**# Lightstreamer Container**

docker run -d --name ls-server -p 80:8080 lightstreamer:latest

**# Ghost Container**

docker run -d --name ghost-container -p 3001:2368 -e NODE\_ENV=development ghost:latest

**# Apache HTTP Server Container**

**# cd in new folder then run the command:**

docker run --name my-apache-app -p 8080:80 -d -v ${PWD}:/usr/local/apache2/htdocs/ httpd:latest

**# SQL Server Container**

docker run -e ACCEPT\_EULA=Y --name sql -e MSSQL\_SA\_PASSWORD=yourStrongPassword12# -p 1433:1433 -d -v sqldata:/var/opt/mssql mcr.microsoft.com/mssql/server

**\*MariaDB Client and Server in a Network**

docker run -d --network mariadb\_network --name mariadb\_server --env MARIADB\_USER=mariadb\_user --env MARIADB\_PASSWORD=mariadb\_password --env MARIADB\_ROOT\_PASSWORD=mariadb\_root\_password mariadb:latest

docker run -d --network mariadb\_network --name mariadb\_client -e APP\_DB\_HOST=mariadb -e APP\_DB\_USER=mariadb\_user -e APP\_DB\_PASSWD=mariadb\_password -e MARIADB\_ROOT\_PASSWORD=mariadb\_root\_password mariadb:latest

# open container

docker exec -it mariadb\_client bash

# grant access in db

mariadb -h mariadb\_server -u maria\_user -p

# in db

SELECT VERSION();

**Tracker App**

1 - docker login

2 - docker push georgilukanov87/track-app

3 - docker build -t georgilukanov87/track-app

4 - docker build -t georgilukanov87/track-app .

5 - docker push georgilukanov87/track-app

6 - docker run -d --name track-app -p 8080:80 georgilukanov87/track-app

**TaskBoard App: Connect Containers in a Network**

# create network

1 - docker network create taskboard\_network

#run server. NAME and PASSWORD are IMPORTANT!

2 - docker run -e ACCEPT\_EULA=Y -e MSSQL\_SA\_PASSWORD=yourStrongPassword12# -p 1433:1433 -v sqldata:/var/opt/mssql --network taskboard\_network --name sqlserver -d mcr.microsoft.com/mssql/server

# change settings in appsetings.json in /TaskBoard.WebApp

3 - "ConnectionStrings": {

"DefaultConnection": "Server=sqlserver;Database=MyDB;User Id=sa;Password=yourStrongPassword12#;MultipleActiveResultSets=true"

}

#build image locally

4 - docker build . -f ./TaskBoard.Webapp/Dockerfile -t georgilukanov87/taskboard\_app

# run both together! check with : docker network inspect taskboard\_network

5 - docker run -p 5000:80 --name web\_app --network

taskboard\_network georgilukanov87/taskboard\_app

# test <http://localhost:5000/>

dsada

**TODO App**

docker network create react-express

docker network create express-mongo

docker build frontend -f frontend/Dockerfile -t frontend

docker build backend -f backend/Dockerfile -t backend

docker run -d --name frontend -p 3000:3000 --network react-express frontend

docker run -d --name backend --network react-express backend

docker network connect backend express-mongo

docker run -d --name mongo --network express-mongo -v ./data:/data/db mongo:latest