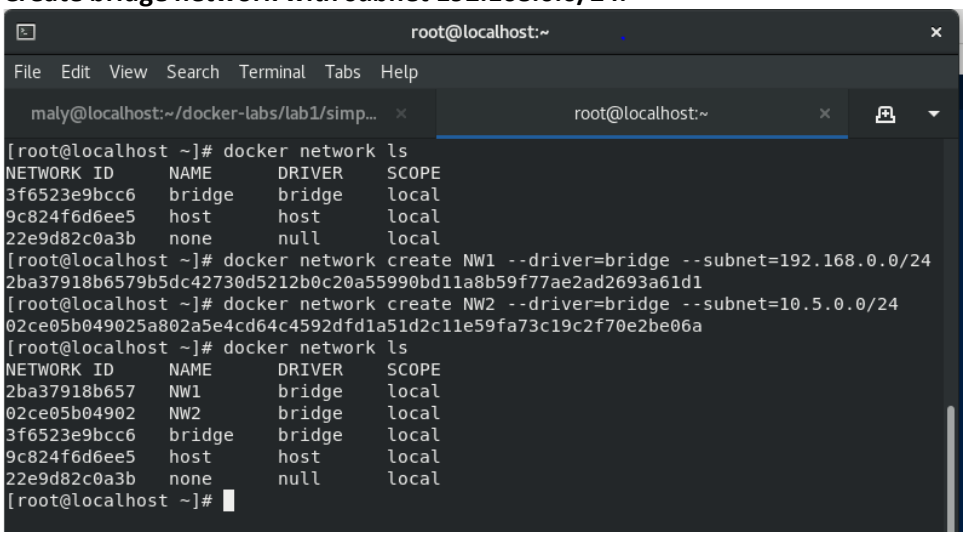
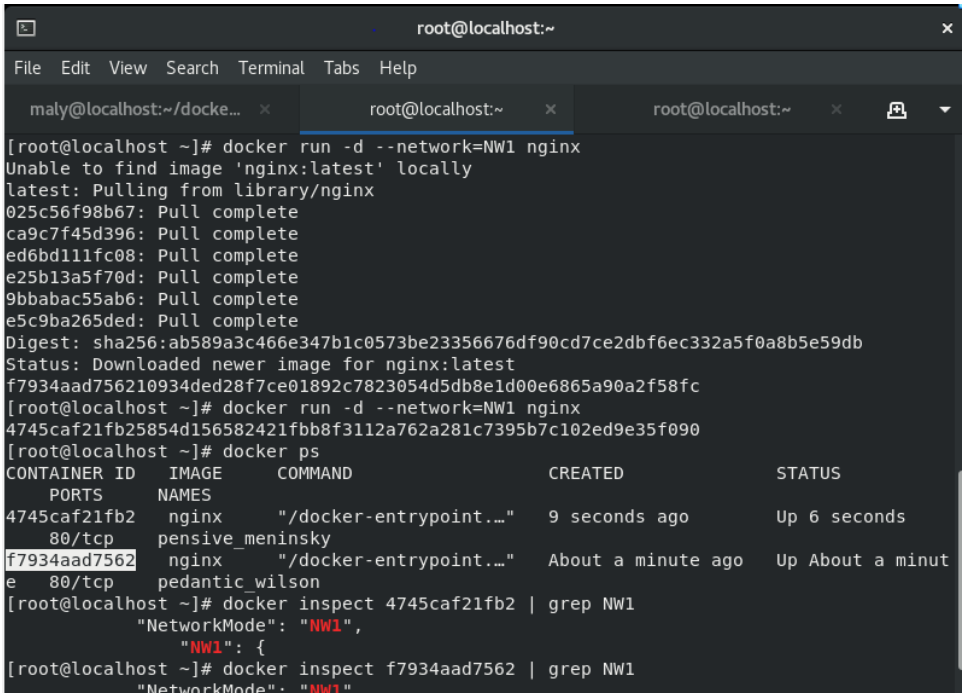
Docker Lab2

1. Problem 1:

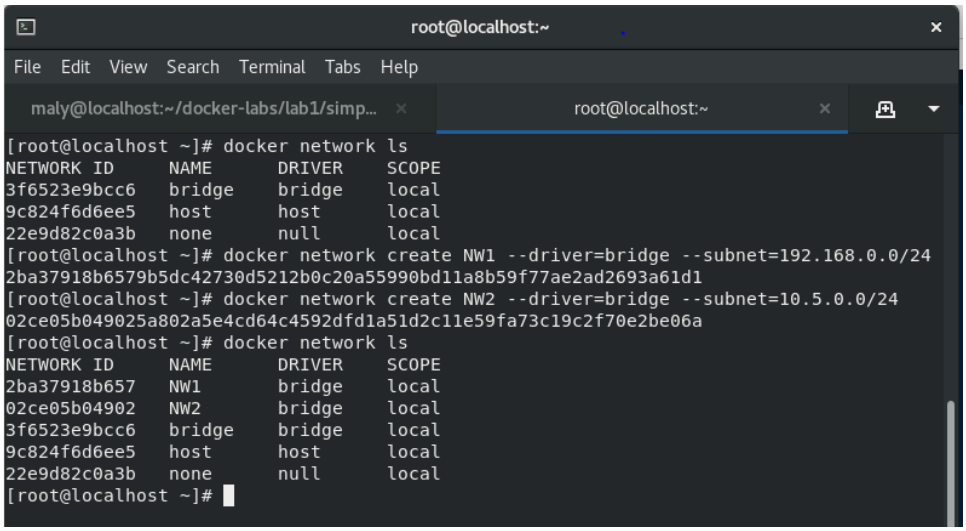
Create bridge network with subnet 192.168.0.0/24.



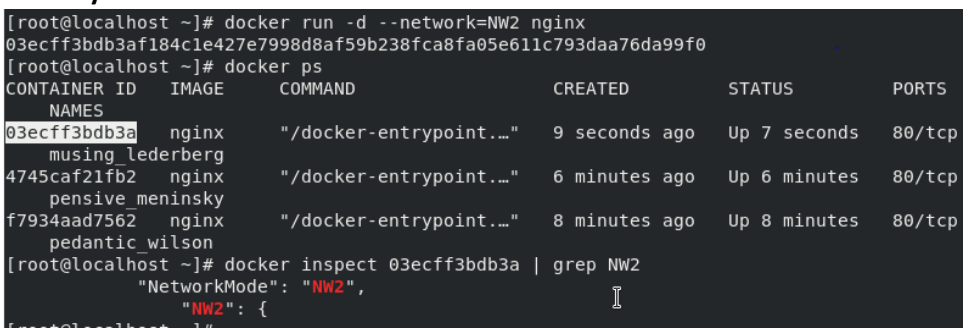
Run 2 containers and attach containers to this network.



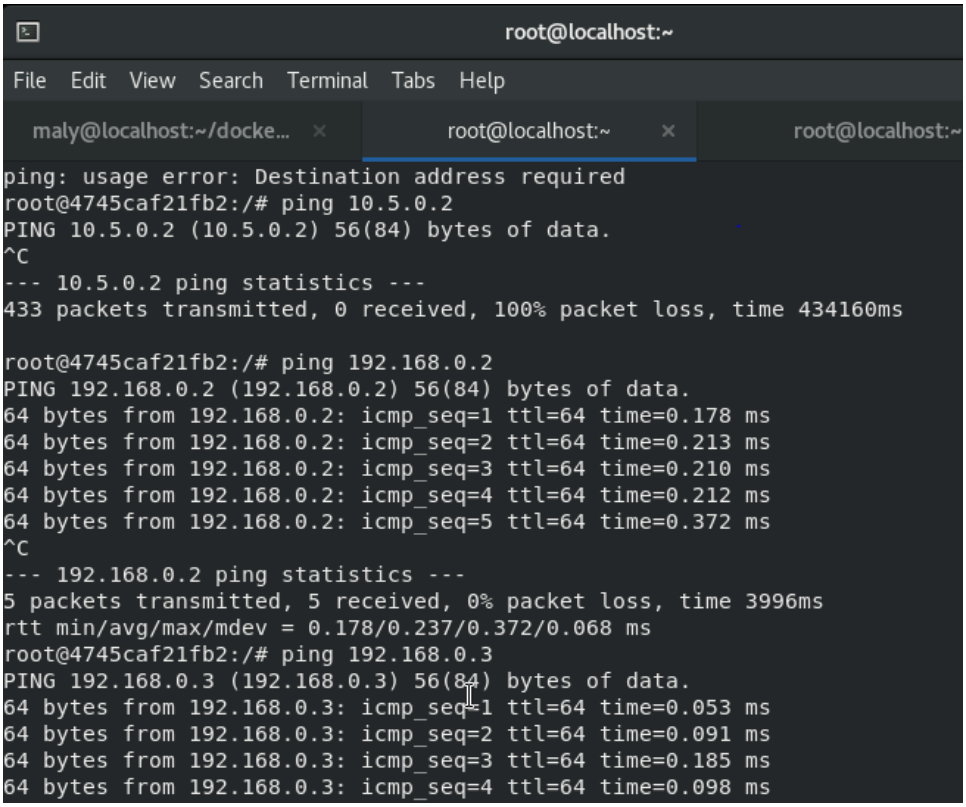
Create another bridge network with subnet 10.5.0.0/24.



Run any container and attach it to the new network.



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



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

home > maly > docker-labs > lab2 > problem2 > helloworld.html > html > head

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Hello, World!</title>
5      </head>
6      <body>
7          <p>Hello, World!</p>
8      </body>
9  </html>
```

Get Started | Dockerfile | helloworld.html

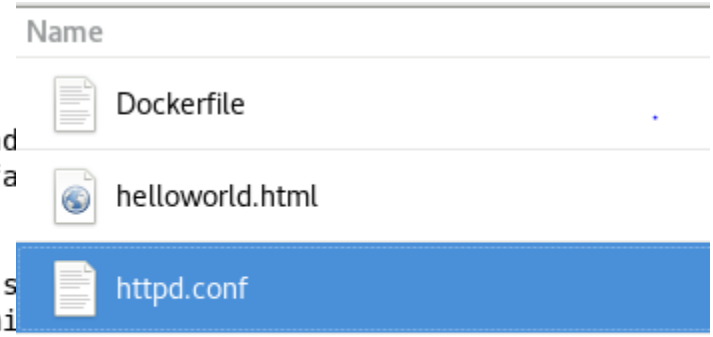
home > maly > docker-labs > lab2 > problem2 > Dockerfile > ...

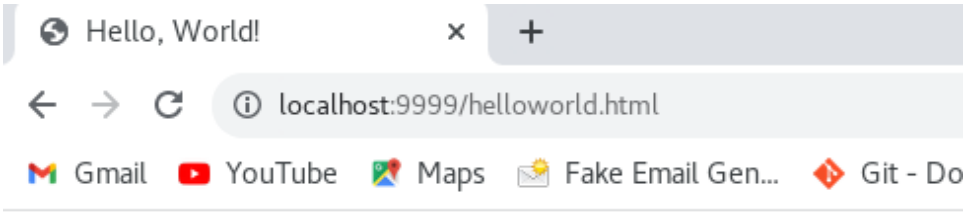
```
1  FROM httpd
2  COPY ./helloworld.html /var/www/html/
3  COPY ./httpd.conf /etc/httpd/conf/httpd.conf
4  ENV CONTAINER=docker
5  EXPOSE 9999
6  CMD echo $CONTAINER && httpd -D FOREGROUND
```

```
[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
 ---> 157dcdf23d6c
Step 2/6 : COPY ./helloworld.html /var/www/html/
 ---> e0995cd3dddd
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 fe9ab1e32c5c
Removing intermediate container fe9ab1e32c5c
 ---> 4b1f4216a9bf
Step 6/6 : CMD echo $CONTAINER && httpd -D FOREGROUND
 ---> Running in 2cc3739aabdd
Removing intermediate container 2cc3739aabdd
 ---> 031ba11ab3c6
Successfully built 031ba11ab3c6
Successfully tagged html-image:latest
[maly@localhost problem2]$ 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
```

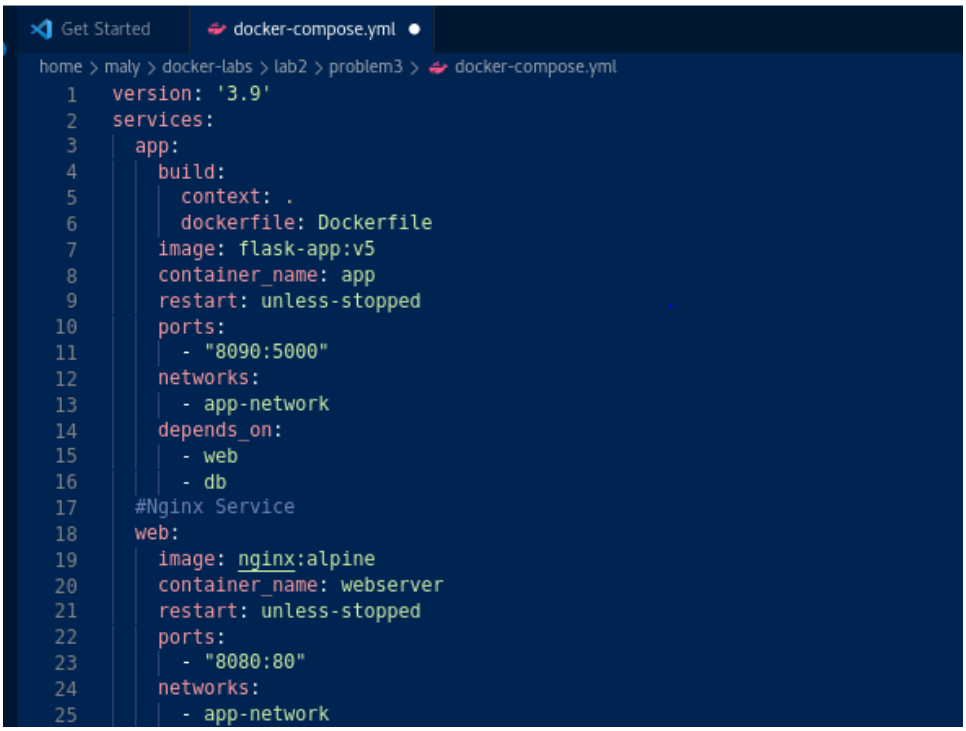


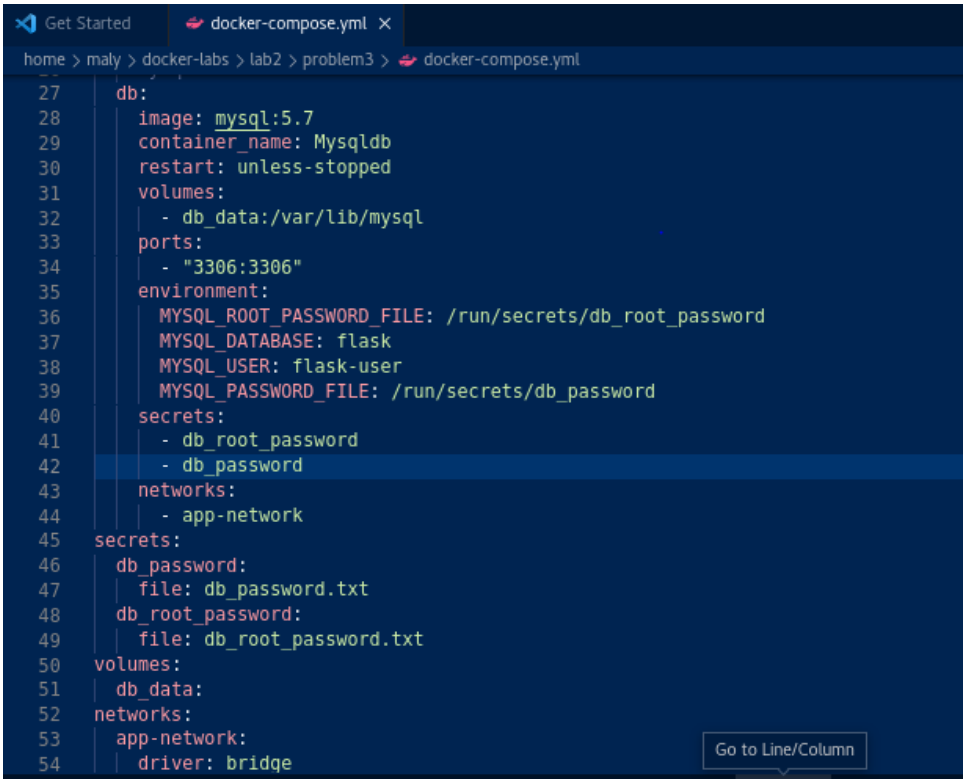


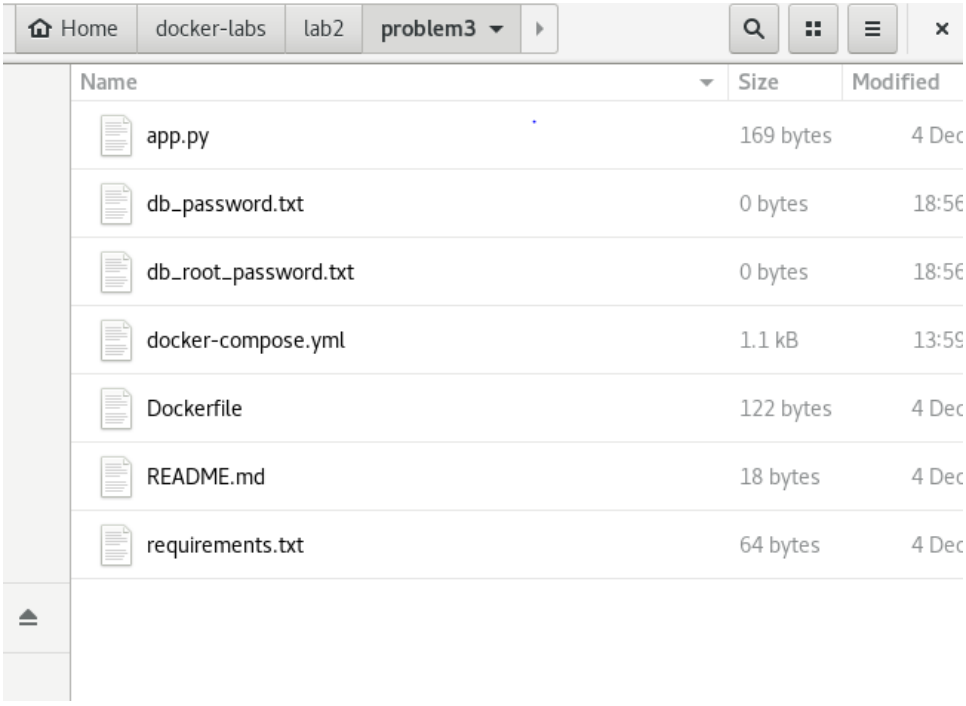
Hello, World!

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

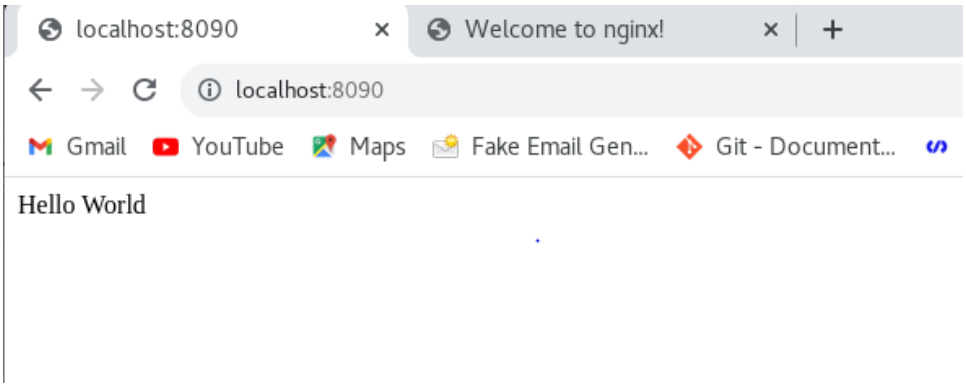


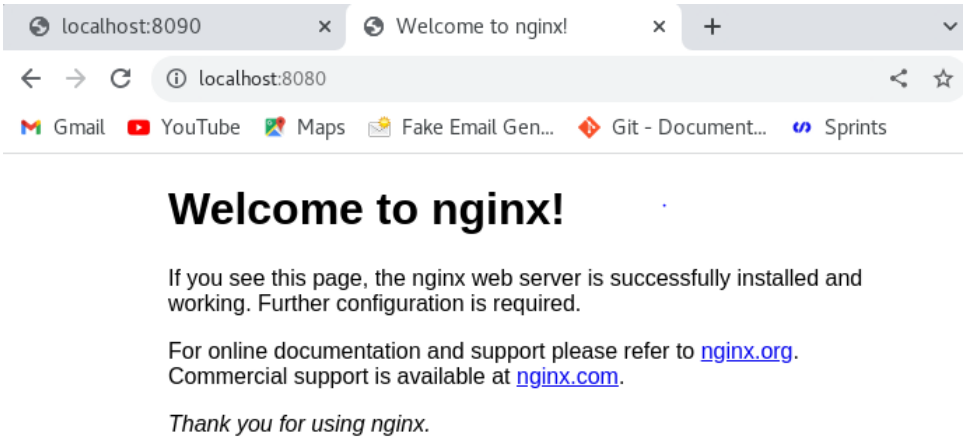





```
[maly@localhost problem3]$ sudo docker compose -f ./docker-compose.yml  up
[sudo] password for maly:
[+] Running 2/2
 ⠿ Container Mysqldb  Created                                                    0.4s
 ⠿ Container app      Created                                                    0.5s
Attaching to Mysqldb, app, webserver
webserver  | /docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configu
ration
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.c
onf
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.
```

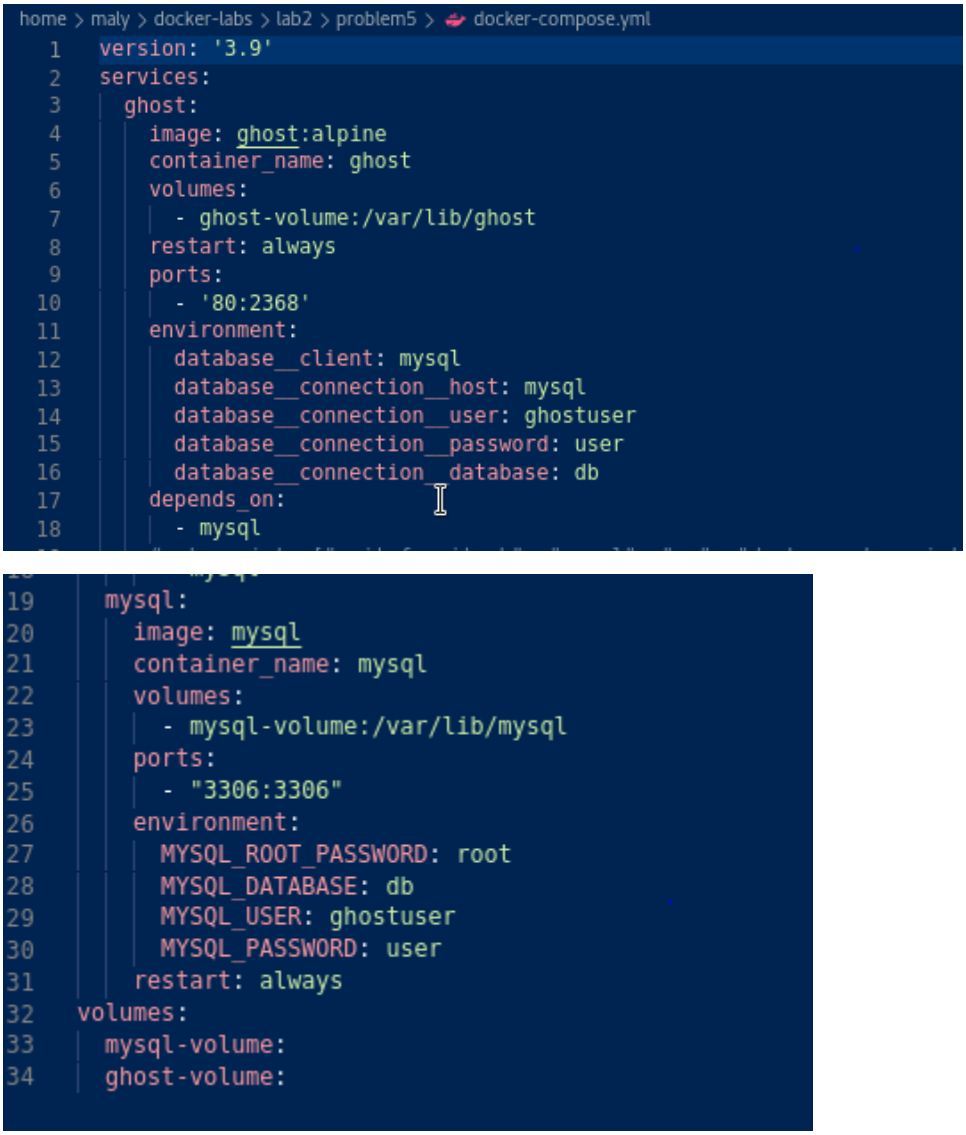





```
[root@localhost ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
                    PORTS                                          NAMES
bdcf392fc442   flask-app:v5   "python app.py"          16 minutes ago   Up 16 mi
nutes               0.0.0.0:8090->5000/tcp, :::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 ~]#
```

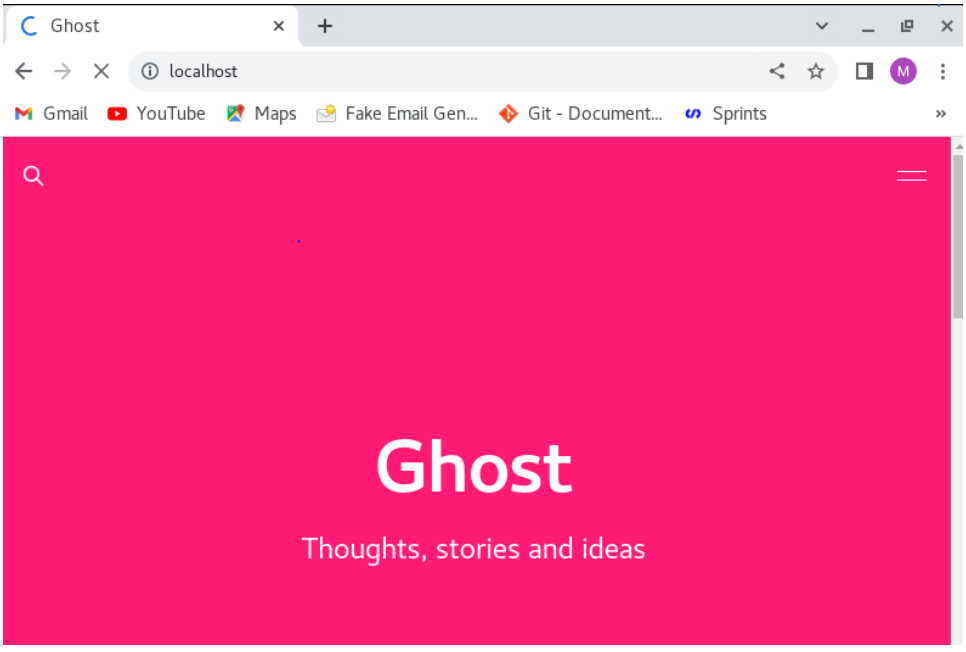
5. Problem 5:

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



```
 ⠿ 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 dockerfil
e.v0: failed to read dockerfile: open /var/lib/docker/tmp/buildkit-mount1539107064/Docker
file: 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 Serve
r 8.0.31-1.el8 started.
mysql  | 2022-12-11 19:35:22+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysq
```

```
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 ru
n 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    031ba11ab3c6   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.

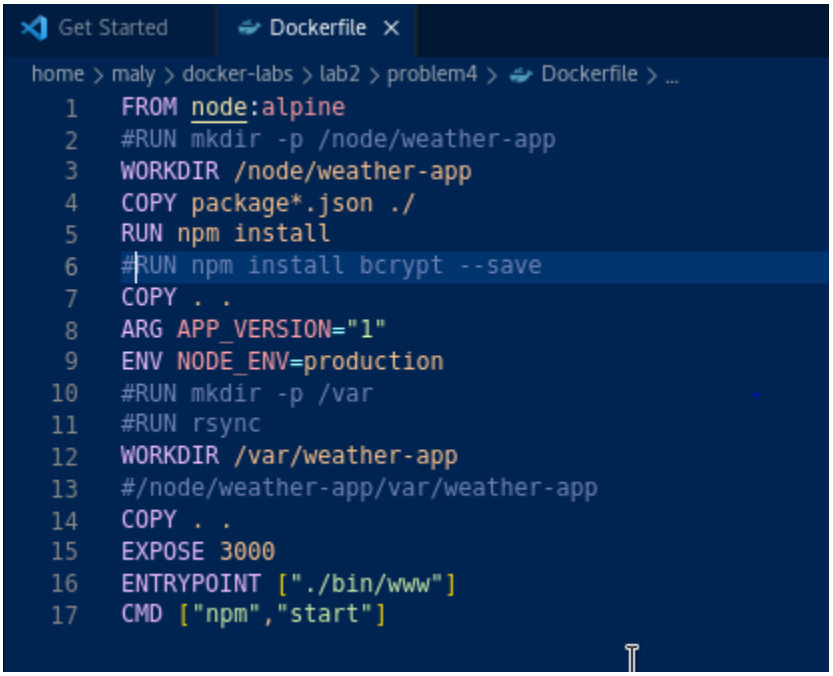
8. Copy /node/weather-app from the source build stage to /var/weather-app.

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
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
 ---> b4f97be6ec1a
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
```