

1-docker run --name nginx -p 38282:8080 nginx:alpine.

```
helmy@ubuntu: ~  
helmy@ubuntu:~$ docker run -d --name nginx -p 38282:8080 nginx:alpine  
b5d5bdf1a771dbbd20d79129f4a3c85401896fe00d543a355eff9464ea8bc505  
helmy@ubuntu:~$ docker ps  
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS  
PORTS          NAMES  
b5d5bdf1a771   nginx:alpine   "/docker-entrypoint. ...." 14 seconds ago Up 13 se  
conds         80/tcp, 0.0.0.0:38282->8080/tcp, :::38282->8080/tcp   nginx  
helmy@ubuntu:~$
```

2-create ubuntu image and check the size of it

```
helmy@ubuntu:~/ubuntu$ touch Dockerfile  
helmy@ubuntu:~/ubuntu$ vim  
vim          vim.basic  vimdiff     vim.tiny    vimtutor  
helmy@ubuntu:~/ubuntu$ vim Dockerfile  
helmy@ubuntu:~/ubuntu$ docker build -t ubuntu .  
[+] Building 0.1s (5/5) FINISHED  
=> [internal] load build definition from Dockerfile                                0.0s  
=> == transferring dockerfile: 58B                                              0.0s  
=> [internal] load .dockerignore                                                0.0s  
=> == transferring context: 2B                                                  0.0s  
=> [internal] load metadata for docker.io/library/ubuntu:latest                0.0s  
=> CACHED [1/1] FROM docker.io/library/ubuntu:latest                          0.0s  
=> exporting to image                                                           0.0s  
=> == exporting layers                                                           0.0s  
=> == writing image sha256:bb59a1c2f7b943b56c8b224b0d0039412d8594a5eebf5      0.0s  
=> == naming to docker.io/library/ubuntu                                       0.0s  
helmy@ubuntu:~/ubuntu$ docker images  
REPOSITORY      TAG          IMAGE ID          CREATED          SIZE  
node            node         bd472681863f     4 days ago      187MB  
<none>          <none>       5c89e3f72667     2 weeks ago     7.33MB  
helloworled2    latest      0fdafaa6892e     2 weeks ago     7.33MB  
hello-world     latest      9c7a54a9a43c     3 weeks ago     13.3kB  
nginx           latest      448a08f1d2f9     3 weeks ago     142MB  
ubuntu          <none>       3b418d7b466a     4 weeks ago     77.8MB  
ubuntu          latest      bb59a1c2f7b9     4 weeks ago     77.8MB  
nginx           alpine      8e75cbc5b25c     8 weeks ago     41MB  
helmy@ubuntu:~/ubuntu$
```

```
Dockerfile  
~/ubuntu  
Open  Save  
1 FROM ubuntu:latest  
2 |
```

3- Run a container named blue-app using image kodekloud/simple-webapp and set the environment variable APP_COLOR to blue. Make the application available on port 38282 on the host. The application listens on port 8080.



Hello from eb5fcd5c9846!

```
helmy@ubuntu: ~  
helmy@ubuntu:~$ docker run --name blue_app -e APP_COLOR=blue -p 38282:8080 kodekloud/simple-webapp  
This is a sample web application that displays a colored background.  
A color can be specified in two ways.  
  
1. As a command line argument with --color as the argument. Accepts one of red, green, blue, blue2, pink, darkblue  
2. As an Environment variable APP_COLOR. Accepts one of red, green, blue, blue2, pink, darkblue  
3. If none of the above then a random color is picked from the above list.  
Note: Command line argument precedes over environment variable.  
  
No Command line argument. Color from environment variable =blue  
* Serving Flask app "app" (lazy loading)  
* Environment: production  
  WARNING: Do not use the development server in a production environment.  
  Use a production WSGI server instead.  
* Debug mode: off  
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

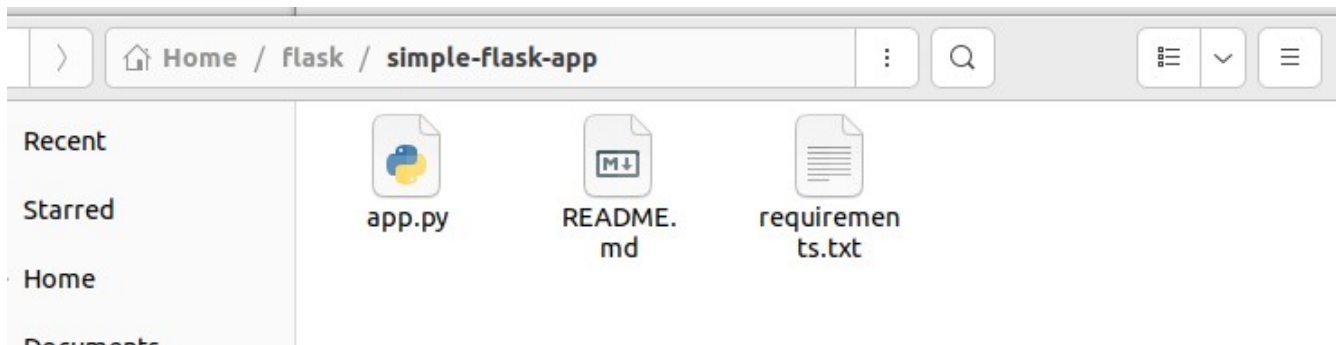
4-Deploy a mysql database using the mysql image and name it mysql-db Set the database password to use db_pass123 then inspect it to check the value

```
helmy@ubuntu: ~  
helmy@ubuntu:~$ docker pull mysql  
Using default tag: latest  
latest: Pulling from library/mysql  
90e2fb2facff: Pull complete  
ba60eb20fd5f: Pull complete  
4f509402d469: Pull complete  
496c2cfa6815: Pull complete  
8ec1dfa9522c: Pull complete  
6dec7ba896f8: Pull complete  
dc9ff75362b0: Pull complete  
73e4682f9014: Pull complete  
9ffdeecd6fb6: Pull complete  
a4346ccfb53f: Pull complete  
434c13bc32de: Pull complete  
Digest: sha256:d6164ff4855b9b3f2c7748c6ec564ccff841f79a7023db0f9293143481a44b6e  
Status: Downloaded newer image for mysql:latest  
docker.io/library/mysql:latest
```

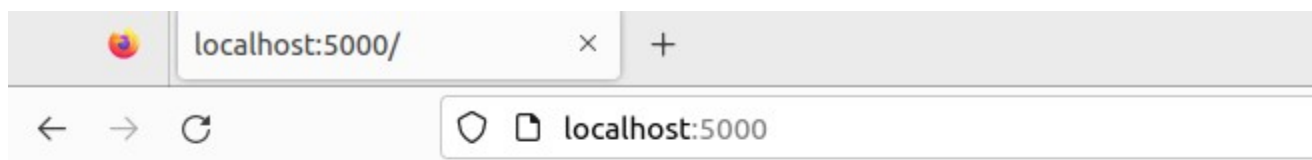
```
helmy@ubuntu: ~  
helmy@ubuntu:~$ docker run --name mysql-db -e MYSQL_ROOT_PASSWORD=db_pass123 -d mysql  
ccc87cbd3c0e5214946efd980cf4276cba6f3d3416c2d5aaecc534fabad44521  
helmy@ubuntu:~$ docker ps  
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS  
NAME          NAMES  
ccc87cbd3c0e   mysql     "docker-entrypoint.s..." 2 minutes ago Up 2 minutes  3306/  
tcp, 33060/tcp mysql-db  
helmy@ubuntu:~$ docker images  
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE  
mysql           latest     05db07cd74c0  2 days ago    565MB  
node            node       bd472681863f  4 days ago    187MB  
helloworld2     latest     0fdafaa6892e  2 weeks ago   7.33MB  
<none>          <none>     5c89e3f72667  2 weeks ago   7.33MB  
hello-world     latest     9c7a54a9a43c  3 weeks ago   13.3kB  
nginx           latest     448a08f1d2f9  3 weeks ago   142MB  
ubuntu          <none>     3b418d7b466a  4 weeks ago   77.8MB  
ubuntu          latest     bb59a1c2f7b9  4 weeks ago   77.8MB  
nginx           alpine     8e75cbc5b25c  8 weeks ago   41MB  
kodekloud/simple-webapp latest     c6e3cd9aae36  4 years ago   84.8MB  
helmy@ubuntu:~$
```

5- pull the code from <https://github.com/sabreensalama/dockerize-node-app-task> and create a docker file for this flask app

```
helmy@ubuntu: ~/flask
helmy@ubuntu:~/flask$ git clone https://github.com/sabreensalama/simple-flask-app.git
Cloning into 'simple-flask-app'...
remote: Enumerating objects: 12, done.
remote: Total 12 (delta 0), reused 0 (delta 0), pack-reused 12
Receiving objects: 100% (12/12), done.
Resolving deltas: 100% (1/1), done.
helmy@ubuntu:~/flask$
```



```
helmy@ubuntu: ~/flask/simple-flask-app
helmy@ubuntu:~/flask/simple-flask-app$ docker build .
[+] Building 239.8s (9/9) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 154B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/python:3.8                    3.8s
=> [1/4] FROM docker.io/library/python:3.8@sha256:e411647c253b75948394a3 221.0s
=> => resolve docker.io/library/python:3.8@sha256:e411647c253b75948394a3 0.0s
=> => sha256:0869a4e92f01e49dfefb23d9ee3cf58f212fb8613e8 7.56kB / 7.56kB 0.0s
=> => sha256:75256935197ed1bb3b994a77c01efa00349b9010 54.58MB / 54.58MB 90.2s
=> => sha256:e411647c253b75948394a343b13ff32b5674687df0c 1.86kB / 1.86kB 0.0s
=> => sha256:bd73737482dd5575526c7207872963479808d979 55.05MB / 55.05MB 71.3s
=> => sha256:6710592d62aa1338ac1c1c363dedc255659f666c 15.76MB / 15.76MB 23.1s
=> => sha256:6947ab0ea011dd1879a12bcc4e9709371ed4b9fb078 2.01kB / 2.01kB 0.0s
=> => sha256:c1e5026c64577dee4b6bac20b18196f964a41 196.85MB / 196.85MB 200.2s
=> => extracting sha256:bd73737482dd5575526c7207872963479808d979ab2741c3 7.4s
```



Hello World