EPAM University Programs

Cloud&DevOps Fundamentals Autumn 2022

Containerization. Docker.

PERFORMING EXERCISES FROM LECTURE №1

1) **I update the apt package index and install packages to allow apt to use the Docker's repository over HTTPS:**

```
vladyslav@docker:~$ sudo apt-get update
[sudo] password for vladyslav:
Hit:1 http://ua.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metadata [40,7 kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/universe amd64 DEP-11 Metadata [93,9 kB]
Get:7 http://security.ubuntu.com/ubuntu focal-security/multiverse amd64 DEP-11 Metadata [940 B]
Fetched 249 kB in 1s (287 kB/s)
Reading package lists... Done
vladyslav@docker:~$ sudo apt-get install \
>     ca-certificates \
>     curl \
>     gnupg \
>     lsb-release
```

2) **I add the Docker's official GPG key:**

```
vladyslav@docker:~$ sudo mkdir -p /etc/apt/keyrings
vladyslav@docker:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /
etc/apt/keyrings/docker.gpg
vladyslav@docker:~$
```

3) **I use the following command to set up the repository:**

```
vladyslav@docker:~$ echo \
>   "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.
docker.com/linux/ubuntu \
>   $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
vladyslav@docker:~$
```

4) **I update the apt package index:**

```
vladyslav@docker:~$ sudo apt-get update
Hit:1 http://ua.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://ua.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://ua.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:5 https://download.docker.com/linux/ubuntu focal InRelease [57,7 kB]
Get:6 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [22,3 kB]
Fetched 80,0 kB in 1s (82,7 kB/s)
Reading package lists... Done
```

5) **I install Docker Engine, containerd, and Docker Compose:**

```
vladyslav@docker:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin git git-man liberror-perl pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc git-el
  git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin
  docker-scan-plugin git git-man liberror-perl pigz slirp4netns
0 upgraded, 11 newly installed, 0 to remove and 345 not upgraded.
Need to get 117 MB of archives.
After this operation, 466 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

6) **I verify that the Docker Engine installation is successful by running the hello-world image:**

```
vladyslav@docker:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:c77be1d3a47d0caf71a82dd893ee61ce01f32fc758031a6ec4cf1389248bb833
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

7) **I add a user to the “docker” group:**

```
vladyslav@docker:~$ sudo usermod -aG docker vladyslav
vladyslav@docker:~$ id vladyslav
uid=1000(vladyslav) gid=1000(vladyslav) groups=1000(vladyslav),4(adm),24(cdrom),27(sudo),30(dip),46(p
lugdev),120(lpadmin),132(lxd),133(sambashare),996(docker)
```

8) **I create a directory for Dockerfile and create a Dockerfile inside that:**

```
vladyslav@docker:~$ mkdir docker_task
vladyslav@docker:~$ cd docker_task/
vladyslav@docker:~/docker_task$ touch Dockerfile
vladyslav@docker:~/docker_task$
```

9) I fill the file with content:

```
FROM ubuntu:20.04
ENV TZ=Europe/Kiev
RUN apt-get -y update
RUN DEBIAN_FRONTEND="noninteractive" \
    apt-get -y install apache2
RUN echo 'Some text 1' > /var/www/html/index.html
RUN echo 'Some text 2' >> /var/www/html/index.html

CMD ["/usr/sbin/apache2ctl", "-DFOREGROUND"]

EXPOSE 80
```

10) I build an image:

```
vladyslav@docker:~/docker_task$ docker build -t firstapp:v1 .
Sending build context to Docker daemon  2.048kB
Step 1/8 : FROM ubuntu:20.04
```

11) I run a container:

```
vladyslav@docker:~/docker_task$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
firstapp      v1        8b72a90c2e46   8 minutes ago   226MB
ubuntu        20.04     d5447fc01ae6   2 weeks ago     72.8MB
hello-world   latest    feb5d9fea6a5   15 months ago   13.3kB
vladyslav@docker:~/docker_task$ docker run -d -p 80:80 --name first_container firstapp:v1
0f36bfa541b3ddbb0a5122d26c16a1b2586976a383ab23ee467433dd970e9eb5
vladyslav@docker:~/docker_task$ docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED         STATUS         PORTS
                    NAMES
0f36bfa541b3   firstapp:v1   "/usr/sbin/apache2ct…"   4 seconds ago   Up 3 seconds   0.0.0.0:80->80/t
cp, :::80->80/tcp   first_container
```

12) I check the container working using a browser:

Some text 1 Some text 2

13) I create a similar container, but based on centos7:

```
vladyslav@docker:~/docker_task$ cd ..
vladyslav@docker:~$ mkdir docker_task_centos7
vladyslav@docker:~$ cd docker_task_centos7/
vladyslav@docker:~/docker_task_centos7$ touch Dockerfile
vladyslav@docker:~/docker_task_centos7$
```

Dockerfile:

```
FROM centos:7

RUN yum -y update
RUN yum -y install httpd
RUN echo 'Some text 1' > /var/www/html/index.html
RUN echo 'Some text 2' >> /var/www/html/index.html

CMD ["/usr/sbin/httpd", "-DFOREGROUND"]

EXPOSE 80
```

```
vladyslav@docker:~/docker_task_centos7$ docker build -t app_centos .
Sending build context to Docker daemon  2.048kB
Step 1/7 : FROM centos:7
7: Pulling from library/centos
2d473b07cdd5: Pull complete
Digest: sha256:be65f488b7764ad3638f236b7b515b3678369a5124c47b8d32916d6487418ea4
```

```
vladyslav@docker:~/docker_task_centos7$ docker run -d -p 81:80 --name centos_app_container app_centos
c2f292f52e593799f2b0a729ccd638f4003cfc55ee8eb84b47e5136642173e46
vladyslav@docker:~/docker_task_centos7$
```

14) I check the container working using a browser:

Some text 1 Some text 2

15) I clean docker using the “docker system prune -a” command:

```
vladyslav@docker:~/docker_task_centos7$ docker stop centos_app_container first_container
centos_app_container
first_container
vladyslav@docker:~/docker_task_centos7$ docker system prune -a
WARNING! This will remove:
  - all stopped containers
  - all networks not used by at least one container
  - all images without at least one container associated to them
  - all build cache

Are you sure you want to continue? [y/N] y
Deleted Containers:
c2f292f52e593799f2b0a729ccd638f4003cfc55ee8eb84b47e5136642173e46
0f36bfa541b3ddbb0a5122d26c16a1b2586976a383ab23ee467433dd970e9eb5

Deleted Images:
untagged: firstapp:v1
deleted: sha256:8b72a90c2e461c26d2d8fd037f593096f9d97a1f3982e66a544e86465a90e6c5
deleted: sha256:a0561a14f0e102d48c7c5d2eab16eea7909ffd61b9539961896be927a968275c
deleted: sha256:c5923e9d608b137724c0c189db5675c04af85accc2d61f0e2642eb048989c27b
deleted: sha256:9ce9ee3f1c51e246e2241850d40b61cf27eb8bd088a884268d6ca922f05368c5
deleted: sha256:babc70b4eded5252b42ba4cb3f76b67a6701f8f441bd6ba1c148e491dc426e18
deleted: sha256:5137449b494b3c3aa42f745e2012292180bc9d391ce46bd42d2849b5eaffe85f
deleted: sha256:06a5bae8a9f320ba4f86e138e6c23fee3a94b6ffc5d63582019d33206776a759
deleted: sha256:7002cbd7627929e4202b951dd1b2ce272a729390061c588a6ff8ae87ca6cbb9c
deleted: sha256:4a342aa420d77e1d6d1a916e459f2128712b22610f225dcaaa269a6c7c698ee4
deleted: sha256:847c24438920b7b87b36aaa8e33a75a82fa6dea90dd1799ea1924535be104cb9
deleted: sha256:7bca02d7e120f4a5e242bac4e9b12020c073014b6fee9784781286636160eab9
deleted: sha256:d5447fc01ae62c20beffbfa50bc51b2797f9d7ebae031b8c2245b5be8ff1c75b
deleted: sha256:0002c93bdb3704dd9e36ce5153ef637f84de253015f3ee330468dccdeacad60b
untagged: app_centos:latest
deleted: sha256:b1efc8333574c71a05b04b0c10738e78bccc7bc7fcbd40ae420d50f727b9037b
deleted: sha256:53aab35610841aee658f5ba5dac63e14b1295a7a3383307a858c4aaba61b5f93
deleted: sha256:9a36e279b3c0e9d29d36fb7e4b72b644e402c34de2d888823653f4104f89991c
deleted: sha256:66660a28e2a2fb3aecbfa7b210e7d81c421e06dcf13fa92dbf6369260054fd08
deleted: sha256:4d803ed380d4f5fccf2570837a5b8c7e012193a5b8210cafa57ae3d719bf695f
deleted: sha256:deb031f195f43ce077db2d011ca82db7a258e22209c9c781722bce874457f195
deleted: sha256:7aeca0235ab4354aaaaaa88548e20d7dcd07599c5f8f338199dca41903194b19
deleted: sha256:e8447e8de3e5ba3683658b19469d34b44128c6acde4d5450134ecaf5b123a6b5
deleted: sha256:130527f199a312dea3e2a877e15a669daef3bdd0b6c599a0434f0da6f397a52f
deleted: sha256:fda98c41d09738222d564b145c1f6f8ead66a57c0d3f16cf661c82d67e09f02b
deleted: sha256:eeb6ee3f44bd0b5103bb561b4c16bcb82328cfe5809ab675bb17ab3a16c517c9
deleted: sha256:174f5685490326fc0a1c0f5570b8663732189b327007e47ff13d2ca59673db02

Total reclaimed space: 986.6MB
```

PERFORMING EXERCISES FROM LECTURE №2

1) I create a directory called flask-app to store the application files there:

```
vladyslav@docker:~/Lecture2$ mkdir flask-app
vladyslav@docker:~/Lecture2$ cd flask-app/
vladyslav@docker:~/Lecture2/flask-app$ touch app.py
vladyslav@docker:~/Lecture2/flask-app$ touch requirements.txt
vladyslav@docker:~/Lecture2/flask-app$ mkdir templates
vladyslav@docker:~/Lecture2/flask-app$ touch templates/index.html
vladyslav@docker:~/Lecture2/flask-app$ touch Dockerfile
vladyslav@docker:~/Lecture2/flask-app$
```

2) I add content to the app.py file:

```
from flask import Flask, render_template
import random

app = Flask(__name__)
images = [
    "https://images.pexels.com/photos/9754/mountains-clouds-forest-fog.jpg?auto=compress&cs=tinysrgb&
w=1260&h=750&dpr=1",
    "https://images.pexels.com/photos/1666012/pexels-photo-1666012.jpeg?auto=compress&cs=tinysrgb&w=1
260&h=750&dpr=1",
    "https://images.pexels.com/photos/1450082/pexels-photo-1450082.jpeg?auto=compress&cs=tinysrgb&w=1
260&h=750&dpr=1",
    "https://images.pexels.com/photos/552785/pexels-photo-552785.jpeg?auto=compress&cs=tinysrgb&w=126
0&h=750&dpr=1",
    "https://images.pexels.com/photos/346529/pexels-photo-346529.jpeg?auto=compress&cs=tinysrgb&w=126
0&h=750&dpr=1",
    "https://images.pexels.com/photos/629161/pexels-photo-629161.jpeg?auto=compress&cs=tinysrgb&w=126
0&h=750&dpr=1"
        ]

@app.route('/')
def index():
    url = random.choice(images)
    return render_template('index.html', url=url)

if __name__ == "__main__":
    app.run(host="0.0.0.0")
```

3) I write a Dockerfile code:

```
FROM alpine:3.5

RUN apk add --update py2-pip

COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt

COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/

EXPOSE 5000

CMD ["python", "/usr/src/app/app.py"]
```

4) I fill the index.html file with content:

```
<html>
  <head>
    <style type="text/css">
      body {
        background: black;
        color: green;
      }
      div.container {
        max-width: 800px;
        margin: 50px auto;
        border: 20px solid white;
        padding: 10px;
        text-align: center;
      }
      div.container img {
        width: 90%;
        max-height: auto;
      }
      h4 {
        font-size: 25px;
      }
    </style>
  </head>
  <body>
     <div class="container">
       <h4>Python Application</h4>
       <h4>Random images of mountains</h4>
       <br>
       <img src="{{url}}">
     </div>
  </body>
</html>
```

5) I write requirements in the requirements.txt file:

```
Flask==0.10.1
~
~
```

6) I build an image:

```
vladyslav@docker:~/Lecture2/flask-app$ docker build -t vladyslavteron/flask_app .
Sending build context to Docker daemon  6.144kB
Step 1/8 : FROM alpine:3.5
3.5: Pulling from library/alpine
8cae0e1ac61c: Pull complete
Digest: sha256:66952b313e51c3bd1987d7c4ddf5dba9bc0fb6e524eed2448fa660246b3e76ec
Status: Downloaded newer image for alpine:3.5
 ---> f80194ae2e0c
Step 2/8 : RUN apk add --update py2-pip
 ---> Running in 352d50af9bbc
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.5/community/x86_64/APKINDEX.tar.gz
(1/12) Installing libbz2 (1.0.6-r5)
```

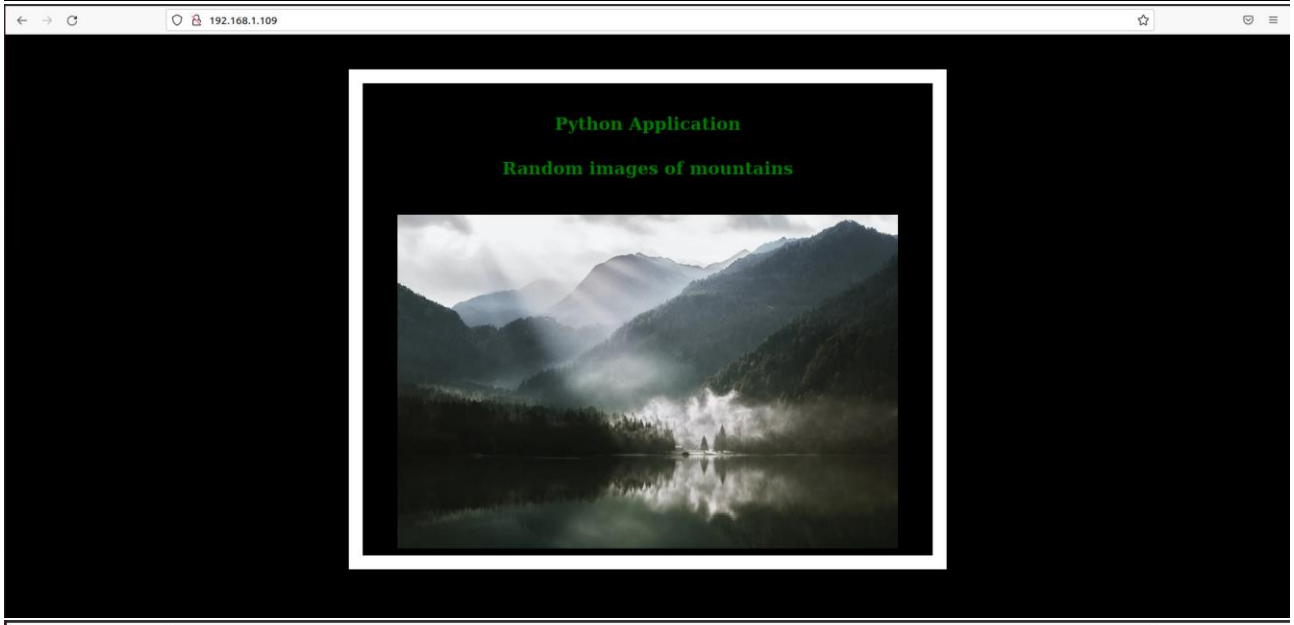
...

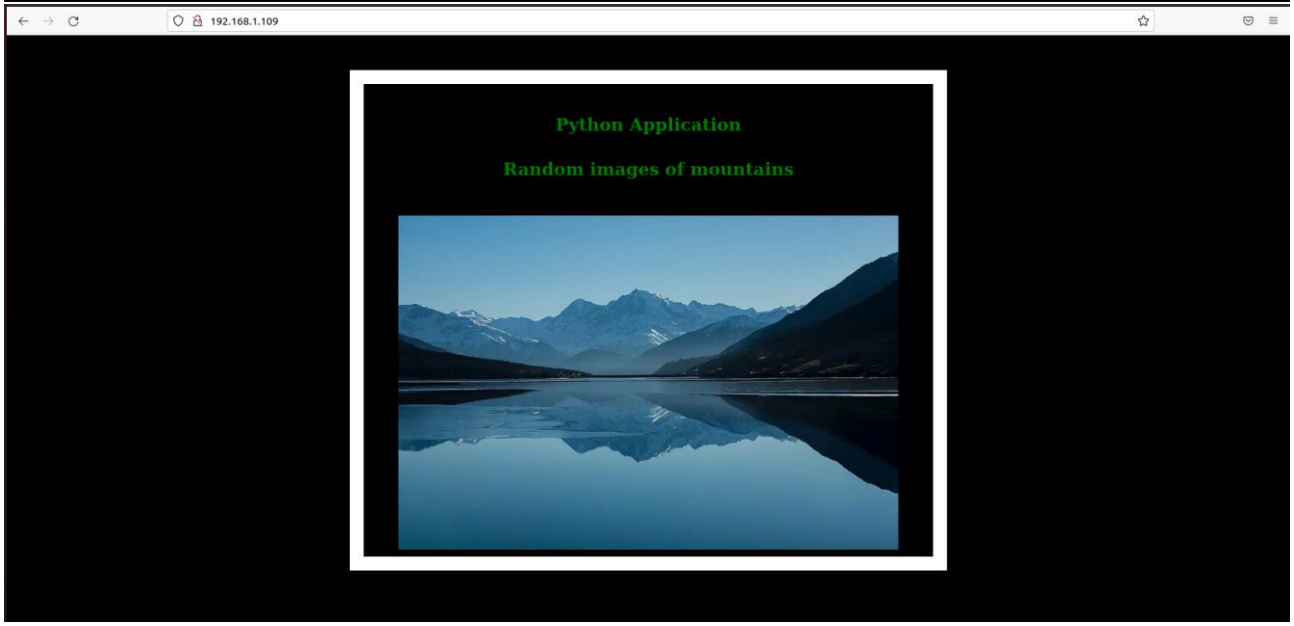
```
Step 5/8 : COPY app.py /usr/src/app/
 ---> a49376b66538
Step 6/8 : COPY templates/index.html /usr/src/app/templates/
 ---> c570d6e85a38
Step 7/8 : EXPOSE 5000
 ---> Running in 331e2c3dbb6b
Removing intermediate container 331e2c3dbb6b
 ---> c63e82e50658
Step 8/8 : CMD ["python", "/usr/src/app/app.py"]
 ---> Running in 68187b91bd3d
Removing intermediate container 68187b91bd3d
 ---> 0b1784199fc9
Successfully built 0b1784199fc9
Successfully tagged vladyslavteron/flask_app:latest
vladyslav@docker:~/Lecture2/flask-app$
```

7) I run a container and check the correctness of the application:

```
vladyslav@docker:~/Lecture2/flask-app$ docker run -p 8000:5000 --name my_flask_app vladyslavteron/fla
sk_app
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

The app displays random images of mountains:





8) I login to DockerHub:

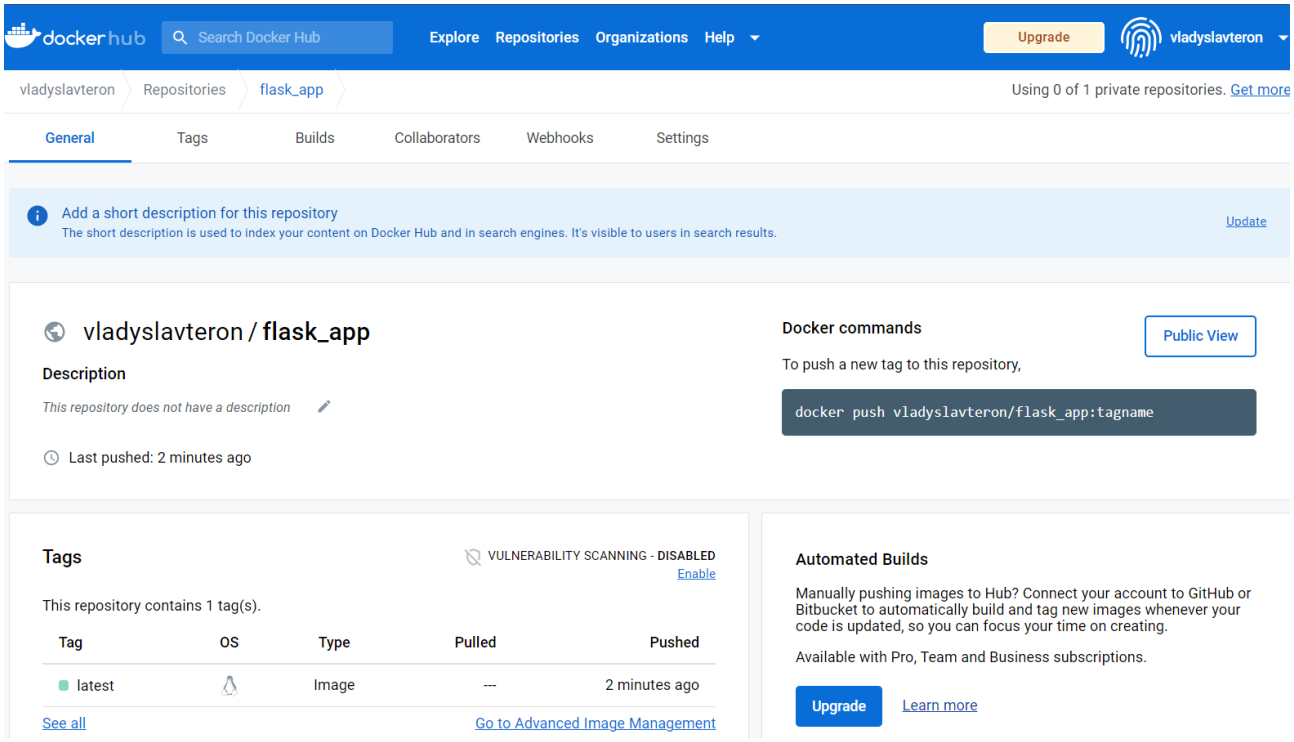
```
vladyslav@docker:~/Lecture2/flask-app$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://h
ub.docker.com to create one.
Username: vladyslavteron
Password:
WARNING! Your password will be stored unencrypted in /home/vladyslav/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

9) I push the image to DockerHub:

```
vladyslav@docker:~/Lecture2/flask-app$ docker push vladyslavteron/flask_app
Using default tag: latest
The push refers to repository [docker.io/vladyslavteron/flask_app]
91d03a4114cb: Pushed
59f21795dac3: Pushed
5e794eab8adc: Pushed
489f1edbc904: Pushed
c0c3988479bb: Pushed
f566c57e6f2d: Mounted from library/alpine
latest: digest: sha256:802ab4027b49e6903d101b81114bb65353a1b53b81b20aede03191a84eb05e62 size: 1571
vladyslav@docker:~/Lecture2/flask-app$
```

Result:



PERFORMING EXERCISES FROM LECTURE №2

1) I create a project directory:

```
vladyslav@docker:~/Lecture3$ mkdir wordpress_app
vladyslav@docker:~/Lecture3$ cd wordpress_app/
vladyslav@docker:~/Lecture3/wordpress_app$
```

2) I create a docker-compose.yml file and fill it with content:

```
vladyslav@docker:~/Lecture3/wordpress_app$ touch docker-compose.yml
vladyslav@docker:~/Lecture3/wordpress_app$
```

docker-compose.yml file content:

```
version: '3.3'

services:
  db:
    image: mysql:5.7
    volumes:
      - db_data:/var/lib/mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: somewordpress
      MYSQL_DATABASE: wordpress
      MYSQL_USER: wordpress
      MYSQL_PASSWORD: wordpress

  wordpress:
    depends_on:
      - db
    image: wordpress:latest
    ports:
      - "8000:80"
    restart: always
    environment:
      WORDPRESS_DB_HOST: db:3306
      WORDPRESS_DB_USER: wordpress
      WORDPRESS_DB_PASSWORD: wordpress
      WORDPRESS_DB_NAME: wordpress

volumes:
  db_data: {}
```

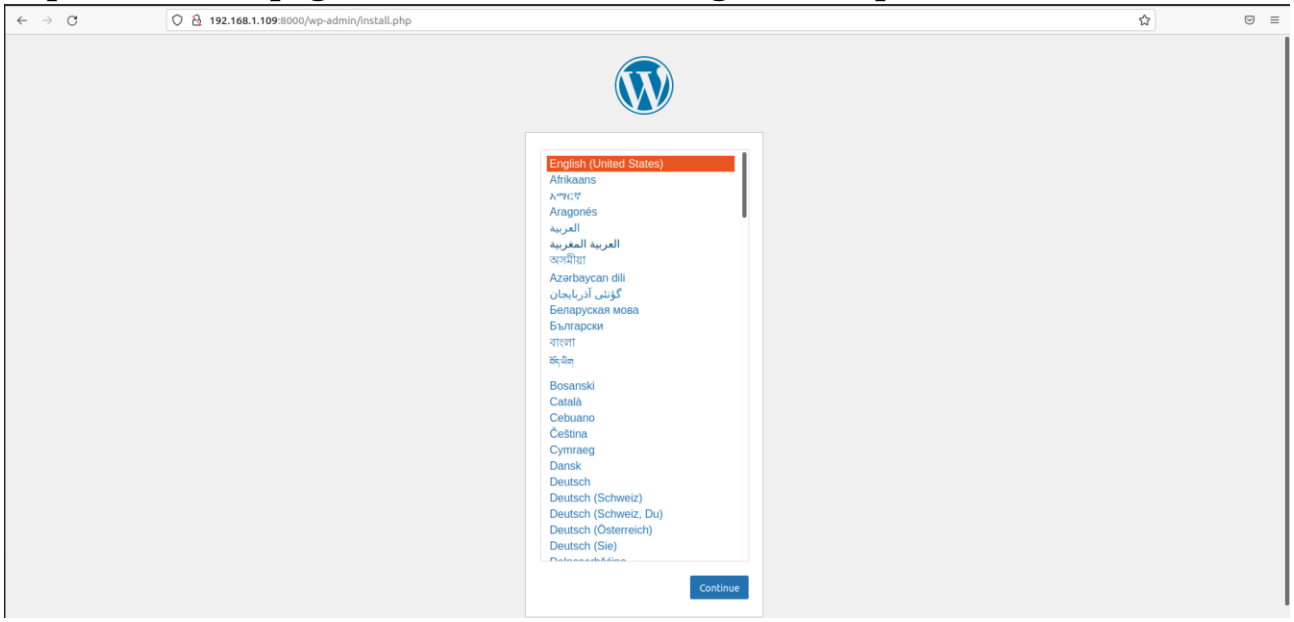
3) I build the project:

```
vladyslav@docker:~/Lecture3/wordpress_app$ docker compose up -d
[+] Running 14/34
 ⠸ wordpress Pulling                                                    35.8s
   ⠿ 3f4ca61aafcd Pull complete                                         22.6s
   ⠿ 460703cf6140 Pull complete                                         22.8s
   ⠸ eba06349db87 Extracting       60.72MB/91.63MB                      33.6s
   ⠸ 9130a4183abd Download complete                                     33.6s
   ⠸ fd60536a0833 Download complete                                     33.6s
```

…

```
[+] Running 4/4
 ⠿ Network wordpress_app_default        Created                          0.4s
 ⠿ Volume "wordpress_app_db_data"       Created                          0.1s
 ⠿ Container wordpress_app-db-1         Started                          5.9s
 ⠿ Container wordpress_app-wordpress-1  Started                          1.6s
vladyslav@docker:~/Lecture3/wordpress_app$
```

4) I open a website page in a browser and configure wordpress:



5) I save configurations and check the correctness of the website:

6) I perform some operations on a volume in Docker:

```
vladyslav@docker:~/Lecture3/wordpress_app$ docker volume create my-vol
my-vol
vladyslav@docker:~/Lecture3/wordpress_app$ docker volume ls
DRIVER    VOLUME NAME
local     60e8d12911698020150bb9e5911289382a745becb717b3fbdd02b2854f790a2f
local     my-vol
local     wordpress_app_db_data
vladyslav@docker:~/Lecture3/wordpress_app$ docker inspect my-vol
[
    {
        "CreatedAt": "2023-01-04T17:30:26+02:00",
        "Driver": "local",
        "Labels": {},
        "Mountpoint": "/var/lib/docker/volumes/my-vol/_data",
        "Name": "my-vol",
        "Options": {},
        "Scope": "local"
    }
]
vladyslav@docker:~/Lecture3/wordpress_app$ docker volume rm my-vol
my-vol
```