**5. Docker Basics**

1. List few benefits of docker

* Isolated and consistent environment

We use several languages, frameworks and dependency libraries for to create our application. So, each language, framework and dependency library have a version. Those versions can be different from a project to project. So, every time we need to setup our environment to run different projects. To solve this problem, we can create docker containers. Inside those containers it creates the environment for the project and there we can create several containers for different projects which need different environments. These containers are isolated from each other and we can run those projects without arising incompatible versioning problems.

* Increased portability

If we containerized our application using docker, we can deploy that docker container it easily on different operating systems and hardware platforms.

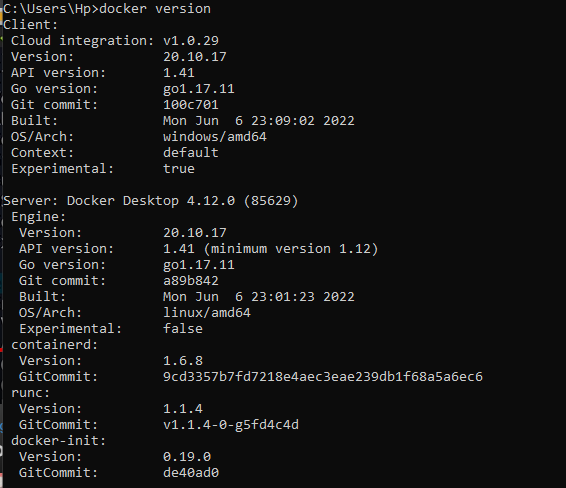
* Scalability and flexibility

Because of the consistent environment, docker images can easily sorted across multiple servers. If our application needed to update during the release, we can conveniently do the changes in docker containers and roll out new containers.

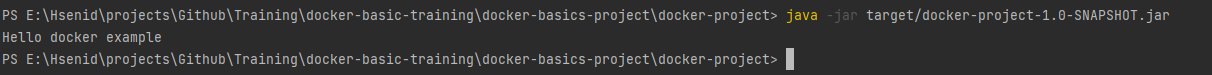
* Security

Because of the isolation of the containers, they cannot access other containers without having authorized access.

1. Check docker version and copy the output



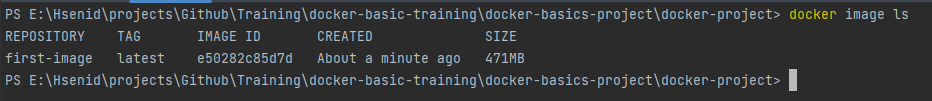
1. Output of generated jar file



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

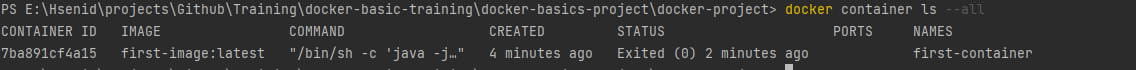
**docker build -t <image name> .**

1. List all the docker images and show output

****

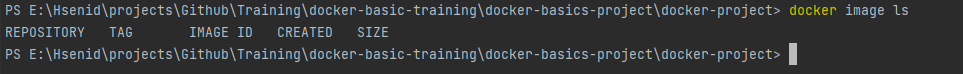
1. Run the created docker image. What is the command you used?

**docker run <image name>**

1. List all the docker containers and show output
2. Remove the docker image. What is the command you used?

**docker rm <image name>**

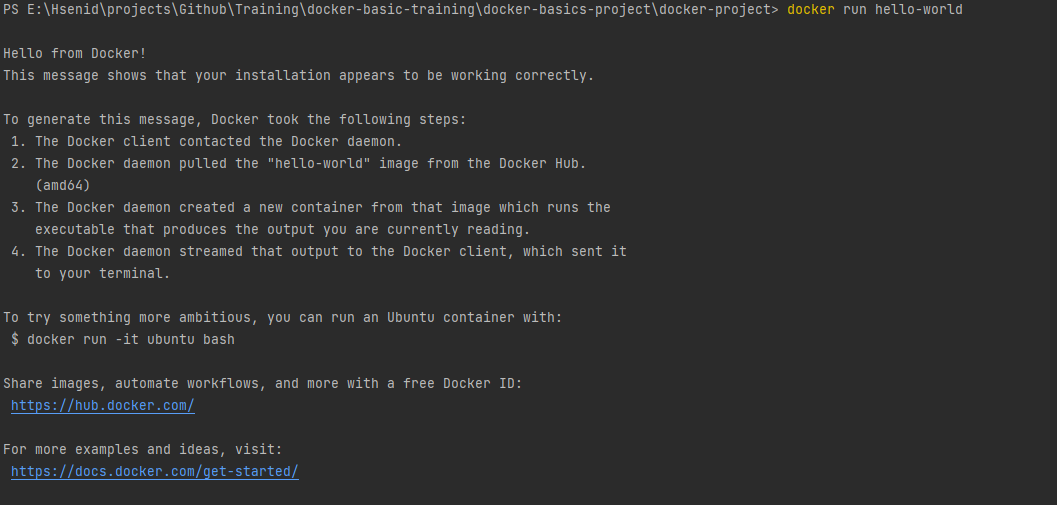
1. List all the docker images and show output

****

1. What is docker hub?

Docker hub is a hosted repository contained container images.

1. Run hello-world image and show output



1. Open mongo shell

Execute the container :

**docker exec -it <CONTAINER\_NAME> bash**

Open Mongo shell :

**mongosh**

1. List mongodb databases

