1. What is Hypervisor? **It is a software that we install on top of hardware for type 1 or on top of host OS for type2. It is used to launch virtuals machines**.
2. What is the difference between type 1 and type 2 Hypervisor? **Configuration difference, Type 1 is install on top of the hardware and type on top of the host OS. Type 1 is mostly by companies and type 2 is used by end users.**
3. What is containerization? **It is the process used to create isolated environment call container where our application will run with all the dependencies and configurations tools.**
4. What is Virtualization? **It is the process used to create virtual representation of something, it can be virtual machines, virtual networks, storage.**
5. What is the difference between virtualization and containerization? **Virtualization it is the process used to create virtual machine, hypervisor is the software used to launch those virtual machine. Virtual machine are heavy because they have their own kernel, they are difficult to boot up. With contenairization, docker engine is the element responsible to launch containers. Containers are light weight because they share the same kernel with the host OS, they are easy to boot, portable, they used less resources and content all the dependencies used to run our application from one computing environment to another one without any issue.**
6. What is Docker and why Docker? **Docker is the tool used to automate the deployment of applications in lightweight containers so that applications can run work effitciently in different environment. Docker because virtualization presented some issues such as the dependencies issues, boot up process, hypervisor license issue, networking issue, isssue between developpers and operations teams.**
7. What are some issues that companies were having before Docker? **Dispute between developpers and operation team, dependencies issues, virtual machine are slow, take time to boot up, heavy, more expensive (required a lot of compute power)**
8. What are some issues or problems that Docker solved? **Dispute between developpers and operation team, dependencies issues, docker is fast, light weight, easy to boot up.**
9. Any idea why a lot of companies are using docker? **Docker because virtualization presented some issues such as the dependencies issues, boot up process, hypervisor license issue, networking issue, isssue between developpers and operations teams.**
10. What are some pros of docker that you are aware of? **Availibility, scalability, portability, easy to boot up, handle dependencies**
11. What is a docker client? **A cmd tool that help us to communicates to the docker deamon**
12. What is a docker host? **It is any machine where docker is installed and run**
13. What is a Docker Hub? **It is a central repository that stores and manages docker images** **provide by docker.**
14. What is a Docker image registry? **Is a central repository for storing and sharing docker images**
15. What is the difference between the public and private registry? **A private repository it is not accessible to the public for security reason and it’s mainly use by compagnies. Public repository are accessible to the public.**
16. What is a docker-machine? **It is tool use to create multiple docker host.**
17. Can you please work me through Docker Architecture?
18. What docker images? **A read only template that content dependencies use to create a docker containers**
19. What is Docker Swarm? **It is an orchestration tool use to manage multiple containers (scalability, availalibity)**
20. What is container orchestration? **Action of managing multiples containers**, process **of managing, deploying multiples containers**
21. It is possible to containerize all applications? **Yes. Every application is based of an image and from those image we can create an isolated environment to containairize our application**
22. What is a DockerFile? **It is a text file that contains all the instructions necessary to build our custom image.**
23. What is the purpose of Docker Compose? **Docker compose is used to launch many containers are the same time.**
24. Are you aware of a Docker Trusted Registry? if yes, what is a Docker Trusted Registry?
25. What are the basic requirements for docker to run on any system**? Docker engine install, CLI, docker hub,**
26. Can you tell the differences between a docker Image and a layer? **A docker image is a template made of layers. Layers is made of instructions.**
27. Describe the lifecycle of the Docker container **running stage, starting stage, killing stage, stop stage, exiting stage**
28. Can you tell me the approach to login into the docker registry. **from the CLI by using docker login and from DOCKER HUB**
29. What is the difference between docker pull and docker run? **With docker pull we download an image from the registry if that image does not exist locally. Where as docker run is used to create a container from an image.**
30. What command can you run to export a docker image as an archive? **Docker export allow us to export the content of a container as tar archive. Docker export -o my container.tar > container**
31. What command can be run to import a pre-exported Docker image into another Docker host? **Docker load**
32. What is the docker command that lists the status of all docker containers? **Docker ps -a**
33. Under what circumstances will you lose data stored in a container? **If you are not persisted your data, and also if the containers died or exit.**
34. Can a paused container be removed from Docker? yes
35. Can you tell the difference between CMD and ENTRYPOINT? **To overwrite the entrypoint we need to write -- entrypoint follow by the argument. The CMD it is easy to iverwrite**
36. Can you tell me what are the purposes of docker –rm (we pass it when we want to remove our automatically), rm, rmi, run, pull, start, and stop?
37. What is a base image in docker? It the template that w
38. If you have permission denied while trying to run docker commands, what can be the problem?
39. What is the proper way to install docker?
40. List the most commonly used instructions in Dockerfile that you are aware of and their purposes
41. What is the best way of deleting a container? **By stopping it first**
42. How many containers you can run in docker and what are the factors influencing this limit? We can run a thousand of containers and it depends on the resources cpu,
43. How will you ensure that container 1 runs before container 2 while using docker-compose?
44. If you wish to use a base image and make modifications or personalize it, how do you do that?
45. How do you create a docker container from an image?
46. How do you list all the running containers? Docker container ls, docker ps
47. Explain various Docker Basic Commands that you are aware of
48. Suppose you have 3 containers running and out of these, you wish to access one of them. How do you access a running container?
49. How to delete an image from the local storage system?
50. How to build a Dockerfile?
51. How to build a Dockerfile if the name is not Dockerfile?
52. How to build a Dockerfile if the name is not Dockerfile and it is not in your current working directory?
53. Do you know why Docker system prune is used? What does it do?
54. What is port forwarding in Docker?
55. What is the difference between free port forwarding and bind port forwarding?
56. What is the difference between a registry and a repository?
57. What are the advantages of AWS ECR compared to the Docker registry?
58. How can you access your application that is running in Docker?
59. Is there a way to identify the status of a Docker container?
60. Will you lose your data, when a docker container exists?
61. Is it better to directly remove the container using the rm command or stop the container followed by removing the container?
62. Suppose you have an application that has many dependent services. Will docker-compose wait for the current container to be ready to move to the running of the next service?
63. How to Backup, Restore, or Migrate data volumes under a Docker container?
64. How is the ENTRYPOINT instruction under Dockerfile different from the RUN instruction?
65. What is a volume in Docker? Why is it useful? It used to save data
66. What is the difference between volume mount and bind mount? **Volume mount is mounted from the volume directory and it is managed by docker and bind mount is mounted from any directory or files in the host. To mount volume we use -v or - -mount that allow us to add other caractheristic to our command such as the type of volume (bind), the source and target.**
67. Where are the docker volumes stored? /**var/lib/docker**
68. What are the different networks in Docker? **Bridge, overlay, host**
69. What is the default network in Docker and why? **Bridge network**
70. How is an overlay network different from a bridge network? **Bridge network is the default network, it allows containers on the same host to interact to each other using the containers names or their ip addresses. Overlay is used to enabled communication between containers from différents hosts.**
71. How have you used Docker in your previous position?
72. It is possible to see processes running inside a container from the Docker host?
73. Can a container restart by itself?
74. It is a good practice to store critical data in the container? if yes, why. If yes, why
75. Is it possible to execute a command in a running container from a Docker host?
76. Why do we say containers are ephemeral?
77. If You are not able to access the application that is running inside a container from the browser, what can be the problem? How can you solve it?
78. If you are not able to push an image into a registry, What can be the problem?
79. If you are not able to run any docker command, what can be the problem?
80. If you run an image to create a container and it exits certainly, what can be the problem?
81. It is a good practice the use free port forwarding in docker? If yes, why. If not, why?
82. What are some image pull policies that you are of in Docker-compose?
83. What are links in Docker-compose?
84. What is the purpose of EXPOSE command in Dockerfile?
85. How do you reduce the size of your image? why is it important?
86. Where do you think Docker is being used?
87. What platforms does docker run on?
88. How is Dockerfile different from Docker Compose?
89. Does the Docker container package up the entire OS?
90. Why Docker containers are so light?
91. Is it possible to have my own private Docker registry? Why is it a good practice?
92. What are the commands to control Docker with Systemd? Systemctl status docker
93. Are you aware of any container scan tools? Datadog, pageduty
94. How do you guys make sure that there is no vulnerability in the images before using them as a base image?
95. Do you guys pull images from the public registry and deploy them directly on the docker host?
96. It is possible to use multiple base images in one Dockerfile? if yes, How can you accomplish that?
97. Is it a good practice to run Docker compose in production? if yes, why. If not, why?