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>
```

- **`docker restart`: Restarts one or more containers.**

```
docker restart <container_id>
```

- **`docker rm`: Removes one or more containers.**

```
docker rm <container_id>
docker rm $(docker ps -a -q)  # Removes all stopped containers
```

- **`docker exec`: Runs a command in a running container.**

```
docker exec -it <container_id> <command>
docker exec -it <container_id> /bin/bash  # Interactive terminal
```

- **`docker logs`: Fetches the logs of a container.**

```
docker logs <container_id>
```

Volumes

- **`docker volume create`: Creates a new volume.**

```
docker volume create <volume_name>
```

- **`docker volume ls`: Lists all volumes.**

```
docker volume ls
```

- **`docker volume rm`: Removes one or more volumes.**

```
docker volume rm <volume_name>
```

Networks

- **`docker network create`: Creates a new network.**

```
docker network create <network_name>
```

- **`docker network ls`: Lists all networks.**

```
docker network ls
```

- **`docker network rm`: Removes one or more networks.**

```
docker network rm <network_name>
```

- **`docker network connect`: Connects a container to a network.**

```
docker network connect <network_name> <container_id>
```

- **`docker network disconnect`: Disconnects a container from a network.**

```
docker network disconnect <network_name> <container_id>
```

Docker Compose

- **`docker-compose up`: Builds, creates, starts, and attaches to containers for a service.**

```
docker-compose up
docker-compose up -d  # Detached mode
```

- **`docker-compose down`: Stops and removes containers, networks, images, and volumes.**

```
docker-compose down
```

- **`docker-compose build`: Builds or rebuilds services.**

```
docker-compose build
```

- **`docker-compose logs`: Shows logs from services.**

```
docker-compose logs
```

- **`docker-compose exec`: Runs a command in a running service container.**

```
docker-compose exec <service_name> <command>
docker-compose exec web /bin/bash
```

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 install --no-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"]
```

Useful Tips

- **Remove all stopped containers:**

```
docker container prune
```

- **Remove all unused images:**

```
docker image prune
```

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

```
docker system prune
```

- **View the Docker system's disk usage**:

```
docker system df
```