Docker Compose with Node Js and MongoDB

**Creating an Image using Docker File**

ubuntu@ip-172-31-18-205:~/Bootcamp/Docker\_compose/easy-node-authentication$ docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

sowmyatcs2306/nodeapp 1.0 cce202b24235 28 seconds ago 916MB

sowmyatcs2306/tomcatwar 1.0 a14c2c4904a3 3 hours ago 493MB

sowmyatcs2306/springboot 1.0 bbe79927d598 4 hours ago 122MB

mysql latest 5ac22cccc3ae 36 hours ago 544MB

hello-world latest bf756fb1ae65 6 months ago 13.3kB

node carbon 8eeadf3757f4 6 months ago 901MB

openjdk 8-jdk-alpine a3562aa0b991 14 months ago 105MB

tomcat 8.0.20-jre8 e88a065848be 5 years ago 493MB

**Docker File**

**FROM node:carbon**

**# Create app directory**

**WORKDIR /usr/src/app**

**# Install app dependencies**

**COPY package\*.json ./**

**RUN npm install**

**# Copy app source code**

**COPY . .**

**#Expose port and start application**

**EXPOSE 8080**

**CMD [ "npm", "start" ]**

**Docker Compose.yaml**

**Services:**

**web:**

**build: .**

**ports:**

**- "8080:8080"**

**depends\_on:**

**- mongo**

**mongo:**

**image: mongo**

**ports:**

**- "27017:27017"**

**Docker Compose Output**

**ubuntu@ip-172-31-18-205:~/Bootcamp/Docker\_compose/easy-node-authentication$ vim docker-compose.yaml**

**ubuntu@ip-172-31-18-205:~/Bootcamp/Docker\_compose/easy-node-authentication$ docker-compose up -d**

**easynodeauthentication\_mongo\_1 is up-to-date**

**Recreating easynodeauthentication\_web\_1**

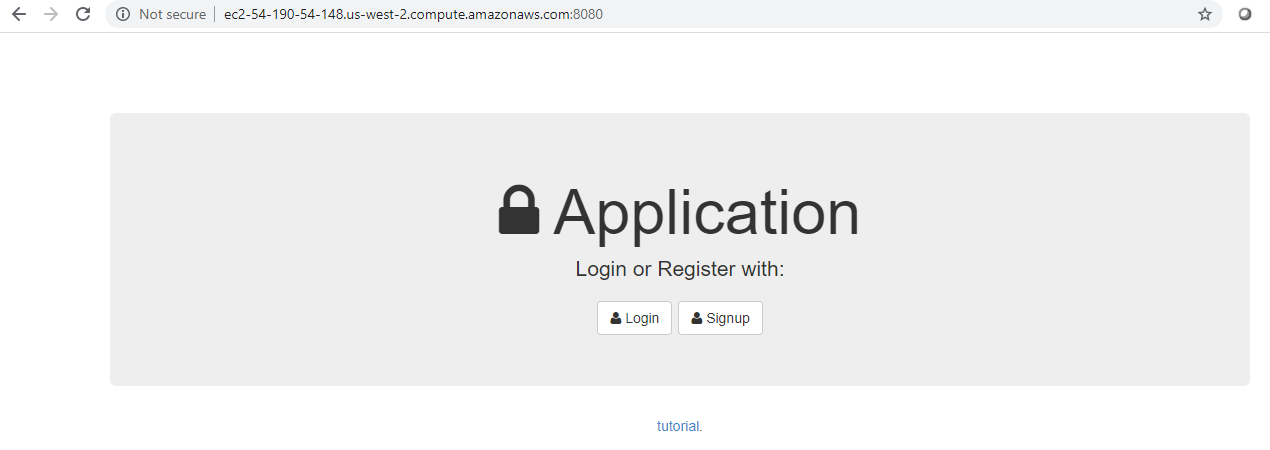
**ubuntu@ip-172-31-18-205:~/Bootcamp/Docker\_compose/easy-node-authentication$ docker ps**

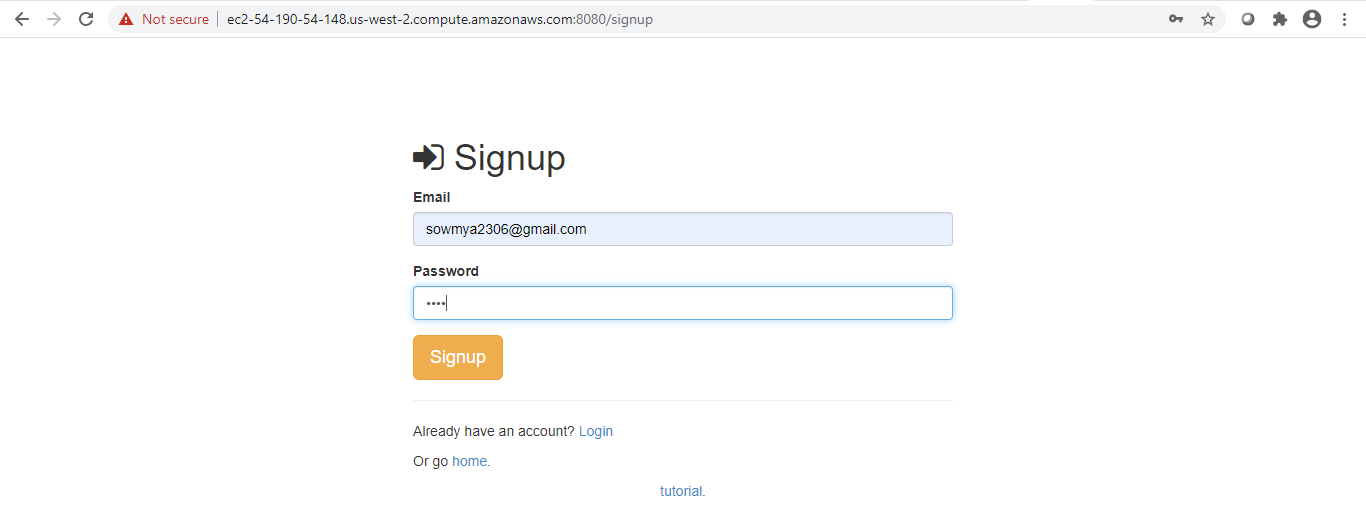
**CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES**

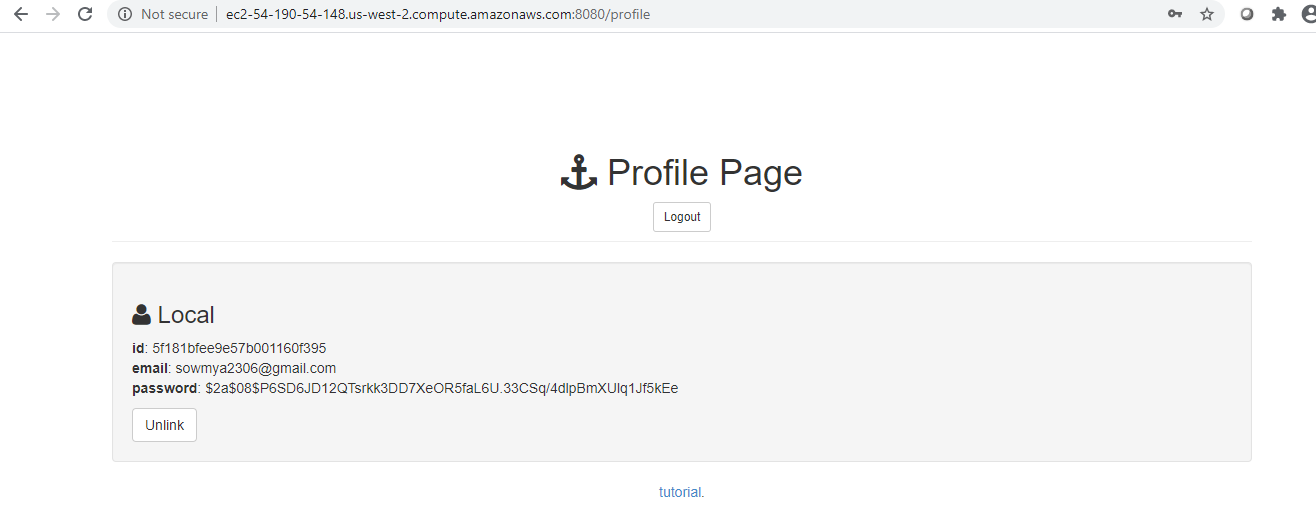
**fa9d04f273b5 easynodeauthentication\_web "docker-entrypoint.s…" 8 seconds ago Up 7 seconds 8080/tcp, 0.0.0.0:9090->9090/tcp easynodeauthentication\_web\_1**

**c705c8625a38 mongo "docker-entrypoint.s…" 52 seconds ago Up 51 seconds 0.0.0.0:27017->27017/tcp easynodeauthentication\_mongo\_1**

**Output**







**Insertion to Mongo DB**

Enable MongoDB's free cloud-based monitoring service, which will then receive and display

metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you

and anyone you share the URL with. MongoDB may use this information to make product

improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()

To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

---

> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

> show local

2020-07-22T11:01:02.227+0000 E QUERY [js] uncaught exception: Error: don't know how to show [local] :

shellHelper.show@src/mongo/shell/utils.js:1139:11

shellHelper@src/mongo/shell/utils.js:790:15

@(shellhelp2):1:1

> use local

switched to db local

> show collections

startup\_log

> use admin

switched to db admin

> show collections

system.version

users

> db.users.find({})

**{ "\_id" : ObjectId("5f181bfee9e57b001160f395"), "local" : { "password" : "$2a$08$P6SD6JD12QTsrkk3DD7XeOR5faL6U.33CSq/4dlpBmXUlq1Jf5kEe", "email" : "sowmya2306@gmail.com" }, "\_\_v" : 0 }**