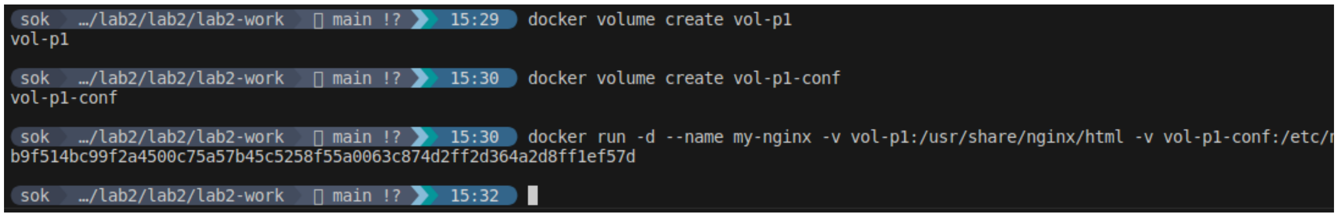
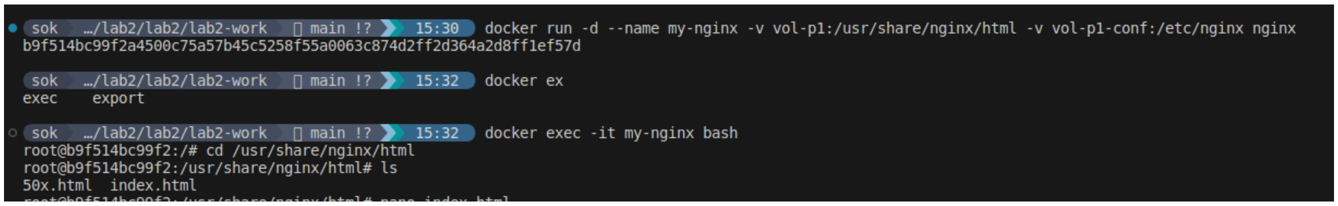
=-=-=-=-=-=-==========-=-=-=-=-============-=-=-=-=====================-=-==-
= Mohamed Ahmed Mohamed Abd elgani =
= Docker lab2 =
===

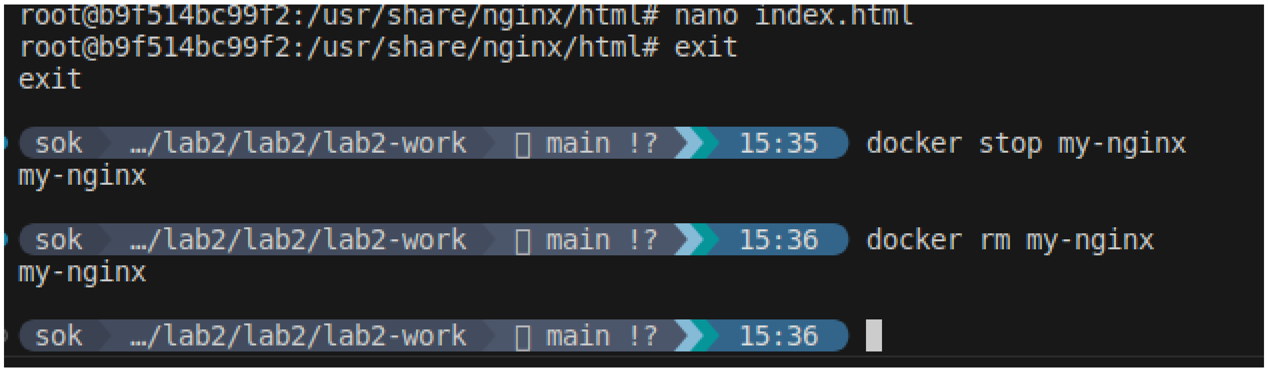
Problem 1:
▪ Run a container nginx with name my-nginx and attach 2 volumes to the container
▪ Volume1 for containing static html file
▪ Volume2 for containing nginx configuration
▪ Edit the html content
▪ Remove the container
▪ Run a new 2 containers with the following:
•Attach the 2 volumes that was attached to the previous container in two different ways (volume mount – bind mount)
•Map port 80 to port 8080 on you host machine
•Access the html files from your browser



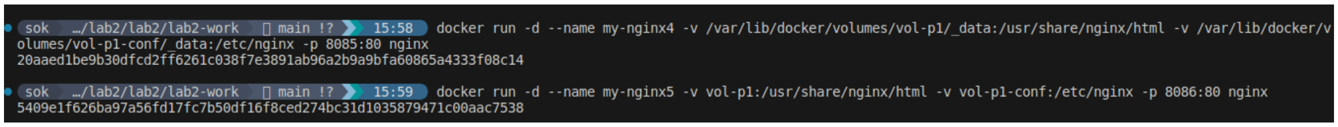
```
sok  …/lab2/lab2/lab2-work   main !?  15:29  docker volume create vol-p1
vol-p1

sok  …/lab2/lab2/lab2-work   main !?  15:30  docker volume create vol-p1-conf
vol-p1-conf

sok  …/lab2/lab2/lab2-work   main !?  15:30  docker run -d --name my-nginx -v vol-p1:/usr/share/nginx/html -v vol-p1-conf:/etc/r
b9f514bc99f2a4500c75a57b45c5258f55a0063c874d2ff2d364a2d8ff1ef57d

sok  …/lab2/lab2/lab2-work   main !?  15:32
```



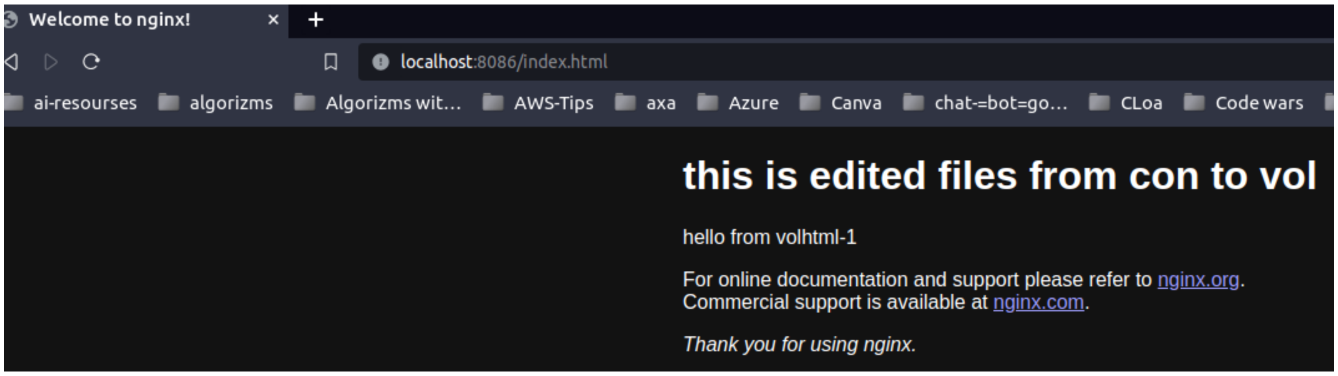
```
sok  …/lab2/lab2/lab2-work   main !?  15:30  docker run -d --name my-nginx -v vol-p1:/usr/share/nginx/html -v vol-p1-conf:/etc/nginx nginx
b9f514bc99f2a4500c75a57b45c5258f55a0063c874d2ff2d364a2d8ff1ef57d

sok  …/lab2/lab2/lab2-work   main !?  15:32  docker ex
exec    export

sok  …/lab2/lab2/lab2-work   main !?  15:32  docker exec -it my-nginx bash
root@b9f514bc99f2:/# cd /usr/share/nginx/html
root@b9f514bc99f2:/usr/share/nginx/html# ls
50x.html  index.html
```



```
root@b9f514bc99f2:/usr/share/nginx/html# nano index.html
root@b9f514bc99f2:/usr/share/nginx/html# exit
exit

sok  …/lab2/lab2/lab2-work   main !?  15:35  docker stop my-nginx
my-nginx

sok  …/lab2/lab2/lab2-work   main !?  15:36  docker rm my-nginx
my-nginx

sok  …/lab2/lab2/lab2-work   main !?  15:36
```



```
sok  …/lab2/lab2/lab2-work   main !?  15:58  docker run -d --name my-nginx4 -v /var/lib/docker/volumes/vol-p1/_data:/usr/share/nginx/html -v /var/lib/docker/v
olumes/vol-p1-conf/_data:/etc/nginx -p 8085:80 nginx
20aaed1be9b30dfcd2ff6261c038f7e3891ab96a2b9a9bfa60865a4333f08c14

sok  …/lab2/lab2/lab2-work   main !?  15:59  docker run -d --name my-nginx5 -v vol-p1:/usr/share/nginx/html -v vol-p1-conf:/etc/nginx -p 8086:80 nginx
5409e1f626ba97a56fd17fc7b50df16f8ced274bc31d1035879471c00aac7538
```

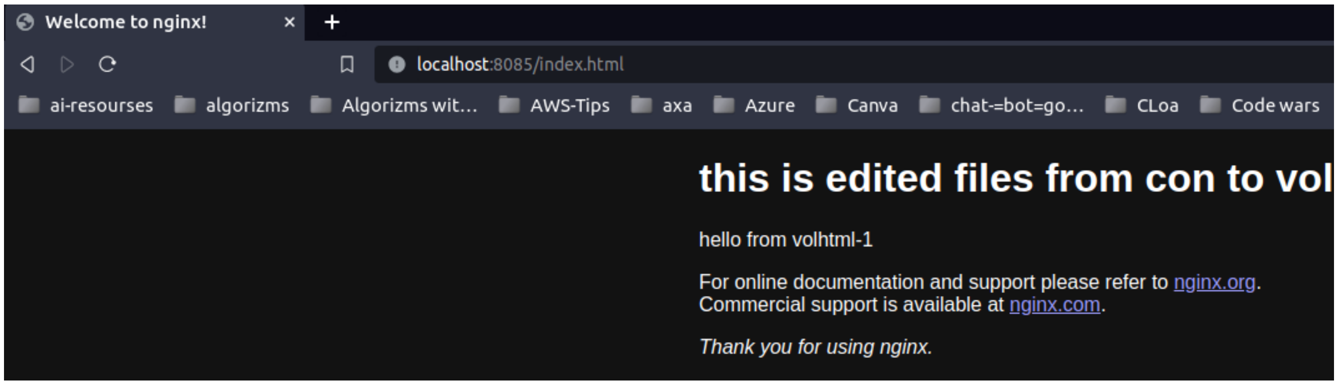


this is edited files from con to vol

hello from volhtml-1

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.



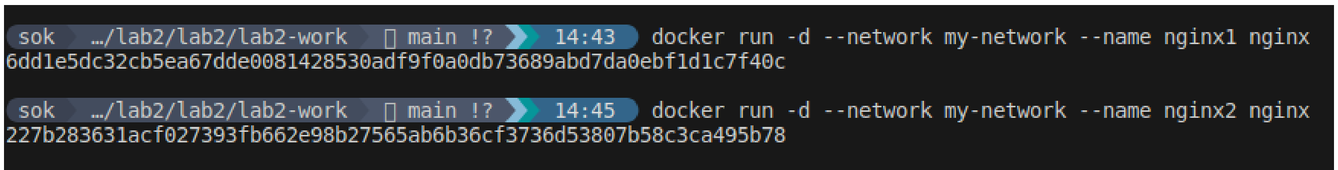
this is edited files from con to vol

hello from volhtml-1

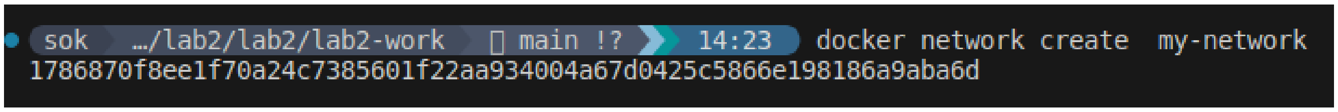
For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

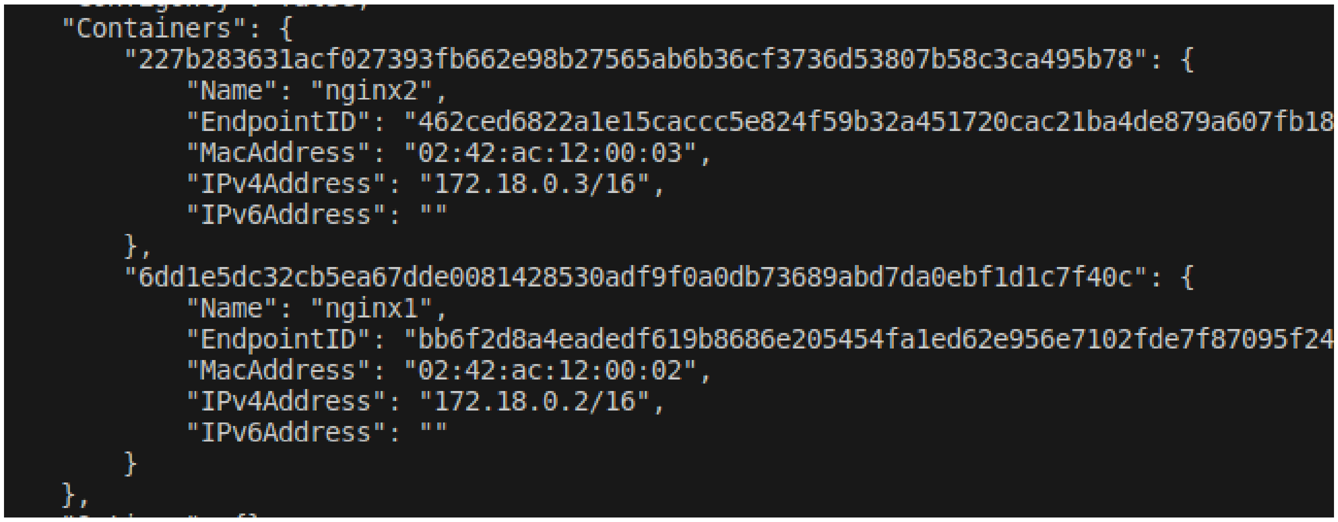
Problem 2:
▪ Create 2 nginx containers with network type bridge, enter to one of
them and use curl command to view the content of the othercontainer.



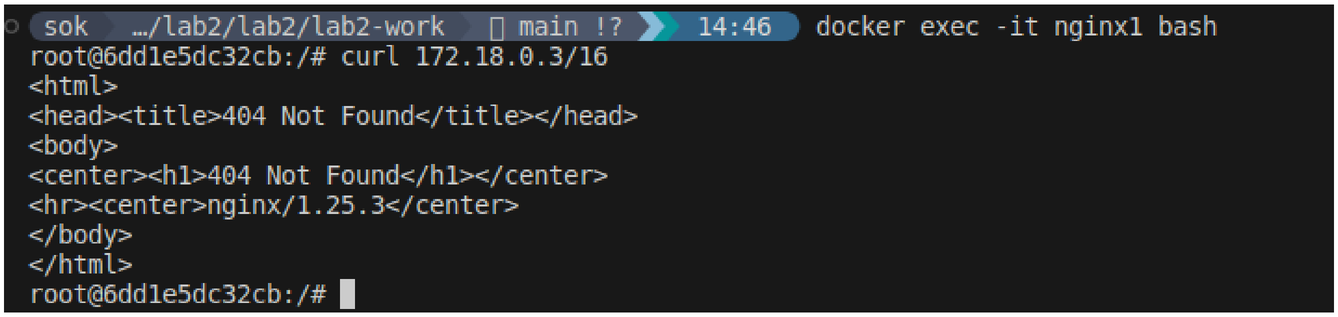
```
 sok  …/lab2/lab2/lab2-work   main !?  14:43  docker run -d --network my-network --name nginx1 nginx
6dd1e5dc32cb5ea67dde0081428530adf9f0a0db73689abd7da0ebf1d1c7f40c

 sok  …/lab2/lab2/lab2-work   main !?  14:45  docker run -d --network my-network --name nginx2 nginx
227b283631acf027393fb662e98b27565ab6b36cf3736d53807b58c3ca495b78
```



```
sok  …/lab2/lab2/lab2-work   main !?  14:23  docker network create  my-network
1786870f8ee1f70a24c7385601f22aa934004a67d0425c5866e198186a9aba6d
```

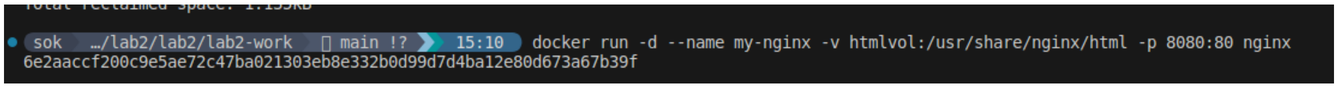


```
    "Containers": {
        "227b283631acf027393fb662e98b27565ab6b36cf3736d53807b58c3ca495b78": {
            "Name": "nginx2",
            "EndpointID": "462ced6822a1e15caccc5e824f59b32a451720cac21ba4de879a607fb18
            "MacAddress": "02:42:ac:12:00:03",
            "IPv4Address": "172.18.0.3/16",
            "IPv6Address": ""
        },
        "6dd1e5dc32cb5ea67dde0081428530adf9f0a0db73689abd7da0ebf1d1c7f40c": {
            "Name": "nginx1",
            "EndpointID": "bb6f2d8a4eadedf619b8686e205454fa1ed62e956e7102fde7f87095f24
            "MacAddress": "02:42:ac:12:00:02",
            "IPv4Address": "172.18.0.2/16",
            "IPv6Address": ""
        }
    },
```

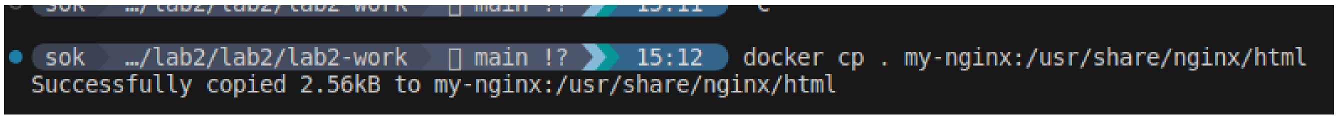


```
sok  …/lab2/lab2/lab2-work   main !?  14:46  docker exec -it nginx1 bash
root@6dd1e5dc32cb:/# curl 172.18.0.3/16
<html>
<head><title>404 Not Found</title></head>
<body>
<center><h1>404 Not Found</h1></center>
<hr><center>nginx/1.25.3</center>
</body>
</html>
root@6dd1e5dc32cb:/#
```

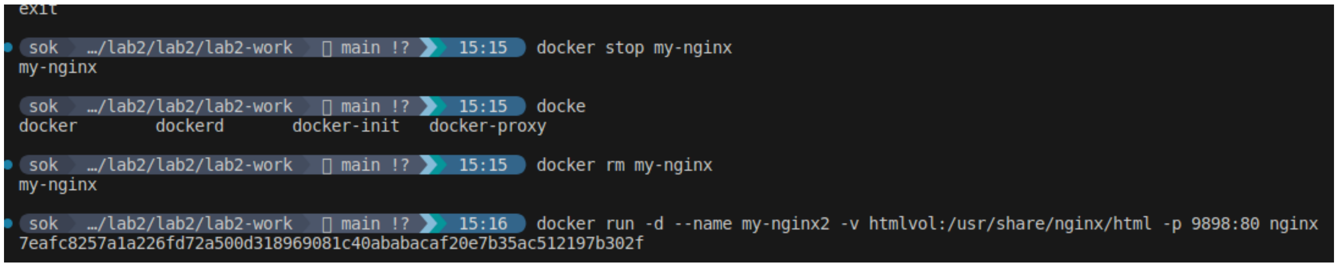
Problem 3:
Run a container Nginx with name my-nginx and attach a volume for containing static html file
Remove the container
Run a new container with the following:
Attach the volume that was attached to the previous container
Map port 80 to port 9898 on you host machine
Access the html files from your browser



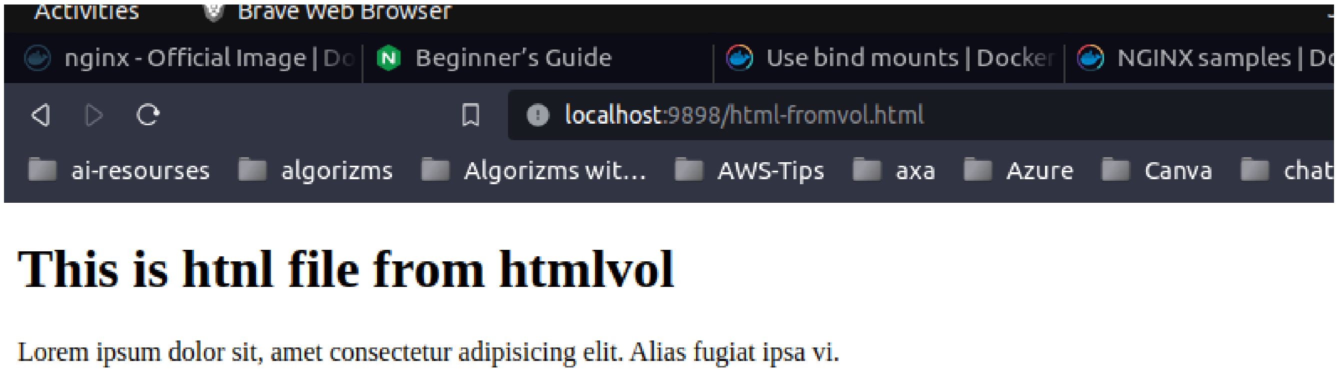
```
sok  …/lab2/lab2/lab2-work   main !?  15:10  docker run -d --name my-nginx -v htmlvol:/usr/share/nginx/html -p 8080:80 nginx
6e2aaccf200c9e5ae72c47ba021303eb8e332b0d99d7d4ba12e80d673a67b39f
```



```
sok  …/lab2/lab2/lab2-work   main !?  15:12  docker cp . my-nginx:/usr/share/nginx/html
Successfully copied 2.56kB to my-nginx:/usr/share/nginx/html
```



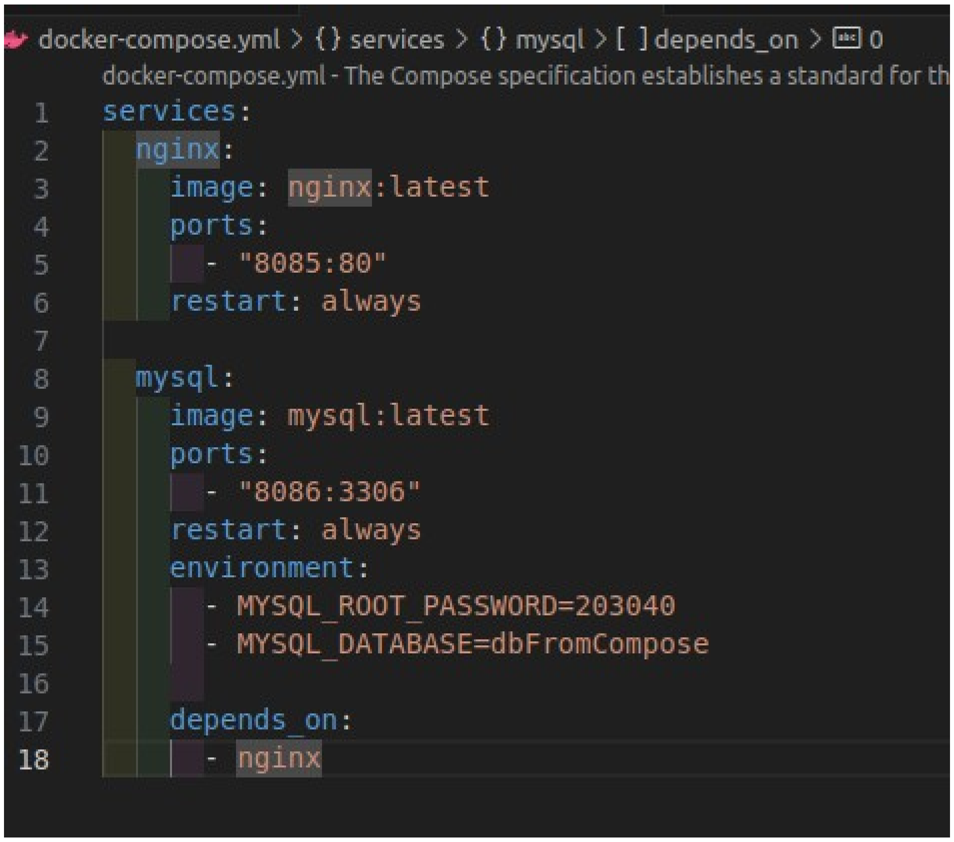
```
exit
sok  …/lab2/lab2/lab2-work   main !?  15:15  docker stop my-nginx
my-nginx

sok  …/lab2/lab2/lab2-work   main !?  15:15  docke
docker        dockerd       docker-init   docker-proxy

sok  …/lab2/lab2/lab2-work   main !?  15:15  docker rm my-nginx
my-nginx

sok  …/lab2/lab2/lab2-work   main !?  15:16  docker run -d --name my-nginx2 -v htmlvol:/usr/share/nginx/html -p 9898:80 nginx
7eafc8257a1a226fd72a500d318969081c40ababacaf20e7b35ac512197b302f
```



This is htnl file from htmlvol

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Alias fugiat ipsa vi.

Problem 4:
▪ Create docker compose with:
Two services (nginx and mysql)
Add needed ports and environments for both services
Mysql service depends on nginx service


docker-compose.yml > {} services > {} mysql > [ ] depends_on > 0
docker-compose.yml - The Compose specification establishes a standard for th

```yaml
services:
  nginx:
    image: nginx:latest
    ports:
      - "8085:80"
    restart: always

  mysql:
    image: mysql:latest
    ports:
      - "8086:3306"
    restart: always
    environment:
      - MYSQL_ROOT_PASSWORD=203040
      - MYSQL_DATABASE=dbFromCompose

    depends_on:
      - nginx
```