Docker Basics

- **Check Docker Installation:**

  ```
  docker version
  ```

- **Download an Image:**

  ```
  docker pull <IMAGE NAME>
  ```

- **Run a Container:**

  ```
  docker run <IMAGE NAME>
  ```

- **Stop a Running Container:**

  ```
  docker stop <CONTAINER ID or NAME>
  ```

- **Start a Stopped Container:**

  ```
  docker start <CONTAINER ID or NAME>
  ```

- **Remove a Container:**

  ```
  docker rm <CONTAINER ID or NAME>
  ```

Dockerfile Explained

A `Dockerfile` is a script of commands that automates the building of Docker images:

- **FROM**: Specifies the base image.
- **RUN**: Executes commands in the container.
- **COPY**: Copies files from the host to the container.
- **CMD**: Sets the default command to run when the container starts.
- **EXPOSE**: Indicates the ports on which a container listens.
- **ENV**: Sets environment variables.

Managing Images and Containers

- **List All Images:**

  ```
  docker image ls
  ```

- **Remove an Image:**

  ```
  docker image rm <IMAGE NAME>
  ```

- **Build an image from Dockerfile:**

  ```
  docker built -t <IMAGE NAME> <PATH>
  ```

Managing Volumes

- **Create a Volume:**

  ```
  docker volume create <VOLUME NAME>
  ```

- **List Volumes:**

  ```
  docker volume list
  ```

- **Remove a Volume:**

  ```
  docker volume rm <NAME>
  ```

Advanced Docker Command

```
docker run --name <NAME> -v <VOLUME NAME>:<PATH IN
   CONTAINER> -p <PC PORT>:<CONTAINER PORT> -e <VAR>=<
   VALUE> <IMAGE NAME>
```

- **-p:** to define port to expose

  ```
  <port on our pc>:<port on docker container>
  ```

- **-e:** create environment variables

  ```
  <variable name>:<value>
  ```

- **-v:** define a volume to use

  ```
  <volume name>:<path in container>
  ```

Docker Compose and docker-compose.yml Explained

`docker-compose.yml` specifies how to set up and run multi-container Docker applications:

```
version: '3.8'
services:
    name of container:
        image: <IMAGE NAME>
        ports:
          - "<PORT ON HOST>:<PORT ON CONTAINER>"
        environment:
          - <ENV_VAR>=<VALUE>

    name of second container:
        build:
            context: FOLDER
            dockerfile: Dockerfile
        container_name: NAME
        restart: always
        volumes:
            - volume1:/folder/in/container
        depends-on:
            - name of container

    volumes:
      volume1:
        name: volume1
      volume2:

    networks:
      default:
        driver: bridge
```

Docker Compose Commands

- **Build Containers:**

  ```
  docker-compose build
  ```

- **Run Containers:**

```
docker-compose up
```

- **Stop and Remove Containers:**

```
docker-compose down
```