

# Day 21 : Docker Important interview Questions

This is [#90DaysofDevops](#) challenge under the guidance of [Shubham Londhe](#) sir.

Day 21 TASK

check this for task:

<https://github.com/LondheShubham153/90DaysOfDevOps/blob/master/2023/day21/tasks.md>

## Docker Interview Questions



### What is Docker?

Docker is an open-source platform that allows developers to create, deploy, and run applications in containers.

### What are the benefits of using Docker?

Docker provides several benefits, such as increased efficiency, portability, scalability, and security. It also simplifies the process of building, testing, and deploying applications.

### How do you create a Docker container?

To create a Docker container, you need to create a Dockerfile that defines the container's configuration, build the image using the Docker build command, and then run the image using the Docker run command.

### **What is a Docker image?**

A Docker image is a lightweight, standalone, executable package that contains everything needed to run an application, including the code, libraries, and dependencies.

### **What is the difference between a Docker image and a container?**

A Docker image is a template or blueprint for a container, whereas a container is a running instance of an image. Images are static, while containers are dynamic.

### **How do you share Docker images?**

Docker images can be shared by pushing them to a Docker registry, such as Docker Hub, and then pulling them from the registry on another machine.

### **What is Docker Hub?**

Docker Hub is a cloud-based repository for storing and sharing Docker images. It provides a central location for users to share and discover images.

### **What is a Dockerfile?**

A Dockerfile is a text file that contains instructions for building a Docker image. It specifies the base image, sets environment variables, copies files into the image, and runs commands to configure the image.

### **What is the difference between ADD and COPY in a Dockerfile?**

ADD and COPY are both used to copy files into a Docker image, but ADD has additional functionality. It can copy files from remote URLs, decompress files, and automatically set the file permissions.

### **What is the difference between CMD and ENTRYPOINT in a Dockerfile?**

CMD specifies the default command to run when a container is started, while ENTRYPOINT specifies the command to run when the container is started and also allows additional arguments to be passed to the command.

### **How do you pass environment variables to a Docker container?**

You can pass environment variables to a Docker container using the `-e` or `--env` option when running the container.

## **What is Docker Compose?**

Docker Compose is a tool for defining and running multi-container Docker applications. It allows you to define the services and dependencies of your application in a single YAML file.

## **How do you scale Docker containers?**

You can scale Docker containers using the Docker Swarm or Kubernetes orchestration tools. These tools allow you to manage and scale containers across multiple nodes.

## **What is Docker Swarm?**

Docker Swarm is a native clustering and orchestration tool for Docker. It allows you to manage and scale Docker containers across multiple hosts.

## **What is Kubernetes?**

Kubernetes is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications.

## **How do you troubleshoot a Docker container?**

You can troubleshoot a Docker container by inspecting the logs using the Docker logs command, attaching to the container using the Docker attach command, or running commands inside the container using the Docker exec command.

## **What is a Docker registry?**

A Docker registry is a storage and distribution system for Docker images. It allows you to store, share, and manage Docker images.

## **What is the difference between a Docker container and a virtual machine?**

A Docker container is a lightweight, isolated environment for running applications, whereas a virtual machine is a full-fledged operating system running on top of a hypervisor. Containers share the host's kernel and only contain the necessary components to run the application.

## **How do you backup and restore Docker containers?**

To backup a Docker container, you can create a Docker image of the container using the Docker commit command. To restore a container from a backup, you can create a new container from the backed-up image using the Docker run command.

## **How do you secure Docker containers?**

To secure Docker containers, you should follow best practices such as keeping the Docker engine and containers up-to-date, limiting access to the Docker daemon, using secure images from trusted sources, and securing the host system. You can also use Docker Security Scanning to identify vulnerabilities in your images.

## **Scenario Based Docker Questions**

### **Scenario 1: You want to create a Docker image for a Python application. How would you do it?**

To create a Docker image for a Python application, you can use a Dockerfile. First, you need to create a Dockerfile that specifies the base image, installs any necessary dependencies, and copies the Python application files into the container. Then, you can build the image using the “docker build” command.

### **Scenario 2: You have created a Docker container and you want to start it. How would you do it?**

To start a Docker container, you can use the “docker run” command followed by the name or ID of the container you want to start.

### **Scenario 3: You have a Docker container running a web server and you want to access the web server from your host machine. How would you do it?**

To access the web server running inside a Docker container, you need to map the container’s port to a port on the host machine using the “-p” option when you start the container. For example, if the web server is running on port 80 inside the container, you can use the command “docker run -p 80:80 [image]” to map port 80 in the container to port 80 on the host machine.

### **Scenario 4: You have multiple Docker containers running different services and you want to link them together so they can communicate with each other. How would you do it?**

To link Docker containers together, you can use the “— link” option when you start a container. For example, if you have a container running a database and a container running a web server that needs to access the database, you can use the command “docker run — link [database container name]:db [web server container name]” to link the two containers together.

### **Scenario 5: You have a Docker container running a database and you want to persist the data even if the container is removed or restarted. How would you do it?**

To persist data in a Docker container, you can use Docker volumes. You can create a volume using the “docker volume create” command and then mount the volume to the container using the “-v” option when you start the container. For example, if you have a container running a database, you can use the command “docker run -v [volume name]:[container path] [image]” to mount the volume to the container.

**Scenario 6: You want to run multiple instances of a Docker container to handle increased traffic. How would you do it?**

To scale Docker containers, you can use Docker Compose or Docker Swarm. With Docker Compose, you can define a “service” that specifies the number of replicas you want to run and then use the “docker-compose up” command to start the containers. With Docker Swarm, you can use the “docker service create” command to create a service and specify the number of replicas you want to run.

**Scenario 7: You want to deploy a Docker container to a remote server. How would you do it?**

To deploy a Docker container to a remote server, you can use Docker Machine to create a new Docker host on the remote server and then use Docker Compose or Docker Swarm to deploy the container to the new host. Alternatively, you can use a container registry like Docker Hub to store the Docker image and then use the “docker pull” command on the remote server to download the image and run it as a container.

**Scenario 8: You want to monitor the performance of a Docker container. How would you do it?**

To monitor the performance of a Docker container, you can use tools like Docker Stats or Docker Inspect. Docker Stats provides real-time information on container resource usage, while Docker Inspect provides more detailed information about a container, including its configuration and network settings. You can also use third-party monitoring tools like Prometheus or Grafana to monitor Docker containers.

**Scenario 9: You have a Docker container running a web server and you want to update the code without stopping the container. How would you do it?**

To update the code running in a Docker container without stopping it, you can use a bind mount or a Docker volume to mount the code directory from the host machine to the container. This will allow you to make changes to the code on the host machine and see the changes immediately in the container.

**Scenario 10: You want to deploy a Docker container to a Kubernetes cluster. How would you do it?**

To deploy a Docker container to a Kubernetes cluster, you can use a Kubernetes Deployment object. The Deployment object specifies the Docker image to use, the number of replicas to run, and other configuration options. You can use the “`kubectl apply`” command to apply the Deployment object to the Kubernetes cluster.

**Scenario 11: You have a Docker container running a web server and you want to expose it to the internet. How would you do it?**

To expose a Docker container to the internet, you can use a reverse proxy like Nginx or Apache. You can configure the reverse proxy to forward requests to the container’s port, which you can map to a port on the host machine using the “-p” option when you start the container.

**Scenario 12: You want to run a Docker container on a server with limited resources. How would you optimize the container for resource usage?**

To optimize a Docker container for resource usage, you can use a smaller base image, remove unnecessary packages and dependencies, and limit the container’s resource usage using the “—memory” and “—cpu” options when you start the container. You can also use Docker Compose or Docker Swarm to orchestrate containers and ensure that they are spread out across multiple hosts to avoid resource contention.

**Scenario 13: You have a Docker container running a Node.js application and you want to debug the application. How would you do it?**

To debug a Node.js application running in a Docker container, you can use the Node.js debugger or a third-party debugger like VS Code. You can use the “—inspect” option when you start the container to expose the Node.js debugger port, which you can then connect to from the host machine using a debugger client.

**Scenario 14: You have a Docker container running a Java application and you want to monitor the application’s JVM. How would you do it?**

To monitor a Java application running in a Docker container, you can use a Java profiler like VisualVM or JProfiler. You can use the “—jvm” option when you start the container to expose the JVM port, which you can then connect to from the host machine using the profiler client.

**Scenario 15: You have a Docker container running a PHP application and you want to profile the application’s performance. How would you do it?**

To profile a PHP application running in a Docker container, you can use a PHP profiler like Xdebug or Blackfire. You can configure the profiler to connect to the container’s PHP server using the container’s IP address and port.

**Scenario 16: You want to deploy a Docker container to a serverless platform like AWS Lambda. How would you do it?**

To deploy a Docker container to AWS Lambda, you can use AWS Fargate, which allows you to run Docker containers in a serverless environment. You can use AWS ECS to create a task definition for the Docker container and then use AWS Lambda to run the task definition.

**Scenario 17: You have a Docker container running a MongoDB database and you want to backup the data. How would you do it?**

To backup a MongoDB database running in a Docker container, you can use the `mongodump` command to create a backup of the database files. You can then copy the backup files to the host machine or another container using a bind mount or a Docker volume.

**Scenario 18: You have a Docker container running a Redis cache and you want to monitor the cache usage. How would you do it?**

To monitor a Redis cache running in a Docker container, you can use a Redis client like Redis CLI or RedisInsight. You can use the “`--link`” option when you start the container to link the container to the Redis client and then connect to the Redis server using the container’s IP address and port.

**Scenario 19: You want to build a Docker image for a Python application. How would you do it?**

To build a Docker image for a Python application, you can create a Dockerfile that specifies the base image to use, installs the necessary Python packages, and copies the application files to the container. You can then use the “`docker build`” command to build the Docker image.

**Scenario 20: You have a Docker container running a MySQL database and you want to scale the database horizontally. How would you do it?**

To scale a MySQL database horizontally, you can use Docker Swarm to orchestrate multiple containers running the MySQL image. You can use the “`--replicas`” option when you create a Docker service to specify the number of replicas to run, and Docker Swarm will automatically distribute the load across the containers. You can also use a load balancer like HAProxy to balance the traffic between the containers.

**Please, feel free to drop any questions in the comments below. I would be happy to answer them.**

**If this post was helpful, please do follow and click the clap**

**\_Thank you for reading**

**\_Rajani**