Answers to Self-check Questions section

Danis Alukaev B19-DS-01

1. Is this statement correct: You can't create multiple containers from the same image?

   This statement is incorrect. On the contrary, the concept of images allows sharing snapshots of the system between multiple users (or even on the same machine) to bring up identical environments.

2. Is this statement correct: All containers running on a single machine share the same operating system kernel, so they start instantly and make more efficient use of RAM.

   This statement is correct. Docker containers run natively on Linux, and share the kernel of the host machine with each other. Each container runs as the discrete process, so that it requires a moderate amount of memory.

3. Is this statement correct: Containers include the application and all of its dependencies, but share the kernel with other containers. They run as an isolated process in userspace on the host operating system. They're also not tied to any specific infrastructure – Docker containers run on any computer, on any infrastructure, and in any cloud.

   This statement is correct.

4. Fill in a blank: _____ is a cloud-hosted service from Docker that provides registry capabilities for public and private content

   > Docker Hub

5. Fill in a blank: _____ is a tool for defining and running multi-container Docker applications.

   > Docker Compose

6. Fill in a blank: _____ is native clustering for Docker. It turns a pool of Docker hosts into a single, virtual Docker host.

   > Docker Swarm

7. Fill in a blank: _____ is a text document that contains all the commands a user could call on the command line to assemble an image

   > Dockerfile

8. Explain Docker command: `docker exec -it container_id bash`

   Command creates a new bash session in container `container_id`. In fact, `docker exec` is used to run executable commands inside the specified container. There are provided options `-i` and `-t`. The former one is used to allocate pseudo-TTY (device that has the functions of a physical terminal without actually being one) whereas the latter one to run the container in interactive mode.

9. Explain Docker command: `docker build -t my_user/repo_name:1.0`

   This command builds the image with a tag `my_user/repo_name:1.0` from the local context.

10. Explain Docker command:

    ```
    docker commit -m "My first update" container_ID user_name/repository_name
    ```

    This command saves a new image by finding the container with id `container_ID` and committing all the changes to a new image name `user_name/repository_name`. The commit message is set to be `"My first update"`.

11. Explain Docker command: `docker push user_name/repository_name`

    This command pushes the image to the repository `repository_name` of user `user_name` in Docker Hub registry.

12. Explain Docker command: `docker ps`

    This command lists **all running** Docker containers.

13. Explain Docker command: `docker images`

    This command lists all the top-level Docker images available locally, their repositories, tags, and sizes.

14. Explain Docker command: `docker ps -a`

    This command lists **all** the Docker containers. By default (without `-a` option) it shows only running containers.

15. Which command runs a Docker container?

    In order to run container from particular Docker image, one can use following command:

    ```
    $ docker run <IMAGE>
    ```

    Note that one needs to specify executable commands either in `docker run`, or have an entrypoint script inside this image, e.g. specified in the Dockerfile.

16. Which command stops the Docker container?

    In order to stop particular Docker container, one can use following command:

    ```
    $ docker stop <CONTAINER ID>
    ```

17. Which command deletes a Docker container?

    In order to delete particular Docker container, one can use following command:

    ```
    $ docker rm <CONTAINER ID>
    ```

18. Which command deletes a Docker image?

    In order to delete particular Docker image, one can use following command:

    ```
    $ docker image rm <IMAGE>
    ```

    One also can delete all dangling images using

    ```
    $ docker image prune
    ```