Docker Udemy

- [ ] `TLS handshake timeout` error
    - [ ] I've encountered many times on push and pull of images
    - [ ] see **attached file** (Handshake timeout example (on push)) with console output on **push attempts**
        - [ ] in this example it managed to push 5 out of 6 images without trying to push the unpushed image again
        - [ ] when executed push-all-images-to-docker-hub.sh script in commit 8ca2b4e used when I reproduced clickup subtash 13 (previous one)

.sh scripts

- [ ] attempts-until-pull-or-push-all-images.sh
    - [ ] on image push make sure all of them are push and for the ones that failed try to push them again
    - [ ] if image not pushed/pulled from docker hub try again until it's pushed/pulled using if and white statements inside .sh scripts - cause `TLS handshake timeout`

```
docker compose -f ../docker-compose.yml up -d
```

- [ ] openjdk:8-jdk-alpine image is needed for mvn package (that build the spring boot project images on local docker) so in the script it's pulled before mvn package
- [ ] scripts to clean up my remote Docker Hub remote
    - [ ] tried delete-one-image-from-docker-hub.sh
        - [ ] it doesn't work but useful to see how args or flags are passed inside an .sh script
    - [ ] see commit below, so the commit before (with hash 533e667719c587c58fe1de8c1254cdb58a37c5a6) is the last one containing the folder with these scripts

```
b06e8d1cd064abfc70eba548f03b81986437a422 (HEAD -> master, origin/master) Deleted not-working-delete-from-
docker-hub folder. Need to try these delete sctipts some other time (to clean up my remote Docker hub repo).
```

Script Examples

- [ ] create .sh file
- [ ] single and multiple line comments
- [ ] if and white statements
- [ ] Using Functions: Parameters and Return Values
- [ ] list all files (inside current folder)

```
ls -l
```

- [ ] chmod -x before run
- [ ] implement scripts for
    - [ ] no fail pull (when/before docker compose file run)
    - [ ] no fail push
- [ ] Create an array from a text file in Bash
    - [ ] get it's size and iterate

```
my_len=${#myArray[@]}
for (( i = 0 ; i < my_len ; i++))
do
    echo "Element [$i]: ${myArray[$i]}"
done
```

Scripts usage in git bash cli:

- [ ] that work

```
$ ./clean-local-docker-env.sh
$ ./attempts-until-pull-or-push-all-images.sh -a push
$ ./attempts-until-pull-or-push-all-images.sh -a project_pull
$ ./attempts-until-pull-or-push-all-images.sh -a official_pull
$ ./restart-all-containers-locally-from-scratch.sh
$ ./restart-all-containers-locally-from-scratch.sh -a no_maven_build
```

- ☐ that will fail with a message

```
$ ./attempts-until-pull-or-push-all-images.sh
```

Flows or scenarios (3)

<u>(1)Push 2 images test</u>

- ☐ delete all images in remote Docker Hub repo https://hub.docker.com/repositories (from the cosdin account)
- ☐ open git bash in the folder `C:\javaDev\workspace\07-micro-1\sh-scripts-and-mvn-aggregate` or `/c/javaDev/workspace/07-micro-1/sh-scripts-and-mvn-aggregate` as listed in the git bash cli
- ☐ run in git bash: clean local, this stops running containers, deletes them, deletes all images and used networks on local Docker

```
$ ./clean-local-docker-env.sh
```

- ☐ create 2 images (only): openjdk and micro-1 images by opening git bash to `/c/javaDev/workspace/07-micro-1` and run

```
mvn clean package -DskipTests
```

- ☐ run in git bash: push the 2 images (only) to the remote Docker Hub repo, for the images that don't exist on local Docker but the names are in the projects_image_arr array in the script the "Image not exists locally so cannot push it." message is printed in cli

```
$ ./attempts-until-pull-or-push-all-images.sh -a push
```

<u>(2)Start containers **with maven build**</u>

- ☐ Run in git bash

```
$ ./restart-all-containers-locally-from-scratch.sh
```

  - ☐ steps
    1. cleans the local Docker environment
    2. pulls the **official** images from Docker Hub by calling `./attempts[...]` script which does attempt when pulling images one by one, if one attempt fails due to bad connection (`TLS handshake timeout` error that I've encountered many times on push and pull of images explicitly without this script or when docker compose pulls the images it needs for the containers (see step 4)) it will do a new attempt until it succeeds
    3. runs the maven build to create the **spring boot project** Docker images locally, this needs the image `openjdk:8-jdk-alpine` pulled in step 2 (because it's in the Dockerfile of all of the 6 spring boot projects)
    4. starts the containers calling the docker compose file, in this step all the necessary images should be present locally:
       - ☐ the official images downloaded in step 2
       - ☐ the spring boot projects images build in step 3
  - ☐ includes the calls:

```
./clean-local-docker-env.sh
[...]
./attempts-until-pull-or-push-all-images.sh -a official_pull
[...]
mvn clean package -DskipTests
[...]
docker compose -f ../docker-compose.yml up -d
```

  - ☐ look in Docker Desktop
    - ☐ there are 11 images with the "in use" status and the openjdk image appers with no status
    - ☐ all 11 containers are green (running)
  - ☐ refresh the Docker Hub remote repo
    - ☐ in the cosdin account delete all images from the page https://hub.docker.com/repositories
    - ☐ push all images from local Docker on Docker Hub remote repo

```
$ ./attempts-until-pull-or-push-all-images.sh -a push
```

- ☐ in the page https://hub.docker.com/repositories you should find the 6 **spring boot project** images built with `mvn clean package -DskipTests` previously and pushed with `./attempts-until-pull-or-push-all-images.sh -a push`

(3)Start containers **without maven build**

☐ Run in git bash

```
$ ./restart-all-containers-locally-from-scratch.sh -a no_maven_build
```

- ☐ steps
  1. cleans the local Docker environment
  2. pulls the 6 **spring boot project** images from remote Docker Hub repo
  3. pulls the **official** images from Docker Hub
  4. starts the containers calling the docker compose file
- ☐ includes the calls:

```
./clean-local-docker-env.sh
[...]
./attempts-until-pull-or-push-all-images.sh -a project_pull
[...]
./attempts-until-pull-or-push-all-images.sh -a official_pull
[...]
docker compose -f ../docker-compose.yml up -d
```

☐ all 11 containers get restarted on docker or laptop restart because there is `restart: always` in the docker compose file

Last commits:

☐ log pretty (remove `-n4` to print all commts not only the last 4 commits only)

```
$ git log --graph --pretty=format:'%C(auto)%h%d (%cr) %cn <%ce> %s' --all -n4
```

☐ 07-micro-1:

```
8227873 (origin/master, master) (23 hours ago) Cosmin Dinu <cdcdd15@gmail.com> Put together one script for repo
basic (status) info for all 6 microservice git repos.
```