

SP23 - INFO - I535 Docker A2

Step 0: Launch a Virtual Machine that contains Docker

Created instance with given name and parameters

1. Flavor/instance size: m3.small
2. Root disk size : 20 GB
3. Number of instances : 1
4. Enable web desktop : Yes
5. Advanced options: hide

Step 1: Hello World

This was the first time I was working with Docker, and I tried to implement docker first on my local Pc with VS code. Installation was successful but hosting a web-based application failed to try.

Having experience with Linux, it was easier to follow instructions and proceed with the given task

Docker allows you to package your application and all its dependencies into a container that can run anywhere, whether it be on a developer's laptop or a production server. This means you can easily move your application between different environments without having to worry about differences in operating systems or dependencies.

```
The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Last login: Sat Feb 25 15:49:20 2023
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

exouser@docker-practice-amhaske:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:6e8b6f026e0b9c419ea0fd02d3905dd0952ad1feea67543f525c73a0a790fefb
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

Step 2: Docker Image

```
exouser@docker-practice-amhaske:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
guacamole/guacamole  <none>             a385e28f9fd6       8 months ago       642MB
guacamole/guacd-dev  <none>             315a12ba560b       9 months ago       228MB
hello-world          latest             feb5d9fea6a5       17 months ago      13.3kB
exouser@docker-practice-amhaske:~$
```

Step 3: Create the First Container

```
exouser@docker-practice-amhaske:~$ sudo docker run hello-world
Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
exouser@docker-practice-amhaske:~$
```

The first three steps were easy to implement because of a previous assignment, Docker containers are lightweight and can be started and stopped quickly, which can save time and resources compared to traditional virtualization.

Docker containers provide a layer of isolation between the application and the host operating system, which can help improve security by limiting the impact of any potential security vulnerabilities.

Create and run an app

Step 4: Create a Directory and Navigate to the Directory

```
mkdir docker-assignment  
cd docker-assignment
```

Step 5: Clone the sample app from GitHub

```
git clone https://github.com/dockersamples/node-bulletin-board.git
```

Step 6: Navigate to the directory which has the Docker file

```
cd node-bulletin-board/bulletin-board-app
```

Step 7: Build the docker image using the docker file in the directory using the build command

```
sudo docker build .
```

Docker provides a consistent runtime environment for your application, which can help reduce issues caused by differences in operating systems or dependencies. This also makes it easier to reproduce and debug issues that occur in production.

Step 8: Verify Docker Image

```
exouser@docker-practice-amhaske:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>             15425711000c       2 hours ago        269MB
node                current-slim        d636cb0562fb       3 days ago         248MB
guacamole/guacamole <none>             a385e28f9fd6       8 months ago       642MB
guacamole/guacd-dev <none>             315a12ba560b       9 months ago       228MB
hello-world         latest             feb5d9fea6a5       17 months ago      13.3kB
exouser@docker-practice-amhaske:~$
```

When implementing a web application with Docker, some common difficulties that may arise include configuring the Dockerfile and managing directories correctly, managing dependencies, and ensuring the application runs smoothly within the container.

Step 9: Start the application to the local host

```
sudo docker run -d -p 8080:8080 15425711000c
```

Step 10: List the containers and verify the container is up and running

```
exouser@docker-practice-amhaske:~$ sudo docker container ls
CONTAINER ID   IMAGE                  COMMAND                  CREATED        STATUS        PORTS
2d39efc9d6ea   15425711000c          "docker-entrypoint.s..." About an hour ago Up About an hour   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp
terminated_lewin
39ad70e8fcd4   guacamole/guacamole   "/opt/guacamole/bin/..." 4 hours ago   Up 4 hours     0.0.0.0:49528->8080/tcp, :::49528->8080/tcp
guacamole_exo-guac-guacamole_1
5f6d14630e2a   guacamole/guacd-dev   "/bin/sh -c '/usr/lo..." 4 hours ago   Up 4 hours (healthy) 4822/tcp
guacamole_exo-guac-guacd_1
exouser@docker-practice-amhaske:~$
```

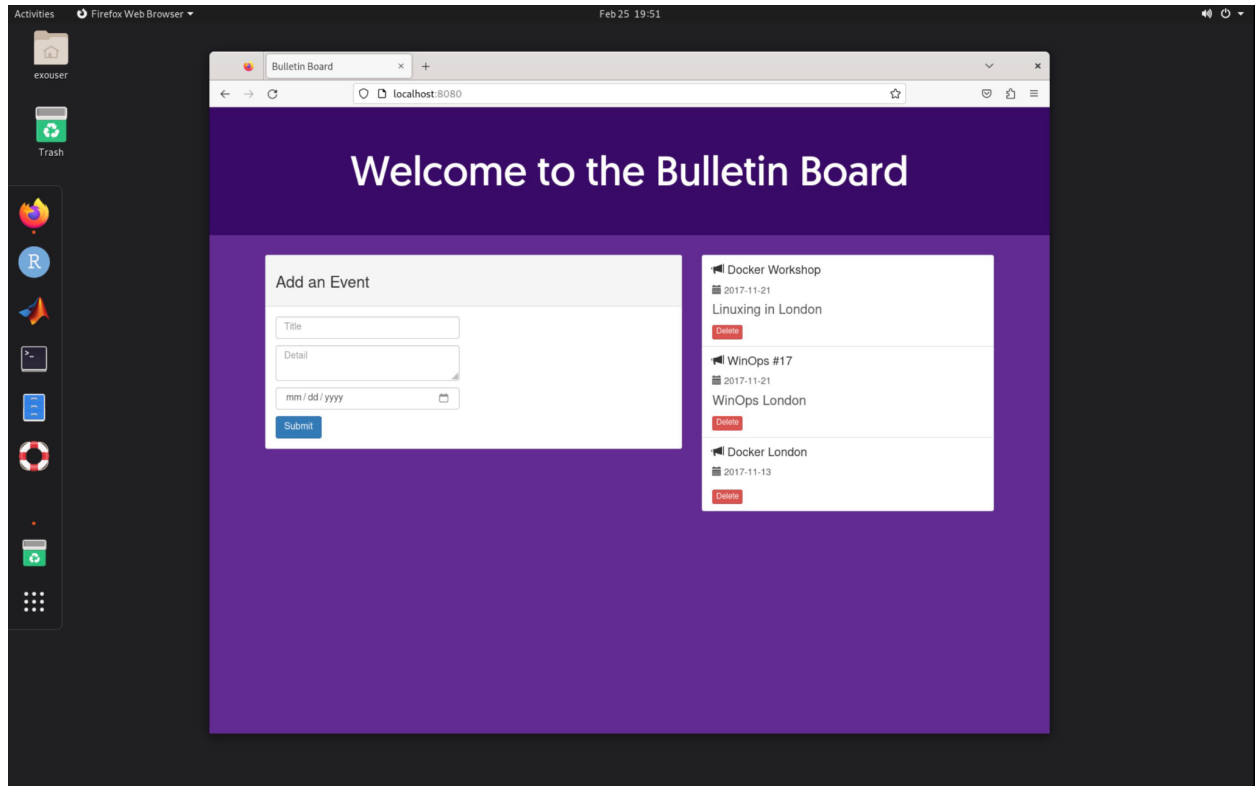
Other benefits rather than those mentioned above are:

1. Isolation and reproducibility of the environment
2. Portability across different machines and operating systems
3. Efficient resource utilization
4. Simplified deployment and scaling
5. Easy testing and debugging
6. Faster development cycles
7. Simplified management and maintenance.

Step 11: Deployed on the Local Host

Docker provides a consistent runtime environment for your application, which can help reduce issues caused by differences in operating systems or dependencies. This also makes it easier to reproduce and debug issues that occur in production.

```
wget http://localhost:8080/
```



- Any troubleshooting that you needed to do

My experience with Docker was good, I did take the help of TA

Issue: Connectivity of application and Local host

Solution: Reconstructed the application with a different directory, as there were some errors in managing directories