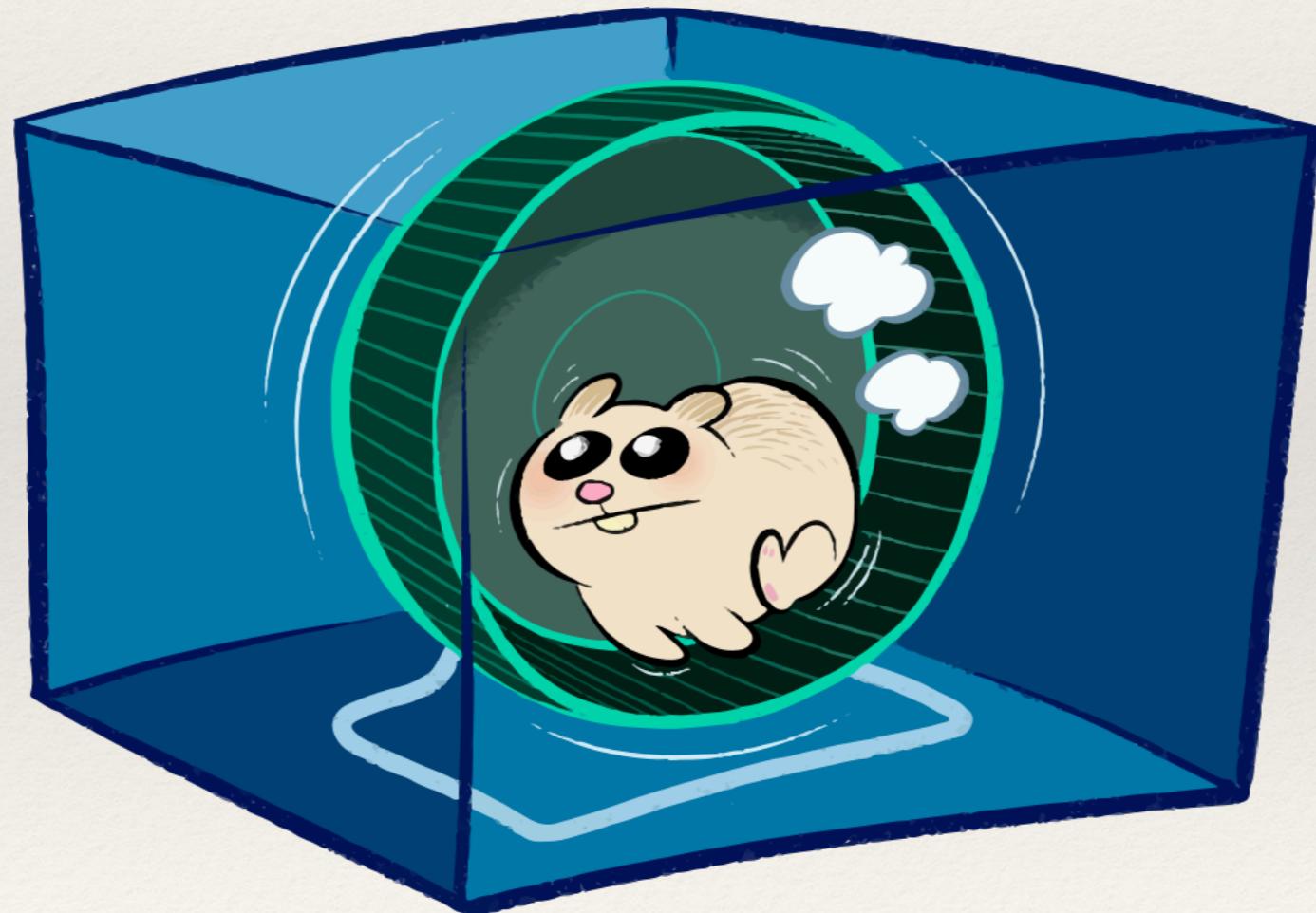


Docker Quiz

Ganesh & Hari
ganesh@codeops.tech
hari@codeops.tech

1. Question

What is the name of the container runtime used by Docker? (hint: it is from Open Container Initiative)



1. Answer

RunC (see <https://runc.io/>)

```
$ docker info | grep -i runc
Runtimes: runc
Default Runtime: runc
```



2. Question

What happens when you execute this on the command-line?

```
docker run debian /bin/sh
```

- A. A prompt from the shell of created container will be thrown to you
- B. A container is created and then exited immediately
- C. A container is created and executes in the detached mode; you can attach to it later using the container id
- D. Docker CLI issues the error: Error response from daemon: No command specified.

2. Answer

When you execute this command,

docker run debian /bin/sh

A container is created and then exited immediately.

```
$ docker ps -a
CONTAINER ID        IMAGE       COMMAND
CREATED             STATUS      NAMES
4c12998fd392      debian     "/bin/bash"
6 seconds ago      Exited (0)  5 seconds ago
                                         sick_panini
```

3. Question

What happens when you execute this on the command-line with -p option?

```
$ docker run -p 80-87:80 --name nginx3 -d nginx
```

- A. The port 80 in the container is mapped to ports 80 and 87 in the host machine
- B. The port 80 in the hostmachine is mapped to ports 80 and 87 in the container
- C. The port 80 in the container is mapped to a port in the range 80 to 87 in the host machine
- D. Docker CLI will issue the error: “Invalid syntax: Using hyphen is not allowed with -P command”

3. Answer

The option “-p 80-87:80” maps the port 80 in the container to a port (based on the availability) in the range 80 to 87 in the host machine; this is called “port range”

Example:

```
docker port nginx3  
80/tcp -> 0.0.0.0:82
```

4. Question

Which command do you use “to find layers and their sizes” in an image using Docker CLI?

- A. Use “docker images -layers <<imageid>>” command
- B. Use “docker layers <<imageid>>” command
- C. Use “docker history <<imageid>>” command
- D. There is no way you can find layers and their sizes using Docker CLI
- you need to use external tools

4. Answer

To find layers and their sizes in an image using Docker CLI, use“docker history <<imageid>> command.

IMAGE	CREATED	CREATED BY	SIZE
COMMENT			
106e303be3a4	2 weeks ago	/bin/sh -c #(nop) ENTRYPOINT ["/usr/bin/cadvi	0 B
<missing>	2 weeks ago	/bin/sh -c #(nop) EXPOSE 8080/tcp	0 B
<missing>	2 weeks ago	/bin/sh -c #(nop) ADD file:1bde294f31142b3dee	25.87 MB
<missing>	2 weeks ago	/bin/sh -c apk --no-cache add ca-certificates	17.13 MB
<missing>	2 weeks ago	/bin/sh -c #(nop) ENV GLIBC_VERSION=2.23-r3	0 B
<missing>	2 weeks ago	/bin/sh -c #(nop) MAINTAINER dengnan@google.c	0 B
<missing>	3 months ago	/bin/sh -c #(nop) ADD file:852e9d0cb9d906535a	4.799 MB

5. Question

Which command do you use “recreate the Dockerfile that was used to build that image” from a given image id/tag using Docker CLI?

- A. Use “docker images -dockerfile <<imageid>>” command
- B. Use “docker build -reverse <<imageid>>” command
- C. Use “docker history --no-trunc --out:<filename> <<imageid>>” command
- D. There is no way to recreate the Dockerfile that was used to build that image from a given image id/tag using Docker CLI

5. Answer

There is **NO** way to recreate the Dockerfile that was used to build that image from a given image id/tag using Docker CLI.

Think about Makefile: can you recreate the Makefile that was used to build that executable file? No.

However, you can see the commands used to create the layers in the image. Pass “—no-trunc” option to “docker history” command.

Example: “`docker history --no-trunc google/cadvisor`”

Try it now!

6. Question

You are creating a new container with this command:

```
docker run -d --name myubuntu ubuntu /bin/sh -c "while true; do echo current date and  
time is: $(date); sleep 10; done"
```

Which network is the “myubuntu” container attached to?

- A. Bridge network
- B. Overlay network
- C. Custom network
- D. Host network
- E. None (not connected to any network)

6. Answer

Bridge network. By default, a newly created container is attached to the bridge network (unless a different network is specified, for example, using the “—network” option with the docker run command).

```
$ docker network inspect bridge
[
  {
    "Name": "bridge",
    // ...
    "Containers": {
      "04579b88a74c981ae854261dffc7ab17328c28bb6fafec0f9c1e9431e77b3b27": {
        "Name": "myubuntu",
        "EndpointID": "8a0e7a2559eac35eb60a90e85554679de276bd1ba39ff3a4083301d08e9ee384",
        "MacAddress": "02:42:ac:11:00:03",
        "IPv4Address": "172.17.0.3/16",
        "IPv6Address": ""
      },
      // ...
      // ...
    }
  ]
]
```

7. Question

Which one of the following is a recommended Docker BEST PRACTICE?

- A. Prefer using docker commit or docker import instead of using docker build using Dockerfiles (for creating custom/new images)
- B. Use --link for linking containers explicitly on the same host instead of using default bridge network for inter container communication (ICC)
- C. Put ADD commands later in the Dockerfile because the source files or executables may change but the earlier layers will not change (to avoid “cache bursting”)
- D. Use explicit file/dir paths (e.g., -v /usr/mydir:/usr/mydir) instead of using named volumes created using “docker volume create” command

7. Answer

Avoid cache bursting and make your Dockerfile cache friendly. Example: Put ADD commands later in the Dockerfile - because the source files or executables may change but the earlier layers will not change

Other three are bad practices:

- ❖ Prefer using docker build using Dockerfiles (for creating custom/ new images) instead of using docker commit or docker import
- ❖ Using default bridge network for inter container communication (ICC) instead of --link for linking containers explicitly on the same host
- ❖ Use named volumes created using “docker volume create” command instead of explicit file/dir paths (e.g., -v /usr/mydir:/usr/mydir)

8. Question

When you run this command, what will be the PID of /bin/sh?

`docker run -it alpine /bin/sh`

- A. PID 1
- B. Same as the PID of the docker process in the host
- C. PID 0
- D. Don't know ;) Its randomly assigned by Docker

8. Answer

PID 1

```
docker run -it alpine /bin/sh
```

```
$ docker run -it alpine /bin/sh
/ # ps
 PID  USER      TIME  COMMAND
   1  root      0:00  /bin/sh
   7  root      0:00  ps
/ #
```

9. Question

What does this command do?

“docker run --privileged -d docker:dind”

- A. It runs “docker within docker”
- B. It runs the monitoring tool for docker
- C. It is equivalent to “docker exec” and attach to a running container
- D. It is for debugging docker containers

9. Answer

“`docker run --privileged -d docker:dind`”

“`docker:dind`” is the official “Docker in Docker base image”

Upcoming bootcamps

<https://www.townscript.com/e/modernarch> (5th Nov)

<https://www.townscript.com/e/solid> (19th Nov)

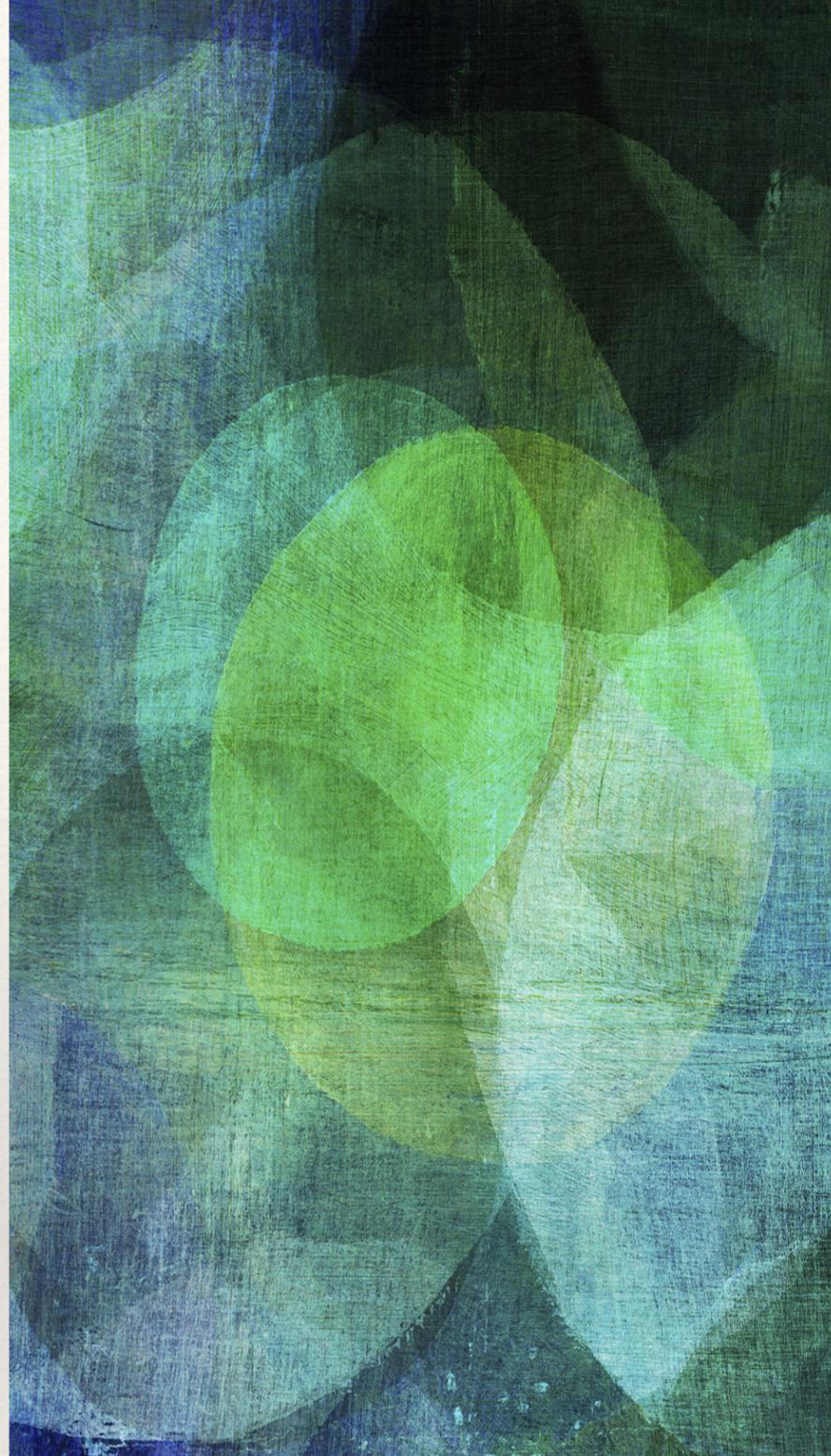
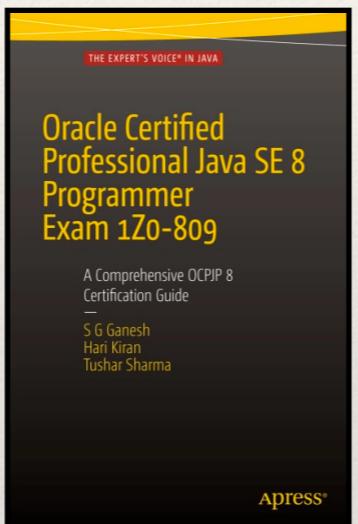
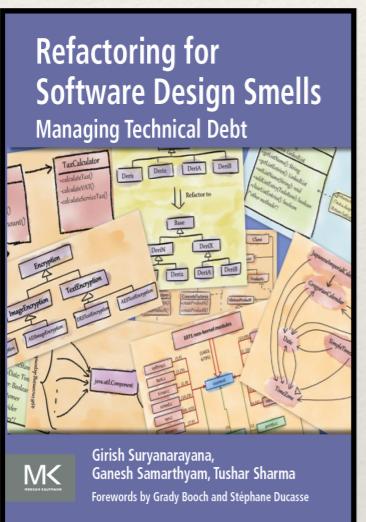
<https://www.townscript.com/e/internetofthings> (26th Nov)

<https://www.townscript.com/e/software-architecture> (10 Oct)

Meetups



- <http://www.meetup.com/Core-Java-Meetup-Bangalore/> (12th Nov, lastminute.com)
- <https://www.meetup.com/Mobile-App-Developers-Bangalore-Meetup/> (26th Nov, Work-In)
- <http://www.meetup.com/JavaScript-Meetup-Bangalore/> (19th Nov, HPE)
- <http://www.meetup.com/Container-Developers-Meetup-Bangalore/> (Dec 3, Microsoft)
- <http://www.meetup.com/CloudOps-Meetup-Bangalore/> (12th Nov, lastminute.com)



ganesh@codeops.tech

www.codeops.tech

+91 98801 64463

[@GSamarthyam](https://twitter.com/GSamarthyam)

slideshare.net/sgganesh

bit.ly/ganeshsg