**5. Docker Basics**

1. List few benefits of docker

**Portability and consistency:** Docker containers can be run on any system with a Docker engine, ensuring that your applications run consistently and reliably on any platform**.**

**Isolation:** Docker containers run in isolation from each other and the host system, reducing the risk of conflicts and improving security.

**Scalability:** Docker makes it easy to scale applications by adding or removing containers, allowing you to handle increased demand.

**Automation:** Docker automates the deployment, testing, and publishing of applications, reducing the time and effort required to get your applications to market.

**Efficiency:** Docker containers are lightweight and share the host system's resources, reducing the resource overhead and increasing efficiency.

**Community:** Docker has a large and growing community of developers and users, providing support and resources for building, testing, and deploying applications.

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

**Docker version 20.10.21, build baeda1f**

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)

**To create a JAR file from a Maven project in IntelliJ IDEA, go to the Maven Tool Window (View → Tool Windows → Maven), expand your project in the tree, expand Lifecycle, and then double-click on package. Maven will compile your package, and the compiled JAR file will be written to the target/ directory**

1. Run the generated jar file inside target directory with command line
2. Display the output

**Hello World !**

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

**docker build -t <image\_name> .**

**docker build -t dockjava**

**<image\_name> is the desired name for the Docker image, and . specifies the current directory where the Dockerfile is located.**

1. List all the docker images and show output

**Docker images**

**Text

Description automatically generated**

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

**docker run <tagname>**

**docker run dockjava**

**docker run –name javacont dockjava**

1. List all the docker images and show output

Text

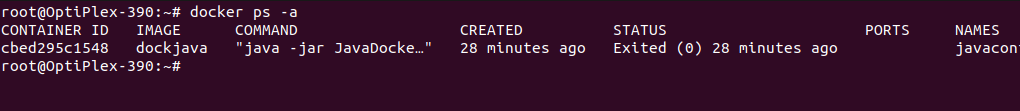
Description automatically generated

1. Stop the docker container?

**docker stop <container name>**

**docker stop javacont**

1. List all the docker containers and show output



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

**docker rmi dockjava**

**docker rm javacont**

1. List all the docker images and show output

Text

Description automatically generated

1. What is docker hub?

**Docker Hub is a cloud-based repository service provided by Docker Inc. that allows users to store and distribute Docker images. It provides a centralized platform for users to store, manage, and share Docker images, making it easier to collaborate with others and distribute software.**

1. Pull hello-world image from docker hub
2. Run hello-world image and show output

Text

Description automatically generated

1. Pull and run mongodb as docker container

**docker pull mongo**

**docker run --name cont-mongo -d mongo**

**This will pull the latest version of the MongoDB image from the Docker Hub repository and run it as a container named "cont-mongo". The** -d **option is used to run the container in the background.**

1. Open mongo shell

**docker exec -it cont-mongo bash**

**mongosh**

1. List mongodb databases

Text

Description automatically generated

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

https://github.com/Anuka-R98/Docker\_Java\_Basic

