

Docker Lab2

1. Problem 1:

Create bridge network with subnet 192.168.0.0/24.

```
root@localhost:~  
File Edit View Search Terminal Tabs Help  
maly@localhost:~/docker-labs/lab1/simp... x root@localhost:~ x  
[root@localhost ~]# docker network ls  
NETWORK ID NAME DRIVER SCOPE  
3f6523e9bcc6 bridge bridge local  
9c824f6d6ee5 host host local  
22e9d82c0a3b none null local  
[root@localhost ~]# docker network create NW1 --driver=bridge --subnet=192.168.0.0/24  
2ba37918b6579b5dc42730d5212b0c20a55990bd11a8b59f77ae2ad2693a61d1  
[root@localhost ~]# docker network create NW2 --driver=bridge --subnet=10.5.0.0/24  
02ce05b049025a802a5e4cd64c4592dfd1a51d2c11e59fa73c19c2f70e2be06a  
[root@localhost ~]# docker network ls  
NETWORK ID NAME DRIVER SCOPE  
2ba37918b657 NW1 bridge local  
02ce05b04902 NW2 bridge local  
3f6523e9bcc6 bridge bridge local  
9c824f6d6ee5 host host local  
22e9d82c0a3b none null local  
[root@localhost ~]#
```

Run 2 containers and attach containers to this network.

```
root@localhost:~  
File Edit View Search Terminal Tabs Help  
maly@localhost:~/docke... x root@localhost:~ x root@localhost:~ x  
[root@localhost ~]# docker run -d --network=NW1 nginx  
Unable to find image 'nginx:latest' locally  
latest: Pulling from library/nginx  
025c56f98b67: Pull complete  
ca9c7f45d396: Pull complete  
ed6bd111fc08: Pull complete  
e25b13a5f70d: Pull complete  
9bbabac55ab6: Pull complete  
e5c9ba265ded: Pull complete  
Digest: sha256:ab589a3c466e347b1c0573be23356676df90cd7ce2dbf6ec332a5f0a8b5e59db  
Status: Downloaded newer image for nginx:latest  
f7934aad756210934ded28f7ce01892c7823054d5db8e1d00e6865a90a2f58fc  
[root@localhost ~]# docker run -d --network=NW1 nginx  
4745caf21fb25854d156582421fbb8f3112a762a281c7395b7c102ed9e35f090  
[root@localhost ~]# docker ps  
CONTAINER ID IMAGE COMMAND CREATED STATUS  
PORTS NAMES  
4745caf21fb2 nginx "/docker-entrypoint..." 9 seconds ago Up 6 seconds  
80/tcp pensive_meninsky  
f7934aad7562 nginx "/docker-entrypoint..." About a minute ago Up About a minut  
e 80/tcp pedantic_wilson  
[root@localhost ~]# docker inspect 4745caf21fb2 | grep NW1  
"NetworkMode": "NW1",  
"NW1": {  
[root@localhost ~]# docker inspect f7934aad7562 | grep NW1  
"NetworkMode": "NW1"
```

Create another bridge network with subnet 10.5.0.0/24.

```
root@localhost:~  
File Edit View Search Terminal Tabs Help  
maly@localhost:~/docker-labs/lab1/simp... x root@localhost:~ x  
[root@localhost ~]# docker network ls  
NETWORK ID          NAME                DRIVER              SCOPE  
3f6523e9bcc6        bridge             bridge              local  
9c824f6d6ee5        host               host                local  
22e9d82c0a3b        none              null                local  
[root@localhost ~]# docker network create NW1 --driver=bridge --subnet=192.168.0.0/24  
2ba37918b6579b5dc42730d5212b0c20a55990bd11a8b59f77ae2ad2693a61d1  
[root@localhost ~]# docker network create NW2 --driver=bridge --subnet=10.5.0.0/24  
02ce05b049025a802a5e4cd64c4592dfd1a51d2c11e59fa73c19c2f70e2be06a  
[root@localhost ~]# docker network ls  
NETWORK ID          NAME                DRIVER              SCOPE  
2ba37918b657        NW1                 bridge              local  
02ce05b04902        NW2                 bridge              local  
3f6523e9bcc6        bridge             bridge              local  
9c824f6d6ee5        host               host                local  
22e9d82c0a3b        none              null                local  
[root@localhost ~]#
```

Run any container and attach it to the new network.

```
[root@localhost ~]# docker run -d --network=NW2 nginx  
03ecff3bdb3af184c1e427e7998d8af59b238fca8fa05e611c793daa76da99f0  
[root@localhost ~]# docker ps  
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS  
NAMES  
03ecff3bdb3a   nginx     "/docker-entrypoint..." 9 seconds ago  Up 7 seconds  80/tcp  
musing_lederberg  
4745caf21fb2   nginx     "/docker-entrypoint..." 6 minutes ago  Up 6 minutes  80/tcp  
pensive_meninsky  
f7934aad7562   nginx     "/docker-entrypoint..." 8 minutes ago  Up 8 minutes  80/tcp  
pedantic_wilson  
[root@localhost ~]# docker inspect 03ecff3bdb3a | grep NW2  
    "NetworkMode": "NW2",  
    "NW2": {  
[root@localhost ~]#
```

Make sure that the containers at different network can't ping each other

```
root@localhost:~  
File Edit View Search Terminal Tabs Help  
maly@localhost:~/docke... x root@localhost:~ x root@localhost:~  
ping: usage error: Destination address required  
root@4745caf21fb2:/# ping 10.5.0.2  
PING 10.5.0.2 (10.5.0.2) 56(84) bytes of data.  
^C  
--- 10.5.0.2 ping statistics ---  
433 packets transmitted, 0 received, 100% packet loss, time 434160ms  
  
root@4745caf21fb2:/# ping 192.168.0.2  
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.  
64 bytes from 192.168.0.2: icmp_seq=1 ttl=64 time=0.178 ms  
64 bytes from 192.168.0.2: icmp_seq=2 ttl=64 time=0.213 ms  
64 bytes from 192.168.0.2: icmp_seq=3 ttl=64 time=0.210 ms  
64 bytes from 192.168.0.2: icmp_seq=4 ttl=64 time=0.212 ms  
64 bytes from 192.168.0.2: icmp_seq=5 ttl=64 time=0.372 ms  
^C  
--- 192.168.0.2 ping statistics ---  
5 packets transmitted, 5 received, 0% packet loss, time 3996ms  
rtt min/avg/max/mdev = 0.178/0.237/0.372/0.068 ms  
root@4745caf21fb2:/# ping 192.168.0.3  
PING 192.168.0.3 (192.168.0.3) 56(84) bytes of data.  
64 bytes from 192.168.0.3: icmp_seq=1 ttl=64 time=0.053 ms  
64 bytes from 192.168.0.3: icmp_seq=2 ttl=64 time=0.091 ms  
64 bytes from 192.168.0.3: icmp_seq=3 ttl=64 time=0.185 ms  
64 bytes from 192.168.0.3: icmp_seq=4 ttl=64 time=0.098 ms
```

2. Problem 2:

Create static html file

Write Dockerfile to build image based on httpd to host the html file and specify the following

Copy the html file.

Copy a new configuration file to listen on port 9999 instead of 80

Open the port 9999 in the container

Add environment variable CONTAINER with value docker.

Add startup command to echo the variable

```
Get Started | helloworld.html x
home > maly > docker-labs > lab2 > problem2 > helloworld.html > html > head
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |   |   <title>Hello, World!</title>
5  |   |
6  |   |   </head>
7  |   |   <body>
8  |   |   |   <p>Hello, World!</p>
9  |   |   </body>
10 |   </html>
```

```
Get Started | Dockerfile x | helloworld.html
home > maly > docker-labs > lab2 > problem2 > Dockerfile > ...
1  FROM httpd
2  COPY ./helloworld.html /var/www/html/
3  COPY ./httpd.conf /etc/httpd/conf/httpd.conf
4  ENV CONTAINER=docker
5  EXPOSE 9999
6  CMD echo $CONTAINER && httpd -D FOREGROUND
7
```

```
[maly@localhost problem2]$ sudo docker build -t html-image .
[sudo] password for maly:
Sending build context to Docker daemon 15.87kB
Step 1/6 : FROM httpd
--> 157dcd623d6c
Step 2/6 : COPY ./helloworld.html /var/www/html/
--> e0995cd3dddc
Step 3/6 : COPY ./httpd.conf /etc/httpd/conf/httpd.conf
--> c9e3ffc164bb
Step 4/6 : ENV CONTAINER=docker
--> Running in 24cf8aa850dc
Removing intermediate container 24cf8aa850dc
--> 052788a31e7e
Step 5/6 : EXPOSE 9999
--> Running in fe9able32c5c
Removing intermediate container fe9able32c5c
--> 4b1f4216a9bf
Step 6/6 : CMD echo $CONTAINER && httpd -D FOREGROUND
--> Running in 2cc3739aabdd
Removing intermediate container 2cc3739aabdd
--> 031ballab3c6
Successfully built 031ballab3c6
Successfully tagged html-image:latest
[maly@localhost problem2]$ sudo docker run -p 9999:9999 html-image
docker
```

```
#
ServerRoot "/etc/httpd"
#
# Listen: Allows you to bind
# ports, instead of the defa
# directive.
#
# Change this to Listen on s
# prevent Apache from glommi
#
#Listen 12.34.56.78:80
Listen 9999
```

Name



Dockerfile



helloworld.html



httpd.conf

Hello, World!



localhost:9999/helloworld.html



Gmail



YouTube



Maps



Fake Email Gen...



Git - Do

Hello, World!

3. Problem 3:

Create a docker compose to setup web container (flask app from lab1 if not exist) and nginx, MySQL, the app container depends on nginx and MySQL

Add volume for MySQL dB

```
Get Started docker-compose.yml
home > maly > docker-labs > lab2 > problem3 > docker-compose.yml
1  version: '3.9'
2  services:
3    app:
4      build:
5        context: .
6        dockerfile: Dockerfile
7      image: flask-app:v5
8      container_name: app
9      restart: unless-stopped
10     ports:
11       - "8090:5000"
12     networks:
13       - app-network
14     depends_on:
15       - web
16       - db
17   #Nginx Service
18   web:
19     image: nginx:alpine
20     container_name: webserver
21     restart: unless-stopped
22     ports:
23       - "8080:80"
24     networks:
25       - app-network
```

```
Get Started docker-compose.yml x
home > maly > docker-labs > lab2 > problem3 > docker-compose.yml
27  db:
28    image: mysql:5.7
29    container_name: Mysqldb
30    restart: unless-stopped
31    volumes:
32      - db_data:/var/lib/mysql
33    ports:
34      - "3306:3306"
35    environment:
36      MYSQL_ROOT_PASSWORD_FILE: /run/secrets/db_root_password
37      MYSQL_DATABASE: flask
38      MYSQL_USER: flask-user
39      MYSQL_PASSWORD_FILE: /run/secrets/db_password
40    secrets:
41      - db_root_password
42      - db_password
43    networks:
44      - app-network
45  secrets:
46    db_password:
47      file: db_password.txt
48    db_root_password:
49      file: db_root_password.txt
50  volumes:
51    db_data:
52  networks:
53    app-network:
54      driver: bridge
```

Go to Line/Column








Home

docker-labs

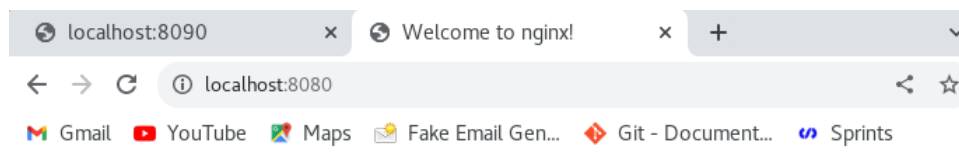
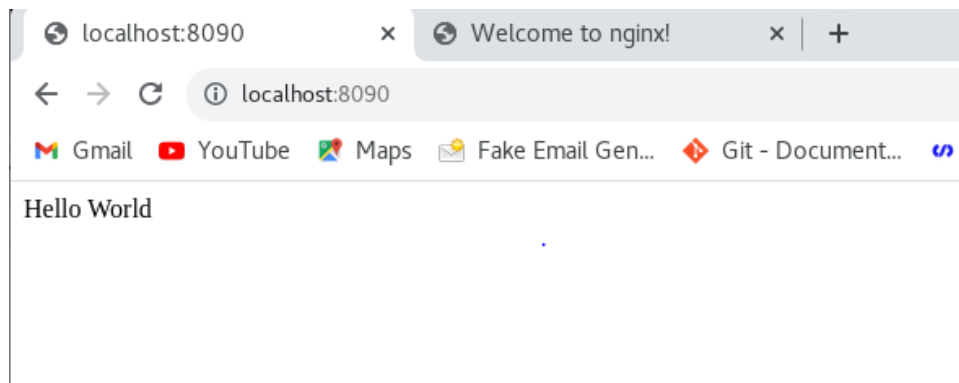
lab2

problem3

Q

Name	Size	Modified
 app.py	169 bytes	4 Dec
 db_password.txt	0 bytes	18:56
 db_root_password.txt	0 bytes	18:56
 docker-compose.yml	1.1 kB	13:59
 Dockerfile	122 bytes	4 Dec
 README.md	18 bytes	4 Dec
 requirements.txt	64 bytes	4 Dec

```
[maly@localhost problem3]$ sudo docker compose -f ./docker-compose.yml up
[sudo] password for maly:
[+] Running 2/2
  :: Container Mysqldb Created                                0.4s
  :: Container app Created                                    0.5s
Attaching to Mysqldb, app, webserver
webserver | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
webserver | /docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
webserver | /docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
webserver | 10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
Mysqldb   | 2022-12-11 16:56:40+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.40-1.el7 started.
webserver | 10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
webserver | /docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
webserver | /docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
webserver | /docker-entrypoint.sh: Configuration complete; ready for start up
webserver | 2022/12/11 16:56:41 [notice] 1#1: using the "epoll" event method
webserver | 2022/12/11 16:56:41 [notice] 1#1: nginx/1.23.2
webserver | 2022/12/11 16:56:41 [notice] 1#1: built by gcc 11.2.1 20220219 (Alpine 11.2.1_git20220219)
webserver | 2022/12/11 16:56:41 [notice] 1#1: OS: Linux 4.18.0-408.el8.x86_64
webserver | 2022/12/11 16:56:41 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
webserver | 2022/12/11 16:56:41 [notice] 1#1: start worker processes
webserver | 2022/12/11 16:56:41 [notice] 1#1: start worker process 30
Mysqldb   | 2022-12-11 16:56:44+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
Mysqldb   | 2022-12-11 16:56:44+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 5.7.40-1.el7 started
```



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

```
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
bdcf392fc442   flask-app:v5   "python app.py"         16 minutes ago Up 16 mi
nutes         0.0.0.0:8090->5000/tcp, :::8090->5000/tcp   app
f2d22405b51c   mysql:5.7      "docker-entrypoint.s..." 16 minutes ago Restarti
ng (1) 10 seconds ago           Mysqldb
b4a0b7356af4   nginx:alpine   "/docker-entrypoint...." 28 minutes ago Up 16 mi
nutes         0.0.0.0:8080->80/tcp, :::8080->80/tcp       webserver
[root@localhost ~]#
```


5. Problem 5:

Use docker compose to deploy ghost platform (image: ghost:1- alpine) (Ghost is a free and open-source blogging platform written in JavaScript) Use MySQL database instead of SQLite

```
home > maly > docker-labs > lab2 > problem5 > docker-compose.yml
1  version: '3.9'
2  services:
3    ghost:
4      image: ghost:alpine
5      container_name: ghost
6      volumes:
7        - ghost-volume:/var/lib/ghost
8      restart: always
9      ports:
10       - '80:2368'
11      environment:
12        database__client: mysql
13        database__connection__host: mysql
14        database__connection__user: ghostuser
15        database__connection__password: user
16        database__connection__database: db
17      depends_on:
18        - mysql
19
20    mysql:
21      image: mysql
22      container_name: mysql
23      volumes:
24        - mysql-volume:/var/lib/mysql
25      ports:
26        - "3306:3306"
27      environment:
28        MYSQL_ROOT_PASSWORD: root
29        MYSQL_DATABASE: db
30        MYSQL_USER: ghostuser
31        MYSQL_PASSWORD: user
32      restart: always
33
34  volumes:
35    mysql-volume:
36    ghost-volume:
```

```

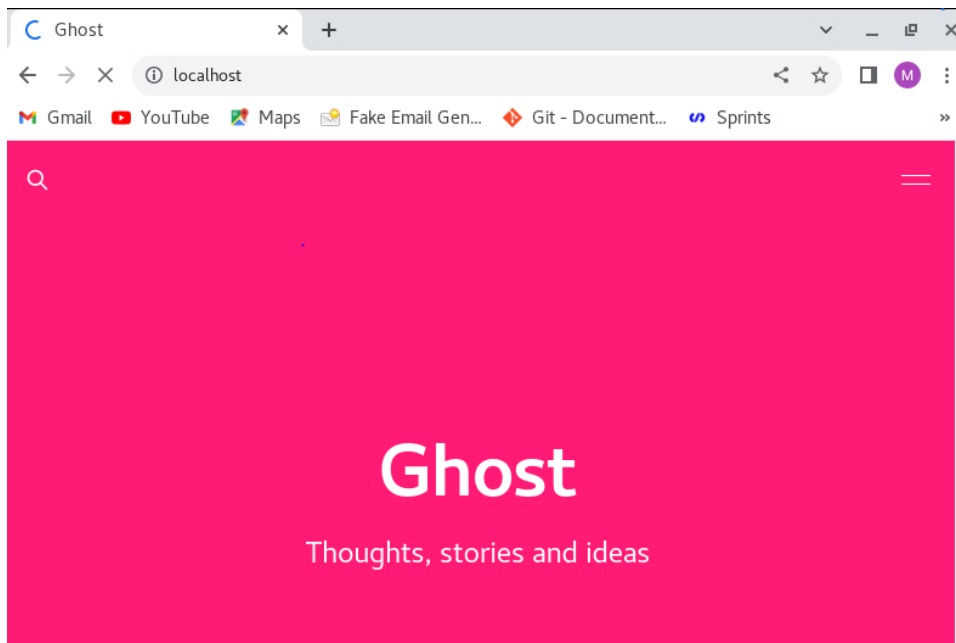
# ghost Pulled 421.2s
# 025c56f98b67 Already exists 0.0s
# 64161be65588 Pull complete 4.2s
# b25f1bb722e1 Pull complete 60.1s
# b76e155def5b Pull complete 63.0s
# d59a1322f524 Pull complete 63.8s
# 6caabdd0cf89 Pull complete 65.6s
# 757dae1792b9 Pull complete 127.1s
# 17853d26dfe8 Pull complete 405.4s
# f48b60a247ce Pull complete 415.3s
[+] Building 1.8s (2/2) FINISHED
=> [internal] load build definition from Dockerfile 1.2s
=> => transferring dockerfile: 2B 0.1s
=> [internal] load .dockerignore 1.0s
=> => transferring context: 2B 0.1s
failed to solve: rpc error: code = Unknown desc = failed to solve with frontend dockerfile.v0: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount1539107064/Dockerfile: no such file or directory
[maly@localhost problem5]$ sudo docker compose -f ./docker-compose.yml up
[sudo] password for maly:
[+] Running 5/5
# Network problem5_default Created 5.0s
# Volume "problem5_ghost-volume" Created 0.1s
# Volume "problem5_mysql-volume" Created 0.0s
# Container mysql Created 1.2s
# Container ghost Created 294.2s
Attaching to ghost, mysql
mysql | 2022-12-11 19:35:04+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
mysql | 2022-12-11 19:35:22+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'

```

```

ghost | [2022-12-11 19:40:37] INFO Ghost booted in 123.888s
ghost | [2022-12-11 19:40:37] INFO Adding offloaded job to the queue
ghost | [2022-12-11 19:40:37] INFO Scheduling job update-check at 56 18 8 * * *. Next run on: Mon Dec 12 2022 08:18:56 GMT+0000 (Coordinated Universal Time)
ghost | [2022-12-11 19:41:11] INFO "GET /favicon.ico" 200 21ms
ghost | [2022-12-11 19:41:12] INFO "GET /" 200 3735ms
ghost | [2022-12-11 19:41:12] INFO "GET /assets/built/screen.css?v=5f6b240d0c" 200 546ms
ghost | [2022-12-11 19:41:12] INFO "GET /public/cards.min.css?v=5f6b240d0c" 200 598ms
ghost | [2022-12-11 19:41:13] INFO "GET /assets/built/casper.js?v=5f6b240d0c" 200 570ms
ghost | [2022-12-11 19:41:13] INFO "GET /public/cards.min.js?v=5f6b240d0c" 200 580ms
ghost | [2022-12-11 19:41:13] INFO "GET /public/member-attribution.min.js?v=5f6b240d0c" 200 589ms
ghost | [2022-12-11 19:43:24] INFO "GET /favicon.ico" 200 54ms

```



```
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
[root@localhost ~]# docker images
REPOSITORY    TAG        IMAGE ID      CREATED      SIZE
ghost         latest    99526a7a09bc  3 days ago  573MB
python        latest    539eccd5ee4e  3 days ago  932MB
mysql         5.7       d410f4167eea  4 days ago  495MB
mysql         latest    7484689f290f  4 days ago  538MB
httpd         latest    157dcdf23d6c  5 days ago  145MB
nginx         latest    ac8efec875ce  5 days ago  142MB
html-image    latest    031ballab3c6  7 days ago  145MB
flask-app     v5        c457c1c576d1  7 days ago  952MB
nginx         alpine    19dd4d73108a  4 weeks ago 23.5MB
ubuntu        latest    a8780b506fa4  5 weeks ago 77.8MB
kodekloud/simple-webapp latest    c6e3cd9aae36  4 years ago 84.8MB
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS    NAMES
375650d74df9   ghost     "docker-entrypoint.s..." 17 minutes ago Up 9 minutes
0.0.0.0:80->2368/tcp, :::80->2368/tcp    ghost
981bffc58c82   mysql     "docker-entrypoint.s..." 17 minutes ago Up 12 minutes
0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp    mysql
```

4. Problem 4:

Create a Dockerfile to deploy weather-app

application(https://github.com/sabreensalama/Good_Reads_App):






1. You will use node:alpine as the base image.
2. create a directory for the source code: /node/weather-app.
3. Add the code from src to /node/weather-app on the image.
4. Set /node/weather-app as the working directory.
5. Execute a npm install to install the dependencies.
6. Create an argument called APP_VERSION that will set the version of the application.
7. Set an environment variable called NODE_ENV and set it to production.
8. Copy /node/weather-app from the source build stage to /var/weather-app.
9. Set /var/weather-app as the working directory.
10. Expose port 3000.
11. Set ./bin/www as the entrypoint. C2 General
12. Build the image

```
Get Started Dockerfile X
home > maly > docker-labs > lab2 > problem4 > Dockerfile > ...
1 FROM node:alpine
2 #RUN mkdir -p /node/weather-app
3 WORKDIR /node/weather-app
4 COPY package*.json ./
5 RUN npm install
6 #RUN npm install bcrypt --save
7 COPY . .
8 ARG APP_VERSION="1"
9 ENV NODE_ENV=production
10 #RUN mkdir -p /var
11 #RUN rsync
12 WORKDIR /var/weather-app
13 #/node/weather-app/var/weather-app
14 COPY . .
15 EXPOSE 3000
16 ENTRYPOINT [".bin/www"]
17 CMD ["npm","start"]
```

```

[maly@localhost problem4]$ sudo docker build -t weatherapp-image:v2 .
Sending build context to Docker daemon 137.2kB
Step 1/13 : FROM node:alpine
----> f01bef8b5e92
Step 2/13 : WORKDIR /node/weather-app
----> Using cache
----> b4f97be6ecla
Step 3/13 : COPY package*.json ./
----> Using cache
----> 121cd993878a
Step 4/13 : RUN npm install
----> Using cache
----> 22650edbbac5
Step 5/13 : RUN npm install bcrypt --save
----> Using cache
----> e207ce4a0e35
Step 6/13 : COPY . .
----> 95f0f45ef2df
Step 7/13 : ARG APP_VERSION="1"
----> Running in 6775875fb1a2
Removing intermediate container 6775875fb1a2
----> 5574ace4dcf3
Step 8/13 : ENV NODE_ENV=production
----> Running in d7f6cc535641
Removing intermediate container d7f6cc535641
----> 0ada4d20ea4c
Step 9/13 : WORKDIR /var/weather-app
----> Running in 01b548100c4c
Removing intermediate container 01b548100c4c
----> 89332e71dc9d

```

Home	docker-labs	lab2	problem4 ▾	bin ▶	
Name					S
	app.js				2
	bin				1
	Dockerfile				3
	package.json				8
	package-lock.json				1