Docker Udemy

- □ Legat de subtask Side projects on Dell laptop #j35q10
- □ Ignite server nodes as Docker containers documentation
  - □ https://apacheignite.readme.io/docs/docker-deployment
  - □ https://ignite.apache.org/docs/latest/installation/installing-using-docker
- □ lists are empty

```
docker network list
docker image list
docker container list -a
```

- □ Create new Docker network

```
docker network create spring-ignite-net
```

- □ Start Ignite server container in Windows CMD (not in Git Bash)

```
docker run -d --network spring-ignite-net --publish 10800:10800 -p 8080:8080 -p 42112:42112 -p 47500:47500 -p
47100:47100 -v C:/javaDev/libs/:/opt/ignite/apache-ignite/libs/user_libs -v C:/javaDev/apache-
ignite/examples/config:/ignite-configs -e CONFIG_URI=/ignite-configs/example-ignite.xml
apacheignite/ignite:2.7.5
```

- □ Create new Maven Spring Boot (client) Project
  - □ name 06-spring-boot-ignite
  - □ Replace "mains" with REST endpoints
  - □ mvn package to create docker image
- □ Start container from spring boot ignite client project
  - □ From image cosdin/spring-ignite:0.0.1-SNAPSHOT (created with mvn package with the docker maven plugin)
  - □ With -v to mount the ignite config xml into the container and -e to set or override the path to xml from local machine with the one in the container
  - □ For this container I used the same config xml mounted for the ignite server node container from the local filesys from C:/javaDev/apache-ignite/examples/config:/ignite-configs

```
docker run -d -p 5000:5000 --network=spring-ignite-net -v C:/javaDev/apache-ignite/examples/config:/ignite-
configs -e IGNITE_CONFIG_XML_PATH=/ignite-configs/example-ignite.xml cosdin/spring-ignite:0.0.1-SNAPSHOT
```

- □ Spring boot ignite client project 06-spring-boot-ignite performs all crud (against an ignite server node started as docker container) (the path to ignite config xml is different between the 2 set from the IGNITE_CONFIG_XML_PATH property from application.properties file)
  - □ started as spring boot project from CMD as usual
  - □ started as docker container
- □ Create Gradle libs project containing the entity class only
  - □ necessary for Ignite server node container classpath
  - □ name 06-spring-boot-ignite-libs-gradle
  - □ sync with git
- □ Start 2 Docker containers in new Docker network
  - □ 1 Ignite server nodes
  - □ 1 Spring Boot Ignite client
- □ Check the 2 containers are present in the Docker network

```
docker inspect spring-ignite-net
```

- □ Ignite docker containers are started from an image with specific version apacheignite/ignite:2.7.5, version 2.7.5 is used in the Ignite client libs from the Java projects
- □ View records with DBeaver in Ignite distributed cache
- □ Add files from local machine filesys in docker container filesys with -v. In this case used to mount ignite config xml and to .jar user libraries
- □ Explore filesys of docker container (check if ignite xml config file and .jar are mounted into container)

```
docker exec -it zealous_buck bash
docker exec -it zealous_buck sh
```

- □ exit container filesys explore - use the escape sequence CTRL+P followed by CTRL+Q
- □ start server node locally from cmd

```
ignite.bat C:/javaDev/apache-ignite/examples/config/example-ignite.xml
```

- □ How to Fix java.lang.UnsupportedClassVersionError thrown in Eclipse and in Ignite container when loading class from custom classpath
  - □ for Eclipse: went to Windows - Preferences - Java - Compiler - Compiler compliance level from 12 to 1.8

```
java.lang.UnsupportedClassVersionError: ignite/spring/EmployeeDTO has been compiled by a more recent version of
the Java Runtime (class file version 56.0), this version of the Java Runtime only recognizes class file
versions up to 52.0]
```

  - □ For Gradle project in build.gradle

```
apply plugin: 'java'
compileJava {
    sourceCompatibility = '1.8'
    targetCompatibility = '1.8'
}
```

- □ remove all containers, images and created custom network