

## hSenid Training – 6.2 Containerization

### Docker exercise commands used

- *Install Docker*
- *Create a new Java project with Maven*
- *Create a main class and print “Hello docker example”*
- *Create a jar file for the project (inside target directory) -*

*Command – mvn package*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< org.example:Docker-exercise >-----
[INFO] Building Docker-exercise 1.0-SNAPSHOT
[INFO]   from pom.xml
[INFO] -----[ jar ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Docker-exercise ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ Docker-exercise ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise\target\classes
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ Docker-exercise ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ Docker-exercise ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Docker-exercise ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Docker-exercise ---
[INFO] Building jar: C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise\target\Docker-exercise-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] -----
[INFO] Total time:  3.087 s
[INFO] Finished at: 2024-07-08T09:42:03+05:30
```

- *Run the generated jar file inside target directory with command line*

*Command – cd .\target\*

*java -jar Docker-exercise-1.0-SNAPSHOT.jar*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise\target> java -jar Docker-exercise-1.0-SNAPSHOT.jar
Hello docker example
```

- Create a docker image for the java project.
- Run the created docker image.

Command - `docker build -t docker-example .`

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> docker build -t docker-example .
[*] Building 41.1s (9/9) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 223B
=> [internal] load metadata for docker.io/library/openjdk:21
=> [auth] library/openjdk:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/3] FROM docker.io/library/openjdk:21@sha256:af9de795d1f8d3b6172f6c55ca9ba1c5768baa11bb2dc8af7045c7db9d4c33ac
=> => resolve docker.io/library/openjdk:21@sha256:af9de795d1f8d3b6172f6c55ca9ba1c5768baa11bb2dc8af7045c7db9d4c33ac
=> => sha256:079114de2be199f2ae0f7766ac0187d24a0c3a2d658fc51bffc6af5b8bd85469 4.42kB / 4.42kB
=> => sha256:af9de795d1f8d3b6172f6c55ca9ba1c5768baa11bb2dc8af7045c7db9d4c33ac 1.04kB / 1.04kB
=> => sha256:c67f402f77197f2e6ae84ff1fca868699ce3b38bfa78604524051420fa2e4383 954B / 954B
=> => sha256:0eab4e2287a59db0ae2d401e107a120e21ac3a291b097faffb1af38a1bc773c 15.03MB / 15.03MB
=> => sha256:5262579e8e45cb87fdc8fb6182d30da3c9e4f1036e02223508f287899ea434c0 44.96MB / 44.96MB
=> => sha256:7c002e8f606286a649b6f6cc6420c9056f7d3075fe3094b9cc33a715ff609335 203.93MB / 203.93MB
=> => extracting sha256:5262579e8e45cb87fdc8fb6182d30da3c9e4f1036e02223508f287899ea434c0 2.0s
=> => extracting sha256:0eab4e2287a59db0ae2d401e107a120e21ac3a291b097faffb1af38a1bc773c 0.5s
=> => extracting sha256:7c002e8f606286a649b6f6cc6420c9056f7d3075fe3094b9cc33a715ff609335 2.0s
=> [internal] load build context
=> => transferring context: 2.46kB
=> [2/3] WORKDIR /app
=> [3/3] COPY target/Docker-exercise-1.0-SNAPSHOT.jar /app/Docker-exercise-1.0-SNAPSHOT.jar
=> => exporting to image
=> => exporting layers
=> => writing image sha256:1bab14f26eb5b6a7d285c4b8a1d3162629ed1cfd5ea2e3c10785769a6c0e9916
=> => naming to docker.io/library/docker-example
View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/mrst42o4hv98wLz9awakzqe9
What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview
```

- List all the docker images and show output

Command – `docker images`  
`docker image ls`

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
docker-example      latest          1bab14f26eb5   2 minutes ago  504MB
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> docker image ls
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
docker-example      latest          1bab14f26eb5   5 minutes ago  504MB
```

- Remove the docker image.

Command - `docker rmi 1bab14f26eb5`

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> docker rmi 1bab14f26eb5
Untagged: docker-example:latest
Deleted: sha256:1bab14f26eb5b6a7d285c4b8a1d3162629ed1cfd5ea2e3c10785769a6c0e9916
```

- *List all the docker images and show output*

*Command – docker images*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> docker images ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

- *Pull hello-world image from docker hub*

*Command – docker pull hello-world*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:94323f3e5e09a8b9515d74337010375a456c909543e1ff1538f5116d38ab3989
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview hello-world
```

- *Run hello-world image and show output*

*Command - docker run hello-world*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docker\Docker-exercise> 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/
```

- *List all the docker images and show output*

*Command – docker images*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docke\Exercise> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
hello-world	latest	d2c94e258dcb	14 months ago	13.3kB

- *Pull and run mongodb as docker container*

*Command – docker pull mongo*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docke\Exercise> docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
3713021b0277: Pull complete
39bdcacccd97: Pull complete
d6b691142508: Pull complete
bcc1924dee6d: Pull complete
091a7990873d: Pull complete
77e5254f6ae8: Pull complete
403f753f5920: Pull complete
88cd53ea307c: Pull complete
Digest: sha256:1cd3951000020c1cb1757868e6cfd82667f57d80bb31fed8b585e26a8a1d960f
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker scout quickview mongo
```

*Command - docker run --name testmongo -d -p 27017:27017 mongo*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docke\Exercise> docker run --name testmongo -d -p 27017:27017 mongo
aa2105fa95388231541a94ac9c7301c518d96f8657c390a2701f14b0e54ac234
```

- *Open mongo shell*

*Command - docker exec -it testmongo mongosh*

```
PS C:\Users\altha\OneDrive\Desktop\Althaf\Hsenid\Docke\Exercise> docker exec -it testmongo mongosh
Current Mongosh Log ID: 668b8f911735eff436149f47
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.10
Using MongoDB:      7.0.12
Using Mongosh:      2.2.10

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
  The server generated these startup warnings when booting
  2024-07-08T06:15:34.180+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
  2024-07-08T06:15:35.389+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
  2024-07-08T06:15:35.389+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never' in this binary version
  2024-07-08T06:15:35.390+00:00: vm.max_map_count is too low
  -----
test>
```

- *List mongodb databases*

*Command – show databases*

```
test> show databases
admin    40.00 KiB
config   72.00 KiB
local    40.00 KiB
test>
```