

Docker Practice Questions

Docker Practice Sheet

1) Build a simple Docker image

Scenario:

Your team wants a tiny Java program containerized for a demo that prints “Hello Docker”.

Task:

Create a Dockerfile and build an image.

Answer:

Dockerfile:

Dockerfile

FROM eclipse-temurin:17-jdk-alpine

COPY Main.java .

RUN javac Main.java

CMD ["java", "Main"]

Build:

Shell

docker build -t hello-docker:1.0 .

2) Run the container in attached & detached modes

Scenario:

You want to see program output once, but run it silently in the background next time.

Answer:

Attached:

Shell

docker run hello-docker:1.0

Detached:

Shell

docker run -d hello-docker:1.0

Docker Practice Questions

3) Add a custom ENTRYPOINT

Scenario:

A Spring Boot JAR must always run when the container starts.

Answer:

Dockerfile

```
ENTRYPOINT ["java", "-jar", "app.jar"]
```

4) Override default port

Scenario:

Your app is configured for port 8080, but QA wants it on port 9090 at runtime.

Answer:

Dockerfile:

Dockerfile

```
EXPOSE 8080
```

Run with port override:

Shell

```
docker run -p 9090:9090 myapp:1.0 --server.port=9090
```

5) Pass environment variables

Scenario:

You need to run the app in dev environment without modifying the image.

Answer:

Shell

```
docker run -e APP_ENV=dev alpine sh -c "echo $APP_ENV"
```

Output:

dev

6) Inspect container logs

Scenario:

Docker Practice Questions

Your app is not responding; you need to review logs from a detached container.

Answer:

Run in background:

Shell

```
docker run -d --name myapp myapp:1.0
```

View logs:

Shell

```
docker logs myapp
```

7) Clean up unused images

Scenario:

Disk space is low; remove unused images but keep active ones.

Answer:

Shell

```
docker image prune
```

Remove all unused:

Shell

```
docker image prune -a
```

8) Use a named volume

Scenario:

You want PostgreSQL data to persist even if the container is deleted.

Answer:

Shell

```
docker volume create pgdata
```

```
docker run -d \
```

```
--name pg \
```

```
-e POSTGRES_PASSWORD=admin \
```

Docker Practice Questions

```
-v pgdata:/var/lib/postgresql/data \
postgres:16
```

9) Create a Docker Compose file

Scenario:

You want the team to run a single service with one simple command.

Answer:

docker-compose.yml

YAML

```
services:
```

```
app:
```

```
image: hello-docker:1.0
```

Run:

Shell

```
docker compose up
```

10) Start two services with Compose

Scenario:

Your Java backend must connect to a PostgreSQL database.

Answer:

docker-compose.yml

YAML

```
services:
```

```
db:
```

```
image: postgres:16
```

```
environment:
```

```
POSTGRES_PASSWORD: admin
```

```
app:
```

Docker Practice Questions

image: myapp:1.0

depends_on:

- db

Run:

Shell

docker compose up

11) Scale a service

Scenario:

You want to simulate load balancing by running 3 copies of a web service.

Answer:

Shell

docker compose up -d --scale web=3

12) Add a .dockerignore

Scenario:

Docker builds are slow because unnecessary files (like target/) are copied.

Answer:

.dockerignore

target/

.git/

13) Test a multi-stage build

Scenario:

Your current Java image is too large; you want a smaller runtime image.

Answer:

Multi-stage Dockerfile:

Dockerfile

FROM maven:3.9-eclipse-temurin-17 AS build

Docker Practice Questions

COPY . .

RUN mvn package -DskipTests

FROM eclipse-temurin:17-jre

COPY --from=build target/app.jar app.jar

CMD ["java", "-jar", "app.jar"]

Check size:

Shell

docker images

14) Inspect image layers

Scenario:

You want to understand how your Dockerfile layers are structured.

Answer:

Shell

docker history myapp:1.0

15) Debug a failing container

Scenario:

The app container exits immediately; you need to troubleshoot inside it.

Answer:

Shell

docker run -it --entrypoint sh myapp:1.0

Inside, run checks:

Shell

ls

java -version

cat logs/error.log

Docker Practice Questions

Docker Practice Sheet

1. Which instruction is best for setting the default executable of a container while still allowing runtime args to be appended?

A. CMD (shell form)
B. ENTRYPOINT (exec form)
C. RUN
D. ENV

Answer: B.

Explanation: ENTRYPOINT (exec form) defines the main executable; CMD supplies default arguments that can be overridden.

2. What does -d do in docker run -d myimage?

A. Runs the container with an attached TTY
B. Runs the container in detached mode
C. Deletes the container on exit
D. Enables debug logs

Answer: B.

Explanation: -d detaches the container from the terminal and runs it in the background.

3. In a Dockerfile, which pair correctly copies a build artifact from a previous stage?

A. COPY app.jar /app/
B. ADD --from=builder /app/target/*.jar /app/
C. COPY --from=builder /app/target/*.jar /app/
D. RUN cp /builder/app.jar /app/

Answer: C.

Explanation: Multi-stage builds use COPY --from=<stage> to bring artifacts from another stage.

4. What does .dockerignore affect during docker build?

A. Files available inside the running container at runtime
B. Files available in the build context sent to the Docker daemon
C. The container's runtime environment variables
D. The image's final layers

Answer: B.

Explanation: .dockerignore reduces the build context uploaded to the daemon, improving performance and cache behavior.

Docker Practice Questions

5. Which statement about EXPOSE 8080 is correct?

- A. It publishes port 8080 to the host automatically.
- B. It documents the container's intended port; you still need -p to publish.
- C. It forces the app to listen on 8080.
- D. It sets a firewall rule on the host.

Answer: B.

Explanation: EXPOSE is documentation/metadata; use -p host:container to publish.

6. You want to run an interactive shell in a running container. Which command is correct?

- A. docker attach <container>
- B. docker exec -it <container> /bin/sh
- C. docker run -d <image> /bin/bash
- D. docker top <container>

Answer: B.

Explanation: docker exec -it starts a process (shell) inside an existing container with TTY.

7. Which option persists database data across container recreations most portably for dev?

- A. Anonymous volume
- B. Named volume
- C. Bind mount to /tmp
- D. ENV variables

Answer: B.

Explanation: Named volumes are portable and managed by Docker, persisting data independent of container lifecycle.

8. In Docker Compose, what does depends_on guarantee by default?

- A. The dependency service is healthy before starting the dependent
- B. Strict startup order only
- C. A network policy between services
- D. Nothing—it is ignored

Answer: B.

Explanation: Basic depends_on controls start order; to wait for health, use healthchecks and condition: service_healthy (Compose v2 semantics).

9. Which is true about ARG vs ENV?

- A. ARG is available at runtime; ENV only at build time

Docker Practice Questions

- B. Both are only build-time
- C. ARG is build-time; ENV is available at runtime to the container
- D. ARG and ENV are equivalent

Answer: C.

Explanation: ARG exists during build steps; ENV persists in the image and is visible at container runtime.

10. **What is the safest way to roll back from app:latest to app:v1.0.0?**

- A. Rebuild the previous commit
- B. Retag latest as v1.0.0
- C. Stop current container and run app:v1.0.0 by tag
- D. Delete all images and start from scratch

Answer: C.

Explanation: Keep immutable version tags and run the known-good tag.

11. **Which command shows real-time logs of a detached container?**

- A. docker ps -f logs
- B. docker attach -f
- C. docker logs -f <container>
- D. docker inspect -f

Answer: C.

Explanation: docker logs -f tails logs continuously.

12. **Which Dockerfile ordering best leverages build cache for Maven projects?**

- A. COPY src, then RUN mvn dependency:go-offline, then COPY pom.xml
- B. COPY pom.xml, run dependency step, then COPY src and build
- C. COPY src, then COPY pom.xml, then build
- D. Order does not matter

Answer: B.

Explanation: Put the infrequently changing dependency step before source code to maximize cache reuse.

13. **What does docker compose up -d --scale web=3 do?**

- A. Creates a Swarm service with 3 replicas
- B. Starts 3 containers for the web service defined in Compose
- C. Clones the image three times on disk
- D. Creates 3 networks for load balancing

Answer: B.

Docker Practice Questions

Explanation: Compose can run multiple containers per service; it is not Swarm orchestration.

14. Which is the best justification for multi-stage builds for Java apps?

- A. Faster runtime startup
- B. Smaller runtime image by excluding build tools
- C. Automatic tests during build
- D. Easier environment variable management

Answer: B.

Explanation: Multi-stage builds copy only the built artifacts into a lean runtime image.
