> exit #exit Docker remote bash shell

```
>  docker exec -it mongodb bash
root@e1d5ffe938fb:/# uname -a
Linux e1d5ffe938fb 4.19.76-linuxkit #1 SMP Thu Oct 17 19:31:58 UTC 2019 x86_64 x86_64 GNU
root@e1d5ffe938fb:/# mongo
MongoDB shell version v4.2.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("da0278fc-2851-4510-8548-89d54d39d5d0") }
MongoDB server version: 4.2.6
Welcome to the MongoDB shell
```

```
> use admin
switched to db admin
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
```

Note: You can press **Ctrl+D** to signal "exit" instead of typing out the word. Over time, shortcuts like this will save you time.

Notice the container is running MongoDB version 4.2.6. It is possible to use different versions by specifying the version number on the Docker pull command.

Use the Docker stop command to stop the container and free up resources (memory, processor) on your computer:

> docker stop mongodb

NOTE: to restart the mongodb (or any other container that is stopped) use

> docker start mongodb

Optional
You can visit the Docker Hub and try taking other images for a spin. One important concept to note: Containers are intended to be smaller and lighter-weight than full virtual machines, so graphical interfaces are not possible. If, for example, you wished to try out Ubuntu and get the full experience, you should instead use a virtual machine since VMs support the full set of hardware on your machine – mouse, USB memory sticks, printers, scanners, and so on.