

Homework 2 - Docker Extended Topics

These exercises focus on Docker networking, volumes, bind mounts, Docker-in-Docker, resource limits, and restart policies.

Exercise 1

1. List all Docker networks.
2. Inspect the default Bridge network.
3. create a **new bridge user-defined network**.
4. Run a container attached to it and inspect its IP.

```
ubuntu@k8s-instance-6:~$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
c8fdc79ccbfb3       bridge              bridge               local
5d251adbccf2        dockernet           bridge               local
bc25f39870f7        host                host                 local
758346aa0c1a        none                null                 local
ubuntu@k8s-instance-6:~$ docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "c8fdc79ccbfb36551c7c8446eb9913637e0b5dda928e436b5309
```

Exercise 2

```
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.101/0.118/0.150/0.019 ms
# ping 172.19.0.4
PING 172.19.0.4 (172.19.0.4) 56(84) bytes of data:
64 bytes from 172.19.0.4: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 172.19.0.4: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 172.19.0.4: icmp_seq=3 ttl=64 time=0.073 ms
^C
--- 172.19.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.073/0.083/0.105/0.015 ms
# hostname -I
172.19.0.4
# ping nginx1
PING nginx1 (172.19.0.3) 56(84) bytes of data:
64 bytes from nginx1.my_custom_network (172.19.0.3): icmp_seq=1 ttl=64 time=0.084 ms
64 bytes from nginx1.my_custom_network (172.19.0.3): icmp_seq=2 ttl=64 time=0.101 ms
64 bytes from nginx1.my_custom_network (172.19.0.3): icmp_seq=3 ttl=64 time=0.101 ms
64 bytes from nginx1.my_custom_network (172.19.0.3): icmp_seq=4 ttl=64 time=0.101 ms
64 bytes from nginx1.my_custom_network (172.19.0.3): icmp_seq=5 ttl=64 time=0.102 ms
64 bytes from nginx1.my_custom_network (172.19.0.3): icmp_seq=6 ttl=64 time=0.117 ms
^C
--- nginx1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5138ms
rtt min/avg/max/mdev = 0.084/0.101/0.117/0.009 ms
# --- 172.19.0.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.101/0.118/0.150/0.019 ms
# ping 172.19.0.4
PING 172.19.0.4 (172.19.0.4) 56(84) bytes of data:
64 bytes from 172.19.0.4: icmp_seq=1 ttl=64 time=0.105 ms
64 bytes from 172.19.0.4: icmp_seq=2 ttl=64 time=0.073 ms
64 bytes from 172.19.0.4: icmp_seq=3 ttl=64 time=0.073 ms
^C
--- 172.19.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2027ms
rtt min/avg/max/mdev = 0.073/0.083/0.105/0.015 ms
# hostname -I
172.19.0.3
# ping nginx3
PING nginx3 (172.19.0.4) 56(84) bytes of data:
64 bytes from nginx3.my_custom_network (172.19.0.4): icmp_seq=1 ttl=64 time=0.089 ms
64 bytes from nginx3.my_custom_network (172.19.0.4): icmp_seq=2 ttl=64 time=0.103 ms
64 bytes from nginx3.my_custom_network (172.19.0.4): icmp_seq=3 ttl=64 time=0.103 ms
64 bytes from nginx3.my_custom_network (172.19.0.4): icmp_seq=4 ttl=64 time=0.103 ms
64 bytes from nginx3.my_custom_network (172.19.0.4): icmp_seq=5 ttl=64 time=0.103 ms
64 bytes from nginx3.my_custom_network (172.19.0.4): icmp_seq=6 ttl=64 time=0.103 ms
^C
--- nginx3 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5118ms
rtt min/avg/max/mdev = 0.089/0.097/0.103/0.006 ms
root@363c1144f81d:/# hostname -I
172.19.0.3
root@363c1144f81d:/# ping 172.19.0.4
PING 172.19.0.4 (172.19.0.4) 56(84) bytes of data:
64 bytes from 172.19.0.4: icmp_seq=1 ttl=64 time=0.089 ms
64 bytes from 172.19.0.4: icmp_seq=2 ttl=64 time=0.103 ms
64 bytes from 172.19.0.4: icmp_seq=3 ttl=64 time=0.100 ms
^C
--- 172.19.0.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2070ms
rtt min/avg/max/mdev = 0.089/0.097/0.103/0.006 ms
root@363c1144f81d:/#
```

1. Run two Nginx containers which have to be connected to that user-defined network created in Exercise 1.
2. Use `ping` within both containers to test communication each other by container name.

Exercise 3

1. Create a Docker volume: `mydata`.
2. Run a container using the volume.
3. Write a file inside `/data` from the container, then:
 1. Stop the container.
 2. Relaunch to verify persistence.

```
ubuntu@k8s-instance-6:~$ docker volume create my_data_volume
my_data_volume
ubuntu@k8s-instance-6:~$ docker run -dit --name vol_container -v my_data_volume:/data alpine
f0973978b47cd3ce09589c38684cb7f336d6773da8773f6784ff3edade685044
ubuntu@k8s-instance-6:~$ docker exec -it vol_container sh
/ # echo "Persistent data" > /data/hello.txt
/ # exit
ubuntu@k8s-instance-6:~$ docker stop vol_container
vol_container
ubuntu@k8s-instance-6:~$
```

```
ubuntu@k8s-instance-6:~$ docker run -dit --name vol_container -v my_data_volume:/data alpine
f0973978b47cd3ce09589c38684cb7f336d6773da8773f6784ff3edade685044
ubuntu@k8s-instance-6:~$ docker exec -it vol_container sh
/ # echo "Persistent data" > /data/hello.txt
/ # exit
ubuntu@k8s-instance-6:~$ docker stop vol_container
vol_container
ubuntu@k8s-instance-6:~$ docker start vol_container
vol_container
ubuntu@k8s-instance-6:~$ docker exec -it vol_container sh
/ # docker exec -it vol_container sh
sh: docker: not found
/ # cat /data/hello.txt
Persistent data
/ #
```

Exercise 4

1. Create a directory on your host.
2. Run a container with a bind mount.
3. Create a file in `/mnt` from the container and check the host.

```
ubuntu@k8s-instance-6:~$ mkdir -p ~/docker_bind_test
ubuntu@k8s-instance-6:~$ docker run -dit --name bind_container -v ~/docker_bind_test:/mnt alpine
83d773883eb54dc6a48762b55b85c61ef82897268f3e5f43644e5977945c452b
ubuntu@k8s-instance-6:~$ docker exec -it bind_container sh
/ # echo "Hello from container" > /mnt/testfile.txt
/ # exit
ubuntu@k8s-instance-6:~$ cat ~/docker_bind_test/testfile.txt
Hello from container
ubuntu@k8s-instance-6:~$
```

Exercise 5

1. Create a file in a named volume.
2. Create a file using a bind mount.
3. Observe where data is stored on the host with `docker volume inspect` and `ls`

```
ubuntu@k8s-instance-6:~$  
ubuntu@k8s-instance-6:~$ docker volume create vol_demo_01  
vol_demo_01  
ubuntu@k8s-instance-6:~$ docker run -dit --name vol_container_01 -v vol_demo_01:/app_data alpine  
68b90f454a2f132bf4c0775b37ba4cf02d4333b1abd80e1646875160cd22f0fa  
ubuntu@k8s-instance-6:~$ docker exec -it vol_container_01 sh -c "echo 'Contenido del volumen' > /app_data/archivo_vol.txt"  
ubuntu@k8s-instance-6:~$ mkdir -p ~/bind_demo_01  
ubuntu@k8s-instance-6:~$ docker run -dit --name bind_container_01 -v ~/bind_demo_01:/mnt alpine  
f9206517af8d92323da3ccc0103a8011fdf49131ca5501cbdc4659fdde47a31a  
ubuntu@k8s-instance-6:~$ docker exec -it bind_container_01 sh -c "echo 'Contenido del bind mount' > /mnt/archivo_bind.txt"  
ubuntu@k8s-instance-6:~$
```

```
ubuntu@k8s-instance-6:~$  
ubuntu@k8s-instance-6:~$ docker volume inspect vol_demo_01  
[  
  {  
    "CreatedAt": "2025-04-30T19:40:47-04:00",  
    "Driver": "local",  
    "Labels": null,  
    "Mountpoint": "/var/lib/docker/volumes/vol_demo_01/_data",  
    "Name": "vol_demo_01",  
    "Options": null,  
    "Scope": "local"  
  }  
]  
ubuntu@k8s-instance-6:~$ sudo ls /var/lib/docker/volumes/vol_demo_01/_data  
archivo_vol.txt  
ubuntu@k8s-instance-6:~$ ls ~/bind_demo_01  
archivo_bind.txt  
ubuntu@k8s-instance-6:~$ cat ~/bind_demo_01/archivo_bind.txt  
Contenido del bind mount  
ubuntu@k8s-instance-6:~$
```

Exercise 6

1. Run an Ubuntu container with the necessary options to enable Docker in Docker (DinD).
2. Exec into the container and run `docker version`

```
ubuntu@k8s-instance-6:~$ docker run -dit \
> --name dind_ubuntu \
> -v /var/run/docker.sock:/var/run/docker.sock \
> ubuntu
d384a031a75273d04b9e35d8f866ec8ed4bb0243e1eef010bbdcdeb1e57d7dcd
ubuntu@k8s-instance-6:~$ docker exec -it dind_ubuntu bash
root@d384a031a752:/# docker exec -it dind_ubuntu bash
bash: docker: command not found
root@d384a031a752:/# apt update
Get:1 http://archive.ubuntu.com/ubuntu noble InRelease [256 kB]
```

```
Processing triggers for ca-certificates (20240203) ...
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
root@d384a031a752:/# docker version
Client:
 Version:           26.1.3
 API version:       1.45
 Go version:        go1.22.2
 Git commit:        26.1.3-0ubuntu1~24.04.1
 Built:             Mon Oct 14 14:29:26 2024
 OS/Arch:           linux/amd64
 Context:           default

Server: Docker Engine - Community
 Engine:
  Version:           28.1.1
  API version:       1.49 (minimum version 1.24)
  Go version:        go1.23.8
  Git commit:        01f442b
  Built:             Fri Apr 18 09:52:14 2025
  OS/Arch:           linux/amd64
  Experimental:      false
 containerd:
  Version:           1.7.27
  GitCommit:        05044ec0a9a75232cad458027ca83437aae3f4da
 runc:
  Version:           1.2.5
  GitCommit:        v1.2.5-0-g59923ef
 docker-init:
  Version:           0.19.0
  GitCommit:        de40ad0
root@d384a031a752:/#
```

Exercise 7

1. Run a container with memory and CPU limits:

1. Memory = 256m

2. CPU = 0.5

2. Check resource usage stats .

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIPS
69d040c943a5	limited-container	0.00%	500KiB / 256MiB	0.19%	446B / 126B	0B / 0B	1
d384a031a752	dind_ubuntu	0.00%	30.61MiB / 7.749GiB	0.39%	138MB / 2.85MB	0B / 625MB	2
f9206517af8d	bind_container_01	0.00%	508KiB / 7.749GiB	0.01%	740B / 126B	0B / 4.1kB	1
68b90f454a2f	vol_container_01	0.00%	516KiB / 7.749GiB	0.01%	866B / 126B	0B / 4.1kB	1
eee46658f669	my_container	0.00%	508KiB / 7.749GiB	0.01%	992B / 126B	0B / 4.1kB	1
83d773883eb5	bind_container	0.00%	512KiB / 7.749GiB	0.01%	1.12kB / 126B	0B / 12.3kB	1
f0973978b47c	vol_container	0.00%	508KiB / 7.749GiB	0.01%	1.24kB / 126B	0B / 0B	1
80de8b52e78d	nginx3	0.00%	5.738MiB / 7.749GiB	0.07%	9.85MB / 261kB	0B / 31.4MB	5
363c1144f81d	nginx1	0.00%	6.41MiB / 7.749GiB	0.08%	9.85MB / 227kB	0B / 31.4MB	6
8d00e73420e5	my_test_container	0.00%	504KiB / 7.749GiB	0.01%	1.35kB / 126B	0B / 0B	1

4. Check disk usage (docker system).

```
ubuntu@k8s-instance-6:~$ docker system df
TYPE                TOTAL        ACTIVE        SIZE          RECLAIMABLE
Images              4             4             647.1MB       0B (0%)
Containers          18           15           551.4MB       56.85MB (10%)
Local Volumes       4             4            133.8MB       0B (0%)
Build Cache         0             0              0B           0B
```

```
ubuntu@k8s-instance-6:~$ docker run -dit \
> --name limited-nginx \
> --memory=256m \
> --cpus=0.5 \
> -p 8080:80 \
> nginx
3d6c5162cfd60af5b2c9b7345b0ca6dbf2b4a360b6d516154146dfbdfbe2e6e4
ubuntu@k8s-instance-6:~$ docker stats limited-nginx
```

Exercise 8

1. Run a container with policy `--restart on -failure` .

2. Kill the container and observe how it restarts


```
ubuntu@k8s-instance-6:~$ docker run -dit \
> --name failing-container \
> --restart=on-failure \
> alpine sh -c "sleep 5 && exit 1"
8a557e4cc59a0ace16c06476a0ccc8c211fc8ba3ce210fde6e530eb0ce4be1108
ubuntu@k8s-instance-6:~$ docker kill failing-container
failing-container
ubuntu@k8s-instance-6:~$ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
8a557e4cc59a	alpine	"sh -c 'sleep 5 && e..."	39 seconds ago	Exited (137) 7 seconds ago
3d6c5162cfd6	nginx	"/docker-entrypoint..."	About a minute ago	Up About a minute
69d040c943a5	alpine	"sh"	4 minutes ago	Up 4 minutes

3. Try with the policy `--restart=unless-stopped`

```
ubuntu@k8s-instance-6:~$ docker run -dit \
> --name persistent-nginx \
> --restart=unless-stopped \
> nginx
d12d85afdc630eaaef9c698a00c444b67990ab8497268dc314ea740aa6ed7142
ubuntu@k8s-instance-6:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
d12d85afdc63	nginx	"/docker-entrypoint..."	8 seconds ago	Up

4. Reboot the system and see what happens.

```
e99d6647747f nginx "/docker-entrypoint..." 5 days ago Up 5 days 80/tcp webserver4
c16aebbd6bc6 nginx "/docker-entrypoint..." 5 days ago Up 5 days 80/tcp webserver
7d9cf8c9c726 nginx "/docker-entrypoint..." 6 days ago Up 6 days 0.0.0.0:80->80/tcp, [::]:80->80/tcp docker-nginx
786b9961c76a nginx "/docker-entrypoint..." 6 days ago Up 6 days 80/tcp nginx2
ubuntu@k8s-instance-6:~$ sudo reboot
Broadcast message from root@k8s-instance-6 on pts/1 (Thu 2025-05-01 13:27:35 -04):
The system will reboot now!

ubuntu@k8s-instance-6:~$ Connection to 10.26.32.177 closed by remote host.
Connection to 10.26.32.177 closed.
noemi@NGUZMANO-DH01:~$ ssh -i k8s-instance-6 ubuntu@10.26.32.177
ubuntu@k8s-instance-6:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
d12d85afdc63	nginx	"/docker-entrypoint..."	3 minutes ago	Up About a minute	80/tcp	persistent-nginx
8a557e4cc59a	alpine	"sh -c 'sleep 5 && e..."	5 minutes ago	Restarting (1) 15 seconds ago		failing-container

Exercise 9

1. Create a network `dbnet`.

2. Create a volume `dbdata`.
3. Run a MariaDB container with the following requirements:
 1. Attached to volume `dbdata`.
 2. Attached to network `dbnet`.
 3. Do NOT expose ANY port.

```

8a557e4cc59a alpine sh -c sleep 5 && e... 5 minutes ago Restart
ubuntu@k8s-instance-6:~$ docker network create my_custom_net
cba4e2b6f6375656f0295071252f0ac188fe333d4a326d5104c7309ed10ff7d1
ubuntu@k8s-instance-6:~$ docker volume create mariadb_data
mariadb_data
ubuntu@k8s-instance-6:~$ docker run -dit \
> --name my_mariadb \
> --network my_custom_net \
> --mount source=mariadb_data,target=/var/lib/mysql \
> -e MARIADB_ROOT_PASSWORD=my-secret-pw \
> mariadb
Unable to find image 'mariadb:latest' locally
latest: Pulling from library/mariadb
2726e237d1a3: Already exists
0b86886c6aaa: Pull complete

```

```

41321948940c69a320f7077d6d1a3866358ee20871d2600f54040e5fe513dc
ubuntu@k8s-instance-6:~$ docker network inspect my_custom_net
[
  {
    "Name": "my_custom_net",
    "Id": "cba4e2b6f6375656f0295071252f0ac188fe333d4a326d5104c7309ed10ff7d1",
    "Created": "2025-05-01T13:35:06.955909885-04:00",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv4": true,
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",

```

```

ubuntu@k8s-instance-6:~$ docker volume inspect mariadb_data
[
  {
    "CreatedAt": "2025-05-01T13:35:14-04:00",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/mariadb_data/_data",
    "Name": "mariadb_data",
    "Options": null,
    "Scope": "local"
  }
]
ubuntu@k8s-instance-6:~$ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
41321948940c   mariadb   "docker-entrypoint.s..." 51 seconds ago Up 50 seconds 3306/tcp                 my_mariadb
d12d85afdc63   nginx     "/docker-entrypoint..." 10 minutes ago Up 7 minutes    80/tcp                  persistent-nginx
8a557e4cc59a   alpine    "sh -c 'sleep 5 && e..." 12 minutes ago Restarting (1) 55 seconds ago failing-container

```

Exercise 10

1. Run a PHPMyAdmin container with the following requirements:
 1. Attached to network `dbnet` (created in Exercise 9).

2. Use a bind mount to persist the web app configuration.

```
ubuntu@k8s-instance-6:~$ mkdir -p ~/phpmyadmin_config
ubuntu@k8s-instance-6:~$ docker run -dit \
> --name my_phpmyadmin \
> --network my_custom_net \
> --mount type=bind,source=$HOME/phpmyadmin_config,target=/etc/phpmyadmin \
> -e PMA_HOST=my_mariadb \
> -e PMA_PORT=3306 \
> -p 8081:80 \
> phpmyadmin
Unable to find image 'phpmyadmin:latest' locally
latest: Pulling from library/phpmyadmin
254e724d7786: Pull complete
```

3. Linked to the previous MariaDB container (created in Exercise 9)
4. Open a browser to display the PHPMYAdmin Login Form.

```
ubuntu@k8s-instance-6:~$ ping http://localhost:8081
ping: http://localhost:8081: Name or service not known
ubuntu@k8s-instance-6:~$ curl http://127.0.0.1:8081
<!doctype html>
<html lang="en" dir="ltr">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="referrer" content="same-origin">
  <meta name="robots" content="noindex,nofollow,notranslate">
  <meta name="google" content="notranslate">
  <style id="cfs-style">html{display: none;}</style>
  <link rel="icon" href="favicon.ico" type="image/x-icon">
  <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
  <link rel="stylesheet" type="text/css" href="/themes/pmahomme/jquery/jquery-ui.css">
  <link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/lib/codemirror.css">
  <link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/addon/hint/show-hi">
  <link rel="stylesheet" type="text/css" href="/js/vendor/codemirror/addon/lint/lint.cs">
  <link rel="stylesheet" type="text/css" href="/themes/pmahomme/css/theme.css?v=5.2.2">
  <title>phpMyAdmin</title>
  <script data-cfasync="false" type="text/javascript" src="/js/vendor/jquery/jquery.m">
  <script data-cfasync="false" type="text/javascript" src="/js/vendor/jquery/jquery-mig">
  <script data-cfasync="false" type="text/javascript" src="/js/vendor/sprintf.js?v=5.2.>
  <script data-cfasync="false" type="text/javascript" src="/js/dist/ajax.js?v=5.2.2"></>
  <script data-cfasync="false" type="text/javascript" src="/js/dist/keyhandler.js?v=5.2">
  <script data-cfasync="false" type="text/javascript" src="/js/vendor/jquery/jquery-ui.>
  <script data-cfasync="false" type="text/javascript" src="/js/dist/name-conflict-fixes">
  <script data-cfasync="false" type="text/javascript" src="/js/vendor/bootstrap/bootstr
```

5. Login with the DB credentials.

10.26.32.177:8081

jala  AWS  cloud  JIRA  JALA  diagramas  team  VC 185  pyth




Welcome to phpMyAdmin

Language

English



Log in 

Username:

root

Password:

.....

Log in

6.