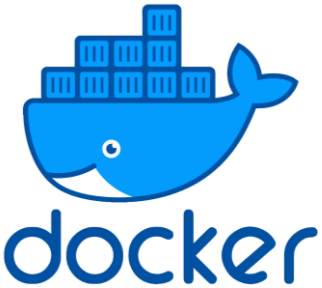

50 Docker Commands For Container Management, Image Manipulation, Networking & More

Container Management:

1. **Run a Container:**

   ```
   docker run [OPTIONS] IMAGE[:TAG] [COMMAND] [ARG...]
   ```

2. **List Running Containers:**

   ```
   docker ps
   ```

3. **List All Containers:**

   ```
   docker ps -a
   ```

4. **Stop a Running Container:**

   ```
   docker stop CONTAINER_ID
   ```

5. **Remove a Container:**

   ```
   docker rm CONTAINER_ID
   ```

6. **Remove All Stopped Containers:**

   ```
   docker container prune
   ```

7. **Inspect Container Details:**

   ```
   docker inspect CONTAINER_ID
   ```

8. **Attach to a Running Container:**

   ```
   docker exec -it CONTAINER_ID /bin/bash
   ```

Image Manipulation:

9. **List Local Images:**

   ```
   docker images
   ```

10. **Pull an Image from Docker Hub:**

    ```
    docker pull IMAGE[:TAG]
    ```

11. **Build an Image from Dockerfile:**

    ```
    docker build -t IMAGE_NAME:TAG PATH_TO_DOCKERFILE
    ```

12. **Remove an Image:**

    ```
    docker rmi IMAGE_ID
    ```

13. **Remove All Unused Images:**

    ```
    docker image prune
    ```

Networking:

14. **List Networks:**

    ```
    docker network ls
    ```

15. **Inspect Network Details:**

    ```
    docker network inspect NETWORK_ID
    ```

16. **Create a Bridge Network:**

```
docker network create --driver bridge NETWORK_NAME
```

17. **Connect Container to Network:**

```
docker network connect NETWORK_NAME CONTAINER_NAME
```

18. **Disconnect Container from Network:**

```
docker network disconnect NETWORK_NAME CONTAINER_NAME
```

Volume Management:

19. **List Volumes:**

```
docker volume ls
```

20. **Inspect Volume Details:**

```
docker volume inspect VOLUME_NAME
```

21. **Create a Volume:**

```
docker volume create VOLUME_NAME
```

22. **Remove a Volume:**

```
docker volume rm VOLUME_NAME
```

Container Logs:

23. **View Container Logs:**

```
docker logs CONTAINER_ID
```

24. **Tail Container Logs:**

```
docker logs -f CONTAINER_ID
```

Docker Compose:

25. **Run Docker Compose:**

    ```
    docker-compose up -d
    ```

26. **Stop Docker Compose Services:**

    ```
    docker-compose down
    ```

27. **Build and Run Docker Compose:**

    ```
    docker-compose up --build -d
    ```

Docker System:

28. **Display System-Wide Information:**

    ```
    docker info
    ```

29. **Show Docker Disk Usage:**

    ```
    docker system df
    ```

30. **Remove All Unused Data:**

    ```
    docker system prune
    ```

Docker Registry:

31. **Login to Docker Hub:**

    ```
    docker login
    ```

32. **Push Image to Docker Hub:**

    ```
    docker push IMAGE[:TAG]
    ```

33. **Pull Image from Private Registry:**

```
docker pull REGISTRY_URL/IMAGE[:TAG]
```

Docker Swarm:

34. **Initialize Docker Swarm:**

```
docker swarm init
```

35. **Join Node to Swarm:**

```
docker swarm join --token TOKEN IP:PORT
```

36. **List Nodes in Swarm:**

```
docker node ls
```

37. **Create a Service:**

```
docker service create [OPTIONS] IMAGE[:TAG] [COMMAND] [ARG...]
```

38. **Scale a Service:**

```
docker service scale SERVICE_NAME=REPLICAS
```

39. **Inspect Service Details:**

```
docker service inspect SERVICE_NAME
```

40. **Remove a Service:**

```
docker service rm SERVICE_NAME
```

Docker Security:

41. **Check Container Vulnerabilities:**

```
docker scan IMAGE[:TAG]
```

42. **Run Container with Security Options:**

```
docker run --security-opt seccomp=unconfined --cap-add=SYS_PTRACE -it
IMAGE[:TAG]
```

43. **Run Container with Readonly Filesystem:**

```
docker run --read-only -it IMAGE[:TAG]
```

Docker Stats:

44. **Display Real-time Container Resource Usage:**

```
docker stats CONTAINER_ID
```

Docker Events:

45. **Monitor Docker Events:**

```
docker events
```

Docker Debugging:

46. **Inspect Docker Bridge Network:**

```
docker network inspect bridge
```

47. **View Docker Daemon Logs:**

```
journalctl -u docker
```

48. **Check Docker Version:**

```
docker version
```

Miscellaneous:

49. **Copy Files between Host and Container:**

```
docker cp SOURCE_PATH CONTAINER_ID:DEST_PATH
```

50. **Create a Custom Docker Bridge Network:**

```
docker network create --driver bridge --subnet=SUBNET_NAME
CUSTOM_NETWORK_NAME
```

This cheat sheet covers a wide range of Docker commands for container management, image manipulation, networking, and more. Customize commands based on your specific use case and requirements.