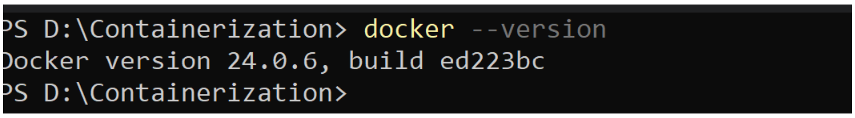
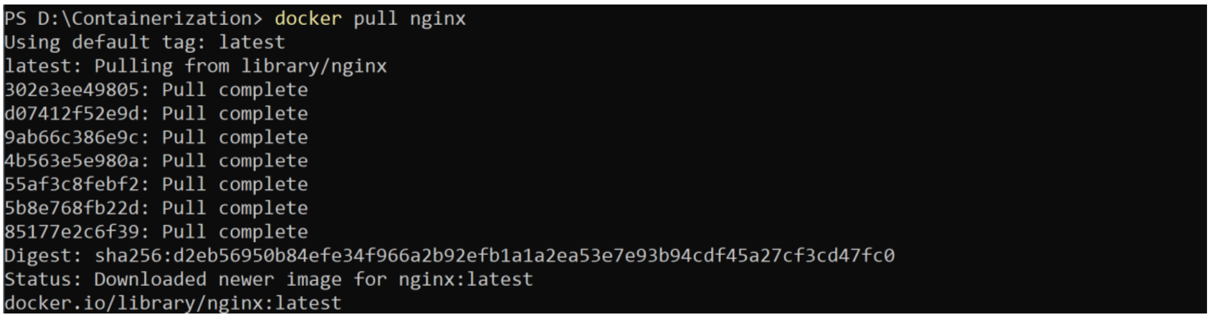
DOCKER DOCUMENTATION

1. **Verify Installation**



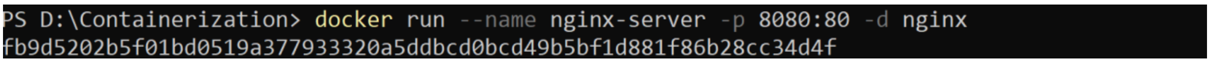
```
PS D:\Containerization> docker --version
Docker version 24.0.6, build ed223bc
PS D:\Containerization>
```

2. **Pull the Nginx Image**



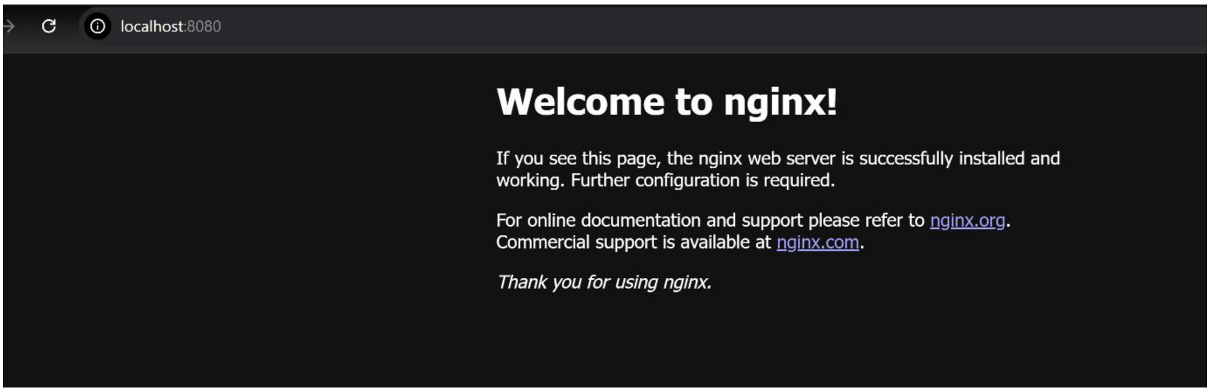
```
PS D:\Containerization> docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
302e3ee49805: Pull complete
d07412f52e9d: Pull complete
9ab66c386e9c: Pull complete
4b563e5e980a: Pull complete
55af3c8febf2: Pull complete
5b8e768fb22d: Pull complete
85177e2c6f39: Pull complete
Digest: sha256:d2eb56950b84efe34f966a2b92efb1a1a2ea53e7e93b94cdf45a27cf3cd47fc0
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

3. **Run Nginx Container**



```
PS D:\Containerization> docker run --name nginx-server -p 8080:80 -d nginx
fb9d5202b5f01bd0519a377933320a5ddbcd0bcd49b5bf1d881f86b28cc34d4f
```

4. **Accessing Nginx to my web browser**



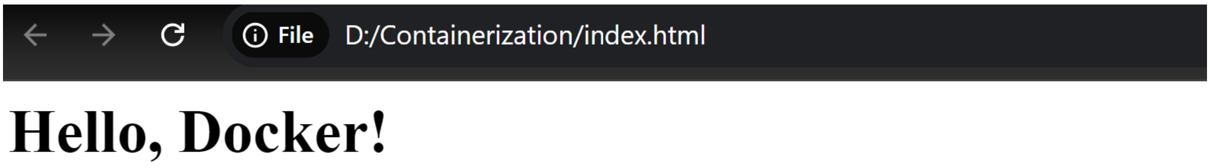
Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

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

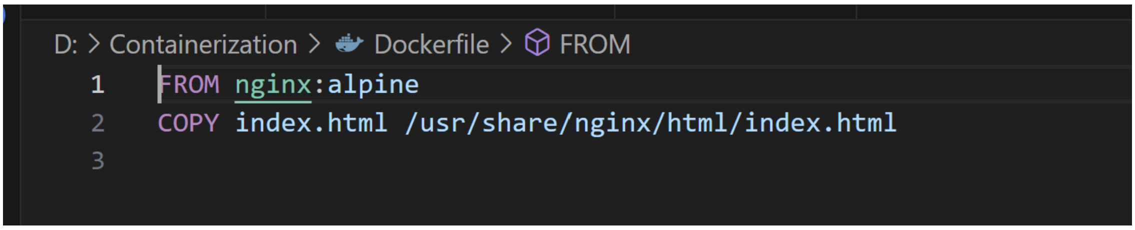
Thank you for using nginx.

5. **Created index.html**



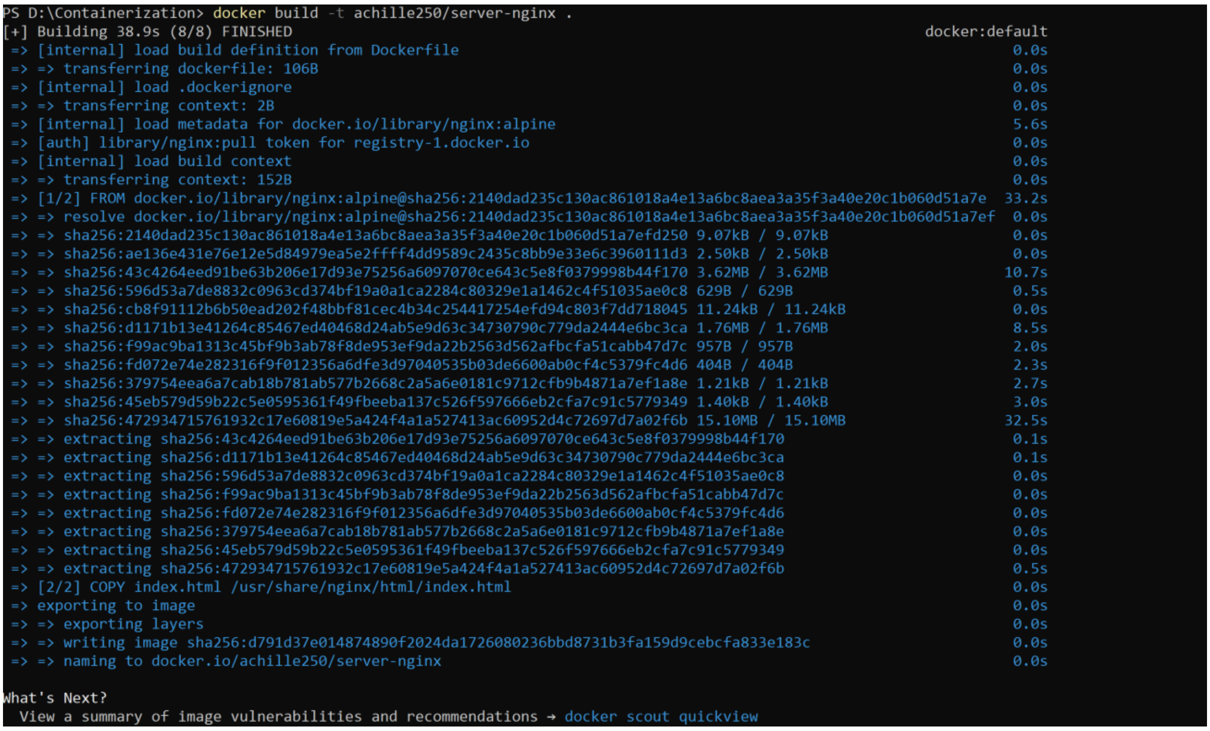
Hello, Docker!

6. Write a Dockerfile



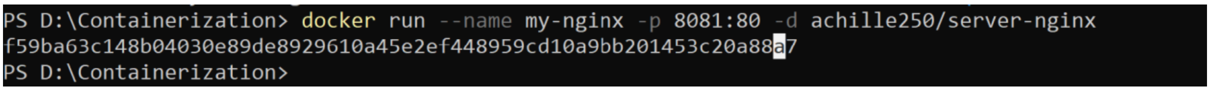
```
FROM nginx:alpine
COPY index.html /usr/share/nginx/html/index.html

```

7. Build the Docker Image



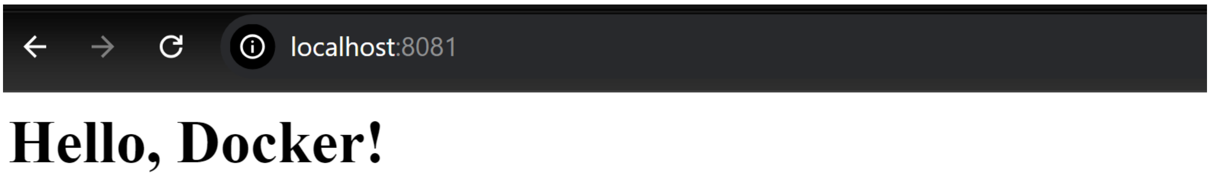
```
PS D:\Containerization> docker build -t achille250/server-nginx .
[+] Building 38.9s (8/8) FINISHED                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                           0.0s
 => => transferring dockerfile: 106B                                                                           0.0s
 => [internal] load .dockerignore                                                                              0.0s
 => => transferring context: 2B                                                                                0.0s
 => [internal] load metadata for docker.io/library/nginx:alpine                                                5.6s
 => [auth] library/nginx:pull token for registry-1.docker.io                                                   0.0s
 => [internal] load build context                                                                              0.0s
 => => transferring context: 152B                                                                              0.0s
 => [1/2] FROM docker.io/library/nginx:alpine@sha256:2140dad235c130ac861018a4e13a6bc8aea3a35f3a40e20c1b060d51a7e  33.2s
 => => resolve docker.io/library/nginx:alpine@sha256:2140dad235c130ac861018a4e13a6bc8aea3a35f3a40e20c1b060d51a7ef  0.0s
 => => sha256:2140dad235c130ac861018a4e13a6bc8aea3a35f3a40e20c1b060d51a7efd250 9.07kB / 9.07kB                  0.0s
 => => sha256:ae136e431e76e12e5d84979ea5e2ffff4dd9589c2435c8bb9e33e6c3960111d3 2.50kB / 2.50kB                  0.0s
 => => sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170 3.62MB / 3.62MB                 10.7s
 => => sha256:596d53a7de8832c0963cd374bf19a0a1ca2284c80329e1a1462c4f51035ae0c8 629B / 629B                      0.5s
 => => sha256:cb8f91112b6b50ead202f48bbf81cec4b34c254417254efd94c803f7dd718045 11.24kB / 11.24kB                0.0s
 => => sha256:d1171b13e41264c85467ed40468d24ab5e9d63c34730790c779da2444e6bc3ca 1.76MB / 1.76MB                  8.5s
 => => sha256:f99ac9ba1313c45bf9b3ab78f8de953ef9da22b2563d562afbcfa51cabb47d7c 957B / 957B                      2.0s
 => => sha256:fd072e74e282316f9f012356a6dfe3d97040535b03de6600ab0cf4c5379fc4d6 404B / 404B                      2.3s
 => => sha256:379754eea6a7cab18b781ab577b2668c2a5a6e0181c9712cfb9b4871a7ef1a8e 1.21kB / 1.21kB                  2.7s
 => => sha256:45eb579d59b22c5e0595361f49fbeeba137c526f597666eb2cfa7c91c5779349 1.40kB / 1.40kB                  3.0s
 => => sha256:472934715761932c17e60819e5a424f4a1a527413ac60952d4c72697d7a02f6b 15.10MB / 15.10MB               32.5s
 => => extracting sha256:43c4264eed91be63b206e17d93e75256a6097070ce643c5e8f0379998b44f170                       0.1s
 => => extracting sha256:d1171b13e41264c85467ed40468d24ab5e9d63c34730790c779da2444e6bc3ca                       0.1s
 => => extracting sha256:596d53a7de8832c0963cd374bf19a0a1ca2284c80329e1a1462c4f51035ae0c8                       0.0s
 => => extracting sha256:f99ac9ba1313c45bf9b3ab78f8de953ef9da22b2563d562afbcfa51cabb47d7c                       0.0s
 => => extracting sha256:fd072e74e282316f9f012356a6dfe3d97040535b03de6600ab0cf4c5379fc4d6                       0.0s
 => => extracting sha256:379754eea6a7cab18b781ab577b2668c2a5a6e0181c9712cfb9b4871a7ef1a8e                       0.0s
 => => extracting sha256:45eb579d59b22c5e0595361f49fbeeba137c526f597666eb2cfa7c91c5779349                       0.0s
 => => extracting sha256:472934715761932c17e60819e5a424f4a1a527413ac60952d4c72697d7a02f6b                       0.5s
 => [2/2] COPY index.html /usr/share/nginx/html/index.html                                                     0.0s
 => exporting to image                                                                                         0.0s
 => => exporting layers                                                                                        0.0s
 => => writing image sha256:d791d37e014874890f2024da1726080236bbd8731b3fa159d9cebcfa833e183c                    0.0s
 => => naming to docker.io/achille250/server-nginx                                                             0.0s

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview
```

8. Run the Docker Image



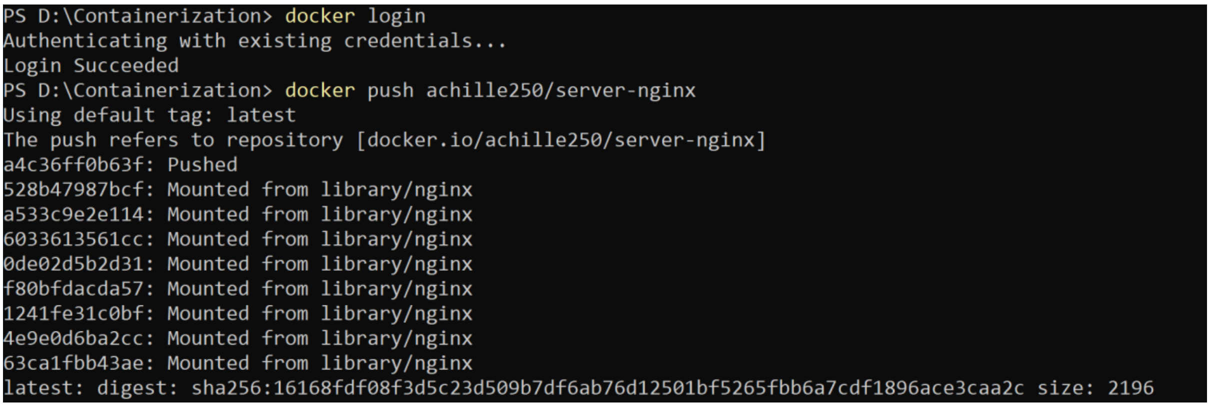
```
PS D:\Containerization> docker run --name my-nginx -p 8081:80 -d achille250/server-nginx
f59ba63c148b04030e89de8929610a45e2ef448959cd10a9bb201453c20a88a7
PS D:\Containerization>
```

9. Verify if it works on localhost

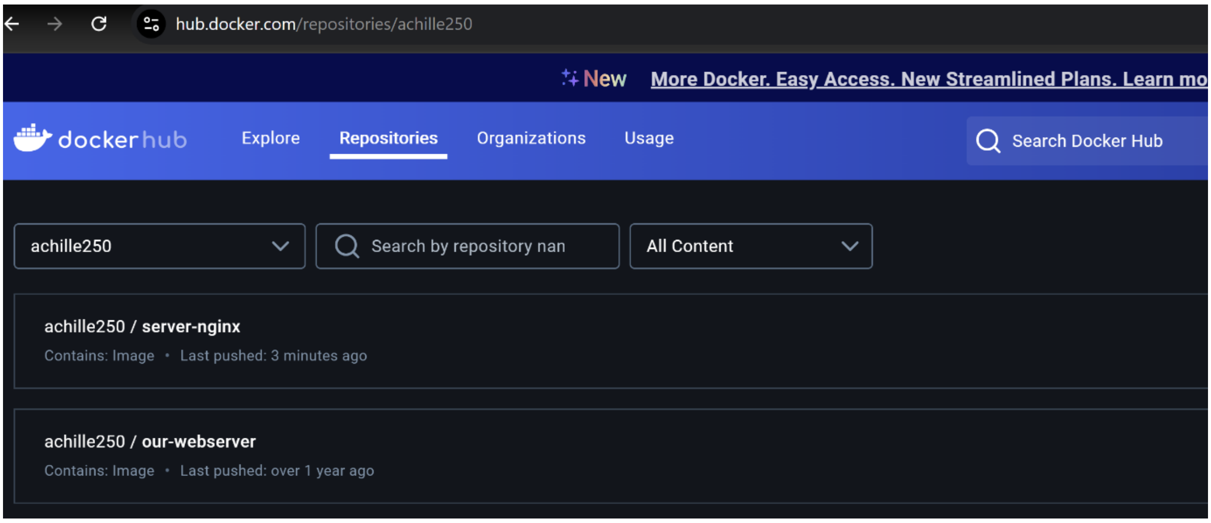


Hello, Docker!

10. Push Docker image to DockerHub



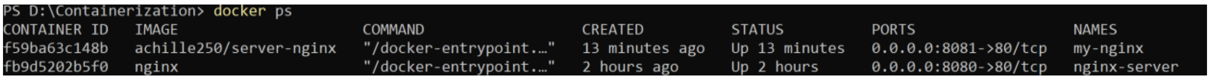
```
PS D:\Containerization> docker login
Authenticating with existing credentials...
Login Succeeded
PS D:\Containerization> docker push achille250/server-nginx
Using default tag: latest
The push refers to repository [docker.io/achille250/server-nginx]
a4c36ff0b63f: Pushed
528b47987bcf: Mounted from library/nginx
a533c9e2e114: Mounted from library/nginx
6033613561cc: Mounted from library/nginx
0de02d5b2d31: Mounted from library/nginx
f80bfdacda57: Mounted from library/nginx
1241fe31c0bf: Mounted from library/nginx
4e9e0d6ba2cc: Mounted from library/nginx
63ca1fbb43ae: Mounted from library/nginx
latest: digest: sha256:16168fdf08f3d5c23d509b7df6ab76d12501bf5265fbb6a7cdf1896ace3caa2c size: 2196
```



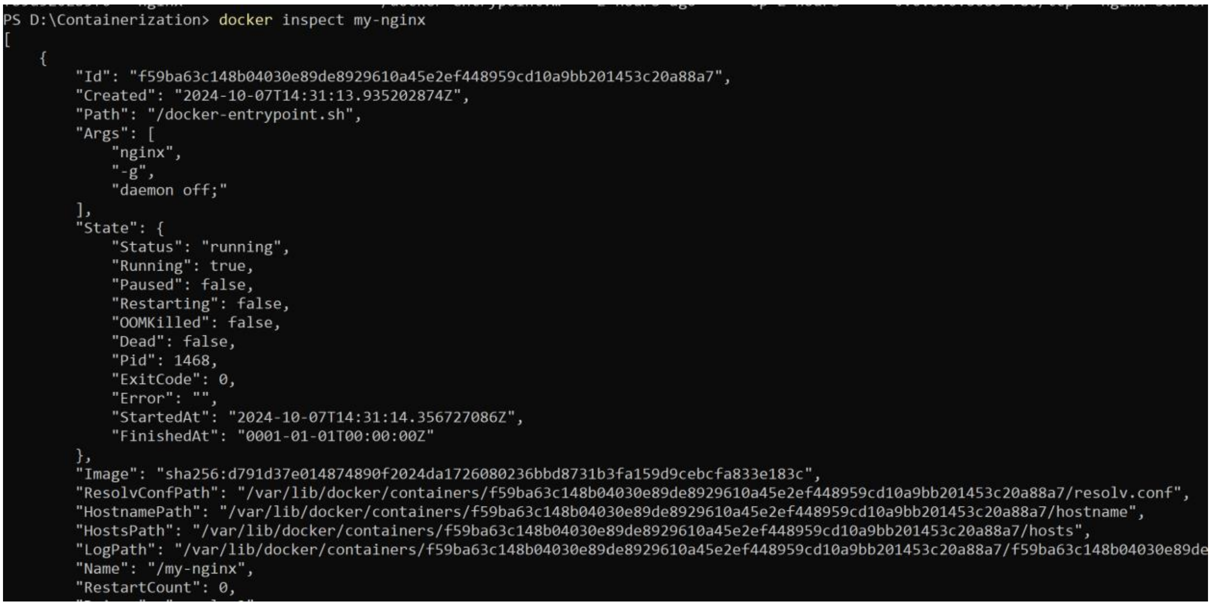
achille250 / server-nginx
Contains: Image • Last pushed: 3 minutes ago

achille250 / our-webserver
Contains: Image • Last pushed: over 1 year ago

11. List and inspect a running container

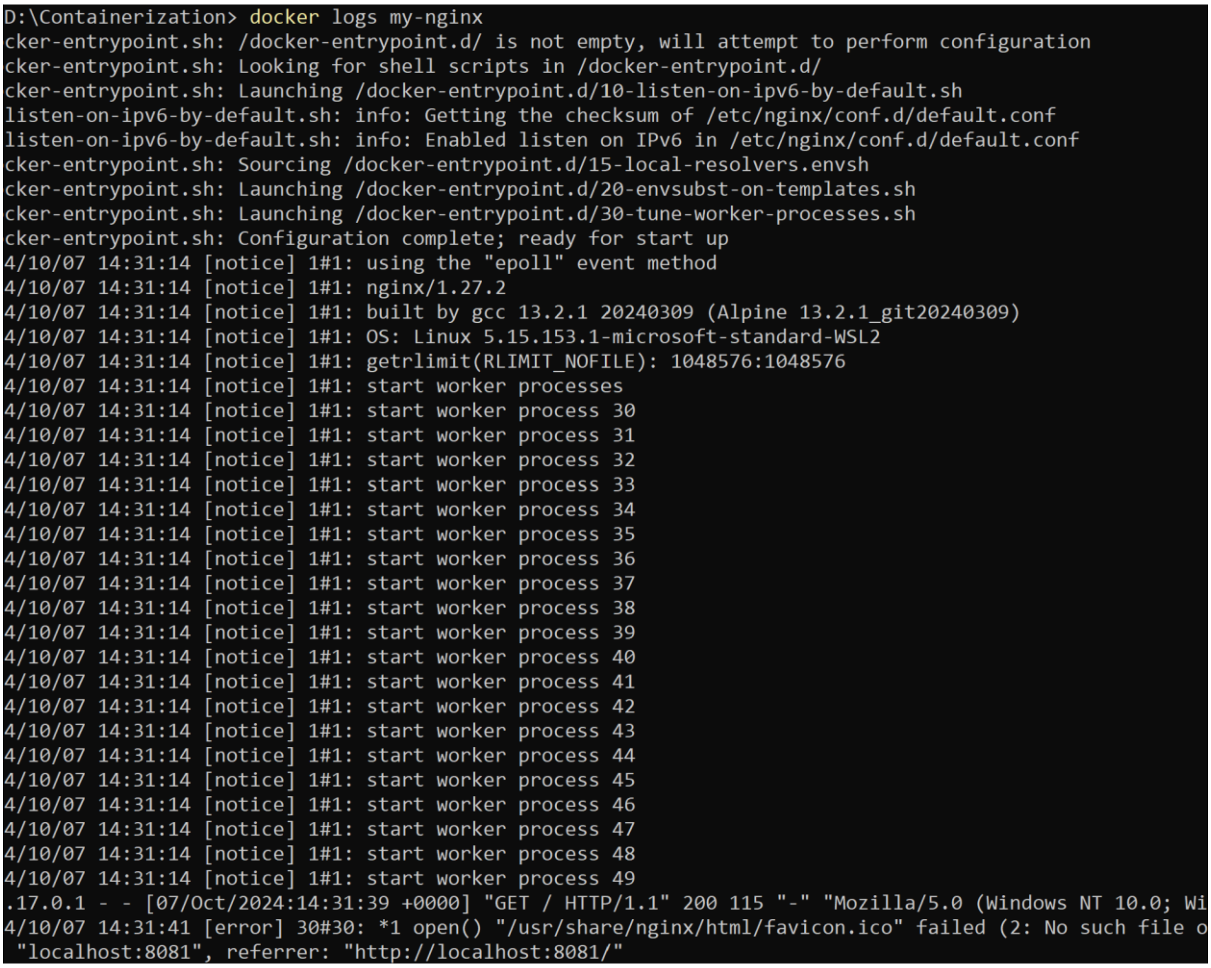


```
PS D:\Containerization> docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS          PORTS                  NAMES
f59ba63c148b   achille250/server-nginx  "/docker-entrypoint.…"   13 minutes ago   Up 13 minutes   0.0.0.0:8081->80/tcp   my-nginx
fb9d5202b5f0   nginx                    "/docker-entrypoint.…"   2 hours ago      Up 2 hours      0.0.0.0:8080->80/tcp   nginx-server
```



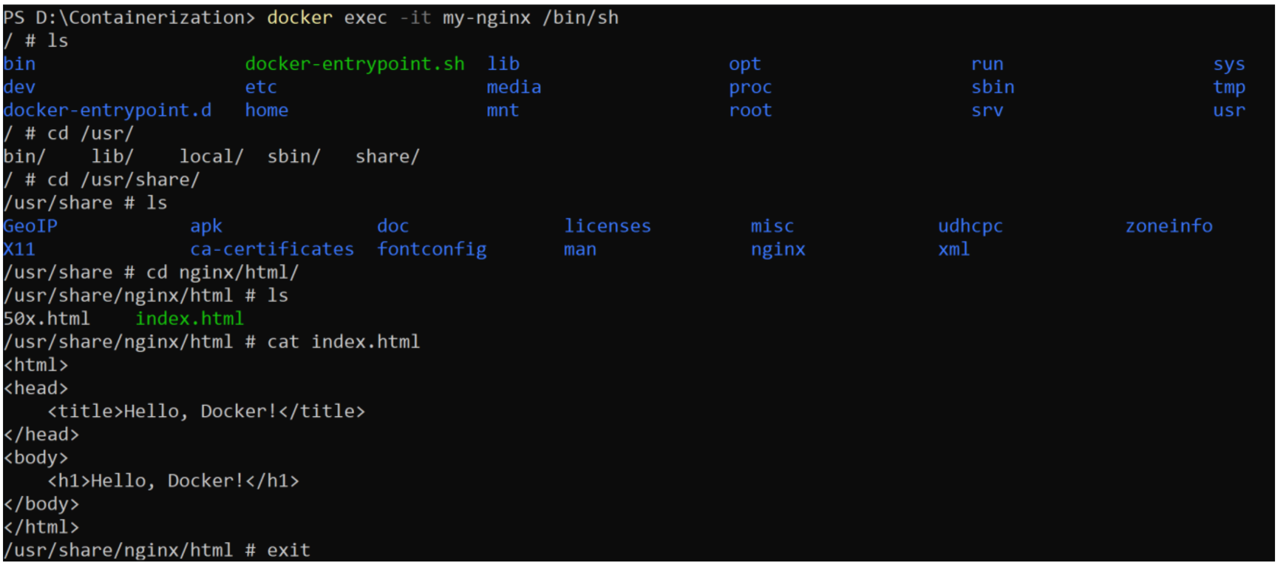
```
PS D:\Containerization> docker inspect my-nginx
[
    {
        "Id": "f59ba63c148b04030e89de8929610a45e2ef448959cd10a9bb201453c20a88a7",
        "Created": "2024-10-07T14:31:13.935202874Z",
        "Path": "/docker-entrypoint.sh",
        "Args": [
            "nginx",
            "-g",
            "daemon off;"
        ],
        "State": {
            "Status": "running",
            "Running": true,
            "Paused": false,
            "Restarting": false,
            "OOMKilled": false,
            "Dead": false,
            "Pid": 1468,
            "ExitCode": 0,
            "Error": "",
            "StartedAt": "2024-10-07T14:31:14.356727086Z",
            "FinishedAt": "0001-01-01T00:00:00Z"
        },
        "Image": "sha256:d791d37e014874890f2024da1726080236bbd8731b3fa159d9cebcfa833e183c",
        "ResolvConfPath": "/var/lib/docker/containers/f59ba63c148b04030e89de8929610a45e2ef448959cd10a9bb201453c20a88a7/resolv.conf",
        "HostnamePath": "/var/lib/docker/containers/f59ba63c148b04030e89de8929610a45e2ef448959cd10a9bb201453c20a88a7/hostname",
        "HostsPath": "/var/lib/docker/containers/f59ba63c148b04030e89de8929610a45e2ef448959cd10a9bb201453c20a88a7/hosts",
        "LogPath": "/var/lib/docker/containers/f59ba63c148b04030e89de8929610a45e2ef448959cd10a9bb201453c20a88a7/f59ba63c148b04030e89de
        "Name": "/my-nginx",
        "RestartCount": 0,
```

12. View the Logs of a Container



```
D:\Containerization> docker logs my-nginx
cker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
cker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
cker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
cker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
cker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
cker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
cker-entrypoint.sh: Configuration complete; ready for start up
4/10/07 14:31:14 [notice] 1#1: using the "epoll" event method
4/10/07 14:31:14 [notice] 1#1: nginx/1.27.2
4/10/07 14:31:14 [notice] 1#1: built by gcc 13.2.1 20240309 (Alpine 13.2.1_git20240309)
4/10/07 14:31:14 [notice] 1#1: OS: Linux 5.15.153.1-microsoft-standard-WSL2
4/10/07 14:31:14 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
4/10/07 14:31:14 [notice] 1#1: start worker processes
4/10/07 14:31:14 [notice] 1#1: start worker process 30
4/10/07 14:31:14 [notice] 1#1: start worker process 31
4/10/07 14:31:14 [notice] 1#1: start worker process 32
4/10/07 14:31:14 [notice] 1#1: start worker process 33
4/10/07 14:31:14 [notice] 1#1: start worker process 34
4/10/07 14:31:14 [notice] 1#1: start worker process 35
4/10/07 14:31:14 [notice] 1#1: start worker process 36
4/10/07 14:31:14 [notice] 1#1: start worker process 37
4/10/07 14:31:14 [notice] 1#1: start worker process 38
4/10/07 14:31:14 [notice] 1#1: start worker process 39
4/10/07 14:31:14 [notice] 1#1: start worker process 40
4/10/07 14:31:14 [notice] 1#1: start worker process 41
4/10/07 14:31:14 [notice] 1#1: start worker process 42
4/10/07 14:31:14 [notice] 1#1: start worker process 43
4/10/07 14:31:14 [notice] 1#1: start worker process 44
4/10/07 14:31:14 [notice] 1#1: start worker process 45
4/10/07 14:31:14 [notice] 1#1: start worker process 46
4/10/07 14:31:14 [notice] 1#1: start worker process 47
4/10/07 14:31:14 [notice] 1#1: start worker process 48
4/10/07 14:31:14 [notice] 1#1: start worker process 49
.17.0.1 - - [07/Oct/2024:14:31:39 +0000] "GET / HTTP/1.1" 200 115 "-" "Mozilla/5.0 (Windows NT 10.0; Wi
4/10/07 14:31:41 [error] 30#30: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file o
 "localhost:8081", referrer: "http://localhost:8081/"
```

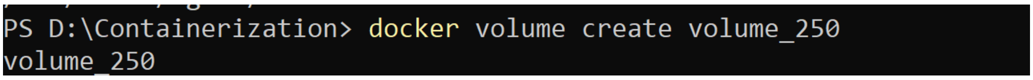
13. Execute Commands in a Running Container



```
PS D:\Containerization> docker exec -it my-nginx /bin/sh
/ # ls
bin                   docker-entrypoint.sh  lib                   opt                   run                   sys
dev                   etc                   media                 proc                  sbin                  tmp
docker-entrypoint.d   home                  mnt                   root                  srv                   usr
/ # cd /usr/
bin/    lib/    local/  sbin/   share/
/ # cd /usr/share/
/usr/share # ls
GeoIP            apk              doc              licenses         misc             udhcpc           zoneinfo
X11              ca-certificates  fontconfig       man              nginx            xml
/usr/share # cd nginx/html/
/usr/share/nginx/html # ls
50x.html    index.html
/usr/share/nginx/html # cat index.html
<html>
<head>
    <title>Hello, Docker!</title>
</head>
<body>
    <h1>Hello, Docker!</h1>
</body>
</html>
/usr/share/nginx/html # exit
```

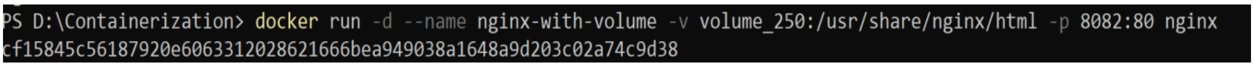
14. Create a Docker Volume and Mount it to a Container

- **Create a volume**



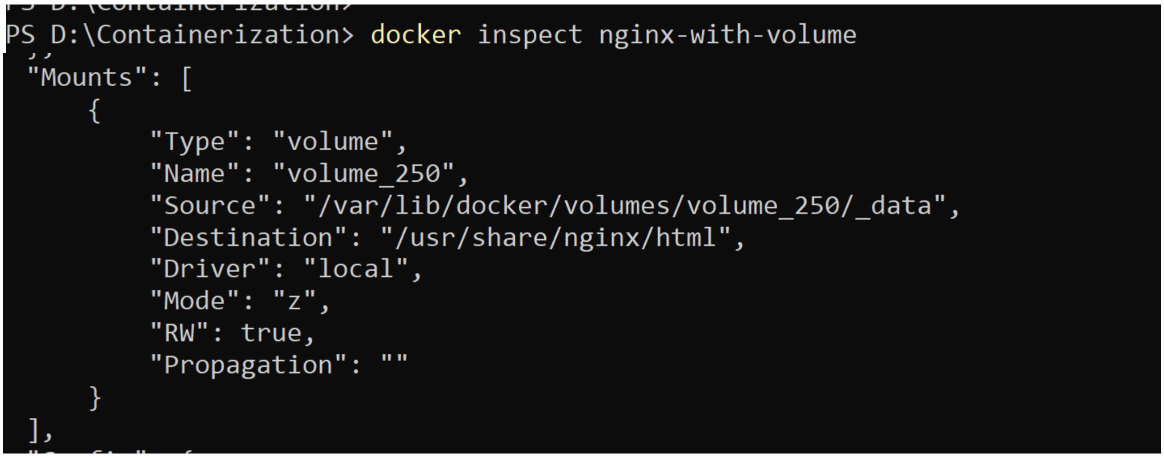
```
PS D:\Containerization> docker volume create volume_250
volume_250
```

- **Run a container and mount the volume**



```
PS D:\Containerization> docker run -d --name nginx-with-volume -v volume_250:/usr/share/nginx/html -p 8082:80 nginx
cf15845c56187920e6063312028621666bea949038a1648a9d203c02a74c9d38
```

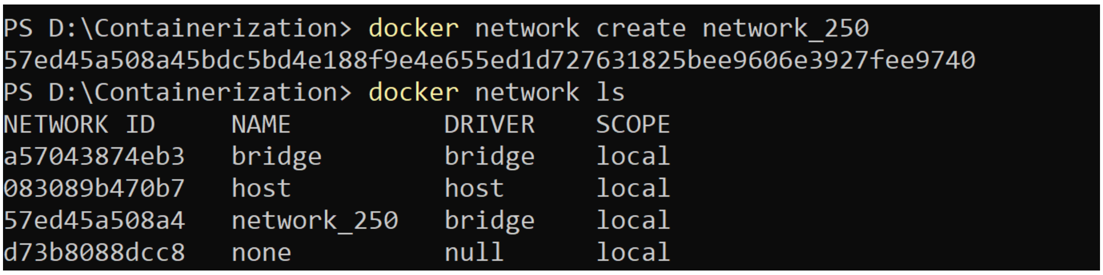
- **Inspect the container to verify that the volume is mounted**



```
PS D:\Containerization> docker inspect nginx-with-volume
 "Mounts": [
     {
         "Type": "volume",
         "Name": "volume_250",
         "Source": "/var/lib/docker/volumes/volume_250/_data",
         "Destination": "/usr/share/nginx/html",
         "Driver": "local",
         "Mode": "z",
         "RW": true,
         "Propagation": ""
     }
 ],
```

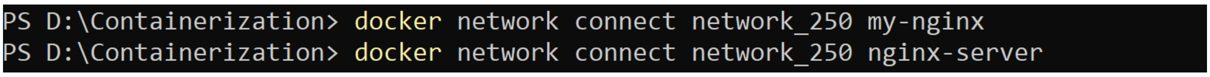
15. Create a Docker Network and Attach Two Containers to It

- **Create a docker network**



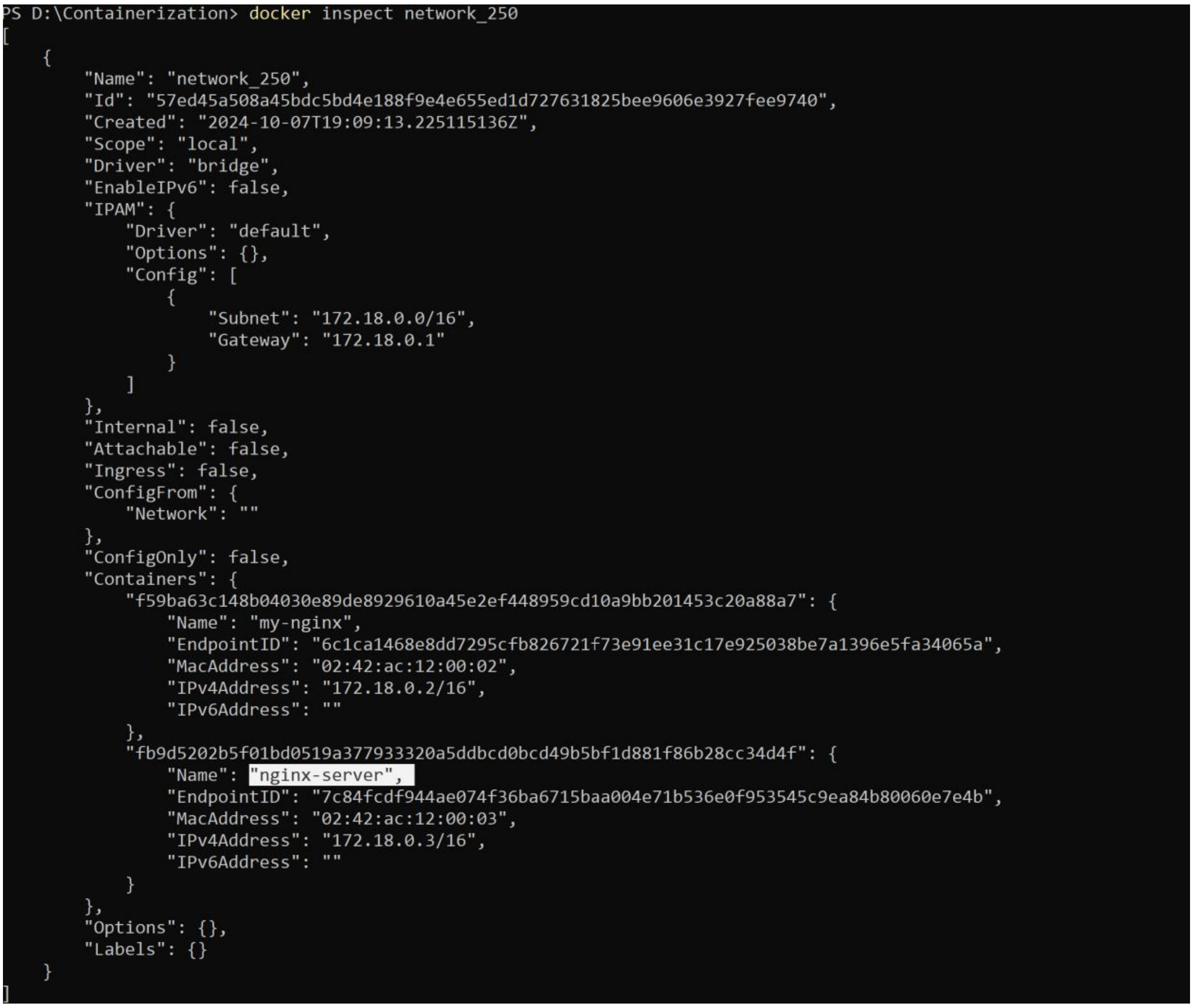
```
PS D:\Containerization> docker network create network_250
57ed45a508a45bdc5bd4e188f9e4e655ed1d727631825bee9606e3927fee9740
PS D:\Containerization> docker network ls
NETWORK ID     NAME          DRIVER    SCOPE
a57043874eb3   bridge        bridge    local
083089b470b7   host          host      local
57ed45a508a4   network_250   bridge    local
d73b8088dcc8   none          null      local
```

- **Attach the network to the existing containers**



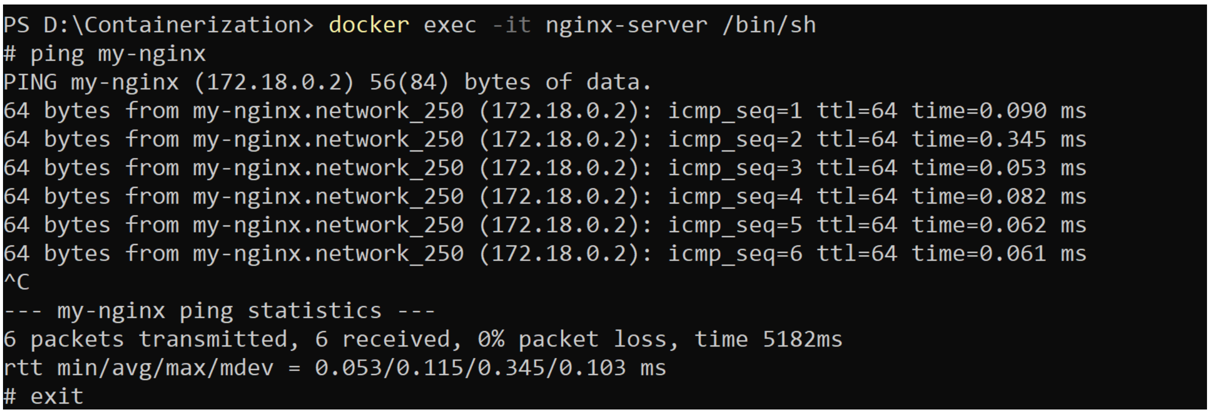
```
PS D:\Containerization> docker network connect network_250 my-nginx
PS D:\Containerization> docker network connect network_250 nginx-server
```

- **Verify the network connection**



```
PS D:\Containerization> docker inspect network_250
[
    {
        "Name": "network_250",
        "Id": "57ed45a508a45bdc5bd4e188f9e4e655ed1d727631825bee9606e3927fee9740",
        "Created": "2024-10-07T19:09:13.225115136Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "f59ba63c148b04030e89de8929610a45e2ef448959cd10a9bb201453c20a88a7": {
                "Name": "my-nginx",
                "EndpointID": "6c1ca1468e8dd7295cfb826721f73e91ee31c17e925038be7a1396e5fa34065a",
                "MacAddress": "02:42:ac:12:00:02",
                "IPv4Address": "172.18.0.2/16",
                "IPv6Address": ""
            },
            "fb9d5202b5f01bd0519a377933320a5ddbcd0bcd49b5bf1d881f86b28cc34d4f": {
                "Name": "nginx-server",
                "EndpointID": "7c84fcdf944ae074f36ba6715baa004e71b536e0f953545c9ea84b80060e7e4b",
                "MacAddress": "02:42:ac:12:00:03",
                "IPv4Address": "172.18.0.3/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {}
    }
]
```

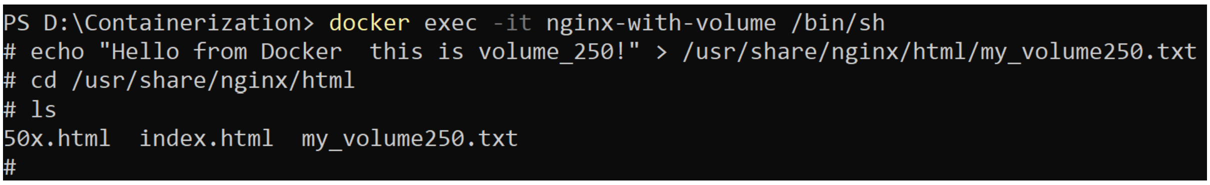
16. Ping the Other Container to Test Network Connection



```
PS D:\Containerization> docker exec -it nginx-server /bin/sh
# ping my-nginx
PING my-nginx (172.18.0.2) 56(84) bytes of data.
64 bytes from my-nginx.network_250 (172.18.0.2): icmp_seq=1 ttl=64 time=0.090 ms
64 bytes from my-nginx.network_250 (172.18.0.2): icmp_seq=2 ttl=64 time=0.345 ms
64 bytes from my-nginx.network_250 (172.18.0.2): icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from my-nginx.network_250 (172.18.0.2): icmp_seq=4 ttl=64 time=0.082 ms
64 bytes from my-nginx.network_250 (172.18.0.2): icmp_seq=5 ttl=64 time=0.062 ms
64 bytes from my-nginx.network_250 (172.18.0.2): icmp_seq=6 ttl=64 time=0.061 ms
^C
--- my-nginx ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5182ms
rtt min/avg/max/mdev = 0.053/0.115/0.345/0.103 ms
# exit
```

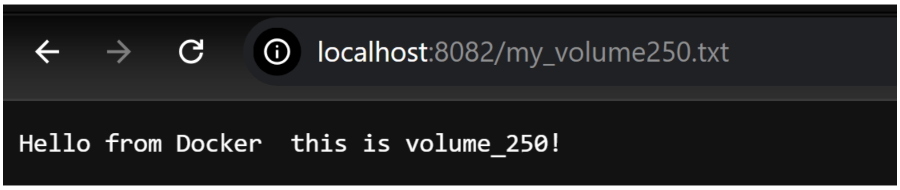
17. Use volume to keep file beyond the container lifetime

- **Write a file to the volume**



```
PS D:\Containerization> docker exec -it nginx-with-volume /bin/sh
# echo "Hello from Docker  this is volume_250!" > /usr/share/nginx/html/my_volume250.txt
# cd /usr/share/nginx/html
# ls
50x.html  index.html  my_volume250.txt
#
```

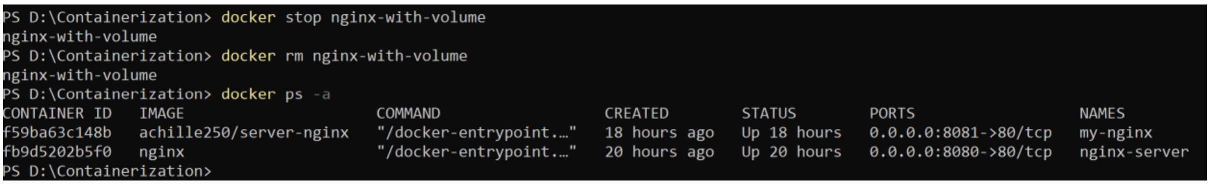
- **Check the file on host machine**



localhost:8082/my_volume250.txt

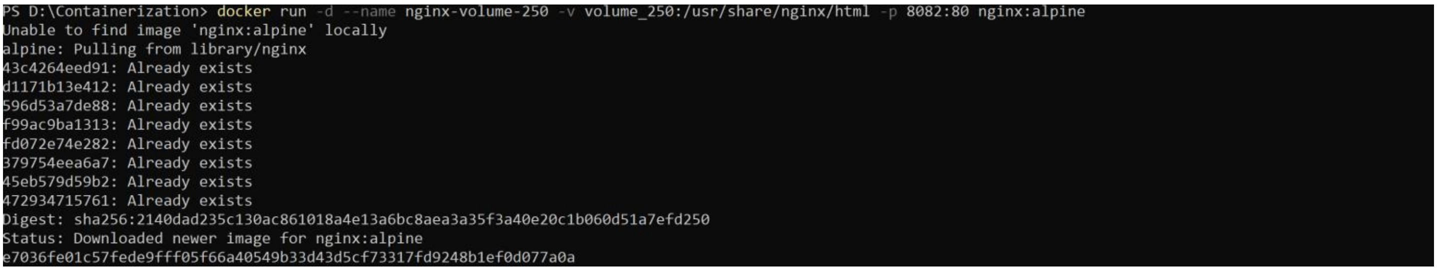
```
Hello from Docker  this is volume_250!
```

- **Stop and remove the container**



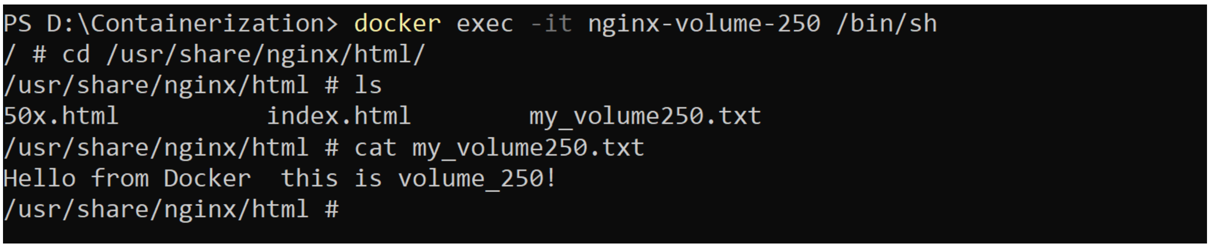
```
PS D:\Containerization> docker stop nginx-with-volume
nginx-with-volume
PS D:\Containerization> docker rm nginx-with-volume
nginx-with-volume
PS D:\Containerization> docker ps -a
CONTAINER ID   IMAGE                   COMMAND                  CREATED        STATUS        PORTS                  NAMES
f59ba63c148b   achille250/server-nginx "/docker-entrypoint.…"   18 hours ago   Up 18 hours   0.0.0.0:8081->80/tcp   my-nginx
fb9d5202b5f0   nginx                   "/docker-entrypoint.…"   20 hours ago   Up 20 hours   0.0.0.0:8080->80/tcp   nginx-server
PS D:\Containerization>
```

- **Start a new container with the same volume**



```
PS D:\Containerization> docker run -d --name nginx-volume-250 -v volume_250:/usr/share/nginx/html -p 8082:80 nginx:alpine
Unable to find image 'nginx:alpine' locally
alpine: Pulling from library/nginx
43c4264eed91: Already exists
d1171b13e412: Already exists
596d53a7de88: Already exists
f99ac9ba1313: Already exists
fd072e74e282: Already exists
379754eea6a7: Already exists
45eb579d59b2: Already exists
472934715761: Already exists
Digest: sha256:2140dad235c130ac861018a4e13a6bc8aea3a35f3a40e20c1b060d51a7efd250
Status: Downloaded newer image for nginx:alpine
e7036fe01c57fede9fff05f66a40549b33d43d5cf73317fd9248b1ef0d077a0a
```

- **Check if the file still exists**



```
PS D:\Containerization> docker exec -it nginx-volume-250 /bin/sh
/ # cd /usr/share/nginx/html/
/usr/share/nginx/html # ls
50x.html           index.html          my_volume250.txt
/usr/share/nginx/html # cat my_volume250.txt
Hello from Docker  this is volume_250!
/usr/share/nginx/html #
```