Docker Questions with answers

b.angelard

September 2019

(*) Easy question
(**) Basic understanding of docker
(***) Good understanding of docker features
(****) Complicated to explain simply or complex feature

1 Closed questions and QCM

1. (*) Docker containers :
   - Docker containers are tied to a specific infrastructure
   - Docker containers run only on the OS you created them on
   - Docker containers are isolated processes
   - Docker containers from the same image share the same variables
   - How to check for docker Version / Docker compose versions
2. (***) Which of the following commands are real docker commands, and explain them
   - docker help
   - docker man
   - docker info
   - docker logs
   - docker ps
   - docker ls
   - docker exec
   - docker cat
   - docker start
   - docker run
3. (*) Will you lose your data, when a docker container exits ?

- Yes
- no

And if "Yes", which one

4. (**) When you delete/remove a container, Which of those propositions apply
   - Package installed through apt-get / pip
   - Files created in a non-mounted directory
   - Files created in a mounted directory
   - Running programs are killed
   - Running programs continue until you stop them even after docker rm
5. (*) How many containers can run per host
   - One
   - The amount of core you have on your machine
   - Twice the amount of core
   - As much as there is space on disk
6. (****) Which command allow you to show binded ports / exposed ports of all container
   - docker ps
   - docker ps -a
7. (****) Explain the following commands
   - docker image ls
   - docker create
   - docker start
   - docker run
   - docker ls
   - docker inspect
   - docker logs
   - docker stop
   - docker kill
   - docker rm
8. (****) Considering the following line from docker ps

CONTAINER_ID	IMAGE	NAMES
2b2dc59ebc39	karpathos_ssh	ben_container

Which commands are valid

- docker start ben_container
- docker start 2b2dc
- docker rm 2b2dc59ebc39
- docker build 2b2dc59ebc39
- docker run -it 2b2dc59ebc39 bash

2 Answer Closed questions

1. pass
2. pass
3. pass
4. pass
5. pass
6. pass
7. pass
8. pass
9. create — Create a container from an image.
   start — Start an existing container.
   run — Create a new container and start it.
   ls — List running containers.
   inspect — See lots of info about a container.
   logs — Print logs.
   stop — Gracefully stop running container.
   kill —Stop main process in container abruptly.
   rm— Delete a stopped container.
10. - Valid
    - Valid - As long as the id is identifiable by those 5 first characters it is fine
    - Valid
    - False - This is a container not an image
    - False - This is a container not an image. You have to use "docker exec"

3 Open Questions

1. (*) What is containerization?
2. (*) Can all application be containerized. Why ?
3. (*) Does docker instance share the kernel or have their own virtual space
4. (*) Can we deploy updates and upgrades on the fly (meaning just replacing an old docker image by a new one) in production.
5. (*) Is docker scalable? what does it mean ?
6. (**) What is its benefice for installations ?
7. (*) What is Docker?
8. (*) What is an image ? A container ?
9. (*) What is a Dockerfile?
10. (*) How do you build a dockerfile
11. (**) Can a docker container restart by itself in case of a bug/crash ? *If he says no : Ask him if it's by default or juste impossible → Expecting "–restart=True"*
12. (****) Explain docker swarm
13. (****) Fill the following file to create and copy a directory

```
FROM nvidia/cudagl:10.0-devel-ubuntu16.04 as dev
[...] \# Define the directory on docker
[...] \# Copy the current dir
```

14. (***) What is the difference between virtual machines and docker containers
15. (**) Can you explain the following dockerfile

```
FROM python:2.7-slim

WORKDIR /app

COPY . /app

RUN pip install --trusted-host pypi.python.org -r requirements.txt

EXPOSE 80

ENV NAME World

CMD ["python", "app.py"]
```

16. (***) How can you map the port 8080 from your machine to the port 4444 from docker

17. (*) The machine Zira expose ARTPLAN on the port 9000, how can you access it from your machine's internet explorer

18. (***) Can you explain the following docker-compose.yml

```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image details
    image: username/repo:tag
    deploy:
      replicas: 5
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
    networks:
      - webnet
networks:
  webnet:
```

4 Answers

Won't be given to trainee

1. Usually, in the software development process, code developed on one machine might not work perfectly fine on any other machine because of the dependencies. This problem was solved by the containerization concept. So basically, an application that is being developed and deployed is bundled and wrapped together with all its configuration files and dependencies. This bundle is called a container. Now when you wish to run the application on another system, the container is deployed which will give a bug-free environment as all the dependencies and libraries are wrapped together.

2. Yes, you have no restriction as long as your computer can launch it, so does docker

3. Shared

4. yes, that's the point

5. You can increase and automatically distribute container replicas

6. Docker is a containerization platform which packages your application and all its dependencies together in the form of containers so as to ensure that your application works seamlessly in any environment, which ensures your installation will be fine.

7. Docker is a containerization platform which packages your application and all its dependencies together in the form of containers so as to ensure that your application works seamlessly in any environment

8. **Docker containers** include the application and all of its dependencies. *It shares the kernel with other containers, running as isolated processes in user space on the host operating system* (Maybe too complicated for simple question). Docker containers run on any computer, on any infrastructure, and in any cloud. Docker containers are basically runtime instances of Docker images.

   Docker image is the source of Docker container. Docker images are used to create containers.

9. Docker can build images automatically by reading the instructions from a file called Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

10. docker build Dockerfile

11. It won't restart automatically by default but you can enforce it through –restart=True

12. Docker Swarm is native gathering for docker which helps you to a group of Docker hosts into a single and virtual docker host.

13. ```
    WORKDIR /app
    COPY . /app
    ```

14. A container runs natively on Linux and shares the kernel of the host machine with other containers. It runs a discrete process, taking no more memory than any other executable, making it lightweight.

    By contrast, a virtual machine (VM) runs a full-blown "guest" operating system with virtual access to host resources through a hypervisor

15. ```
    # Use an official Python runtime as a parent image
    FROM python:2.7-slim

    # Set the working directory to /app
    ```

```
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python", "app.py"]
```

16. DOcker run -p 8080:4444 dock

17. localhost:9000

18. This docker-compose.yml file tells Docker to do the following:

    Pull the image we uploaded in step 2 from the registry.

    Run 5 instances of that image as a service called web, limiting each one to use, at most, 10

    Immediately restart containers if one fails.

    Map port 4000 on the host to web's port 80.

    Instruct web's containers to share port 80 via a load-balanced network called webnet. (Internally, the containers themselves publish to web's port 80 at an ephemeral port.)

    Define the webnet network with the default settings (which is a load-balanced overlay network).