Лабораторна робота 2

Основи DOCKER

Виконала:
студентка групи МІТ-31
Півторак Каріна
Варіант -1

Мета: розглянути принципи модульного тестування програмного забезпечення; навчитися створювати модульні тести з використанням JUnit.

Хід виконання лабораторної роботи

1. Ознайомитися із теоретичним матеріалом

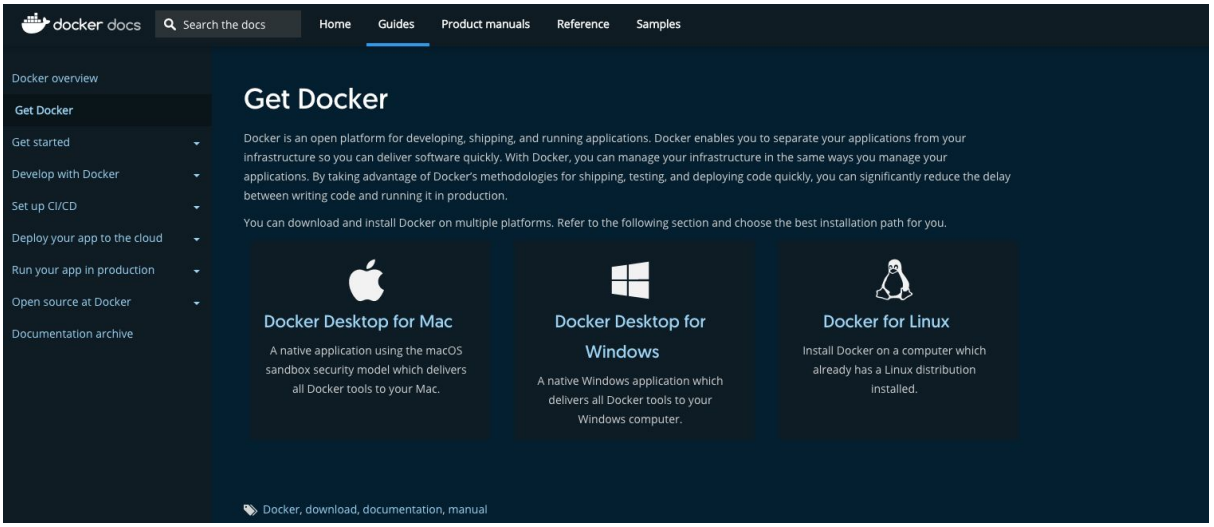
Docker являє собою систему управління контейнерами. Вона дозволяє «упакувати» додаток або веб-сайт з усім його оточенням і залежностями в контейнер, яким в подальшому можна легко і просто управляти: переносити на інший сервер, масштабувати, оновлювати.

Docker був написаний на мові програмування Go і випущений в 2013 році. Спочатку він працював тільки з Linux-системами, проте на даний момент його можна використовувати також в Windows і macOS. Незважаючи на те, що проект є відносно новим, Докер широко використовується багатьма фахівцями і продовжує завойовувати популярність.

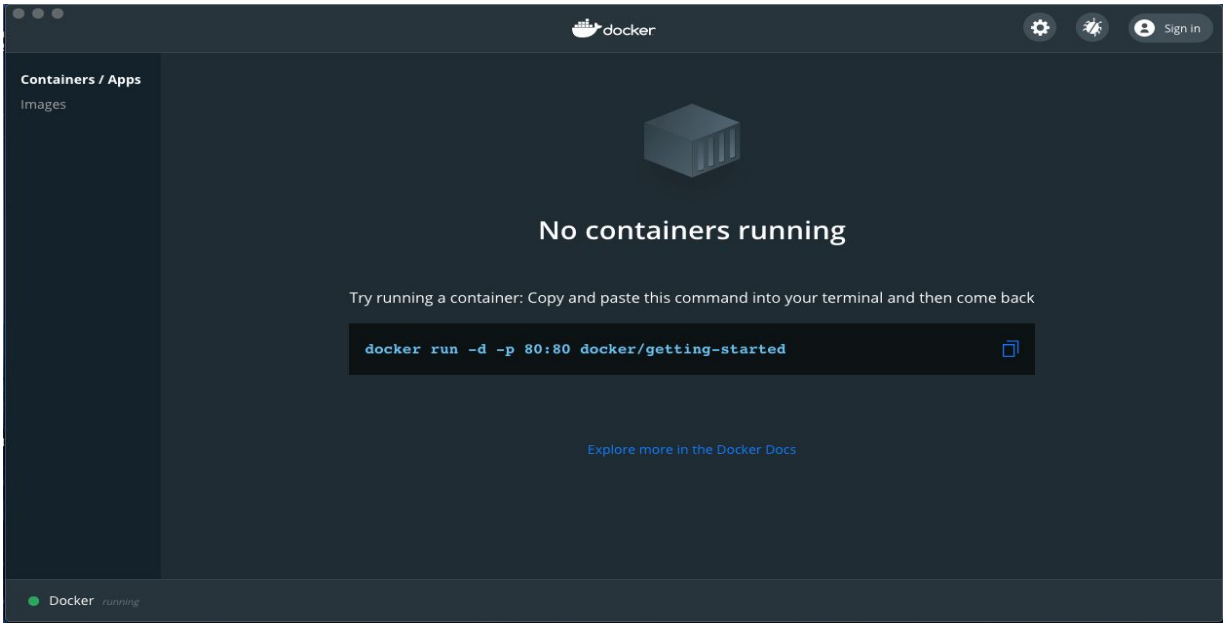
Важливою частиною екосистеми Docker є Docker Hub - відкритий репозиторій образів контейнерів. У ньому можна знайти десятки готових додатків від офіційних розробників. Серед них - nginx, MySQL, Apache, Gitlab, Redmine, Elasticsearch, Jenkins і інші.

2. Встановлення Docker

Переходимо за посиланням https://docs.docker.com/get-docker/



Встановлюємо необхідну версію, відкриваємо застосунок



перевіряємо встановлення Docker Machine

```
[kari@iMac-Kari: cheers2019 [master] » docker-machine
fish: Unknown command docker-machine
```

Оскільки отримуємо у відповідь Unknown command docker-machine, встановлюємо Docker Machine за допомогою brew

```
[kari@iMac-Kari: cheers2019 [master] » brew install docker-machine
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> Updated Formulae
Updated 1 formula.

==> Downloading https://homebrew.bintray.com/bottles/docker-machine-0.16.2.mojave.bottle.tar.gz
==> Downloading from https://d29vzk4ow07wi7.cloudfront.net/cc56a9c37702ecaeea1a5034326d87fa145fbc4cb613d
######################################################################## 100.0%
==> Pouring docker-machine-0.16.2.mojave.bottle.tar.gz
==> Caveats
Bash completion has been installed to:
  /usr/local/etc/bash_completion.d

zsh completions have been installed to:
  /usr/local/share/zsh/site-functions

To have launchd start docker-machine now and restart at login:
  brew services start docker-machine
Or, if you don't want/need a background service you can just run:
  docker-machine start
==> Summary
  /usr/local/Cellar/docker-machine/0.16.2: 11 files, 36MB
```

Вводимо команду ще раз

```
[kari@iMac-Kari: cheers2019 [master] » docker-machine -v
docker-machine version 0.16.2, build bd45ab1
kari@iMac-Kari: cheers2019 [master] »
```

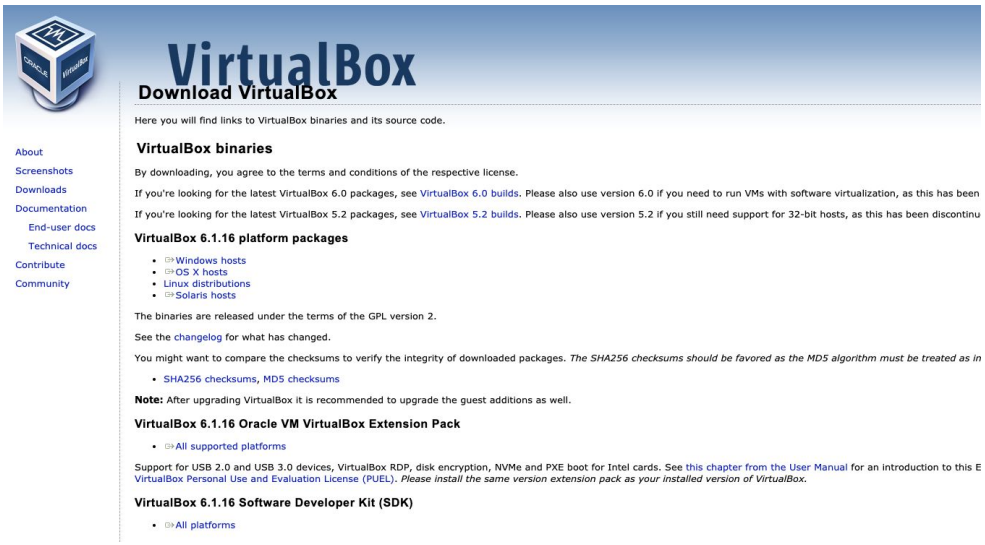
Переглядаємо список наявних докер-машин

```
[kari@iMac-Kari: cheers2019 [master] » docker-machine ls
NAME   ACTIVE   DRIVER   STATE   URL   SWARM   DOCKER   ERRORS
```

Створюємо машину командою docker-machine create --driver virtualbox karidm

```
kari@iMac-Kari: cheers2019 [master] » docker-machine create --driver virtualbox karidm
Running pre-create checks...
Error with pre-create check: "VBoxManage not found. Make sure VirtualBox is installed an
d VBoxManage is in the path"
kari@iMac-Kari: cheers2019 [master] »                                    3 [16:38:05]
```

Щоб виконати команду потрібно встановити VirtualBox (https://www.virtualbox.org/wiki/Downloads)



VirtualBox

Download VirtualBox

Here you will find links to VirtualBox binaries and its source code.

About
Screenshots
Downloads
Documentation
End-user docs
Technical docs
Contribute
Community

VirtualBox binaries

By downloading, you agree to the terms and conditions of the respective license.

If you're looking for the latest VirtualBox 6.0 packages, see VirtualBox 6.0 builds. Please also use version 6.0 if you need to run VMs with software virtualization, as this has been

If you're looking for the latest VirtualBox 5.2 packages, see VirtualBox 5.2 builds. Please also use version 5.2 if you still need support for 32-bit hosts, as this has been discontinu

VirtualBox 6.1.16 platform packages

- Windows hosts
- OS X hosts
- Linux distributions
- Solaris hosts

The binaries are released under the terms of the GPL version 2.

See the changelog for what has changed.

You might want to compare the checksums to verify the integrity of downloaded packages. *The SHA256 checksums should be favored as the MD5 algorithm must be treated as in*

- SHA256 checksums, MD5 checksums

Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.

VirtualBox 6.1.16 Oracle VM VirtualBox Extension Pack

- All supported platforms

Support for USB 2.0 and USB 3.0 devices, VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See this chapter from the User Manual for an introduction to this E
VirtualBox Personal Use and Evaluation License (PUEL). *Please install the same version extension pack as your installed version of VirtualBox.*

VirtualBox 6.1.16 Software Developer Kit (SDK)

- All platforms

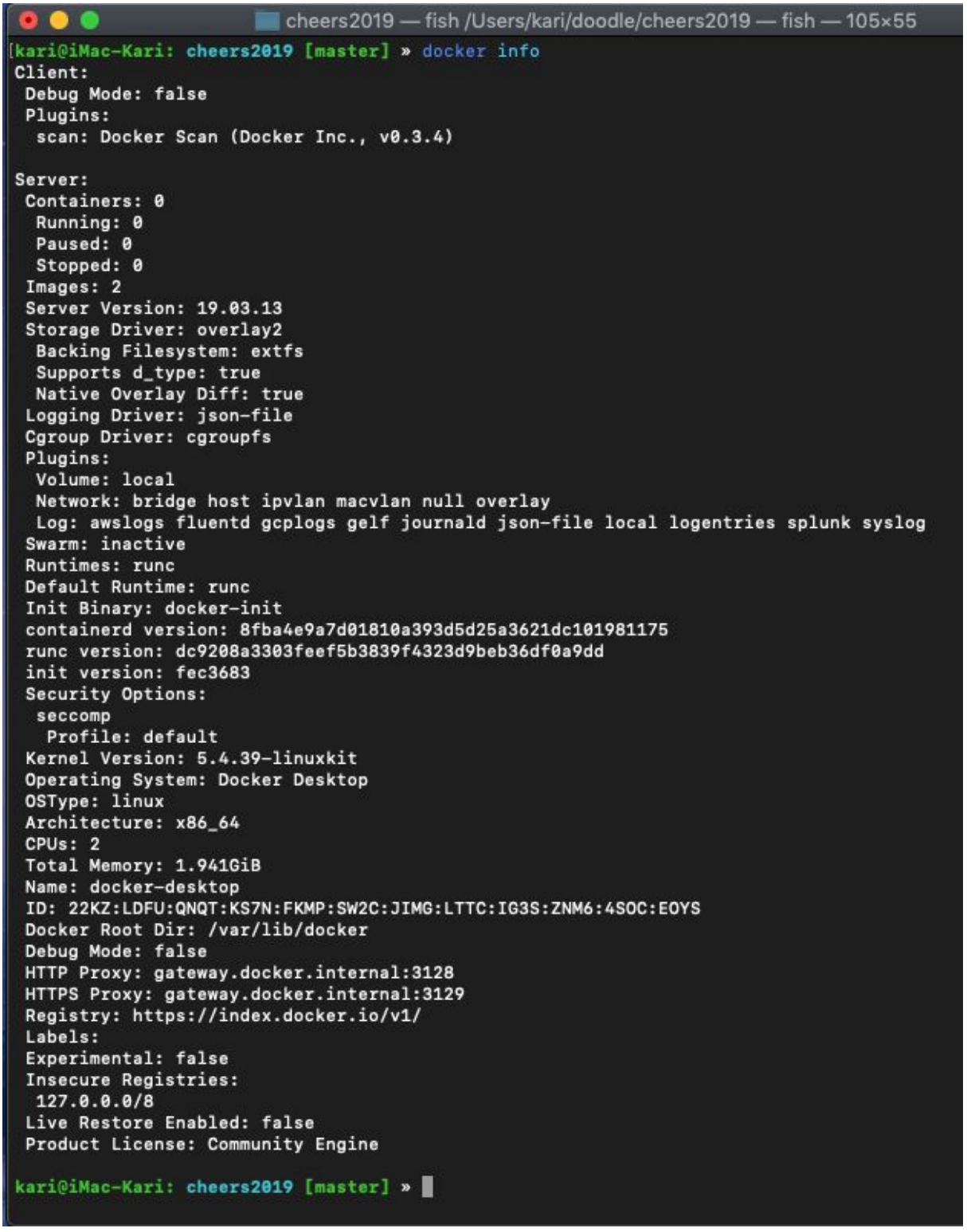
Створюємо докер машину

```
kari@iMac-Kari: cheers2019 [master] » docker-machine create --driver virtualbox karidm
Running pre-create checks...
(karidm) Image cache directory does not exist, creating it at /Users/kari/.docker/machine/cache...
(karidm) No default Boot2Docker ISO found locally, downloading the latest release...
(karidm) Latest release for github.com/boot2docker/boot2docker is v19.03.12
(karidm) Downloading /Users/kari/.docker/machine/cache/boot2docker.iso from https://github.com/boot2docke
r/boot2docker/releases/download/v19.03.12/boot2docker.iso...
(karidm) 0%....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%
Creating machine...
(karidm) Copying /Users/kari/.docker/machine/cache/boot2docker.iso to /Users/kari/.docker/machine/machine
s/karidm/boot2docker.iso...
(karidm) Creating VirtualBox VM...
(karidm) Creating SSH key...
(karidm) Starting the VM...
(karidm) Check network to re-create if needed...
(karidm) Found a new host-only adapter: "vboxnet0"
(karidm) Waiting for an IP...
Waiting for machine to be running, this may take a few minutes...
Detecting operating system of created instance...
Waiting for SSH to be available...
Detecting the provisioner...
Provisioning with boot2docker...
Copying certs to the local machine directory...
Copying certs to the remote machine...
Setting Docker configuration on the remote daemon...
Checking connection to Docker...
Docker is up and running!
To see how to connect your Docker Client to the Docker Engine running on this virtual machine, run: docke
r-machine env karidm
kari@iMac-Kari: cheers2019 [master] »                                    [16:43:08]
```

Отримуємо змінні середовища, що потрібно налаштувати для підключення до докер машини

```
kari@iMac-Kari: cheers2019 [master] » docker-machine env karidm
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.100:2376"
export DOCKER_CERT_PATH="/Users/kari/.docker/machine/machines/karidm"
export DOCKER_MACHINE_NAME="karidm"
# Run this command to configure your shell:
# eval $(docker-machine env karidm)
kari@iMac-Kari: cheers2019 [master] »
```

Вводимо команду для перевірки правильності підключення docker info



3. Запуск контейнера з існуючого образу

Завантажити образ

```
kari@iMac-Kari: ~ » docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
188c0c94c7c5: Already exists
Digest: sha256:c0e9560cda118f9ec63ddefb4a173a2b2a0347082d7dff7dc14272e7841a5b5a
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
kari@iMac-Kari: ~ »
```

Переглядаємо доступні контейнери

```
kari@iMac-Kari: ~ » docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
karispace/cheers2019    latest              c417333a3898        2 hours ago         4.01MB
telegraf                latest              8a2da2d9fedd        9 days ago          279MB
docker/getting-started  latest              67a3629d4d71        3 weeks ago         27.2MB
alpine                  latest              d6e46aa2470d        5 weeks ago         5.57MB
kari@iMac-Kari: ~ »
```

Запускаємо контейнер

```
kari@iMac-Kari: ~ » docker run alpine ls -l
total 56
drwxr-xr-x    2 root     root          4096 Oct 21 09:23 bin
drwxr-xr-x    5 root     root           340 Nov 28 15:17 dev
drwxr-xr-x    1 root     root          4096 Nov 28 15:17 etc
drwxr-xr-x    2 root     root          4096 Oct 21 09:23 home
drwxr-xr-x    7 root     root          4096 Oct 21 09:23 lib
drwxr-xr-x    5 root     root          4096 Oct 21 09:23 media
drwxr-xr-x    2 root     root          4096 Oct 21 09:23 mnt
drwxr-xr-x    2 root     root          4096 Oct 21 09:23 opt
dr-xr-xr-x  124 root     root             0 Nov 28 15:17 proc
drwx------    2 root     root          4096 Oct 21 09:23 root
drwxr-xr-x    2 root     root          4096 Oct 21 09:23 run
drwxr-xr-x    2 root     root          4096 Oct 21 09:23 sbin
drwxr-xr-x    2 root     root          4096 Oct 21 09:23 srv
dr-xr-xr-x   13 root     root             0 Nov 28 15:17 sys
drwxrwxrwt    2 root     root          4096 Oct 21 09:23 tmp
drwxr-xr-x    7 root     root          4096 Oct 21 09:23 usr
drwxr-xr-x   12 root     root          4096 Oct 21 09:23 var
kari@iMac-Kari: ~ »
```

Перелік усіх контейнерів:

```
kari@iMac-Kari: ~ » docker ps -a                                                [17:17:18]
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
        PORTS               NAMES
b6a5f695fc8b        alpine              "ls -l"             About a minute ago  Exited (0) About a minut
e ago                       jolly_lewin
kari@iMac-Kari: ~ »                                                             [17:18:34]
```

Видалення контейнера:

```
kari@iMac-Kari: ~ » docker rm jolly_lewin
jolly_lewin
kari@iMac-Kari: ~ »
```

Виконання команд в контейнері

```
kari@iMac-Kari: ~ » docker run -it alpine
/ # ls
bin    etc    lib    mnt    proc   run    srv    tmp    var
dev    home   media  opt    root   sbin   sys    usr
/ # arch
x86_64
/ # echo "Hello alpine"
Hello alpine
/ # uname -r
5.4.39-linuxkit
/ #
```

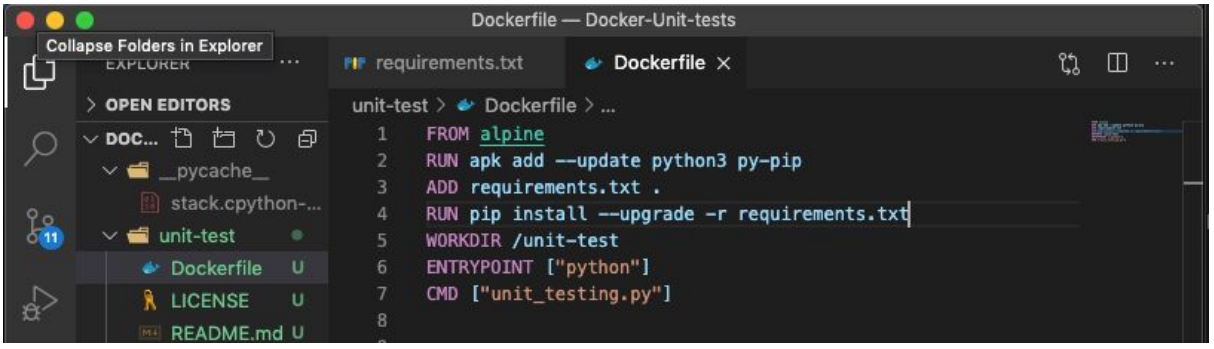
```
kari@iMac-Kari: ~ » docker ps                                                    [17:30:35]
CONTAINER ID   IMAGE     COMMAND     CREATED          STATUS          PORTS     NAMES
32220e0de9d1   alpine    "/bin/sh"   8 minutes ago    Up 7 minutes              distracted_vaughan
kari@iMac-Kari: ~ »
```

Знайти образ контейнеру `hello-world` і запустити
посилання на контейнер: https://hub.docker.com/_/hello-world
Запускаємо:

```
kari@iMac-Kari: ~ » docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:e7c70bb24b462baa86c102610182e3efcb12a04854e8c582838d92970a09f323
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/

kari@iMac-Kari: ~ »
```

4. Створення власного образу

Створюємо докерфайл



Файл requirements.txt містить модулі python, необхідні для запуску вашого застосування. У цьому випадку, коли потрібно встановити `unittest2`

Створюємо образ контейнера

```
[kari@iMac-Kari: unit-test [main]× » docker build -t unit-test:v1.0  .
[+] Building 0.0s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 120B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/alpine:latest
 => [1/5] FROM docker.io/library/alpine
 => [internal] load build context
 => => transferring context: 37B
 => CACHED [2/5] RUN apk add --update python3 py-pip
 => CACHED [3/5] ADD requirements.txt .
 => CACHED [4/5] RUN pip install --upgrade -r requirements.txt
 => CACHED [5/5] WORKDIR /unit-test
 => exporting to image
 => => exporting layers
 => => writing image sha256:b7c30bb15eeaea1f3d5602c3e5836b824182584ef3c1f39bfb394736886c8776
 => => naming to docker.io/library/unit-test:v1.0
kari@iMac-Kari: unit-test [main]× »
```

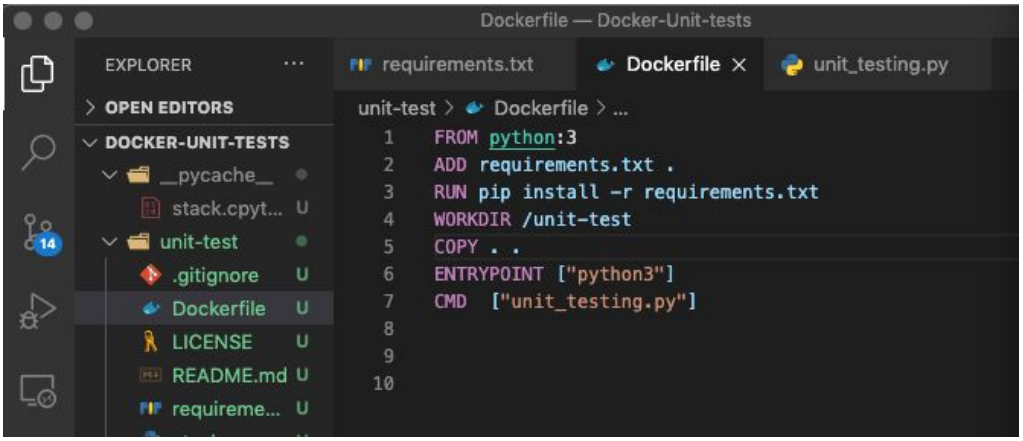
Перегляд всіх образів

```
[kari@iMac-Kari: unit-test [main]× » docker images
REPOSITORY              TAG                 IMAGE ID            CREATED             SIZE
unit-test               latest              b7c30bb15eea        2 minutes ago       61.7MB
unit-test               v1.0                b7c30bb15eea        2 minutes ago       61.7MB
karispace/cheers2019    latest              c417333a3898        3 hours ago         4.01MB
ubuntu                  latest              f643c72bc252        2 days ago          72.9MB
telegraf                latest              8a2da2d9fedd        9 days ago          279MB
docker/getting-started  latest              67a3629d4d71        3 weeks ago         27.2MB
alpine                  latest              d6e46aa2470d        5 weeks ago         5.57MB
hello-world             latest              bf756fb1ae65        11 months ago       13.3kB
kari@iMac-Kari: unit-test [main]× »
```

У мене виникала помилка docker: Error response from daemon: OCI runtime create failed: container_linux.go:349: starting container process caused "exec: \"python\": executable file not found in $PATH": unknown.

```
kari@iMac-Kari: unit-test [main]× » docker run --publish 8000:8080 --detach --name kari_unittests unit-test:v1.0                125 [18:38:16
a10fb02bd261de60c7066cb4a149a5d49a99ab3635359c92f66ebb53822c2f41
docker: Error response from daemon: OCI runtime create failed: container_linux.go:349: starting container process caused "exec: \"python\": execut
ble file not found in $PATH": unknown.
```

Тому, я в докер файлі `python` замінила на `python3`



та створила образ контейнеру з версією v1.1

```
[kari@iMac-Kari: unit-test [main]× » docker build -t unit-test:v1.1  .
[+] Building 0.0s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 120B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/alpine:latest
 => [1/5] FROM docker.io/library/alpine
 => [internal] load build context
 => => transferring context: 37B
 => CACHED [2/5] RUN apk add --update python3 py-pip
 => CACHED [3/5] ADD requirements.txt .
 => CACHED [4/5] RUN pip install --upgrade -r requirements.txt
 => CACHED [5/5] WORKDIR /unit-test
 => exporting to image
 => => exporting layers
 => => writing image sha256:f3833cd0e1ceb9fe4ce36c393d10c37d7a07c19b43ca3f6d1ff90fcc699b5ceb
 => => naming to docker.io/library/unit-test:v1.1
```

```
[kari@iMac-Kari: unit-test [main]× » docker run --publish 8000:8080 --detach --name kari_unittests unit-test:v1.1
docker: Error response from daemon: Conflict. The container name "/kari_unittests" is already in use by container "a10fb02bd261de60c
9a99ab3635359c92f66ebb53822c2f41". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
```

Запускаємо файл із лабораторною 2

```
[kari@iMac-Kari: unit-test [main]× » docker run -it unit-test:v1.3 unit_testing.
.....
----------------------------------------------------------------------
Ran 5 tests in 0.001s

OK
```

Репозиторій: https://github.com/KariSpace/Simple-Unit-tests

Висновок: Завдяки модулю unittest можна створювати прості модульні тести для того, щоб упевнитися, що код відповідає вимогам архітектури та має очікувану поведінку.