Nitin Nagavel

SE 422X

**Individual** Assignment 2

Docker and Containers

Total Weight: 5%

Due Date: Friday March 25

Objectives: You will be getting familiar with Docker by following Docker’s documentation. After completing this assignment you should be more familiar with Docker, containers, and some of the functionality.

**How to Sign up:**

You will need to sign up during part 4 of the tutorial through the docker hub. It is the first step in part 4 and it provides links as to where to sign up. When you click the link it should open up straight to the sign up page where you create a docker id, email, password. If it helps, you can use your ISU email (although not required). This login will help you sign in onto the docker dashboard and play with docker (also in part 4).

Here’s the link to sign up in case: https://hub.docker.com/

**No Credit Card Needed:**

To sign up you don’t need any credit card info. You will just sign up using a free tier account. **If you get to point where credit card information is required, you must have not used the link through the docker documentation**. I do not need to enter any payment information.

**Link to Docker’s Documentation site**:

Please refer to the side menu to the left and do parts 1 - 9 in “Get Started”:

https://docs.docker.com/get-started/

**What to Submit**

* A Single PDF Document with the necessary screenshots for each part
* A one-paragraph summary of what you learned

To complete this assignment follow the documentation and steps at the link above. Below you will find a list of the screenshots you need to provide at each of the 9 parts. Please make sure you provide all the listed screenshots below and provide captions for each.

**List of Required Screenshots**

**Part 1: Getting Started**

* Docker Dashboard with the docker/getting-started container running
* A screenshot of a computer

  Description automatically generated with medium confidence

Application with instance running

**Part 2: Sample Application**

* Dockerfile code

Graphical user interface, text, chat or text message

Description automatically generated

Dockerfile code with the configs

* The web browser of the running application

A screenshot of a computer

Description automatically generated with medium confidence

Sample items hosted on 3000 port

* Docker Dashboard with application container running

A screenshot of a computer

Description automatically generated with medium confidence

2 instances running as given

**Part 3: Update the Application**

* Updated web browser with modifications

Graphical user interface, text, application

Description automatically generated

“You have no todo items”

* Docker Dashboard with new application container running

A screenshot of a computer screen

Description automatically generated with medium confidence

New updated instances running in the Dashboard

**Part 4: Share the application**

* Instance and terminal on Play with Docker site (similar to the documentation screenshot)

A screenshot of a computer

Description automatically generated with medium confidence

Terminal shows the session from the website

**Part 5: Persist the DB**

* Docker Dashboard with Ubuntu container

A screenshot of a computer screen

Description automatically generated with medium confidence

New Ubuntu Container shown in the screencapture!

* Output for command: cat /data.txt

Text

Description automatically generated

Running the terminal in the Ubuntu instance

* Output of random number for command: docker exec <id> cat /data.txt



Running the terminal in the command prompt

* Added items on to-do list with timestamp

A screenshot of a computer

Description automatically generated with medium confidence

Timestamp with the todo list

* Command of stopping/removing container with timestamp

A screenshot of a computer

Description automatically generated with medium confidence

**Stopped and Removed the timestamp**

* To-do list after command with timestamp

A screenshot of a computer

Description automatically generated with medium confidence

Todo list still shows up after stopping the instance

**Bottom Right Corner is the TIME!!**

* Output for command: docker volume inspect

Text

Description automatically generated

Docker volume shows up when inspecting the DB

**Part 6: Use bind mounts**

* Output for command: docker logs -f <id>

Text

Description automatically generated

Docker logs show sqlite database information

* Web browser application after modifications

Graphical user interface, application

Description automatically generated

New webpage!

**Part 7: Multi-container apps**

* Output to verify the todo database is created and connected

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Both Screenshots show the mysql authentication

* Output for command: dig mysql

Graphical user interface, text

Description automatically generated

Dig mysql shows info regarding the server info

* Select statement output from the todo items

Graphical user interface

Description automatically generated with medium confidence

The 3 instances from the website in the mysql database

* Docker Dashboard with mysql container running

Graphical user interface, text

Description automatically generated

Mysql database container running

**Part 8: Use Docker Compose**

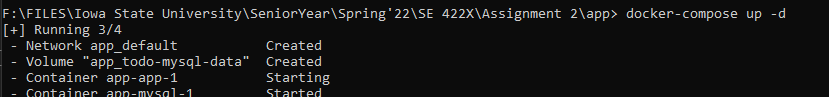
* Entire docker-compose.yml file

Text

Description automatically generated

Config file showing the Docker Compose

* Output for command: docker-compose up



Creates the docker compose

* Docker-compose logs

Text

Description automatically generated

Logs show various information for MySQL spanning thousands of lines

* Docker Dashboard with app group

Graphical user interface

Description automatically generated with medium confidence

App running in the backgroungd

**Part 9: Image-building best practices**

* Docker scan output

Text

Description automatically generated

Scan is running showing the scan being performed

* Docker image history output

Text

Description automatically generated

Image history showing the various dockerfiles being created

* After modifying the package.json file, provide output for the new build

Graphical user interface, text, application

Description automatically generated

Output showing the new build after updating the file

Text

Description automatically generated

The updated title

* After modifying the title, provide output for the new build

Graphical user interface, text

Description automatically generated

Output showing the new build after changing the index file

Text

Description automatically generated

Updated Index file.

Good Luck.

Paragraph:

Throughout this assignment, I learned the importance of Docker when it comes to Cloud Computing. First, Container images are being created through its own custom filesystem. Next, I learned to start a container and setup a running react app using the Dockerfile. Next, I was able to add items through a todo list and I would have to rebuild and start a new container every time a change is made. Then, I was able to push the image to a registry and run the freshly pushed image. Then I realized that we can create an application and survive the restarts of the instances through DBs. I learned later that the SQLite migration was key for entering production. And Finally, I was able to learn how the use Docker Compose through multi container applications. I hope to use my knowledge in my future job and create Docker and run instances!