Deploy Angular Application in Docker Container.

DESCRIPTION

Deploy the Angular application in Docker. The Angular application should be built with the Angular CLI along with Docker Compose for development and production.

Problem Statement Scenario:

HTQual Technology Solutions hired you as a MEAN Stack Developer. The organization decided to implement DevOps to develop and deliver the products. Since HTQual is an Agile organization, they follow Scrum methodology to develop the projects incrementally. The Company decided to develop their website on Mean stack. Since you are the MEAN stack developer, you have to demonstrate that deploying an Angular application on Docker is always a best approach to develop a project and test it incrementally. You agreed upon the following:

Setting up an image for code development

Build the application in Docker and host it in Docker Hub

List the advantages, disadvantages, and document the tasks involved

Your goal is to demonstrate the Angular application and run it in a Docker container.

You must use the following tools:

Docker – To package the application in a Docker container

Node.js – To support the Angular application with the required node modules.

Angular CLI – To execute and bundle the dependencies together.

Linux (Ubuntu) – As a base operating system to start and execute the project.

Following requirements should be met:

Document the step-by-step process from the initial installation to the final production.

Run the Angular application successfully in the Docker container.

Use Docker Compose to manage the Angular application running inside the Docker container.

REPORT

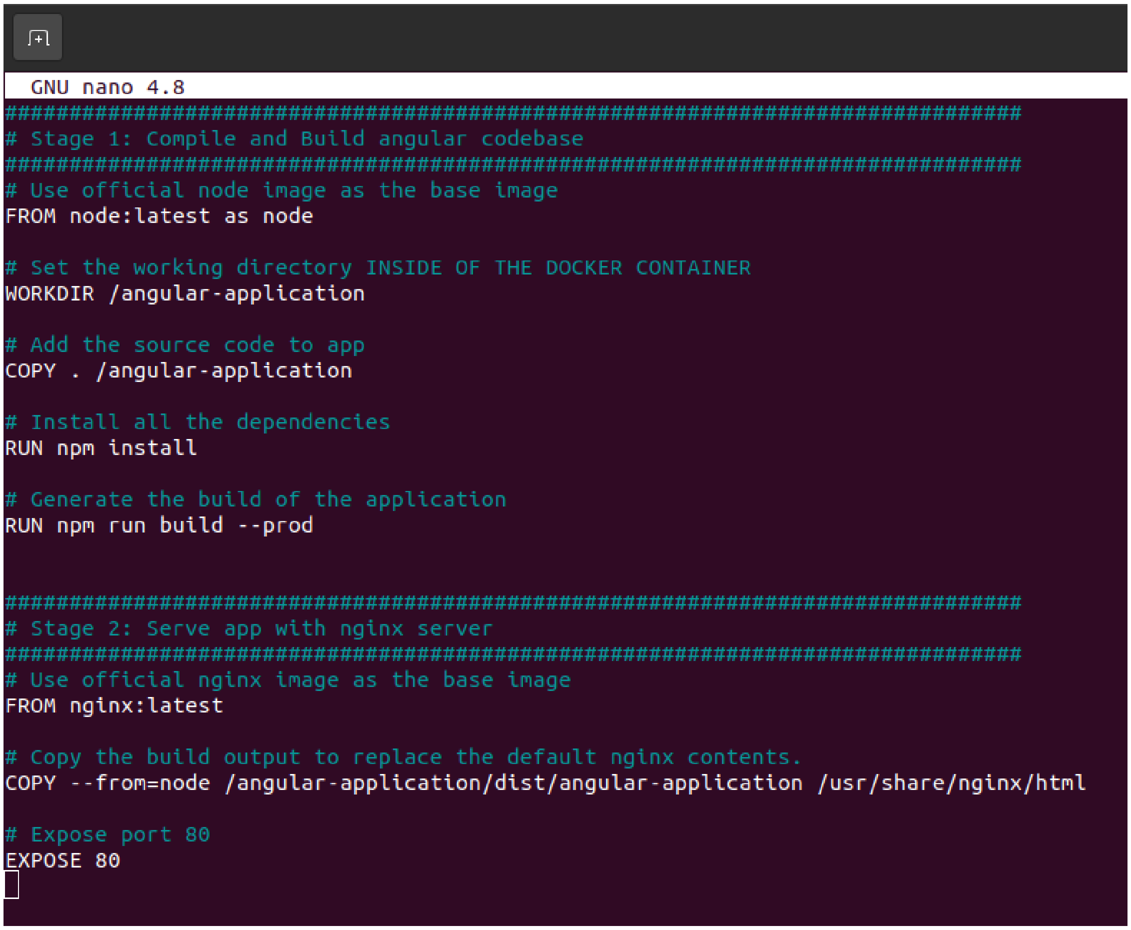
1) Install Docker, Docker Compose, Node.js, TypeScript and Angular:

```
curl -fsSL https://get.docker.com -o get-docker.sh

sudo sh get-docker.sh

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

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

sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose

curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -

sudo apt-get install -y nodejs

npm install typescript -g

npm install -g @angular/cli
```

2) Create the Angular application.

```
sudo ng new angular-application

ng serve --open
```

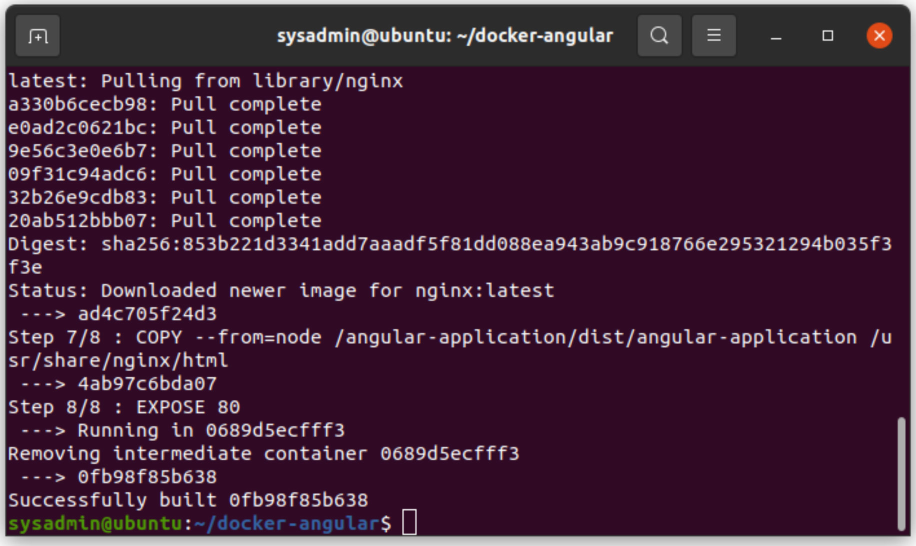
3) Create the Dockerfile



```
  GNU nano 4.8
###############################################################################
# Stage 1: Compile and Build angular codebase
###############################################################################
# Use official node image as the base image
FROM node:latest as node

# Set the working directory INSIDE OF THE DOCKER CONTAINER
WORKDIR /angular-application

# Add the source code to app
COPY . /angular-application

# Install all the dependencies
RUN npm install

# Generate the build of the application
RUN npm run build --prod


###############################################################################
# Stage 2: Serve app with nginx server
###############################################################################
# Use official nginx image as the base image
FROM nginx:latest

# Copy the build output to replace the default nginx contents.
COPY --from=node /angular-application/dist/angular-application /usr/share/nginx/html

# Expose port 80
EXPOSE 80
```

4) Create the Docker image with the Angular application

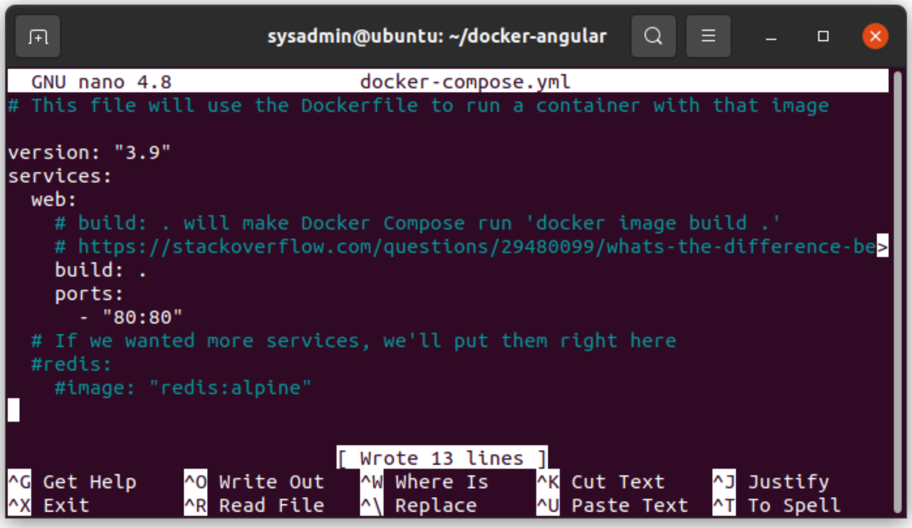
sudo docker image build .



```
latest: Pulling from library/nginx
a330b6cecb98: Pull complete
e0ad2c0621bc: Pull complete
9e56c3e0e6b7: Pull complete
09f31c94adc6: Pull complete
32b26e9cdb83: Pull complete
20ab512bbb07: Pull complete
Digest: sha256:853b221d3341add7aaadf5f81dd088ea943ab9c918766e295321294b035f3
f3e
Status: Downloaded newer image for nginx:latest
 ---> ad4c705f24d3
Step 7/8 : COPY --from=node /angular-application/dist/angular-application /u
sr/share/nginx/html
 ---> 4ab97c6bda07
Step 8/8 : EXPOSE 80
 ---> Running in 0689d5ecfff3
Removing intermediate container 0689d5ecfff3
 ---> 0fb98f85b638
Successfully built 0fb98f85b638
sysadmin@ubuntu:~/docker-angular$ 
```

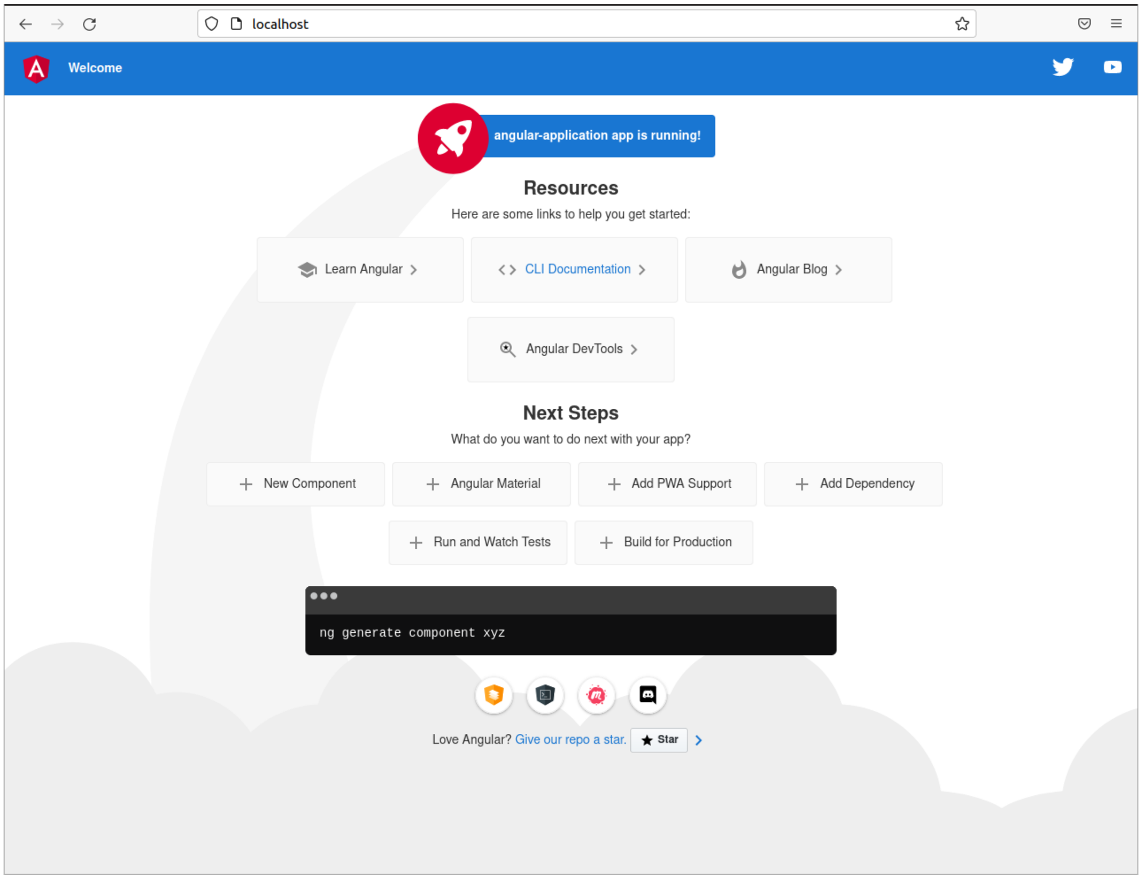
With that, now we know that the Dockerfile has a correct syntax and everything works.

5) Create the Docker Compose file for development and production



```
  GNU nano 4.8                   docker-compose.yml
# This file will use the Dockerfile to run a container with that image

version: "3.9"
services:
  web:
    # build: . will make Docker Compose run 'docker image build .'
    # https://stackoverflow.com/questions/29480099/whats-the-difference-be>
    build: .
    ports:
      - "80:80"
  # If we wanted more services, we'll put them right here
  #redis:
    #image: "redis:alpine"

                            [ Wrote 13 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell
```

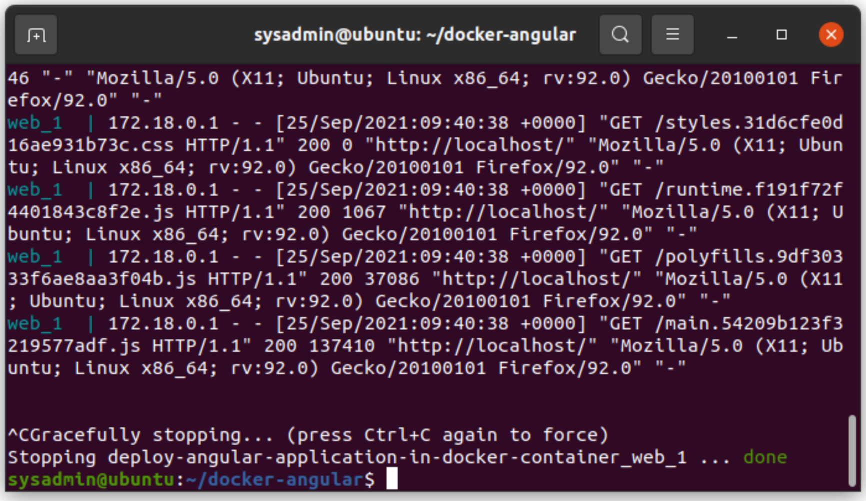
```
sudo docker-compose up
```



```
ng generate component xyz
```

To run the service as a daemon, run sudo docker-compose up -d as -d stands for detached.

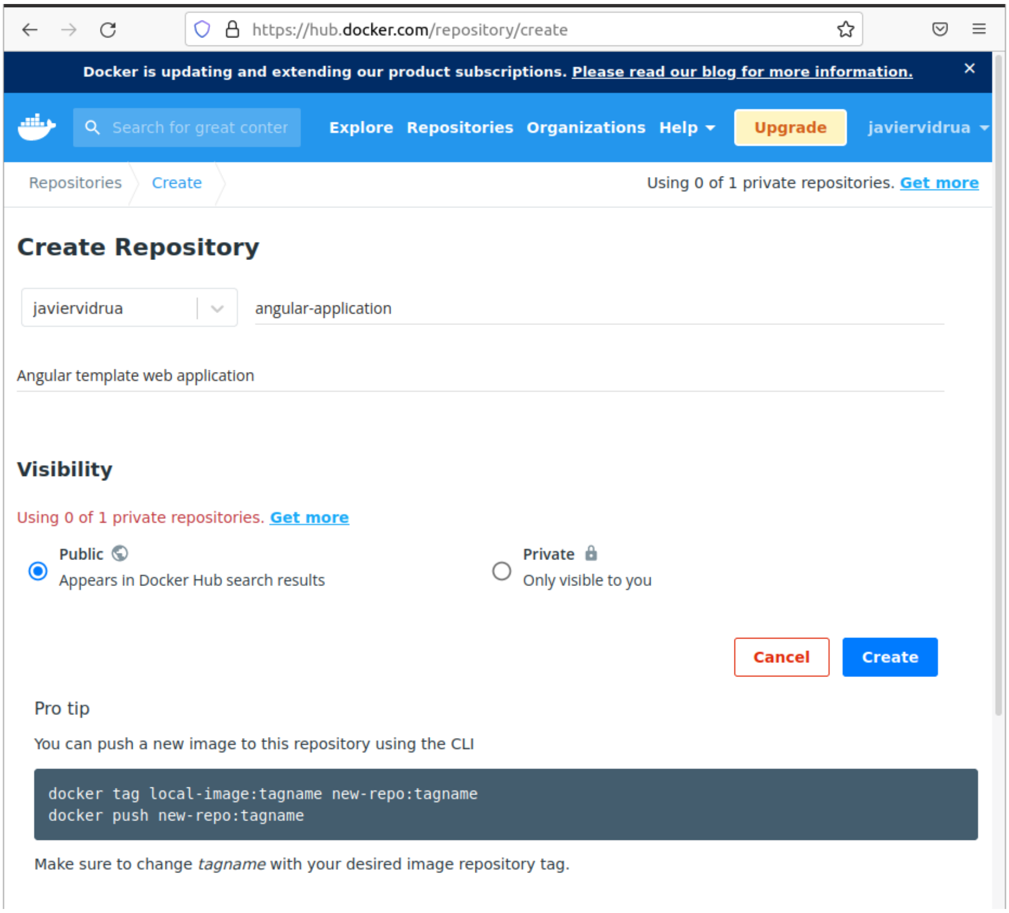
To stop the execution, press Ctrl+C or run sudo docker-compose stop .



```
46 "-" "Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Fir
efox/92.0" "-"
web_1  | 172.18.0.1 - - [25/Sep/2021:09:40:38 +0000] "GET /styles.31d6cfe0d
16ae931b73c.css HTTP/1.1" 200 0 "http://localhost/" "Mozilla/5.0 (X11; Ubun
tu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0" "-"
web_1  | 172.18.0.1 - - [25/Sep/2021:09:40:38 +0000] "GET /runtime.f191f72f
4401843c8f2e.js HTTP/1.1" 200 1067 "http://localhost/" "Mozilla/5.0 (X11; U
buntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0" "-"
web_1  | 172.18.0.1 - - [25/Sep/2021:09:40:38 +0000] "GET /polyfills.9df303
33f6ae8aa3f04b.js HTTP/1.1" 200 37086 "http://localhost/" "Mozilla/5.0 (X11
; Ubuntu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0" "-"
web_1  | 172.18.0.1 - - [25/Sep/2021:09:40:38 +0000] "GET /main.54209b123f3
219577adf.js HTTP/1.1" 200 137410 "http://localhost/" "Mozilla/5.0 (X11; Ub
untu; Linux x86_64; rv:92.0) Gecko/20100101 Firefox/92.0" "-"


^CGracefully stopping... (press Ctrl+C again to force)
Stopping deploy-angular-application-in-docker-container_web_1 ... done
sysadmin@ubuntu:~/docker-angular$
```

6) Host the application in DockerHub

First, we create the repository in DockerHub:



Docker is updating and extending our product subscriptions. Please read our blog for more information.

Using 0 of 1 private repositories. Get more

Create Repository

javiervidrua / angular-application

Angular template web application

Visibility

Using 0 of 1 private repositories. Get more

● Public
Appears in Docker Hub search results

○ Private
Only visible to you

Pro tip

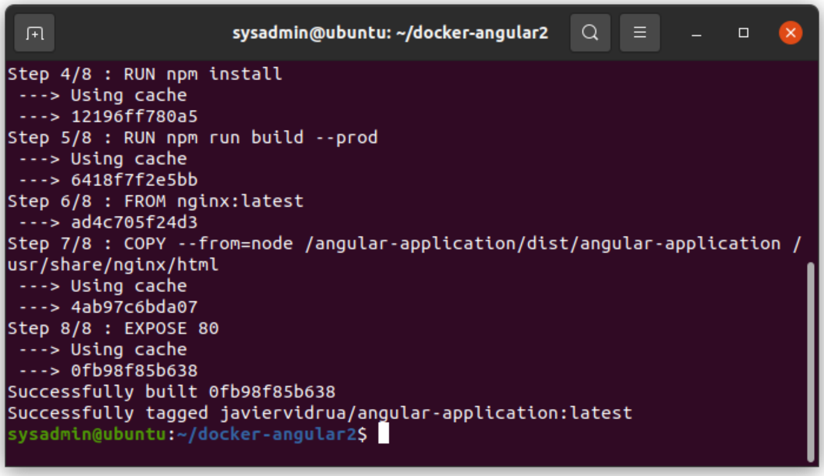
You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

Then, we build the files:

```
sudo docker build -t javiervidrua/angular-application .
```

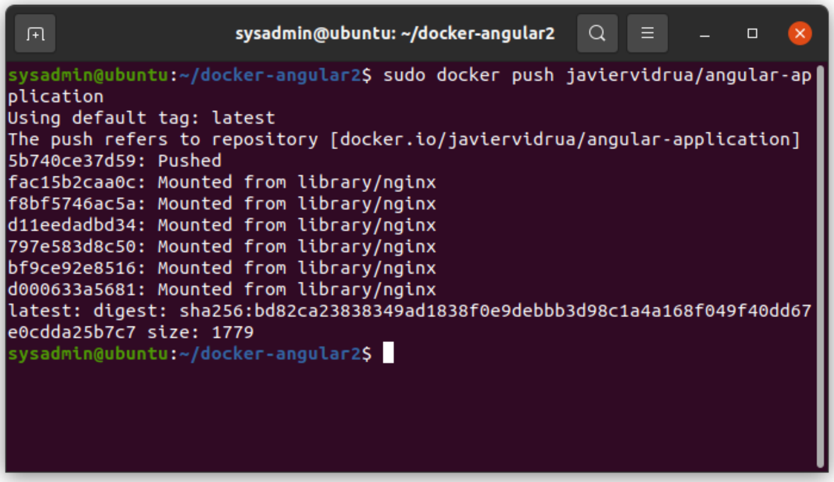


```
Step 4/8 : RUN npm install
 ---> Using cache
 ---> 12196ff780a5
Step 5/8 : RUN npm run build --prod
 ---> Using cache
 ---> 6418f7f2e5bb
Step 6/8 : FROM nginx:latest
 ---> ad4c705f24d3
Step 7/8 : COPY --from=node /angular-application/dist/angular-application /
usr/share/nginx/html
 ---> Using cache
 ---> 4ab97c6bda07
Step 8/8 : EXPOSE 80
 ---> Using cache
 ---> 0fb98f85b638
Successfully built 0fb98f85b638
Successfully tagged javiervidrua/angular-application:latest
sysadmin@ubuntu:~/docker-angular2$ 
```

Then, we upload the files to the repository:

```
sudo docker login
```

```
sudo docker push javiervidrua/angular-application
```



```
sysadmin@ubuntu:~/docker-angular2$ sudo docker push javiervidrua/angular-ap
plication
Using default tag: latest
The push refers to repository [docker.io/javiervidrua/angular-application]
5b740ce37d59: Pushed
fac15b2caa0c: Mounted from library/nginx
f8bf5746ac5a: Mounted from library/nginx
d11eedadbd34: Mounted from library/nginx
797e583d8c50: Mounted from library/nginx
bf9ce92e8516: Mounted from library/nginx
d000633a5681: Mounted from library/nginx
latest: digest: sha256:bd82ca23838349ad1838f0e9debbb3d98c1a4a168f049f40dd67
e0cdda25b7c7 size: 1779
sysadmin@ubuntu:~/docker-angular2$ 
```

7) List the advantages, disadvantages and document the tasks involved

Advantages: Now the application is scalable and portable
Disadvantages: A little configuration is needed

All in all, it’s worth the effort!