**Docker– Question & Answers**

1. **What is Docker, and how does it differ from a virtual machine?**

Docker is a platform for developing, shipping, and running applications using lightweight, portable containers.

* Difference from a Virtual Machine (VM):
  + Docker containers share the host OS kernel, making them lightweight and faster to start compared to VMs.
  + VMs have their own OS and virtualized hardware, leading to more resource overhead.

**2. What is a Docker Image, and how is it related to a Docker container?**  
A Docker Image is a lightweight, standalone, and immutable file that contains the source code, libraries, dependencies, and tools needed to run an application.

* **Relation to Docker Containers:**
  + A container is a runtime instance of a Docker Image. It adds a writable layer on top of the image and runs the application.

**3. What is the purpose of Docker Hub, and how do you use it?**  
Docker Hub is a cloud-based registry for sharing Docker images. It allows developers to store, share, and retrieve images.

* **Usage:**
  + Pull images (docker pull <image-name>).
  + Push images (docker push <image-name>).
  + Search for public images (docker search <image-name>).

**4. What is a Dockerfile, and what is its role in building Docker images?**  
A Dockerfile is a text file containing instructions to create a Docker image. Each line in a Dockerfile represents a step in the build process, such as installing software or copying files.

* **Role:** Automates the image creation process, ensuring consistency and reproducibility.

**5. How do you run a Docker container in detached mode, and what are the benefits of doing so?**

* Command: docker run -d <image-name>
* Benefits:
  + Runs the container in the background, allowing the terminal to remain free.
  + Useful for running long-term services like web servers.

**6. What is Docker Compose, and how is it used to manage multi-container Docker applications?**Docker Compose is a tool for defining and managing multi-container applications using a docker-compose.yml file.

* Usage:
  + Define services, networks, and volumes in a YAML file.
  + Use docker-compose up to start all defined containers.

**7. How do you persist data in a Docker container, and what are the implications of not doing so?**

* Persist Data: Use Docker volumes or bind mounts to store data outside the container (docker run -v /host/path:/container/path <image-name>).
* Implications: Without persistence, data generated inside the container will be lost when the container stops or is removed.

**8. What is a Docker volume, and how is it used to share data between containers?**A Docker volume is a managed storage area for persisting data.

* Usage:
  + Volumes can be shared among multiple containers using the -v flag during container creation.
  + Example: docker run -v my-volume:/data <image-name>.

**9. How do you configure networking for a Docker container, and what are the different networking modes available?**

* Configuration: Use the --network flag during container creation (docker run --network <network-name>).
* Networking Modes:
  + Bridge (default): Containers share a bridge network for communication.
  + Host: Containers share the host's network stack.
  + None: No network access.
  + Custom: User-defined networks for better isolation.

**10. What are some common use cases for Docker, and how can it benefit development and production environments?**

* **Common Use Cases:**
  + Application isolation and consistency.
  + CI/CD pipelines for testing and deployment.
  + Microservices architecture.
  + Scaling applications with container orchestration (e.g., Kubernetes).
* **Benefits:**
  + Speeds up development and deployment.
  + Reduces dependency conflicts.
  + Provides scalability and portability across environments.