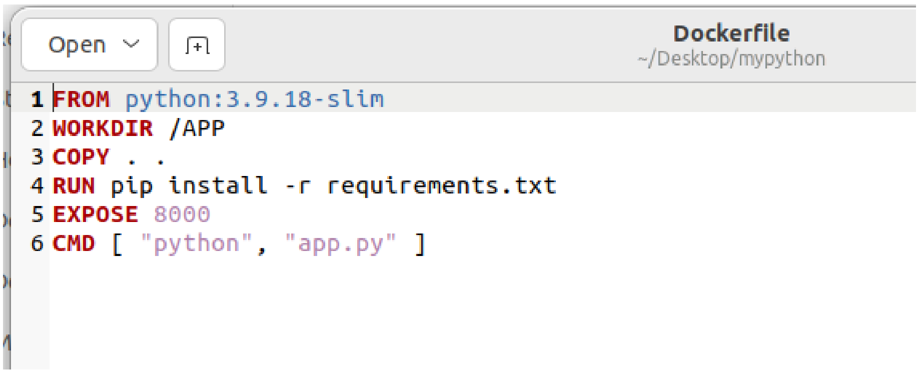
- P1: What do you know about Docker Images Tags and what is the difference

between them?

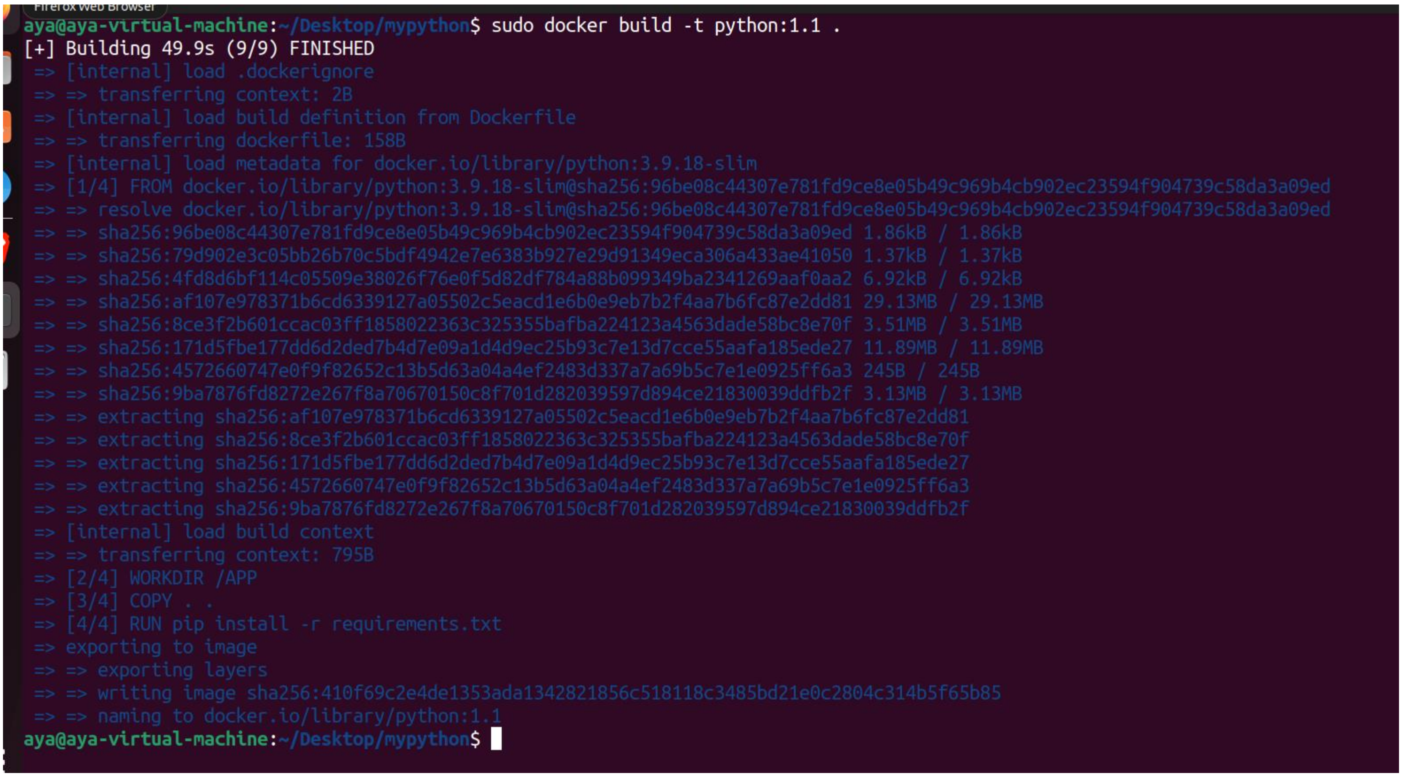
Each tag is a version of an image. When we edit an image. we build a new tag with a different name. there are some tag names which are with a meaning such as “slim” which refers to that this image is smaller than the original one

For example: If you saw ubuntu:slim means that this ubuntu only knows the basic commands and if you need to use the “ping” command, you gotta install it as you’re gonna need to run these commands first “apt-get update” then “apt-get install iputils-ping”

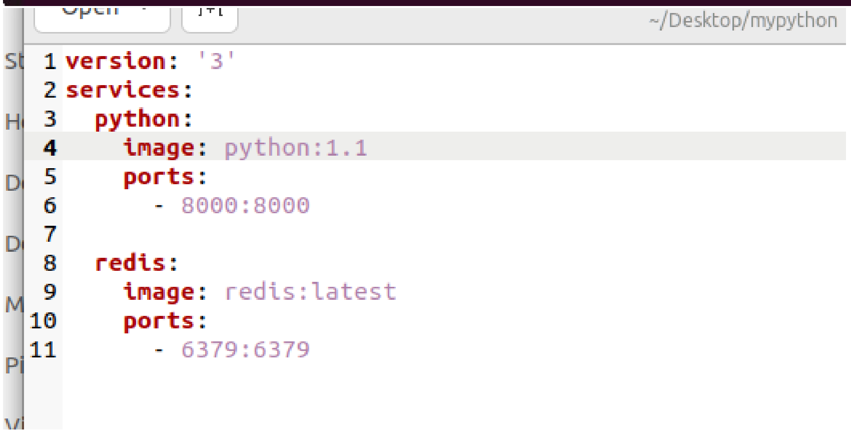
- P2: Create docker image for Python app



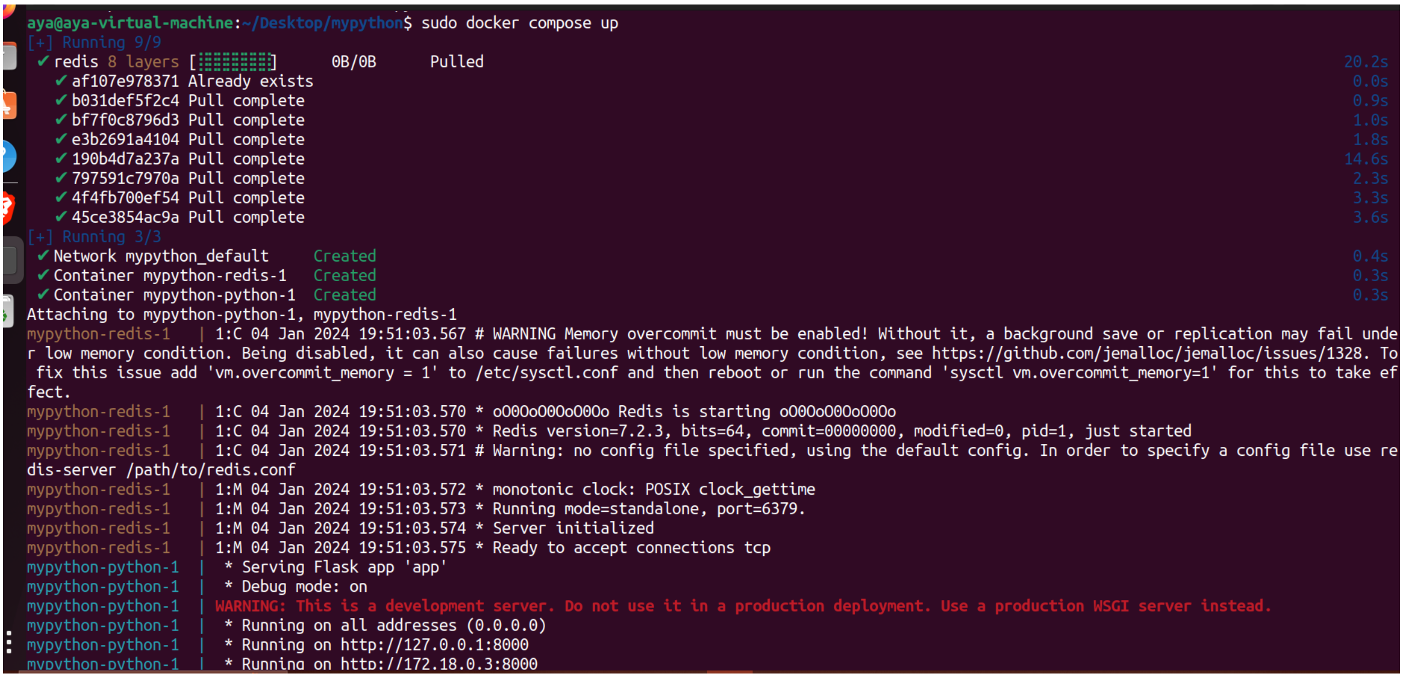
```
FROM python:3.9.18-slim
WORKDIR /APP
COPY . .
RUN pip install -r requirements.txt
EXPOSE 8000
CMD [ "python", "app.py" ]
```



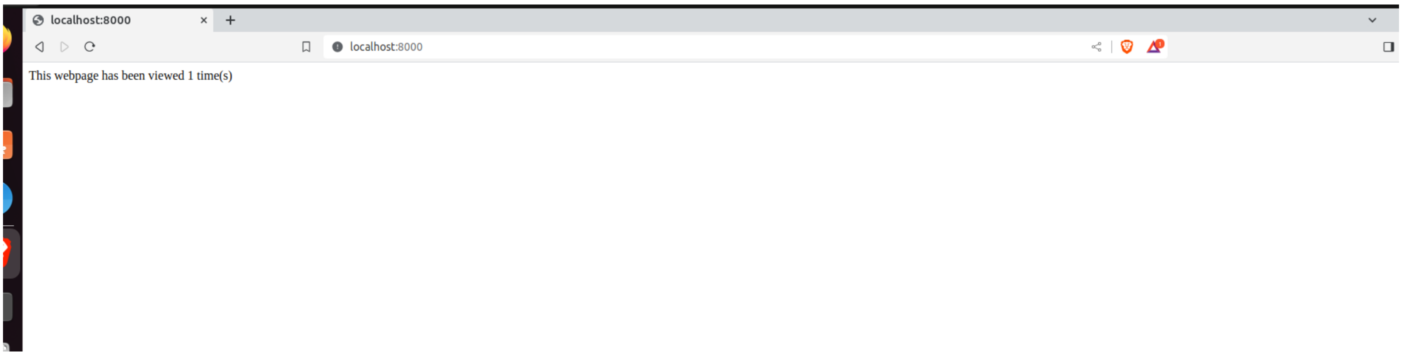
```
aya@aya-virtual-machine:~/Desktop/mypython$ sudo docker build -t python:1.1 .
[+] Building 49.9s (9/9) FINISHED
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 158B
 => [internal] load metadata for docker.io/library/python:3.9.18-slim
 => [1/4] FROM docker.io/library/python:3.9.18-slim@sha256:96be08c44307e781fd9ce8e05b49c969b4cb902ec23594f904739c58da3a09ed
 => => resolve docker.io/library/python:3.9.18-slim@sha256:96be08c44307e781fd9ce8e05b49c969b4cb902ec23594f904739c58da3a09ed
 => => sha256:96be08c44307e781fd9ce8e05b49c969b4cb902ec23594f904739c58da3a09ed 1.86kB / 1.86kB
 => => sha256:79d902e3c05bb26b70c5bdf4942e7e6383b927e29d91349eca306a433ae41050 1.37kB / 1.37kB
 => => sha256:4fd8d6bf114c05509e38026f76e0f5d82df784a88b099349ba2341269aaf0aa2 6.92kB / 6.92kB
 => => sha256:af107e978371b6cd6339127a05502c5eacd1e6b0e9eb7b2f4aa7b6fc87e2dd81 29.13MB / 29.13MB
 => => sha256:8ce3f2b601ccac03ff1858022363c325355bafba224123a4563dade58bc8e70f 3.51MB / 3.51MB
 => => sha256:171d5fbe177dd6d2ded7b4d7e09a1d4d9ec25b93c7e13d7cce55aafa185ede27 11.89MB / 11.89MB
 => => sha256:4572660747e0f9f82652c13b5d63a04a4ef2483d337a7a69b5c7e1e0925ff6a3 245B / 245B
 => => sha256:9ba7876fd8272e267f8a70670150c8f701d282039597d894ce21830039ddfb2f 3.13MB / 3.13MB
 => => extracting sha256:af107e978371b6cd6339127a05502c5eacd1e6b0e9eb7b2f4aa7b6fc87e2dd81
 => => extracting sha256:8ce3f2b601ccac03ff1858022363c325355bafba224123a4563dade58bc8e70f
 => => extracting sha256:171d5fbe177dd6d2ded7b4d7e09a1d4d9ec25b93c7e13d7cce55aafa185ede27
 => => extracting sha256:4572660747e0f9f82652c13b5d63a04a4ef2483d337a7a69b5c7e1e0925ff6a3
 => => extracting sha256:9ba7876fd8272e267f8a70670150c8f701d282039597d894ce21830039ddfb2f
 => [internal] load build context
 => => transferring context: 795B
 => [2/4] WORKDIR /APP
 => [3/4] COPY . .
 => [4/4] RUN pip install -r requirements.txt
 => exporting to image
 => => exporting layers
 => => writing image sha256:410f69c2e4de1353ada1342821856c518118c3485bd21e0c2804c314b5f65b85
 => => naming to docker.io/library/python:1.1
aya@aya-virtual-machine:~/Desktop/mypython$
```



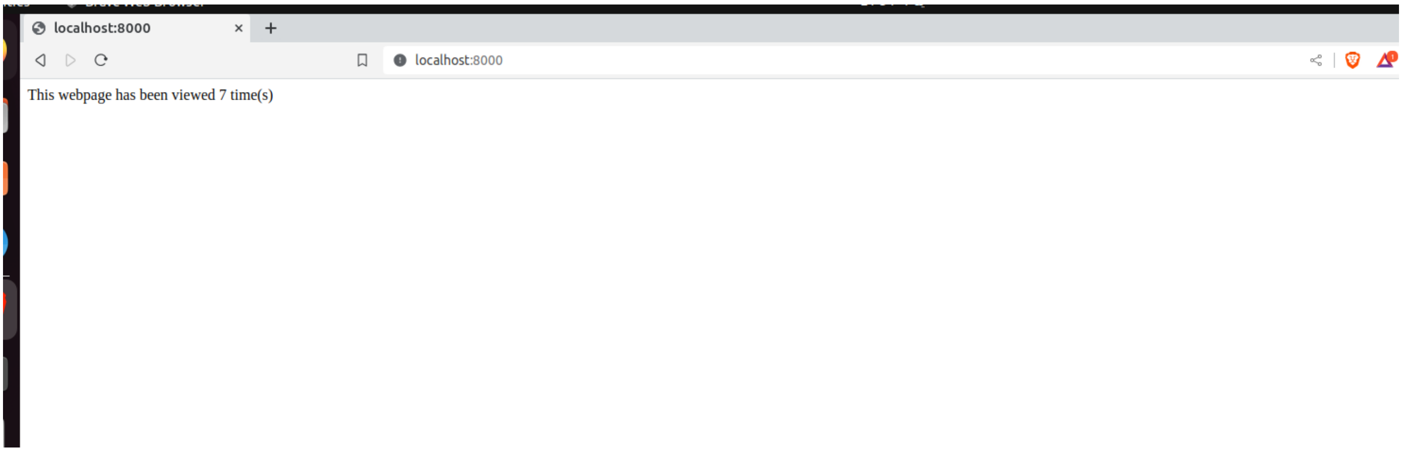
```
version: '3'
services:
  python:
    image: python:1.1
    ports:
      - 8000:8000

  redis:
    image: redis:latest
    ports:
      - 6379:6379
```



```
aya@aya-virtual-machine:~/Desktop/mypython$ sudo docker compose up
[+] Running 9/9
 ✔ redis 8 layers [⣿⣿⣿⣿⣿⣿⣿⣿]      0B/0B      Pulled                                                                         20.2s
   ✔ af107e978371 Already exists                                                                                                0.0s
   ✔ b031def5f2c4 Pull complete                                                                                                 0.9s
   ✔ bf7f0c8796d3 Pull complete                                                                                                 1.0s
   ✔ e3b2691a4104 Pull complete                                                                                                 1.8s
   ✔ 190b4d7a237a Pull complete                                                                                                14.6s
   ✔ 797591c7970a Pull complete                                                                                                 2.3s
   ✔ 4f4fb700ef54 Pull complete                                                                                                 3.3s
   ✔ 45ce3854ac9a Pull complete                                                                                                 3.6s
[+] Running 3/3
 ✔ Network mypython_default     Created                                                                                         0.4s
 ✔ Container mypython-redis-1   Created                                                                                         0.3s
 ✔ Container mypython-python-1  Created                                                                                         0.3s
Attaching to mypython-python-1, mypython-redis-1
mypython-redis-1   | 1:C 04 Jan 2024 19:51:03.567 # WARNING Memory overcommit must be enabled! Without it, a background save or replication may fail unde
r low memory condition. Being disabled, it can also cause failures without low memory condition, see https://github.com/jemalloc/jemalloc/issues/1328. To
 fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take ef
fect.
mypython-redis-1   | 1:C 04 Jan 2024 19:51:03.570 * oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
mypython-redis-1   | 1:C 04 Jan 2024 19:51:03.570 * Redis version=7.2.3, bits=64, commit=00000000, modified=0, pid=1, just started
mypython-redis-1   | 1:C 04 Jan 2024 19:51:03.571 # Warning: no config file specified, using the default config. In order to specify a config file use re
dis-server /path/to/redis.conf
mypython-redis-1   | 1:M 04 Jan 2024 19:51:03.572 * monotonic clock: POSIX clock_gettime
mypython-redis-1   | 1:M 04 Jan 2024 19:51:03.573 * Running mode=standalone, port=6379.
mypython-redis-1   | 1:M 04 Jan 2024 19:51:03.574 * Server initialized
mypython-redis-1   | 1:M 04 Jan 2024 19:51:03.575 * Ready to accept connections tcp
mypython-python-1  |  * Serving Flask app 'app'
mypython-python-1  |  * Debug mode: on
mypython-python-1  | WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
mypython-python-1  |  * Running on all addresses (0.0.0.0)
mypython-python-1  |  * Running on http://127.0.0.1:8000
mypython-python-1  |  * Running on http://172.18.0.3:8000
```

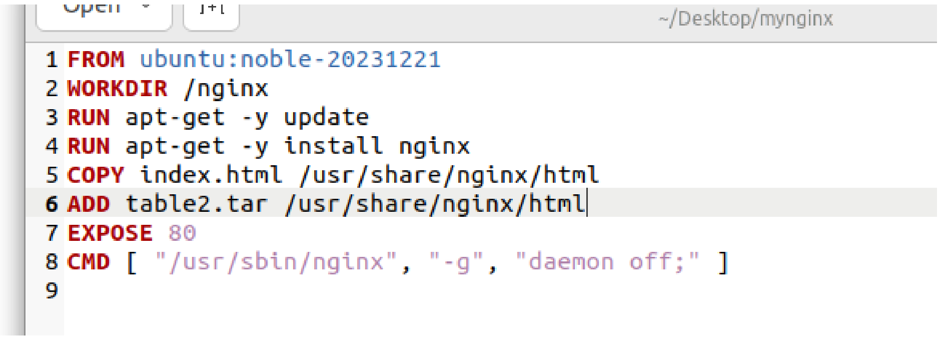


This webpage has been viewed 1 time(s)

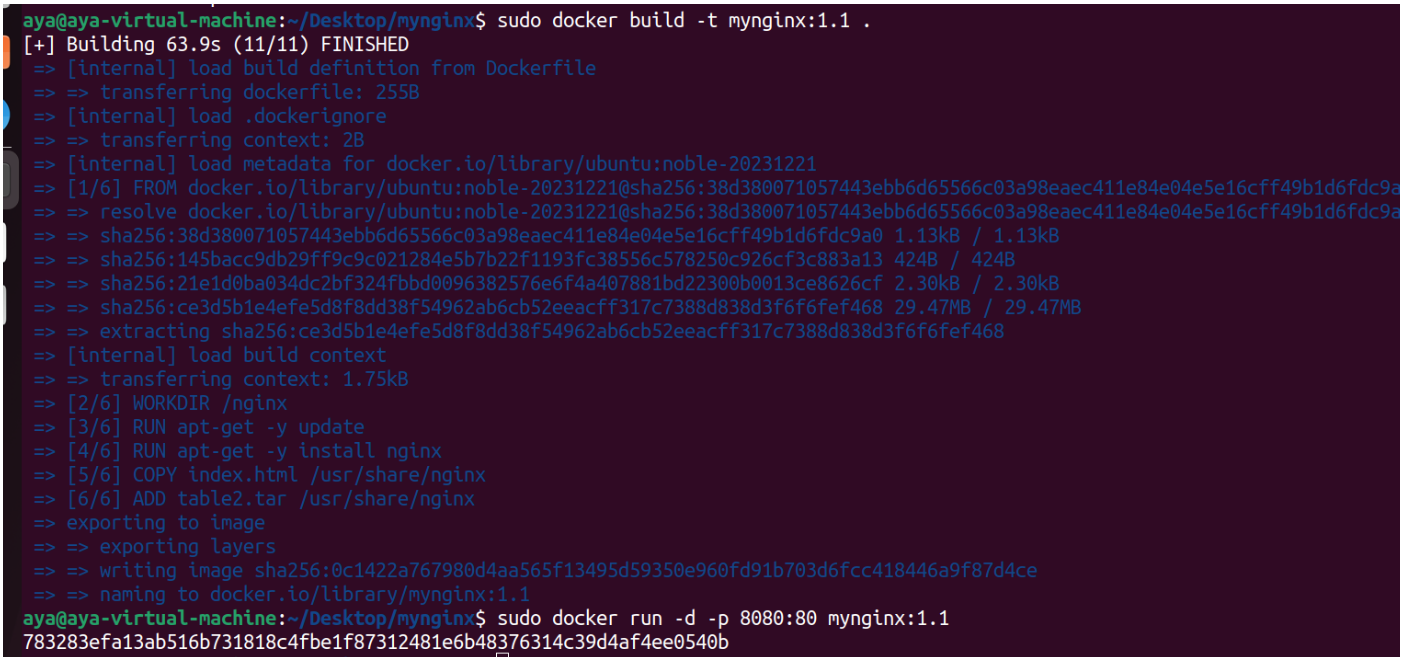


This webpage has been viewed 7 time(s)

- P3: Create your own nginx docker image "NEVER USE FROM nginx"

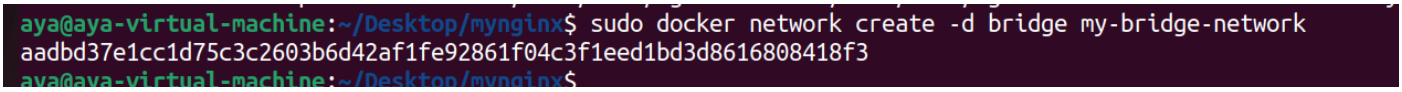


```
FROM ubuntu:noble-20231221
WORKDIR /nginx
RUN apt-get -y update
RUN apt-get -y install nginx
COPY index.html /usr/share/nginx/html
ADD table2.tar /usr/share/nginx/html
EXPOSE 80
CMD [ "/usr/sbin/nginx", "-g", "daemon off;" ]

```

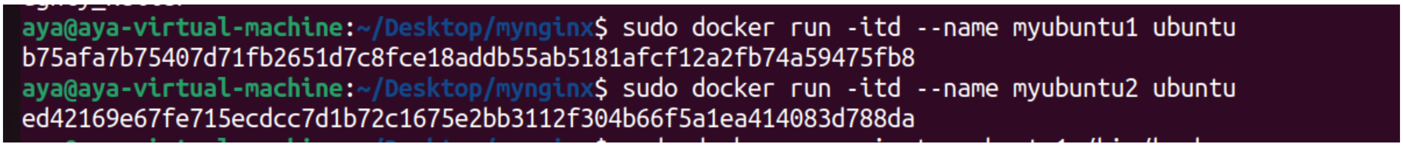


```
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker build -t mynginx:1.1 .
[+] Building 63.9s (11/11) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 255B
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [internal] load metadata for docker.io/library/ubuntu:noble-20231221
 => [1/6] FROM docker.io/library/ubuntu:noble-20231221@sha256:38d380071057443ebb6d65566c03a98eaec411e84e04e5e16cff49b1d6fdc9a
 => => resolve docker.io/library/ubuntu:noble-20231221@sha256:38d380071057443ebb6d65566c03a98eaec411e84e04e5e16cff49b1d6fdc9a
 => => sha256:38d380071057443ebb6d65566c03a98eaec411e84e04e5e16cff49b1d6fdc9a0 1.13kB / 1.13kB
 => => sha256:145bacc9db29ff9c9c021284e5b7b22f1193fc38556c578250c926cf3c883a13 424B / 424B
 => => sha256:21e1d0ba034dc2bf324fbbd0096382576e6f4a407881bd22300b0013ce8626cf 2.30kB / 2.30kB
 => => sha256:ce3d5b1e4efe5d8f8dd38f54962ab6cb52eeacff317c7388d838d3f6f6fef468 29.47MB / 29.47MB
 => => extracting sha256:ce3d5b1e4efe5d8f8dd38f54962ab6cb52eeacff317c7388d838d3f6f6fef468
 => [internal] load build context
 => => transferring context: 1.75kB
 => [2/6] WORKDIR /nginx
 => [3/6] RUN apt-get -y update
 => [4/6] RUN apt-get -y install nginx
 => [5/6] COPY index.html /usr/share/nginx
 => [6/6] ADD table2.tar /usr/share/nginx
 => exporting to image
 => => exporting layers
 => => writing image sha256:0c1422a767980d4aa565f13495d59350e960fd91b703d6fcc418446a9f87d4ce
 => => naming to docker.io/library/mynginx:1.1
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker run -d -p 8080:80 mynginx:1.1
783283efa13ab516b731818c4fbe1f87312481e6b48376314c39d4af4ee0540b
```

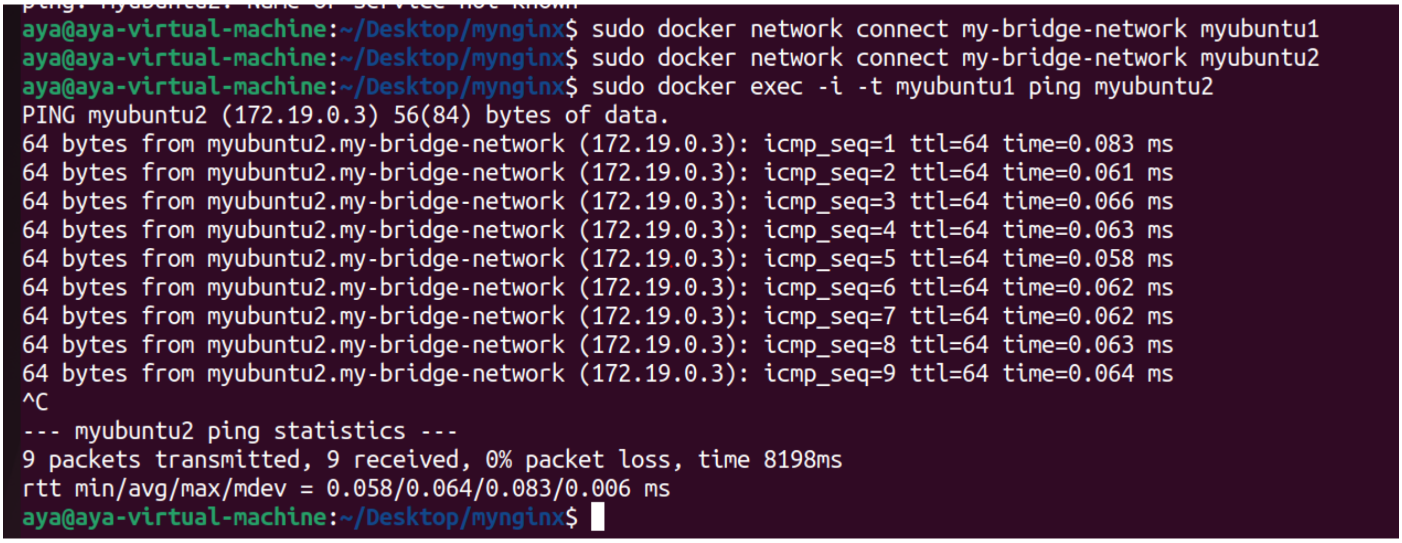
- P4: Create your bridge network, two containers from ubuntu image with different names and try to ping each other using NAME



```
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker network create -d bridge my-bridge-network
aadbd37e1cc1d75c3c2603b6d42af1fe92861f04c3f1eed1bd3d8616808418f3
aya@aya-virtual-machine:~/Desktop/mynginx$
```



```
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker run -itd --name myubuntu1 ubuntu
b75afa7b75407d71fb2651d7c8fce18addb55ab5181afcf12a2fb74a59475fb8
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker run -itd --name myubuntu2 ubuntu
ed42169e67fe715ecdcc7d1b72c1675e2bb3112f304b66f5a1ea414083d788da
```



```
ping: myubuntu2: Name or service not known
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker network connect my-bridge-network myubuntu1
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker network connect my-bridge-network myubuntu2
aya@aya-virtual-machine:~/Desktop/mynginx$ sudo docker exec -i -t myubuntu1 ping myubuntu2
PING myubuntu2 (172.19.0.3) 56(84) bytes of data.
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=1 ttl=64 time=0.083 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=2 ttl=64 time=0.061 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=3 ttl=64 time=0.066 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=4 ttl=64 time=0.063 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=5 ttl=64 time=0.058 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=6 ttl=64 time=0.062 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=7 ttl=64 time=0.062 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=8 ttl=64 time=0.063 ms
64 bytes from myubuntu2.my-bridge-network (172.19.0.3): icmp_seq=9 ttl=64 time=0.064 ms
^C
--- myubuntu2 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8198ms
rtt min/avg/max/mdev = 0.058/0.064/0.083/0.006 ms
aya@aya-virtual-machine:~/Desktop/mynginx$
```