Docker Compose for containerizing a MEAN Stack Application :

Docker Compose can be used to create separate containers (and host them) for each of the stacks in a MEAN stack application. MEAN is the acronym for MongoDB Express Angular & NodeJs.

By using Docker Compose, we can host each of these technologies in separate containers on the same host and get them to communicate with each other. Each container will expose a port for communicating with other containers.

The communication and up-time of these containers will be maintained by Docker Compose.

In our case, we have a full stack application which comprises of MongoDB, ExpressJS, Angular and NodeJS. MongoDB takes care of the back end database, NodeJS and ExpressJS are for server side rendering, and Angular is for front end.

Write a separate dockerfile for building each of the single-container applications. Additionally, we will have to also write a Docker Compose File which will do the actual work. Docker Compose File will execute the different dockerfiles to create the different containers and let them interact with each other.

Creating Docker Containers

1. Dockerfile For Front End

```
FROM node:6
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
COPY package.json /usr/src/app
RUN npm cache clean
RUN npm install
COPY . /usr/src/app
EXPOSE 4200
CMD ["npm","start"]
```

2. Dockerfile for Back End:

```
FROM node:6
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
COPY package.json /usr/src/app
RUN npm cache clean
RUN npm install
COPY . /usr/src/app
EXPOSE 3000
CMD ["npm","start"]
```

If you are using latest versions :

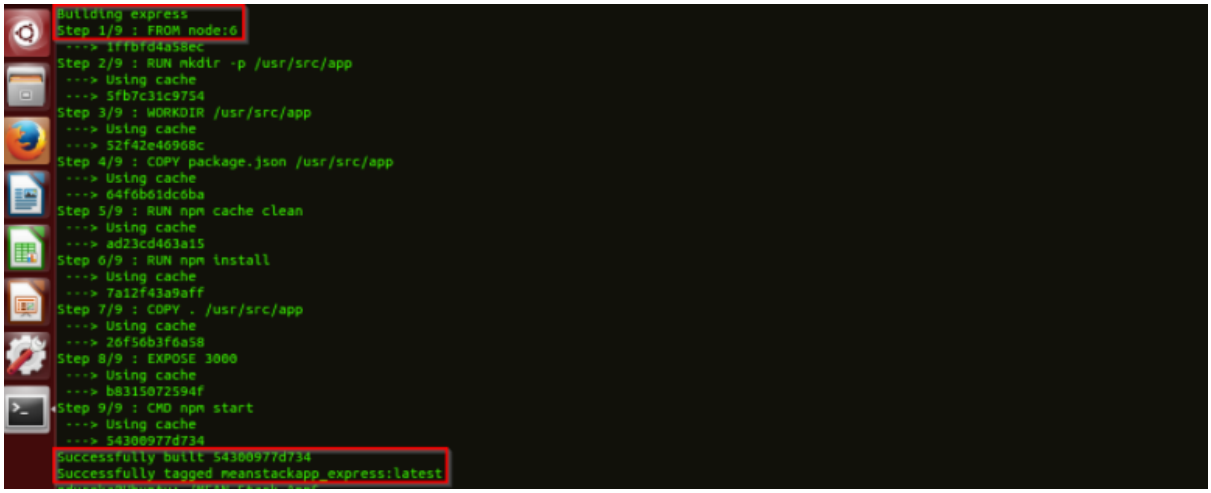
To download Compose, run the below set of commands.

```
sudo curl -L https://github.com/docker/compose/releases/dowr
`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

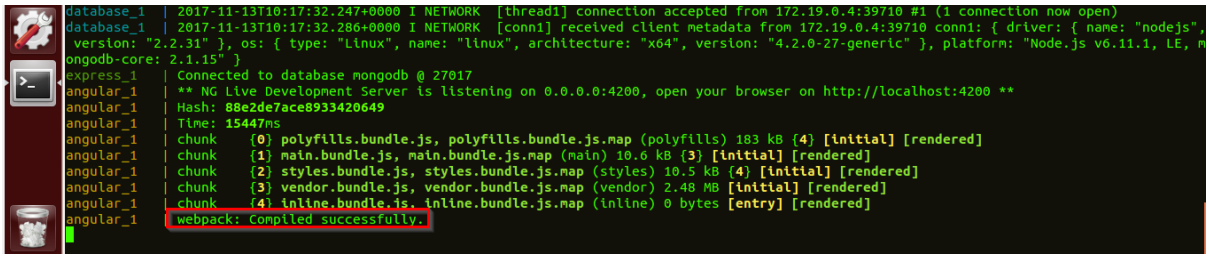
```
sudo edit docker-compose.yml

version: '3.0' # specify docker-compose version

# Define the services/ containers to be run
services:
 angular: # name of the first service
  build: angular-app # specify the directory of the
Dockerfile
  ports:
  - "4200:4200" # specify port mapping

 express: # name of the second service
  build: express-server # specify the directory of
the Dockerfile
  ports:
  - "3000:3000" #specify ports mapping
  links:
  - database # link this service to the database
service

 database: # name of the third service
  image: mongo # specify image to build container
from
  ports:
  - "27017:27017" # specify port forwarding
```

Run these commands from same folder where Docker compose file is present

```
docker-compose build
docker-compose up
```



Now run docker-compose up



Go type the following port numbers in your web browser to interact with the GUI of the MEAN app.

localhost:4200 – *Angular App (Front-end)*
localhost:3000 – *Express Server & NodeJS (Back-end/ Server-side)*
localhost:27017 – MongoDB (Database)

"docker-compose scale='x'" command to easily scale up/ down the number of deployments.

docker-compose scale=5