

به نام خدا

پروژه رایانش ابری

دانشجو: امیر حسین محمدیان
استاد: دکتر طاهری

سیستم عامل استفاده شده arch linux می باشد.
برای نصب داکر روی ارج ابتدا سورس ان را دانلود میکنیم سپس نصب میکنیم .
دستورات لازم در زیر آمده :

```
wget https://download.docker.com/linux/static/stable/x86_64/docker-27.4.0.tgz -qO- | tar xvfz  
- docker/docker --strip-components=1
```

```
mv ./docker /usr/local/bin
```

```
sudo pacman -U ./docker-desktop-x86_64.pkg.tar.zst
```

بعد از نصب برای تست اولیه ان دستور زیر را وارد میکنیم و چیزی طبق تصویر زیر مشاهده می کنیم:

docker run hello-world

```
[amir@myarch ~]$ 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/
```

برای اینکه بعد از هر لاگین در سیستم داکر فعال شود نیاز است ان را در systemctl فعال کنیم برای این کار دستورات زیر را وارد میکنیم:

```
systemctl --user start docker-desktop  
systemctl --user enable docker-desktop
```

برای ادامه کار به docker compose نیز نیاز است که بطور خودکار نصب می شود که با دستور زیر نصب بودن آن را چک میکنیم :

```
docker compose version
```

در صورت نصب نبودن آن را طبق دستورات زیر نصب میکنیم:

```
sudo curl -L "https://github.com/docker/compose/releases/download/v2.18.0/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

```
sudo chmod +x /usr/local/bin/docker-compose
```

```
docker-compose --version
```

حالا یک پروژه CRUD با node js ایجاد می کنیم به این صورت :

```
mkdir programming-languages-crud  
cd programming-languages-crud
```

```
npm init -y
```

```
[amir@myarch ~]$ mkdir programming-languages-crud  
[amir@myarch ~]$ cd programming-languages-crud/  
[amir@myarch programming-languages-crud]$ npm init -y  
(node:5884) ExperimentalWarning: CommonJS module /usr/lib/node_modules/npm/node_modules/debu  
es/npm/node_modules/supports-color/index.js using require().  
Support for loading ES Module in require() is an experimental feature and might change at an  
(Use `node --trace-warnings ...` to show where the warning was created)  
Wrote to /home/amir/programming-languages-crud/package.json:  
  
{  
  "name": "programming-languages-crud",  
  "version": "1.0.0",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "keywords": [],  
  "author": "",  
  "license": "ISC",  
  "description": ""  
}
```

```
npm install express mongoose body-parser cors  
npm install --save-dev nodemon
```

```
[amir@myarch programming-languages-crud]$ npm install express mongoose body-parser cors
(node:6053) ExperimentalWarning: CommonJS module /usr/lib/node_modules/npm/node_modules/debug/src/node
es/npm/node_modules/supports-color/index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

added 91 packages, and audited 92 packages in 27s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
[amir@myarch programming-languages-crud]$ npm install --save-dev nodemon
(node:6083) ExperimentalWarning: CommonJS module /usr/lib/node_modules/npm/node_modules/debug/src/node
es/npm/node_modules/supports-color/index.js using require().
Support for loading ES Module in require() is an experimental feature and might change at any time
(Use `node --trace-warnings ...` to show where the warning was created)

added 29 packages, and audited 121 packages in 12s

19 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
[amir@myarch programming-languages-crud]$
```

و بعد از آن پروژه را در فایل `server.js` می نویسم.

برای داکرایز کردن این پروژه ابتدا یک داکر فایل می سازیم و دستورات زیر را در آن می نویسیم :

FROM node:18

WORKDIR /app

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 5000

CMD ["npm", "start"]

یک فایل به نام `docker-compose.app.yml` ایجاد می کنیم:

version: '3.8'

services:

server:

```
build:
  context: .
container_name: programming-languages-server
ports:
  - '5000:5000'
environment:
  - PORT=5000
  - MONGO_URL=mongodb://mongo:27017/programmingLanguagesDB
networks:
  - app-network
```

```
networks:
  app-network:
    external: true
```

و یک فایل `docker-compose.db.yml` :

```
version: '3.8'

services:
  mongo:
    image: mongo:6
    container_name: mongo
    ports:
      - '27017:27017'
    volumes:
      - mongo-data:/data/db
    networks:
      - app-network
```

```
volumes:
  mongo-data:
```

```
networks:
  app-network:
    external: true
```

ابتدا شبکه‌ای به نام **app-network** ایجاد میکنیم که بین دیتابیس و اپلیکیشن مشترک باشد:

```
docker network create app-network
```

```
[amir@myarch programming-languages-crud]$ docker network create app-network
5f32b3296bbc73903f0f51af463c18a92b30cca3382f242da987aca72a233121
[amir@myarch programming-languages-crud]$
```

فایل `docker-compose.db.yml` را اجرا می کنیم تا دیتابیس در کانتینر خود راه اندازی شود:

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

```
[amir@myarch programming-languages-crud]$ docker-compose -f docker-compose.db.yml up -d
[+] Running 9/9
✔ mongo 8 layers [#####] 0B/0B Pulled
✔ 6414378b6477 Pull complete
✔ e72d05fdcad4 Pull complete
✔ e74f3b867e89 Pull complete
✔ 8dd75d4e021c Pull complete
✔ ee8b12b4d97c Pull complete
✔ 686e3d514bb8 Pull complete
✔ 15eba7714a6a Pull complete
✔ 77ad40620929 Pull complete
[+] Building 0.0s (0/0)
[+] Running 3/3
✔ Network programming-languages-crud_app-network Created
✔ Volume "programming-languages-crud_mongo-data" Created
✔ Container mongo Started
[amir@myarch programming-languages-crud]$
```

فایل `docker-compose.app.yml` را برای اپلیکیشن اجرا می کنیم. این کار شامل ساخت ایمپج و اجرای کانتینر اپلیکیشن است:

`docker-compose -f docker-compose.app.yml up --build`

```
Q
amir@myarchi:~/programming-languages-crud
=> sha256:d71f59450bae3256b97ad4524b6d3cld8b45c9841969a64be675832c4b4f91e6 45.70MB / 45.70MB 73.6s
=> extracting sha256:54c7be425079efba0003054ee884bf72f1ffebca733bedd6f077d2809ee9aa6f 0.7s
=> sha256:9e4ed576e08d82f4ef497f814ca08452a78e77f2037e3318b67ffd8d5af2070a 1.25MB / 1.25MB 75.5s
=> sha256:b22066964b681d46cde8764fadc8b165a88bd968d81cb719fdb047ea0aad0157 446B / 446B 76.2s
=> extracting sha256:7aa8176e6d893aff4b57b2c22574ec2afadff4673b8e0954e275244bfd3d7bc1 2.8s
=> extracting sha256:1523f4b3f5602bf41caf8d8e649536ac0b3e56984c81a9f518fb20c6516ca075 7.2s
=> extracting sha256:4bc45260713570fbd8ba0e2dd4ccd25bf7f9df302d8aac2f2f2d7b8974142cf 0.0s
=> extracting sha256:d71f59450bae3256b97ad4524b6d3cld8b45c9841969a64be675832c4b4f91e6 2.1s
=> extracting sha256:9e4ed576e08d82f4ef497f814ca08452a78e77f2037e3318b67ffd8d5af2070a 0.1s
=> extracting sha256:b22066964b681d46cde8764fadc8b165a88bd968d81cb719fdb047ea0aad0157 0.0s
[server internal] load build context
=> transferring context: 14.10MB 0.3s
=> [server 2/5] WORKDIR /app 0.2s
=> [server 3/5] COPY package*.json ./ 0.0s
=> [server 4/5] RUN npm install 0.1s
=> [server 5/5] COPY . . 9.6s
=> [server] exporting to image 0.7s
=> exporting layers 1.7s
=> writing image sha256:9d2330f62f8303764c711558a5e6faf521f14c426790e6235add162c081004cb 1.7s
=> naming to docker.io/library/programming-languages-crud-server 0.0s
[+] Running 1/1 0.0s
✔ Container programming-languages-server Created 0.1s
Attaching to programming-languages-server
programming-languages-server > programming-languages-crud@1.0.0 start
programming-languages-server > node server.js
(node:19) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since No
de.js Driver version 4.0.0 and will be removed in the next major version
(Use `node --trace-warnings ...` to show where the warning was created)
(node:19) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect si
nce Node.js Driver version 4.0.0 and will be removed in the next major version
Server running on port 5000
Connected to MongoDB
```

برای اطمینان از اجرای درست کانتینرها، وضعیت آن‌ها را بررسی می کنیم:

docker ps

با وارد کردن ادرس زیر در postman می توانیم پروژه را تست کنیم:

http://localhost:5000/languages

برای اینکار ما یک نمونه دیتا رو post کردیم و دوباره get کردیم که به درستی کار میکنه:



