

Docker Assignment:-1

Verify the installation by running a test container.

```
anil@IN-PG02P670:~$ docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:53cc4d415d839c98be39331c948609b659ed725170ad2ca8eb36951288f81b75
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
anil@IN-PG02P670:~$ 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/

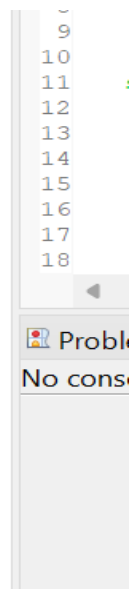
For more examples and ideas, visit:
https://docs.docker.com/get-started/

anil@IN-PG02P670:~$
```

- Create a simple web application (e.g., a Python Flask, DotNet, Java or Node application).

student [boot]

- > src/main/java
- > src/main/resources
- > src/test/java
- > JRE System Library [JavaSE-17]
- > Maven Dependencies
- target/generated-sources/annotations
- target/generated-test-sources/test-annotations
- > src
- > target
- docker-compose.yml
- Dockerfile
- HELP.md
- mvnw
- mvnw.cmd
- pom.xml



- Write a Dockerfile to containerize the application.

```
1 # Use OpenJDK image to run the application
2 FROM openjdk:17
3 WORKDIR /app
4 COPY --from=build /app/target/*.jar app.jar
5 ENTRYPOINT ["java", "-jar", "app.jar"]
6
```

- Write a docker-compose.yml file to define a multi-container application.

```

1 version: '3'
2 services:
3   mongo:
4     image: mongo:5.0
5     container_name: mongo
6     ports:
7       - 27017:27017
8     volumes:
9       - mongo-data:/data/db
10
11  springboot-app:
12    build: .
13    ports:
14      - 8085:8085
15
16    depends_on:
17      - mysql
18    environment:
19      SPRING_DATA_MONGODB_URI: mongodb://mongo:27017/student-management
20
21
22
23 volumes:
24   mongo-data:

```

- Use Docker commands to list, start, stop, and remove containers.

```

CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
226d62ae8ed9   anil647/java-project:latest         "java -jar app.jar"     5 minutes ago Up 5 minutes  0.0.0.0:8085->8085/tcp, :::8085->8085/tcp
anil@IN-PG02P670:~/ld/java-project/student$ docker rm 226d62ae8ed9
Error response from daemon: cannot remove container "/springboot-container": container is running: stop the container before removing or force remove
anil@IN-PG02P670:~/ld/java-project/student$ docker stop 226d62ae8ed9
226d62ae8ed9
anil@IN-PG02P670:~/ld/java-project/student$ docker rm 226d62ae8ed9
226d62ae8ed9
anil@IN-PG02P670:~/ld/java-project/student$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS        NAMES
anil@IN-PG02P670:~/ld/java-project/student$ docker ps --a

```

- Use Docker Compose to bring up the application and ensure all services are running correctly.

```

anil@IN-PG02P670:~/ld$ dir
Java8 java-project student
anil@IN-PG02P670:~/ld$ cd java-project
anil@IN-PG02P670:~/ld/java-project$ cd student
anil@IN-PG02P670:~/ld/java-project/student$ docker compose up --build
WARN[0000] /home/anil/ld/java-project/student/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Building 0.3s (9/9) FINISHED
=> [springboot-app internal] load build definition from Dockerfile                                docker:default
=> => transferring dockerfile: 213B                                                            0.0s
=> [springboot-app internal] load metadata for docker.io/library/openjdk:17                    0.0s
=> [springboot-app internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                                    0.0s
=> [springboot-app internal] load build context                                                  0.0s
=> => transferring context: 94B                                                                    0.0s
=> [springboot-app 1/3] FROM docker.io/library/openjdk:17                                       0.0s
=> CACHED [springboot-app 2/3] WORKDIR /app                                                       0.0s
=> CACHED [springboot-app 3/3] COPY target/student-management-0.0.1-SNAPSHOT.jar app.jar        0.0s
=> [springboot-app] exporting to image                                                            0.0s
=> => exporting layers                                                                              0.0s
=> => writing image sha256:c7cf59cee2001e35919c647ce82d4fb5abellcc7c790719edf3a3c074665dad2    0.0s
=> => naming to docker.io/library/student-springboot-app                                         0.0s
=> [springboot-app] resolving provenance for metadata file                                         0.0s
[+] Running 2/2
  ✓ Container mongo                                     Recreated                                0.1s
  ✓ Container student_springboot-app_1                  Recreated                                0.1s
Attaching to mongo, springboot-app-1
mongo | {"t":{"$date":"2024-08-29T01:46:29.168+00:00"},"s":"I",  "c":"NETWORK",  "id":4915701, "ctx":"-", "msg":"Initializ
ed wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"m
inWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true}}}
mongo | {"t":{"$date":"2024-08-29T01:46:29.177+00:00"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":"-", "msg":"Automatic

```

```

springboot-app-1 | 2024-08-29T01:46:30.831Z INFO 1 --- [student-project] [main] c.n.s.StudentManagementApplication
: Starting StudentManagementApplication v0.0.1-SNAPSHOT using Java 17.0.2 with PID 1 (/app/app.jar started by root in /app)
springboot-app-1 | 2024-08-29T01:46:30.836Z INFO 1 --- [student-project] [main] c.n.s.StudentManagementApplication
: No active profile set, falling back to 1 default profile: "default"
springboot-app-1 | 2024-08-29T01:46:31.835Z INFO 1 --- [student-project] [main] .s.d.r.c.RepositoryConfigurationDelegate
: Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
springboot-app-1 | 2024-08-29T01:46:31.926Z INFO 1 --- [student-project] [main] .s.d.r.c.RepositoryConfigurationDelegate
: Finished Spring Data repository scanning in 85 ms. Found 1 MongoDB repository interface.
springboot-app-1 | 2024-08-29T01:46:32.245Z WARN 1 --- [student-project] [main] trationDelegate$BeanPostProcessorChecker
mongo | {"t":{"$date":"2024-08-29T01:46:33.297+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn1","msg":"client
metadata", "attr":{"remote":"172.19.0.3:38222","client":"conn1","negotiatedCompressors":[],"doc":{"driver":{"name":"mongo-java-drive
r|sync|spring-boot","version":"5.0.1"},"os":{"type":"Linux","name":"Linux","architecture":"amd64","version":"5.10.102.1-microsoft-sta
ndard-WSL2"},"platform":"Java/Oracle Corporation/17.0.2+8-66","env":{"container":{"runtime":"docker"}}}}}
mongo | {"t":{"$date":"2024-08-29T01:46:33.297+00:00"},"s":"I", "c":"NETWORK", "id":51800, "ctx":"conn2","msg":"client
metadata", "attr":{"remote":"172.19.0.3:38224","client":"conn2","negotiatedCompressors":[],"doc":{"driver":{"name":"mongo-java-drive
r|sync|spring-boot","version":"5.0.1"},"os":{"type":"Linux","name":"Linux","architecture":"amd64","version":"5.10.102.1-microsoft-sta
ndard-WSL2"},"platform":"Java/Oracle Corporation/17.0.2+8-66","env":{"container":{"runtime":"docker"}}}}}
springboot-app-1 | 2024-08-29T01:46:33.318Z INFO 1 --- [student-project] [l1]-mongo:27017] org.mongodb.driver.cluster
: Monitor thread successfully connected to server with description ServerDescription{address=mongo:27017, type=STANDALONE, state=CON
NECTED, ok=true, minWireVersion=0, maxWireVersion=13, maxDocumentSize=16777216, logicalSessionTimeoutMinutes=30, roundTripTimeNanos=3
1093609}
springboot-app-1 | 2024-08-29T01:46:34.070Z INFO 1 --- [student-project] [main] o.s.b.w.embedded.tomcat.TomcatWebServer
: Tomcat started on port 8085 (http) with context path '/'
springboot-app-1 | 2024-08-29T01:46:34.110Z INFO 1 --- [student-project] [main] c.n.s.StudentManagementApplication
: Started StudentManagementApplication in 4.05 seconds (process running for 4.928)

```

- **Create a private Docker registry or use Docker Hub.**

A screenshot of the Docker Hub web interface. The top navigation bar is blue with the Docker Hub logo, links for Explore, Repositories (underlined), and Organizations, a search bar, and utility icons. The breadcrumb trail shows 'anil647 / Repositories / java-project / General'. The 'General' tab is selected. The repository name 'anil647/java-project' is shown with a lock icon. It indicates the repository was created about 8 hours ago and lacks a description and category. On the right, the 'Docker commands' section shows the command 'docker push anil647/java-project:tagname'.

- **Push your Docker images to the registry.**

anil647

Repositories

java-project

Tags

Using 1 of 1 private repositories.

General

Tags

Builds

Collaborators

Webhooks

Settings

☐ Sort by

Newest

Filter Tags

Q

Delete

TAG

latest

Last pushed a few seconds ago by anil647

docker pull anil647/java-project:latest

Copy

Digest	OS/ARCH	Last pull	Compressed Size
fbf94cd4e31a	linux/amd64	---	256.8 MB

```

anil@IN-PG02P670:~/ld/java-project/student$ docker build -t anil647/java-project .
[+] Building 0.2s (8/8) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 213B
=> [internal] load metadata for docker.io/library/openjdk:17
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/openjdk:17
=> [internal] load build context
=> => transferring context: 94B
=> CACHED [2/3] WORKDIR /app
=> CACHED [3/3] COPY target/student-management-0.0.1-SNAPSHOT.jar app.jar
=> exporting to image
=> => exporting layers
=> => writing image sha256:ed52a1df40feef0ba55927f588d05ca82455ba53c593cf33d8f65749c8bcd9c8
=> => naming to docker.io/anil647/java-project
anil@IN-PG02P670:~/ld/java-project/student$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/anil/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
anil@IN-PG02P670:~/ld/java-project/student$ docker push anil647/java-project
Using default tag: latest
The push refers to repository [docker.io/anil647/java-project]
74fc293ad3ac: Pushed
06c118e9322d: Layer already exists
dc9fa3d8b576: Layer already exists
27ee19dc88f2: Layer already exists
c8dd973c66670: Layer already exists
latest: digest: sha256:fbf94cd4e31af2b0a6ee65fb2bb8015380427ad46d0f478535260902f29145f4 size: 1373

```

- Pull the images from the registry and run them locally.

```

anil@IN-PG02P670:~/ld/java-project/student$ docker pull anil647/java-project:latest
latest: Pulling from anil647/java-project
Digest: sha256:fbf94cd4e31af2b0a6ee65fb2bb8015380427ad46d0f478535260902f29145f4
Status: Image is up to date for anil647/java-project:latest
docker.io/anil647/java-project:latest
anil@IN-PG02P670:~/ld/java-project/student$ docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
student-springboot-app latest         c7cf59cee200   8 hours ago    501MB
anil647/java-project latest         ed52a1df40fe   8 hours ago    501MB
student_springboot-app latest         70849db8a4a2   10 hours ago   501MB
mysql               latest         a82a8f162e18   5 weeks ago    586MB
mongo               5.0            642d31cc2d52   6 weeks ago    724MB
hello-world         latest         d2c94e258dcb   16 months ago  13.3kB
openjdk             17            5e28ba2b4cdb   2 years ago    471MB
anil@IN-PG02P670:~/ld/java-project/student$ docker build --it ed52a1df40fe
unknown flag: --it
See 'docker buildx build --help'.
anil@IN-PG02P670:~/ld/java-project/student$ docker build ed52a1df40fe
[+] Building 0.0s (0/0)
ERROR: unable to prepare context: path "ed52a1df40fe" not found
anil@IN-PG02P670:~/ld/java-project/student$ docker run -d --name springboot-container -p 8085:8085 anil647/java-project:latest
226d62ae8ed9a942d7dc173773d8a55596ca40cbcb4a3874f32e02ed979b1f64
anil@IN-PG02P670:~/ld/java-project/student$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
226d62ae8ed9   anil647/java-project:latest         "java -jar app.jar"      34 seconds ago Up 33 seconds  0.0.0.0:8085->8085/tcp, :::8085->8085/tcp
springboot-container

```

Bonus Question

- Extend the multi-container application to include a database service and configure the web application to interact with the database.

```
1 version: '3'
2 services:
3   mongo:
4     image: mongo:5.0
5     container_name: mongo
6     ports:
7       - 27017:27017
8     volumes:
9       - mongo-data:/data/db
10
11   springboot-app:
12     build: .
13     ports:
14       - 8085:8085
15
16     depends_on:
17       - mysql
18     environment:
19       SPRING_DATA_MONGODB_URI: mongodb://mongo:27017/student-management
20
21
22
23 volumes:
24   mongo-data:
```