**5. Docker Basics**

1. List few benefits of docker

* Portability:

Docker allows you to package applications and their dependencies into a single container that can run consistently across different environments, such as development, testing, and production. This makes it easier to create, deploy, and run applications consistently across different platforms and reduces "it works on my machine" issues.

* Isolation:

Docker provides containerization, which allows applications to run in isolated environments with their own file systems, processes, and networking.

* Scalability:

Docker makes it easy to create, deploy, and manage multiple instances of containers, allowing applications to be scaled horizontally based on demand.

* Reproducibility:

Docker uses Dockerfiles, which are declarative scripts that specify the application dependencies and configuration. Docker images created from Dockerfiles are immutable and can be version-controlled, allowing for easy reproducibility of application environments across different stages of the software development lifecycle.

* Flexibility:

Docker provides a vast ecosystem of images and containers from Docker Hub, which is a public container image registry.

* Efficiency:

Docker uses containerization, which allows for efficient utilization of system resources by sharing the host OS kernel among containers.

* Collaboration:

Docker provides a standardized format for packaging and distributing applications, which makes it easy to share and collaborate on containerized applications with team members or across different environments.

1. Install docker
2. Check docker version and copy the output

Command: docker version

Client:

Cloud integration: v1.0.31

Version: 20.10.24

API version: 1.41

Go version: go1.19.7

Git commit: 297e128

Built: Tue Apr 4 18:28:08 2023

OS/Arch: windows/amd64

Context: default

Experimental: true

Server: Docker Desktop 4.18.0 (104112)

Engine:

Version: 20.10.24

API version: 1.41 (minimum version 1.12)

Go version: go1.19.7

Git commit: 5d6db84

Built: Tue Apr 4 18:18:42 2023

OS/Arch: linux/amd64

Experimental: false

containerd:

Version: 1.6.18

GitCommit: 2456e983eb9e37e47538f59ea18f2043c9a73640

runc:

Version: 1.1.4

GitCommit: v1.1.4-0-g5fd4c4d

docker-init:

Version: 0.19.0

GitCommit: de40ad0

1. Create a new java project with maven
2. Create a main class and print “Hello docker example”
3. Create a jar file for the project (inside target directory)
4. Run the generated jar file inside target directory with command line
5. Display the output

Text

Description automatically generated

1. Create a docker image for the java project. What is the command you used?

**Dockerfile:**

Text

Description automatically generated

**Command:**

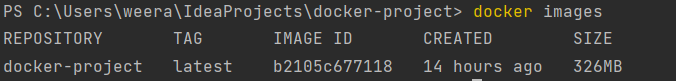
docker build -t docker-project .

1. List all the docker images and show output

**Command:**

docker images

**Output:**

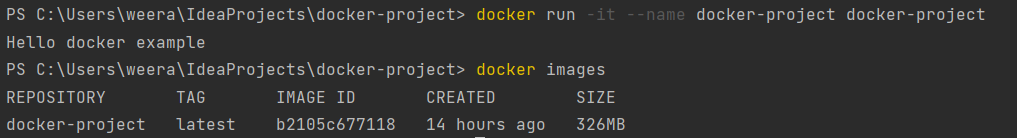


1. Run the created docker image. What is the command you used?

**Command:**

docker run -it –name docker-project docker-project

1. List all the docker images and show output

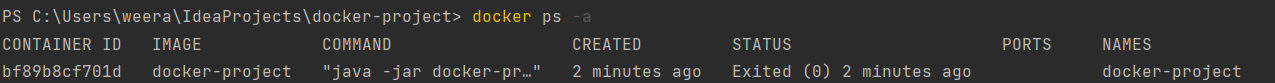


1. Stop the docker container?

**Command:**

docker stop bf89b8cf701d

1. List all the docker containers and show output



1. Remove the docker image. What is the command you used?

To remove the docker image, there should not be any container running based on it. If there is, it should be stopped and removed.

**Command:**

docker stop bf89b8cf701d

docker rm bf89b8cf701d

docker image rm docker-project

1. List all the docker images and show output

Graphical user interface, text

Description automatically generated

1. What is docker hub?

Docker hub is a cloud based registry that allows developers to store, share and distribute docker images. Docker images. It offers a wide range of official and community-contributed images, including popular operating systems, programming languages, databases, web servers, and other tools and applications.

1. Pull hello-world image from docker hub

**Command:**

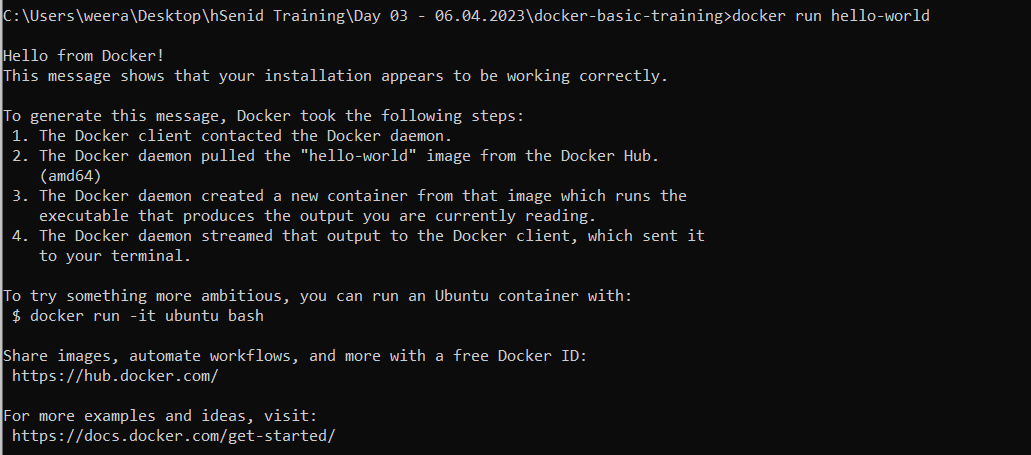
docker pull hello-world

1. Run hello-world image and show output

**Command:**

docker run hello-world

**Output:**

****

1. Pull and run mongodb as docker container

**Command:**

docker pull mongo

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

1. Open mongo shell

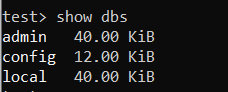
**Command:**

docker exec -it mongodb mongosh

1. List mongodb databases

**Command:**

show dbs



1. Add your codes and answer sheet to a directory named “docker-basic-training” and push it to your training github repository

