Complete Project 1: Docker Containers

Re-submit Assignment

Due Nov 30 by 11:59pm **Points** 75 **Submitting** a website url or a media recording
Available after Nov 20 at 12am

Getting Started:

Introduction

MongoDB is a practical NoSQL database solution. It does not use a fixed data structure, making it scalable and ideal for managing dynamic workloads. MongoDB is well suited for distributed environments, such as Docker containers.

Using Docker and an official MongoDB container image can significantly shorten and simplify the database deployment process.

These instructions will show you **how to deploy a MongoDB instance on a Docker container**.

Prerequisites

- A user with **Administrative/sudo** privileges
- Access to a command line
- A running Docker instance

Step 1: Download MongoDB Image for Docker

Follow the step-by-step instructions below to download the latest official MongoDB image for Docker.

Note: for Mac Users: Please Add "**sudo**" in front of the docker commands.

1. Your Docker service needs to be active and running. You can quickly check the current status by entering the following command in your terminal:
   You need to **Screen Capture** the output of the Docker service is active and running.
   Type the following command:

   ```
   docker service ps -q
   ```

   ```
   sudo docker service ps -q
   ```

2. Proceed to download the latest official Docker image for the MongoDB database:


You need to **Screen Capture** the output

```
docker pull mongo
```

```
sudo docker pull mongo
```

Note: The image indicates that the system used the `latest` tag by default.
Optional:

- To download a specific version of MongoDB, use the same command appended with the version tag. For example:


You need to **Screen Capture** the output

-
```
docker pull mongo:4.2.2
```

```
sudo docker pull mongo:4.2.2
```

3. List the images in your Docker repository with the following command:

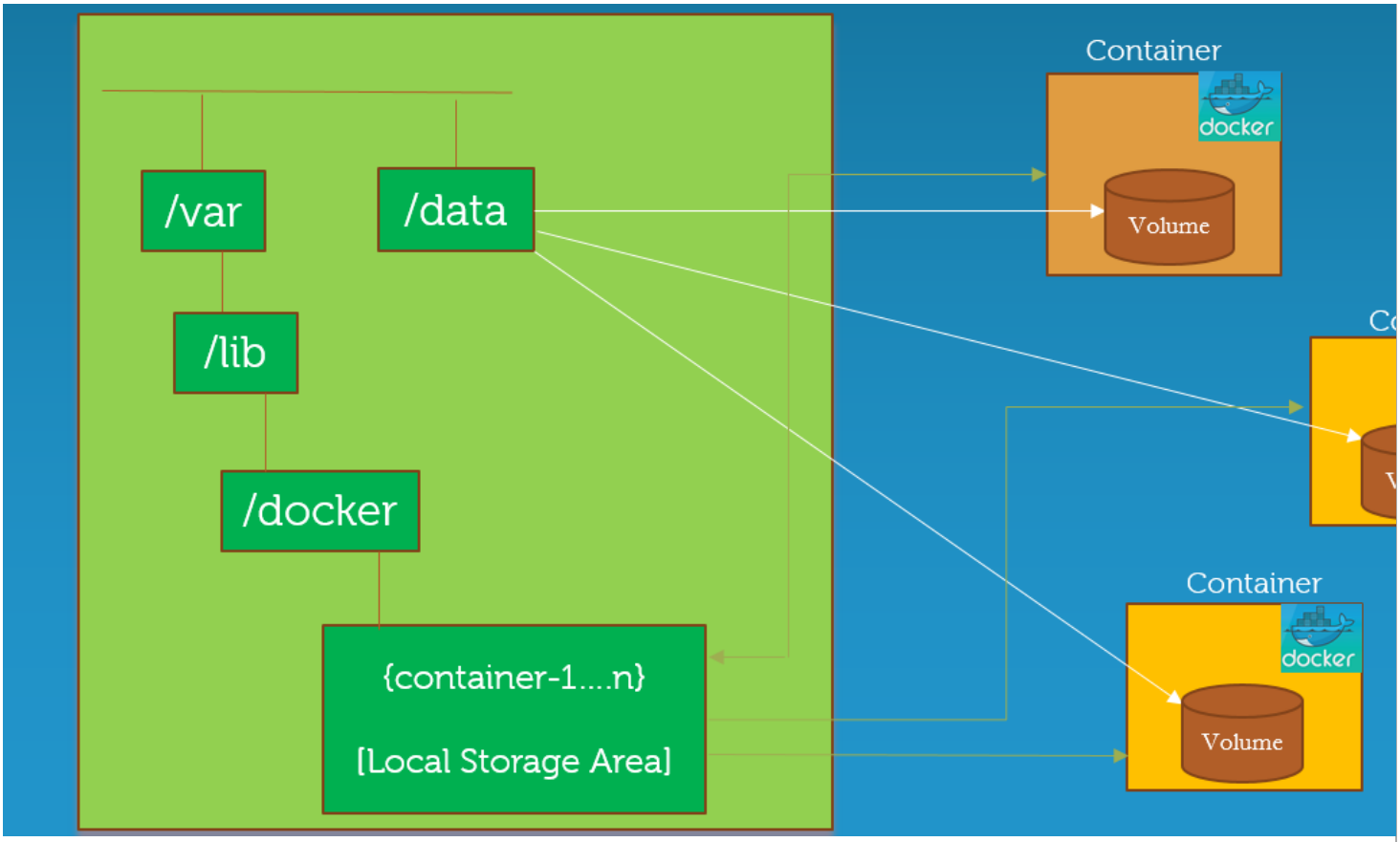

You need to **Screen Capture** the output

```
docker images
```

```
sudo docker images
```

The interface confirms that the MongoDB image is now available.

Volumes:



Deploy MongoDB Container

Note: for Mac Users: Please Add "**sudo**" in front of the docker commands.

By default, MongoDB stores data in the **/data/db** directory within the Docker container. To remedy this, mount a directory from the underlying host system to the container running the MongoDB database. This way, data is stored on your host system and is not going to be erased if a container instance fails.

1. a- Create a **/mongodata** directory on the host system:
   In Winows OS:

```
mkdir mongodata
```

1. b-Create a **/mongodata** directory on the host system
   In Mac OS:

```
sudo mkdir -p /mongodata
```

2. Start the Docker container with the `run` command using the mongo image. The **/data/db** directory in the container is mounted as **/mongodata** on the host. Additionally, this command changes the name of the container to *mongodb*:

You need to **Screen Capture** the output

```
docker run -dit -v mongodata:/data/db --name mongodb -d mongo
```

```
sudo docker run -dit -v mongodata:/data/db --name mongodb -d mongo
```

`-it` – Provides an interactive shell to the Docker container.

`-v` – Use this option to attach the /**mongodata** host volume to the **/data/db** container volume.

`-d` – Starts the container as a background process. Detached mode.

`--name` – Name of the container.

3. Once the MongoDB server starts running in a container, check the status by typing:
   You need to **Screen Capture** the output

```
docker run ps
```

```
sudo docker ps
```

4. Optionally you can specify the MongoDB port explicitly:
   The default port number is **27017,** as can be seen in the output**.**
   You need to **Screen Capture** the output

```
docker run -it -v mongodata:/data/db -p 27017:27017 --name mongodb -d mongo
```

```
sudo docker run -it -v mongodata:/data/db -p 27017:27017 --name mongodb -d mongo
```

5. Always check the Docker log to see the chain of events after making changes:
   You need to **Screen Capture** the output
   The logs provide a wealth of useful information.

```
docker logs mongodb
```

```
sudo docker logs mongodb
```

Start Interactive Docker Terminal (Shell) to Manage

MongoDB Database

1. The container is currently running in **a detached mode**. Connect to the container using the interactive terminal instead:

You need to **Screen Capture** the output

```
docker docker exec -it mongodb
```

```
sudo docker exec -it mongodb bash
```

Start the MongoDB shell by typing `mongo` in the interactive terminal.

The MongoDB shell launches and the prompt is ready to accept your commands.

3. Instead of just typing `mongo`, you can additionally define a specific host and port by typing:

```
mongo -host localhost -port 27017
```

```
show db
```

Shows Default Databases:

Create a new Database:

```
use afs205_test
```

Insert a new record into the database

```
db.user.insert("student":"david martin"})
```

Find the record in the database

```
db.user.find()
```

Show the list of Database:

```
show db
```

With the MongoDB shell, you can now create a database, add collections or manage individual documents.

How to Exit MongoDB and Interactive Shell


You need to **Screen Capture** the output

1. Type `exit` to leave the MongoDB shell and then `exit` once again to leave the Interactive shell.

As an alternative, you can type `quit()` or use **Ctrl-C** to exit the shell.

Stopping and Restarting MongoDB Database

1. The `docker stop` command is a short and clear command that **stops running container instances (https://phoenixnap.com/kb/how-to-list-start-stop-docker-containers)** :


You need to **Screen Capture** the output

```
docker stop mongodb
```

```
sudo docker stop mongodb
```

2. Inspect the list of running Docker containers by typing:


You need to **Screen Capture** the output

```
docker ps
```

```
sudo docker ps
```

3. Containers are started by using the `docker start` command:


You need to **Screen Capture** the output

```
docker start mongodb
```

```
sudo start mongodb
```

4. The list of running containers now confirms that the MongoDB database has been initiated once again:

You need to **Screen Capture** the output

```
docker ps
```

```
sudo docker ps
```

Review:

You now know how to install MongoDB on a Docker container, and you have learned how to access the MongoDB shell to manage databases.

Use Docker to streamline MongoDB database deployment across multiple servers and scale your operations quickly and efficiently.

Validation :

Use the following ref:**https://www.mongodb.com/products/compass, (https://www.mongodb.com/products/compass,)** to install the MongoDB Compass Client. And test your connectivity to MongoDB from the Compass Client.

Submission

You will submit a screencast video explaining every requirement of the assignment; and, you will submit the GitHub URL to your assignment files, so I can review your code.

How to Submit Your GitHub URL:

1. When you're done, stage, commit, and then push your changes to your Github repository.
2. Then, copy the URL. Your URL should look similar to this: https://github.com/username/class/week/project
3. Go back to this assignment in Canvas, click the Submit Assignment button. At the bottom of the page, paste the URL into the Website URL field, and click Submit Assignment.

How to Take a Screenshot:

1. Take a screenshot of the required material, ensuring everything can be seen (Go to https://www.take-a-screenshot.org **(https://www.take-a-screenshot.org/)** to learn how to take a screenshot). You can save your screenshot as an image or paste it to a Word document.
2. Go back to the assignment in Canvas, click the Submit Assignment button. At the bottom of the page, upload your file.

How to Create a Video Screencast:

1. Go back to this assignment in Canvas, click the Submit Assignment button, if you haven't already. At the bottom of the page, select the Studio tab.
2. Take a screencast of the required material, ensuring you clearly display what has been required from the assignment.
3. Select the Embed button, to ensure your screencast link saves to this assignment.

Once **both** your URL and Screenshot/Screencast have been uploaded, click the Submit Assignment button.