

## Docker Inspect hands-on

- Create an Image (InsImage) and Run a Container (InsContainer) using the image.

```
labuser@ip-172-31-9-138:~/test$ docker build -t insimage .
Sending build context to Docker daemon 2.048kB
Step 1/1 : FROM nginx: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
--> a99a39d070bf
Successfully built a99a39d070bf
Successfully tagged insimage:latest
labuser@ip-172-31-9-138:~/test$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
insimage             latest             a99a39d070bf        3 weeks ago        142MB
nginx                latest             a99a39d070bf        3 weeks ago        142MB
labuser@ip-172-31-9-138:~/test$ cat Dockerfile
FROM nginx:latest
labuser@ip-172-31-9-138:~/test$

labuser@ip-172-31-9-138:~/test$ docker run -p 8080:8080 --name=inscontainer insimage
docker: Error response from daemon: driver failed programming external connectivity on endpoint inscontainer (cf4941a060d76392eb93e703f3c023d1c8c7074e9e14ee5fc336f04307048552): Error starting userland proxy: listen tcp 0.0.0.0:8080: bind: address already in use.
ERRO[0000] error waiting for container: context canceled
labuser@ip-172-31-9-138:~/test$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
1e98a16d0f0d       insimage           "/docker-entrypoint..." 13 seconds ago     Created                                inscontainer
labuser@ip-172-31-9-138:~/test$ docker run -p 8081:8080 --name=inscontainer1 insimage
docker: Error response from daemon: driver failed programming external connectivity on endpoint inscontainer1 (d6e4726e1857b8b1d639f6f99c91b4defdb8e5bfc8506deb6857e0a7fb94a4e): Error starting userland proxy: listen tcp 0.0.0.0:8081: bind: address already in use.
ERRO[0000] error waiting for container: context canceled
labuser@ip-172-31-9-138:~/test$ docker ps --a
unknown shorthand flag: '-' in --a
See 'docker ps --help'.
labuser@ip-172-31-9-138:~/test$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
6c4fa987f4a8       insimage           "/docker-entrypoint..." 10 seconds ago     Created                                inscontainer1
1e98a16d0f0d       insimage           "/docker-entrypoint..." About a minute ago   Created                                inscontainer
labuser@ip-172-31-9-138:~/test$
```

- Create a new bridge (InsNetwork) and connect the container to this bridge.

```
labuser@ip-172-31-9-138:~/test$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
6c4fa987f4a8       insimage           "/docker-entrypoint..." 10 seconds ago     Created                                inscontainer1
1e98a16d0f0d       insimage           "/docker-entrypoint..." About a minute ago   Created                                inscontainer
labuser@ip-172-31-9-138:~/test$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
2935df103bac        bridge              bridge              local
77fd590fce13        host                host                local
72637949e8c5        none                null                local
labuser@ip-172-31-9-138:~/test$ docker network create insnetwork
ea51cdd396aaebd4e1585alc911ffd47e16161fa1ec1356669de1e0c0a72a406
labuser@ip-172-31-9-138:~/test$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
2935df103bac        bridge              bridge              local
77fd590fce13        host                host                local
ea51cdd396aa        insnetwork          bridge              local
72637949e8c5        none                null                local
labuser@ip-172-31-9-138:~/test$ docker network connect insnetwork inscontainer
```

```
est$ docker container inspect inscontainer
```

Using this command we can check whether container is connected to network .

```

"Networks": {
  "insnetwork": {
    "IPAMConfig": {},
    "Links": null,
    "Aliases": [
      "1e98a16d0f0d"
    ],
    "NetworkID": "",
    "EndpointID": "",
    "Gateway": "",
    "IPAddress": "",
    "IPPrefixLen": 0,
    "IPv6Gateway": "",
    "GlobalIPv6Address": "",
    "GlobalIPv6PrefixLen": 0,
    "MacAddress": "",
    "DriverOpts": {}
  }
}

```

We can disconnect container from old network .

```
it$ docker network disconnect bridge inscontainer
```

- Create a new storage volume (InsVolume) and mount this on InsContainer.

```

labuser@ip-172-31-9-138:~/test$ docker volume create insvolume
insvolume
labuser@ip-172-31-9-138:~/test$

```

To mount :

```
:$ docker run -v /insvolume --name=inscontainer insimage
```

Check whether volume mount to container .

```
st$ docker container inspect inscontainer
```

```

{
  "Image": "insimage",
  "Volumes": {
    "/insvolume": {}
  },
  "Networks": {

```

- Now Run Docker inspect command on Image, Container, network and volume created.

Try Networking in Docker

Create a new bridge 'bridge\_sample'.

```
$ docker network ls
NETWORK ID        NAME        DRIVER        SCOPE
80e29e9c4327      bridge      bridge        local
27ada7c126c7      host        host          local
e2ab2fce2fdd      none        null          local
$ docker network create bridge_sample
14abe3a7106701d8c9079f3811c96cb8eae0991a7c87a86dec8f6603042e1253
$ docker network ls
NETWORK ID        NAME            DRIVER        SCOPE
80e29e9c4327      bridge          bridge        local
14abe3a71067      bridge_sample   bridge        local
27ada7c126c7      host            host          local
e2ab2fce2fdd      none            null          local
```

Run a couple of images (Cont1 and Cont2) and connect these to the new bridge created. Now try to ping from cont1 to cont2 to verify connectivity.

Create 2 containers .

```
$ docker run -d --name=cont1 training/postgres
Unable to find image 'training/postgres:latest' locally
latest: Pulling from training/postgres
Image docker.io/training/postgres:latest uses outdated schema1 manifest format. Please
re information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
a3ed95caeb02: Pull complete
6e71c809542e: Pull complete
2978d9af87ba: Pull complete
e1bca35b062f: Pull complete
500b6decf741: Pull complete
74b14ef2151f: Pull complete
7afd5ed3826e: Pull complete
3c69bb244f5e: Pull complete

23a27375ed4805cb08b9d37af4911f61d9d18a1851dccc0a2d34678e73e29c349
$ docker run -d --name=cont2 training/webapp python app.py
Unable to find image 'training/webapp:latest' locally
latest: Pulling from training/webapp
Image docker.io/training/webapp:latest uses outdated schema1 manifest
information at https://docs.docker.com/registry/spec/deprecated-schema-v1/
e190868d63f8: Pull complete
909cd34c6fd7: Pull complete
0b9bfabab7c1: Pull complete
a3ed95caeb02: Pull complete
10bbbc0fc0ff: Download complete
fca59b508e9f: Download complete
e7ae2541b15b: Download complete
```

Check network of these 2 containers .

```

"MacAddress": "02:42:ac:12:00:02",
"Networks": {
  "bridge": {
    "IPAMConfig": null,
    "Links": null,
    "Aliases": null,
    "NetworkID": "80e29e9c43278de",
    "EndpointID": "e98c1a35f515ae",
    "Gateway": "172.18.0.1",

```

Both Connected to default bridge network .

```

]
$ docker exec -it cont1 bash
root@23a27375edf8:/# ping 172.18.0.3
PING 172.18.0.3 (172.18.0.3) 56(84) bytes of data.
64 bytes from 172.18.0.3: icmp_seq=1 ttl=64 time=0.166 ms
64 bytes from 172.18.0.3: icmp_seq=2 ttl=64 time=0.112 ms
64 bytes from 172.18.0.3: icmp_seq=3 ttl=64 time=0.109 ms
64 bytes from 172.18.0.3: icmp_seq=4 ttl=64 time=0.111 ms

```

Ping is working since both container are connected to default network .

Connect 1 container to above created network and disconnect from default network . Now try ping .

```

$ docker network disconnect bridge cont1
$ docker network connect bridge_sample cont1

```

```

$ docker exec -it cont1 bash
root@23a27375edf8:/# ping 172.18.0.3
PING 172.18.0.3 (172.18.0.3) 56(84) bytes of data.

```

Ping is not working now .

When we change network ip address also changes .

```

},
"4a2664dbd113394975f2275fc2a1b427142ab6b6f1162e2ccf18e75ec03b432f": {
  "Name": "cont2",
  "EndpointID": "0567526cb9c44714aa9734456ae5456af272c0e350642c1358d5590b8d",
  "MacAddress": "02:42:ac:11:00:03",
  "IPv4Address": "172.17.0.3/16",
  "IPv6Address": ""
}

```

```

root@23a27375edf8:/# ping 172.17.0.3
PING 172.17.0.3 (172.17.0.3) 56(84) bytes of data.
64 bytes from 172.17.0.3: icmp_seq=1 ttl=64 time=0.136 ms
64 bytes from 172.17.0.3: icmp_seq=2 ttl=64 time=0.108 ms
64 bytes from 172.17.0.3: icmp_seq=3 ttl=64 time=0.105 ms
64 bytes from 172.17.0.3: icmp_seq=4 ttl=64 time=0.098 ms
^C
--- 172.17.0.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.098/0.111/0.136/0.019 ms
root@23a27375edf8:/# ping 172.18.0.3

```

Ping is working now when we connect both container to same network .

Once done, stop containers and then remove network, containers, and images using docker commands.

Create a new storage volume 'volume1' and mount this on to a container 'Web'. Create another volume 'volume2'.

```

$ docker volume create --name=volume1
volume1
$ docker volume inspect volume1
[
  {
    "CreatedAt": "2023-02-03T15:41:34Z",
    "Driver": "local",
    "Labels": {},
    "Mountpoint": "/var/lib/docker/volumes/volume1/_

```

```

$ docker run -v /volume1 --name=web nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty
/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: info: Getting th

```

Stop the container and remove volume mounted. Run command to remove unused volume ('volume2').

```

5432/tcp cont1
$ docker stop web
web
$ docker volume rm volume1
volume1
$ docker ps -a

```

```

$ docker volume create volume2
volume2
$ docker volume ls
DRIVER      VOLUME NAME
local       3294f5abf74b5934b9dd091be140bd854b9bf28842e4090cd52b7c1c4f370a84
local       volume2
$ docker volume prune
WARNING! This will remove all local volumes not used by at least one container.

```

## Kickoffs - Docker Challenge

In this challenge, you are going to perform the following tasks:

1. Create a Dockerfile in the path "~/Desktop/Project/kickoffs-itis-docker".
2. Add a build step to dockerize the given Flask application with python alpine3.7 as its base image named "flask\_app\_image" and deploy it on a container named "flask\_app\_container" and expose in port 5005.
3. Check the application deployment by hitting the URL "http://localhost:5005/"

```

labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$ cat Dockerfile
FROM python:alpine3.7
RUN pip install flask
WORKDIR /myapp
COPY ./app.py /myapp/
CMD ["python", "/myapp/app.py"]
labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$

```

```

labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$ cat app.py
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return '<h2>Flask application has been successfully deployed using Docker.</h2>'

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5006)
labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$

```

```
$ docker build -t flask_app_image .
```

```

labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
flask_app_image      latest       75a771030f19     5 minutes ago   92.2MB
python               alpine3.7   00be2573e9f7     4 years ago     81.3MB
labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$

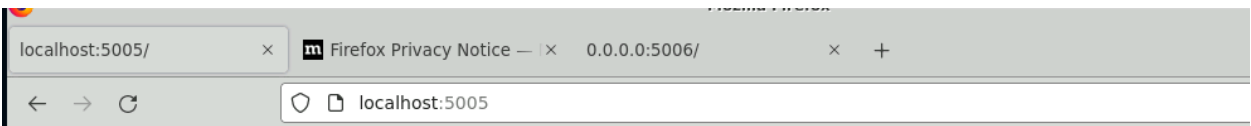
```

```

labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$ docker run -d -p 5005:5006 --name=flask_app_container flask_app_image:late
st

```

```
python3 alpine3.7 00be23/3e9f7 4 years ago 01.3MB
labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
4a590e49295d        flask_app_image:latest  "python /myapp/app.py"  2 minutes ago       Up 2 minutes        0.0.0.0:5005->5006/tcp
p_flask_app_container
4792a37a132b        00be2573e9f7       "/bin/sh -c 'pip ins..."  28 minutes ago       Exited (1) 28 minutes ago
pensive_black
c3b8fec4c081        00be2573e9f7       "/bin/sh -c 'pip ins..."  30 minutes ago       Exited (1) 30 minutes ago
gallant_sutherland
171779df0072        00be2573e9f7       "/bin/sh -c 'pip ins..."  32 minutes ago       Exited (1) 32 minutes ago
gallant_cannon
labuser@ip-172-31-28-203:~/Desktop/Project/kickoffs-itis-docker$
```



**Flask application has been successfully deployed using Docker.**

Which is the Docker command to build a docker image using a docker file in the current directory?

docker build-image

docker build image

docker build .

Which of the following is the correct way to name a docker file?

DOCKERFILE

DokcerFile

dockerfile

Dockerfile

Docker images have a different state and change with time.

False

True

Which of the following commands will start a container based on Ubuntu 14.04 base image?

`docker run ubuntu -v 14.04`

`docker run Ubuntu:14.04`

`docker run ubuntu -version 14.04`

`docker run ubuntu:14.04`

Which is the Docker command to show the version information of the docker components?

`docker v`

`docker detail`

`docker version`

`apt-get update`

What is the command to stop a container?

`docker halt <Container>`

`Docker-stop <Container>`

`docker-stop <Container>`

`docker stop <Container>`



What is the syntax to mount a '/project/data' directory from the docker host into a directory '/data' in the container?

docker run -v from /data:/project/data

docker run -volumes from /data:/project/data

docker run -v /project/data:/data

docker run -v /data:/project/data

The -v flag is used to mount a host folder, and it consists of two fields separated by a colon. The first part is the path in the host machine. The second part is the path in the container. The --name flag (optional) is used to give the container a specific name.

\_\_\_\_\_ is a tool for defining and running multi-container Docker applications.

docker file

docker hub

docker hub

docker compose

Which of the following is true?

docker ps shows all containers by default

docker ps shows all running containers by default

Which of the following commands will install the Docker engine (Community edition) in a Centos server?

apt-get install docker.i.o

yum install docker-ce

apt-get install docker

yum install docker

Which docker command is used to attach to a running container?

docker telnet<container>

docker login <container>

docker ssh <container>

docker attach <container>

Virtual machines are a form of \_\_\_\_\_ type of virtualization.

Hypervisor

Mainframe

Container

VLAN

Using docker compose, we cannot bring up more than one service.

False

True

Which of the following is a valid docker inspect command to display the log path?

`docker inspect -format='{{.Path}}' tomcatContainer`

None of the options

`docker inspect --format='{{.LogPath}}' tomcatContainer`

All the options

`docker inspect -format='{{.Log}}' tomcatContainer`

Docker inspect command is used to \_\_\_\_\_.

troubleshoot errors

return low-level information on docker daemon

display log information

return low-level information on docker objects

Container networking model explains the \_\_\_\_\_ architecture.

Security

Networking

All the options

Storage

What is the Docker command to disconnect a container from the bridge network?

`docker disconnect bridge <container>`

None of the options

`docker network disconnect <container>`

`docker network disconnect bridge <container>`

When a container is started, Docker establishes a bridge between \_\_\_\_\_.

image and container

host machine and container

host machine and image

None of the options

The default Inet address for Docker is \_\_\_\_\_.

172.17.42.1

172.17.42.2

172.17.42.3

There is no default address

The virtual bridge that the Docker establishes is called \_\_\_\_\_.

docker network

docker0

docker bridge

docker1

Configuration changes should be done in \_\_\_\_\_ file to modify the storage driver used.

driver.json

file.json

daemon.json

storage.json

Which Docker command is used to know the current storage driver configured?

docker info

docker bridge

docker version

docker storage

None of the options

Which of the following is correct?

Multiple container layers can be built on top of an image layer

One Container layer can be built on top of an image layer

The container layer cannot be built on top of the image layer

Which of the following is the Docker command to remove unused volumes?

All the options

None of the options

docker volume prune

docker volume delete

docker volume remove

Docker volumes can be backed up/restored/migrated.

True

False

Which of the following are storage drivers supported by Docker for Ubuntu systems?

None of the options

zfs

devicemapper

All the options

overlay2

aufs