DOCKER - E1

1. Download a 19-alpine version node image from DockerHub. Node.js is a JavaScript runtime environment.

```
docker pull node:19-alpine
```

```
root@christianms13:/home/christian# docker pull node:19-alpine
19-alpine: Pulling from library/node
8921db27df28: Pull complete
54670a6b4be2: Pull complete
78512a5ffb79: Pull complete
1ec9d67c6b65: Pull complete
Digest: sha256:ab3603cb7934b21f1ffb522b1a1d538809516c6e4cd73b144716bc1830aad1a6
Status: Downloaded newer image for node:19-alpine
docker.io/library/node:19-alpine
```

2. Show the image downloaded successfully.

```
docker images
```

```
root@christianms13:/home/christian# docker images
REPOSITORY   TAG         IMAGE ID       CREATED      SIZE
node         19-alpine   17299c0421ee   9 days ago   176MB
```

3. Download an image from the nginx web page server (latest version).

```
docker pull nginx
```

```
root@christianms13:/home/christian# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
8740c948ffd4: Pull complete
d2c0556a17c5: Pull complete
c8b9881f2c6a: Pull complete
693c3ffa8f43: Pull complete
8316c5e80e6d: Pull complete
b2fe3577faa4: Pull complete
Digest: sha256:b8f2383a95879e1ae064940d9a200f67a6c79e710ed82ac42263397367e7cc4e
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

4. Show these two images.

```
docker images
```

```
root@christianms13:/home/christian# docker images
REPOSITORY   TAG          IMAGE ID       CREATED      SIZE
nginx        latest       a99a39d070bf   8 days ago   142MB
node         19-alpine    17299c0421ee   9 days ago   176MB
```

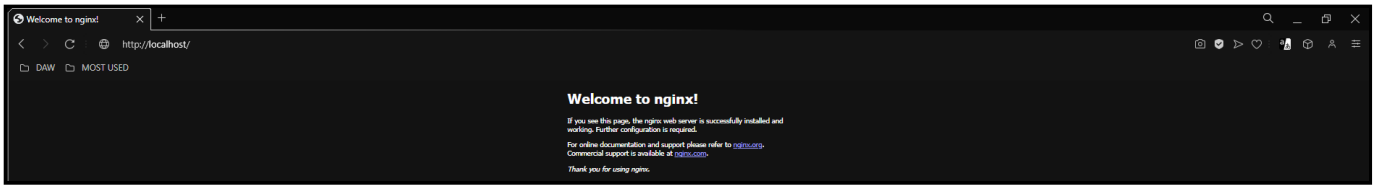
5. Run the nginx container to connect port 80 of this container with requests to port 80 that web browsers use when using http:// requests. What happens if you don't use the "-d" parameter?

```
docker run -p 80:80 nginx
```

```
root@christianms13:/home/christian# docker run -p 80:80 nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2023/01/19 15:48:22 [notice] 1#1: using the "epoll" event method
2023/01/19 15:48:22 [notice] 1#1: nginx/1.23.3
2023/01/19 15:48:22 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2023/01/19 15:48:22 [notice] 1#1: OS: Linux 5.15.79.1-microsoft-standard-WSL2
2023/01/19 15:48:22 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2023/01/19 15:48:22 [notice] 1#1: start worker processes
2023/01/19 15:48:22 [notice] 1#1: start worker process 29
2023/01/19 15:48:22 [notice] 1#1: start worker process 30
2023/01/19 15:48:22 [notice] 1#1: start worker process 31
2023/01/19 15:48:22 [notice] 1#1: start worker process 32
2023/01/19 15:48:22 [notice] 1#1: start worker process 33
2023/01/19 15:48:22 [notice] 1#1: start worker process 34
2023/01/19 15:48:22 [notice] 1#1: start worker process 35
2023/01/19 15:48:22 [notice] 1#1: start worker process 36
2023/01/19 15:48:22 [notice] 1#1: start worker process 37
2023/01/19 15:48:22 [notice] 1#1: start worker process 38
2023/01/19 15:48:22 [notice] 1#1: start worker process 39
2023/01/19 15:48:22 [notice] 1#1: start worker process 40
172.17.0.1 - - [19/Jan/2023:15:51:30 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537.36 OPR/94.0.0.0" "-"
172.17.0.1 - - [19/Jan/2023:15:51:30 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537
.36 OPR/94.0.0.0" "-"
2023/01/19 15:51:30 [error] 29#29: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "local
host", referrer: "http://localhost/"
2023/01/19 15:53:36 [error] 30#30: *4 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "local
host", referrer: "http://localhost/"
172.17.0.1 - - [19/Jan/2023:15:53:36 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537
.36 OPR/94.0.0.0" "-"
172.17.0.1 - - [19/Jan/2023:15:53:43 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/537
.36 OPR/94.0.0.0" "-"
2023/01/19 15:53:43 [error] 30#30: *4 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "local
host", referrer: "http://localhost/"
2023/01/19 15:57:43 [notice] 1#1: signal 3 (SIGQUIT) received, shutting down
2023/01/19 15:57:43 [notice] 30#30: gracefully shutting down
2023/01/19 15:57:43 [notice] 29#29: gracefully shutting down
2023/01/19 15:57:43 [notice] 34#34: gracefully shutting down
2023/01/19 15:57:43 [notice] 30#30: exiting
```

If the "-d" switch is not used, the terminal stays in standby mode, displaying server activities. To be able to run the container in the background, the "-d" parameter is used.

6. Check that the server responds using the IP of the machine itself (localhost) in a web browser.



7. Run the container downloaded in step 1 and name this container "nodejs".

```
docker run -it -d --name nodejs node:19-alpine sh
```

```
root@christianms13:/home/christian# docker run -it -d --name nodejs node:19-alpine sh
ea85fd08fef335c4dc6b60075ec6cfe95499e7aa81843c97c14a729ef7e52005
```

8. Download Ubuntu:18.

```
docker pull ubuntu:18.04
```

```
root@christianms13:~# docker pull ubuntu:18.04
18.04: Pulling from library/ubuntu
a055bf07b5b0: Pull complete
Digest: sha256:c1d0baf2425ecef88a2f0c3543ec43690dc16cc80d3c4e593bb95e4f45390e45
Status: Downloaded newer image for ubuntu:18.04
docker.io/library/ubuntu:18.04
```

9. List all the downloaded images.

```
docker images
```

```
root@christianms13:~# docker images
REPOSITORY   TAG          IMAGE ID       CREATED       SIZE
nginx        latest       a99a39d070bf   9 days ago    142MB
node         19-alpine    17299c0421ee   11 days ago   176MB
ubuntu       18.04        e28a50f651f9   2 weeks ago   63.1MB
```

10. Show all running containers.

```
docker ps
```

```
root@christianms13:~# docker ps
CONTAINER ID   IMAGE           COMMAND                  CREATED      STATUS          PORTS     NAMES
ea85fd08fef3   node:19-alpine   "docker-entrypoint.s…"   4 days ago   Up 14 seconds             nodejs
```

11. Show all containers.

```
docker ps -a
```

```
root@christianms13:~# docker ps -a
CONTAINER ID   IMAGE           COMMAND                  CREATED       STATUS                      PORTS     NAMES
ea85fd08fef3   node:19-alpine   "docker-entrypoint.s…"   27 hours ago   Exited (137) 27 hours ago             nodejs
```

12. Stop all running containers.

```
docker container stop $(docker container ls -a -q)
```

```
root@christianms13:~# docker container stop $(docker container ls -a -q)
ea85fd08fef3
```

13. Now start the container from exercise 8. Call it "Ubuntu18".

```
docker run -d --name Ubuntu18 -it ubuntu:18.04
```

```
root@christianms13:~# docker run -d --name Ubuntu18 -it ubuntu:18.04
62923ac364017d2768655b5dd497569b9e7801f453ec0dfe6bfc691e05691f00
```

14. Access the command console of the "Ubuntu18" container. Use the bash commands interpreter.

```
docker exec -it Ubuntu18 bash
```

```
root@christianms13:~# docker exec -it Ubuntu18 bash
root@62923ac36401:/#
```

15. As you will see, when you start the service again we don't know if it started with errors or correctly. Use the docker logs command to see if the boot of the container from the previous exercise was correct.

```
docker logs
```

```
root@christianms13:~# docker logs
"docker logs" requires exactly 1 argument.
See 'docker logs --help'.

Usage:  docker logs [OPTIONS] CONTAINER

Fetch the logs of a container
```

16. What does the following docker instruction do? Explain its syntax

```
docker run -d --name servidor_nginx -p 80:80 nginx:1.22
```

This statement starts a new container, in the background, with a custom name ("servidor_nginx"), using port 80 and the nginx:1.22 image.

17. What outputs are given by the following execution instructions on a container in march?

a.

```
docker exec servidor_nginx ls /var/log/nginx/
```

It executes the "ls" instruction in a specific directory on the "servidor_nginx" container.

b.

```
docker stop servidor_nginx | docker rm servidor_nginx
```

This instruction, on the one hand, stops the container with the name "servidor_nginx" and, on the other hand, removes the already stopped container.

c.

```
docker stop Ubuntu18
```

This command stops the "Ubuntu18" container.

d.

```
docker start Ubuntu18
```

This command starts the "Ubuntu18" container.

e.

```
docker exec -it Ubuntu18 bash
```

This command allows the user to connect to the "Ubuntu18" container and be able to interact directly with the container's command line.

f.

```
docker rm Ubuntu18
```

This command removes the "Ubuntu18" container. The command doesn't work at first because the container is still active. To be able to remove it, you must first execute the "docker stop Ubuntu18" command.

18. Delete downloaded images related to nginx.

```
docker rmi $(docker images -q nginx)
```

```
root@christianms13:~# docker rmi $(docker images -q nginx)
Untagged: nginx:latest
Untagged: nginx@sha256:b8f2383a95879e1ae064940d9a200f67a6c79e710ed82ac42263397367e7cc4e
Deleted: sha256:a99a39d070bfd1cb60fe65c45dea3a33764dc00a9546bf8dc46cb5a11b1b50e9
Deleted: sha256:937f740376014d8e951ae4f0ff0b0ca64fd958c176510fc0a86dd1b491226a71
Deleted: sha256:a73369e624d6b4fbea1e6046a2f3cf3c7a15c6e0e0012659f410380d8dc70594
Deleted: sha256:bc2dd4cf50f065c3e884872cd30eb888a71f6ab545d8bec196a7662bfbd033a3
Deleted: sha256:c9372c5c9249380700f7af0441de14f14c361b5561ecc876da46f58a7ad71b7b
Deleted: sha256:deea3af05bc845b217a415b240422c19933218422b6d60414c2caeff976e5430
Deleted: sha256:67a4178b7d47beb6a1f697a593bd0c6841c67eb0da00f2badefb05fd30671490
```

19. Download the "node.zip" file. Copy its content to "/home/user/node".

Create an image called web_node based on the contents of the Dockerfile.

Create a container based on the image above with the parameter –d –p 80:3000. Visit the localhost site from any browser.