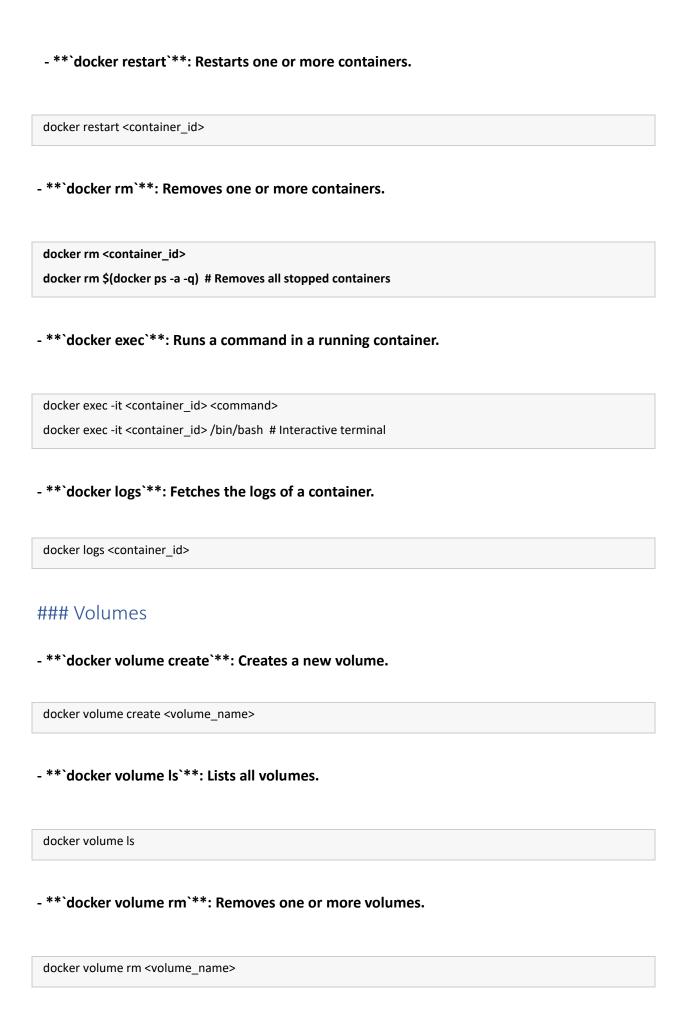
Docker Cheat Sheet

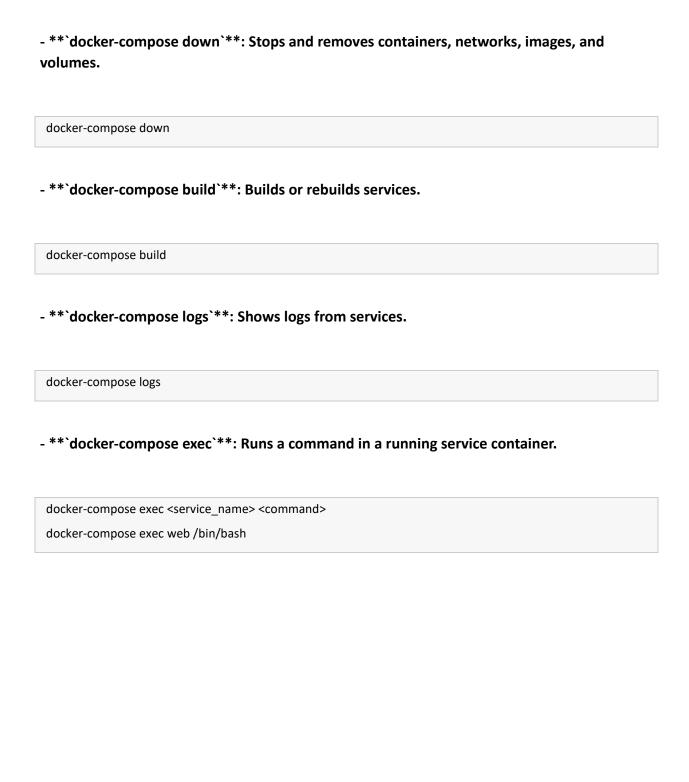
Basic Commands - **`docker --version`**: Shows the Docker version. docker --version - **`docker info`**: Displays system-wide information about Docker. docker info - **`docker help`**: Lists all available Docker commands or provides help for a specific command. docker help docker help <command> ### Images - **`docker pull`**: Downloads an image from a registry. docker pull <image>:<tag> docker pull ubuntu:latest - **`docker images`**: Lists all downloaded images. docker images

- **`docker rmi`**: Removes one or more images. docker rmi <image_id> docker rmi ubuntu:latest - **`docker build`**: Builds an image from a Dockerfile. docker build -t <name>:<tag> <path> docker build -t myapp:latest. ### Containers - **`docker run`**: Runs a command in a new container. docker run <options> <image> <command> docker run -it ubuntu:latest /bin/bash # Interactive terminal docker run -d -p 80:80 nginx # Detached mode - **`docker ps`**: Lists running containers. docker ps docker ps -a # Lists all containers - **`docker stop`**: Stops one or more running containers. docker stop <container_id> - **`docker start`**: Starts one or more stopped containers. docker start <container_id>



Networks

- **`docker network create`**: Creates a new network.
docker network create <network_name></network_name>
- **`docker network Is`**: Lists all networks.
docker network ls
- **`docker network rm`**: Removes one or more networks.
docker network rm <network_name></network_name>
- **`docker network connect`**: Connects a container to a network.
docker network connect <network_name> <container_id></container_id></network_name>
- **`docker network disconnect`**: Disconnects a container from a network.
docker network disconnect <network_name> <container_id></container_id></network_name>
Docker Compose
- **`docker-compose up`**: Builds, creates, starts, and attaches to containers for a service.
docker-compose up
docker-compose up -d # Detached mode



Dockerfile

A `Dockerfile` is a text document that contains all the commands to assemble an image.

Use an official Python runtime as a parent image
FROM python:3.8-slim-buster
Set the working directory in the container
WORKDIR /app
Copy the current directory contents into the container at /app
COPY . /app
Install any needed packages specified in requirements.txt
RUN pip installno-cache-dir -r requirements.txt
Make port 80 available to the world outside this container
EXPOSE 80
Define environment variable
ENV NAME World
Run app.py when the container launches
CMD ["python", "app.py"]
шин 11
Useful Tips

- **Remove all stopped containers**:

docker container prune

- **Remove all unused images**:

docker image prune

docker system prune
- **View the Docker system's disk usage**:
docker system df

- **Remove all unused data (volumes, networks, images, containers)**: