Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Docker Quiz – Hard Level (50 Questions)

Section 1: Basics and Containers

Question 1:

You want to limit a container's memory usage to 512MB to avoid consuming all available system memory. Which option will you use in the `docker run` command?

1. `--memory=512mb`
2. `--limit-memory=512mb`
3. `--mem-limit=512m`
4. `--resource-limit=512m`

Question 2:

You need to run a container in detached mode while naming it `web_app`. Which command achieves this?

1. `docker run --detach --name web_app`
2. `docker container create --detach web_app`
3. `docker create web_app`
4. `docker run web_app --detach`

Question 3:

How can you restart all running containers at once?

1. `docker restart all`
2. `docker restart $(docker ps -q)`
3. `docker container restart`
4. `docker service restart all`

Question 4:

You want to see the real-time resource usage (CPU, memory, etc.) of running containers. Which command will you use?

1. `docker info`
2. `docker stats`
3. `docker ps -a`
4. `docker usage`

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Question 5:

How can you forcefully stop a container that isn't responding to the `docker stop` command?

1. `docker terminate`
2. `docker kill`
3. `docker container delete`
4. `docker force-stop`

Section 2: Docker Images

Question 6:

How can you rebuild a Docker image without using cached layers?

1. `docker build --no-cache`
2. `docker rebuild`
3. `docker build --force`
4. `docker image rebuild`

Question 7:

Which of the following reduces Docker image size during the build process?

1. Use a larger base image for compatibility
2. Use multi-stage builds and smaller base images
3. Include all dependencies in one layer
4. Avoid using a `.dockerignore` file

Question 8:

What command shows the history of all layers in a Docker image?

1. `docker inspect <image>`
2. `docker history <image>`
3. `docker image layers <image>`
4. `docker logs <image>`

Question 9:

You want to remove dangling images. What is the best way?

1. `docker image prune`
2. `docker system prune`

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

3. `docker remove --dangling`
4. `docker image rm --dangling`

Question 10:

How do you view the configuration and metadata of a Docker image?

1. `docker info <image>`
2. `docker inspect <image>`
3. `docker meta <image>`
4. `docker logs <image>`

Section 3: Networking

Question 11:

You need two containers to communicate with each other. What is the best approach?

1. Use the `--link` flag
2. Use `docker network connect`
3. Use a shared volume
4. Use a custom Docker network

Question 12:

How can you inspect the details of a custom Docker network?

1. `docker inspect network <network-id>`
2. `docker network show <network-name>`
3. `docker network inspect <network-name>`
4. `docker network info`

Question 13:

What is the default network driver created by Docker for new containers?

1. `bridge`
2. `host`
3. `none`
4. `overlay`

Question 14:

How can you disconnect a running container from a network?

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

1. `docker container disconnect <container> <network>`
2. `docker network disconnect <network> <container>`
3. `docker network remove <container> <network>`
4. `docker disconnect <network> <container>`

Question 15:

Which Docker network type allows containers on multiple hosts to communicate with each other?

1. `bridge`
2. `overlay`
3. `host`
4. `none`

Section 4: Volumes and Storage

Question 16:

How do you create a named volume in Docker?

1. `docker volume create my_volume`
2. `docker volume my_volume`
3. `docker create --volume my_volume`
4. `docker create volume my_volume`

Question 17:

What happens to the data in a container's anonymous volume after the container is removed?

1. The volume is retained until manually deleted
2. The volume is automatically deleted
3. The volume is archived
4. The volume is renamed

Question 18:

How do you list all Docker volumes?

1. `docker volume list`
2. `docker list volumes`
3. `docker volume ls`
4. `docker volumes`

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Question 19:

You want to backup the contents of a named volume. Which command should you use?

1. `docker volume save`
2. `docker volume export`
3. `docker run --rm -v volume_name:/data -v $(pwd):/backup busybox tar cvf /backup/backup.tar /data`
4. `docker volume archive`

Question 20:

What is the difference between a bind mount and a volume?

1. Bind mounts are managed by Docker, while volumes are tied to host paths
2. Volumes are managed by Docker, while bind mounts use host paths
3. Bind mounts are always persistent, but volumes are not
4. Volumes are more secure than bind mounts

Section 5: Security

Question 21:

What is the best practice for securing sensitive data used by containers?

1. Use Docker secrets
2. Use environment variables directly in the Dockerfile
3. Hardcode sensitive data into the application
4. Use a `.env` file

Question 22:

How do you run a container with the least privileges?

1. Use `--privileged=false`
2. Use the `--user` flag to specify a non-root user
3. Run as root but restrict file permissions
4. Use Docker's default configuration

Question 23:

What does enabling Docker Content Trust (DCT) ensure?

1. Only signed images can be pulled and run
2. Containers cannot access the host filesystem
3. Secrets are encrypted in transit
4. Volumes are securely mounted

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Question 24:

How do you encrypt traffic between Docker Swarm nodes?

1. Docker Swarm encrypts traffic by default
2. Use `--encrypt` flag during service creation
3. Configure an SSL certificate for each container
4. Enable host encryption

Question 25:

What is the purpose of a Docker secret?

1. To store encrypted logs
2. To securely store sensitive data like passwords and API keys
3. To manage container backups
4. To encrypt Docker volumes

Section 6: Orchestration (Docker Swarm and Kubernetes)

Question 26:

How do you scale a Docker Swarm service to 10 replicas?

1. `docker service scale <service-name>=10`
2. `docker swarm scale <service-name> 10`
3. `docker scale <service-name>=10`
4. `docker swarm update <service-name>=10`

Question 27:

How can you rollback a service in Docker Swarm?

1. `docker service rollback`
2. `docker service revert`
3. `docker stack rollback`
4. `docker update --rollback`

Question 28:

What is the purpose of a stack file in Docker Swarm?

1. To store sensitive environment variables
2. To define multi-container applications

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

3. To create Docker secrets
4. To scale services automatically

Question 29:

How do you view the logs of a service in Docker Swarm?

1. `docker service logs`
2. `docker stack logs`
3. `docker logs`
4. `docker swarm logs`

Question 30:

What is a key difference between Docker Compose and Docker Swarm?

1. Compose is for production use, while Swarm is for local development
2. Swarm supports scaling and orchestration, but Compose does not
3. Compose manages clusters, while Swarm manages single nodes
4. Swarm requires YAML files, while Compose does not

Section 7: Performance

Question 31:

How do you monitor a container's CPU and memory usage over time?

1. `docker monitor <container>`
2. `docker stats <container>`
3. `docker top <container>`
4. `docker resource-usage <container>`

Question 32:

What is the effect of using the `--cpu-shares` flag when starting a container?

1. It limits the container to a fixed CPU usage
2. It prioritizes CPU time relative to other containers
3. It assigns a container to a specific CPU core
4. It throttles the CPU frequency

Question 33:

What does the `--ulimit` flag do?

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

1. Sets CPU limits for a container
2. Configures the maximum number of open file descriptors or processes
3. Restricts memory usage
4. Limits I/O bandwidth

Question 34:

Which of the following is the best practice to reduce container startup time?

1. Use smaller base images
2. Disable container health checks
3. Increase the container's memory limit
4. Use the `--privileged` flag

Question 35:

How can you optimize image build times in Docker?

1. Avoid using the `.dockerignore` file
2. Place frequently changing instructions at the top of the Dockerfile
3. Place static instructions at the top of the Dockerfile
4. Use large base images

Section 8: Troubleshooting

Question 36:

How do you view the logs of a failed container?

1. `docker ps --logs`
2. `docker logs <container-id>`
3. `docker inspect logs <container-id>`
4. `docker container logs --failed`

Question 37:

What does the error “no space left on device” indicate in Docker?

1. CPU resources are maxed out
2. A volume or the Docker storage layer has run out of disk space
3. Memory usage has exceeded the limit
4. The container has reached its maximum file limit

Question 38:

How do you debug a container that keeps restarting?

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

1. Use `docker logs` to check logs for errors
2. Inspect the health status using `docker inspect`
3. Remove the container and rebuild it
4. Both 1 and 2

Question 39:

How can you see all events (start, stop, etc.) related to Docker containers?

1. `docker events`
2. `docker history`
3. `docker activity`
4. `docker info`

Question 40:

You suspect a container process is consuming excessive resources. How do you check which processes are running inside it?

1. `docker inspect <container>`
2. `docker ps`
3. `docker exec <container> ps aux`
4. `docker resource-usage <container>`

Section 9: Advanced Networking

Question 41:

Which of the following is required to enable inter-host communication for Docker containers?

1. Use the `host` network driver
2. Use the `overlay` network driver
3. Use a bind mount
4. Use the `bridge` network driver

Question 42:

How can you expose a container on port 8080 to the host?

1. `docker run -p 8080:80`
2. `docker run --expose 8080:80`
3. `docker run -publish 80:8080`
4. `docker run --bind 80:8080`

Question 43:

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

What does the `--network-alias` flag do?

1. Assigns a custom IP address to the container
2. Provides an alternate DNS name for the container
3. Configures port forwarding for the container
4. Links the container to a bridge network

Question 44:

How can you verify connectivity between two containers on the same network?

1. Use `docker inspect` to check IPs and test with `ping`
2. Use `docker network ls` to confirm connectivity
3. Use `docker logs` to view connection details
4. Use `docker stats` to monitor connectivity

Section 10: Miscellaneous

Question 45:

How do you remove all stopped containers, unused networks, and dangling images?

1. `docker cleanup`
2. `docker system prune`
3. `docker prune --all`
4. `docker remove --all`

Question 46:

Which command pauses all processes within a running container?

1. `docker halt <container>`
2. `docker stop <container>`
3. `docker pause <container>`
4. `docker suspend <container>`

Question 47:

How do you update the Docker daemon configuration?

1. Edit `/etc/docker/daemon.json` and restart Docker
2. Use `docker config update`
3. Use `docker daemon update`
4. Modify the Dockerfile

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Question 48:

What is the purpose of the `.dockerignore` file?

1. Exclude files from being copied during the image build process
2. Ignore container logs during runtime
3. Prevent specific containers from being started
4. Specify ignored images

Question 49:

What does `docker save` do?

1. Back up container data
2. Export a Docker image as a tar file
3. Compress container logs
4. Create a volume snapshot

Question 50:

How do you restore a Docker image from a `.tar` backup?

1. `docker import <tar-file>`
2. `docker load < tar-file`
3. `docker restore < tar-file>`
4. `docker unpack < tar-file>`

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Docker Quiz – Hard Level (50 Answers)

Section 1: Basics and Containers

Question 1:

Answer: `--memory=512mb`
Explanation: The `--memory` flag in the `docker run` command allows you to set a memory limit for a container.

Question 2:

Answer: `docker run --detach --name web_app`
Explanation: The `--detach` (`-d`) flag runs a container in the background, and `--name` assigns it a custom name.

Question 3:

Answer: `docker restart $(docker ps -q)`
Explanation: `docker ps -q` lists all running container IDs. Passing it to `docker restart` restarts them all.

Question 4:

Answer: `docker stats`
Explanation: The `docker stats` command displays real-time resource usage for all running containers.

Question 5:

Answer: `docker kill`
Explanation: `docker kill` forcibly stops a container by sending a `SIGKILL` signal.

Section 2: Docker Images

Question 6:

Answer: `docker build --no-cache`
Explanation: The `--no-cache` flag forces a rebuild of all image layers, ignoring cached layers.

Question 7:

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Answer: Use multi-stage builds and smaller base images
Explanation: Multi-stage builds allow you to reduce image size by excluding unnecessary build dependencies from the final image.

Question 8:

Answer: `docker history <image>`
Explanation: The `docker history` command shows the creation history of a Docker image, including each layer.

Question 9:

Answer: `docker image prune`
Explanation: This command removes dangling images (untagged images not associated with any container).

Question 10:

Answer: `docker inspect <image>`
Explanation: The `docker inspect` command provides detailed metadata about a Docker image or container.

Section 3: Networking

Question 11:

Answer: Use a custom Docker network
Explanation: Creating a custom network (e.g., bridge or overlay) allows containers to communicate directly without `--link`, which is deprecated.

Question 12:

Answer: `docker network inspect <network-name>`
Explanation: The `docker network inspect` command displays detailed information about a Docker network.

Question 13:

Answer: `bridge`
Explanation: The default network for new containers is the `bridge` network unless specified otherwise.

Question 14:

Answer: `docker network disconnect <network> <container>`
Explanation: This command removes a container from the specified network.

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Question 15:

Answer: `overlay`
Explanation: The `overlay` network driver enables multi-host communication, suitable for Swarm or Kubernetes.

Section 4: Volumes and Storage

Question 16:

Answer: `docker volume create my_volume`
Explanation: The `docker volume create` command creates a named volume managed by Docker.

Question 17:

Answer: The volume is automatically deleted
Explanation: Anonymous volumes are removed when the container they are attached to is removed unless explicitly retained.

Question 18:

Answer: `docker volume ls`
Explanation: The `docker volume ls` command lists all Docker volumes on the system.

Question 19:

Answer:

docker run --rm -v volume_name:/data -v $(pwd):/backup busybox tar cvf /backup/backup.tar /data

Explanation: This command mounts the volume, copies its contents into a tar archive, and saves it to the current directory.

Question 20:

Answer: Volumes are managed by Docker, while bind mounts use host paths
Explanation: Volumes are Docker-managed and portable, whereas bind mounts directly map host directories.

Section 5: Security

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Question 21:

Answer: Use Docker secrets
Explanation: Docker secrets securely store sensitive data and are only accessible to services running in a Swarm.

Question 22:

Answer: Use the `--user` flag to specify a non-root user
Explanation: Running containers as non-root users is a best practice to minimize security risks.

Question 23:

Answer: Only signed images can be pulled and run
Explanation: Docker Content Trust ensures only signed, verified images are used to improve security.

Question 24:

Answer: Docker Swarm encrypts traffic by default
Explanation: Docker Swarm automatically encrypts network traffic between nodes.

Question 25:

Answer: To securely store sensitive data like passwords and API keys
Explanation: Docker secrets are encrypted at rest and only exposed to containers that require them.

Section 6: Orchestration

Question 26:

Answer: `docker service scale <service-name>=10`
Explanation: This command scales the specified Docker service to 10 replicas.

Question 27:

Answer: `docker service rollback`
Explanation: The `docker service rollback` command reverts a service to its previous version.

Question 28:

Answer: To define multi-container applications
Explanation: A stack file (YAML) defines services, networks, and volumes in a Swarm.

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Question 29:

Answer: `docker service logs`
Explanation: This command shows logs for a specific Swarm service.

Question 30:

Answer: Swarm supports scaling and orchestration, but Compose does not
Explanation: Compose is ideal for local environments, while Swarm enables service scaling and orchestration.

Section 7: Performance

Question 31:

Answer: `docker stats <container>`
Explanation: This command displays real-time CPU, memory, and network usage for containers.

Question 32:

Answer: It prioritizes CPU time relative to other containers
Explanation: `--cpu-shares` defines a container's relative CPU priority compared to others.

Question 33:

Answer: Configures the maximum number of open file descriptors or processes
Explanation: The `--ulimit` flag sets resource usage limits for containers.

Question 34:

Answer: Use smaller base images
Explanation: Using minimal base images like `alpine` reduces container startup time.

Question 35:

Answer: Place static instructions at the top of the Dockerfile
Explanation: Placing rarely-changing instructions early allows Docker to reuse cached layers during builds.

Section 8: Troubleshooting

Question 36:

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Answer: `docker logs <container-id>`
Explanation: This command retrieves logs for any container, including failed ones.

Question 37:

Answer: A volume or the Docker storage layer has run out of disk space
Explanation: The error indicates insufficient disk space for Docker storage.

Question 38:

Answer: Both 1 and 2
Explanation: Checking logs and inspecting container health often reveals the cause of frequent restarts.

Question 39:

Answer: `docker events`
Explanation: This command shows real-time event streams for Docker containers.

Question 40:

Answer: `docker exec <container> ps aux`
Explanation: Running `ps aux` inside the container reveals active processes.

Section 9: Advanced Networking

Question 41:

Answer: Use the `overlay` network driver
Explanation: The `overlay` driver enables inter-host container communication in Docker Swarm.

Question 42:

Answer: `docker run -p 8080:80`
Explanation: The `-p` flag maps a container port to a host port.

Question 43:

Answer: Provides an alternate DNS name for the container
Explanation: `--network-alias` assigns an additional hostname for containers in the same network.

Question 44:

Author: Khadar Basha SHAIK
LinkedIn: https://www.linkedin.com/in/shaikkhadarbasha/
Website: https://newsletter.codetuts.in/

Answer: Use `docker inspect` to check IPs and test with `ping`
Explanation: Inspect the container network settings and verify connectivity using `ping`.

Section 10: Miscellaneous

Question 45:

Answer: `docker system prune`
Explanation: This command removes all stopped containers, dangling images, and unused networks.

Question 46:

Answer: `docker pause <container>`
Explanation: Pausing suspends all container processes without stopping them.

Question 47:

Answer: Edit `/etc/docker/daemon.json` and restart Docker
Explanation: Docker daemon settings are configured in `daemon.json` and require a restart to apply changes.

Question 48:

Answer: Exclude files from being copied during the image build process
Explanation: `.dockerignore` prevents unnecessary files from being included in the image build context.

Question 49:

Answer: Export a Docker image as a tar file
Explanation: `docker save` creates a tar archive of an image that can be shared or backed up.

Question 50:

Answer: `docker load < tar-file`
Explanation: `docker load` restores an image from a tar backup created with `docker save`.