

Introduction to Docker & Docker Compose

Docker

Helps reduce the size of development and provide smaller footprints

Helps teams work seamlessly across applications

Can be deployed anywhere

Lightweight and scalable

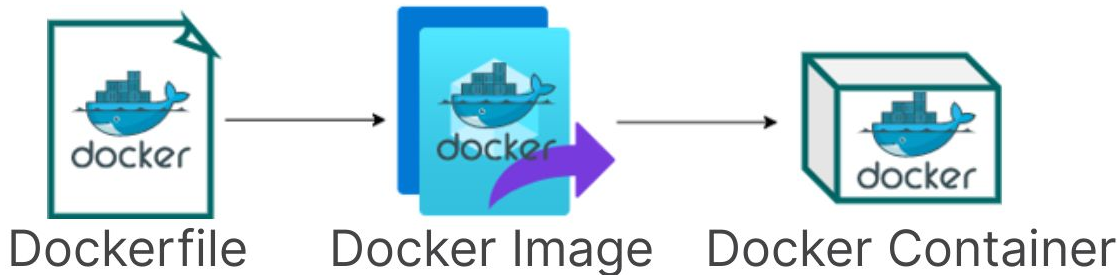
Components include:

Docker Engine

Docker Hub

Docker Compose

Overview of Docker process



Create a **Dockerfile**

Use command **docker build** to build Docker image from Dockerfile

Use command **docker run** to start Docker container and run application

The Dockerfile

```
# syntax=docker/dockerfile:1
FROM node:12-alpine
RUN apk add --no-cache python g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
```

Text file named **Dockerfile** with no extension, containing commands

The Dockerfile

```
# syntax=docker/dockerfile:1
FROM node:12-alpine
RUN apk add --no-cache python g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
```

FROM – indicates base image

The Dockerfile

```
# syntax=docker/dockerfile:1
FROM node:12-alpine
RUN apk add --no-cache python g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
```

RUN – executes commands on top of current image layer

The Dockerfile

```
# syntax=docker/dockerfile:1
FROM node:12-alpine
RUN apk add --no-cache python g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
```

WORKDIR – sets working directory

The Dockerfile

```
# syntax=docker/dockerfile:1
FROM node:12-alpine
RUN apk add --no-cache python g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
```

COPY – copies file/directory from host machine to Docker image

The Dockerfile

```
# syntax=docker/dockerfile:1
FROM node:12-alpine
RUN apk add --no-cache python g++ make
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
```

CMD – executes command in running container, only one CMD per file

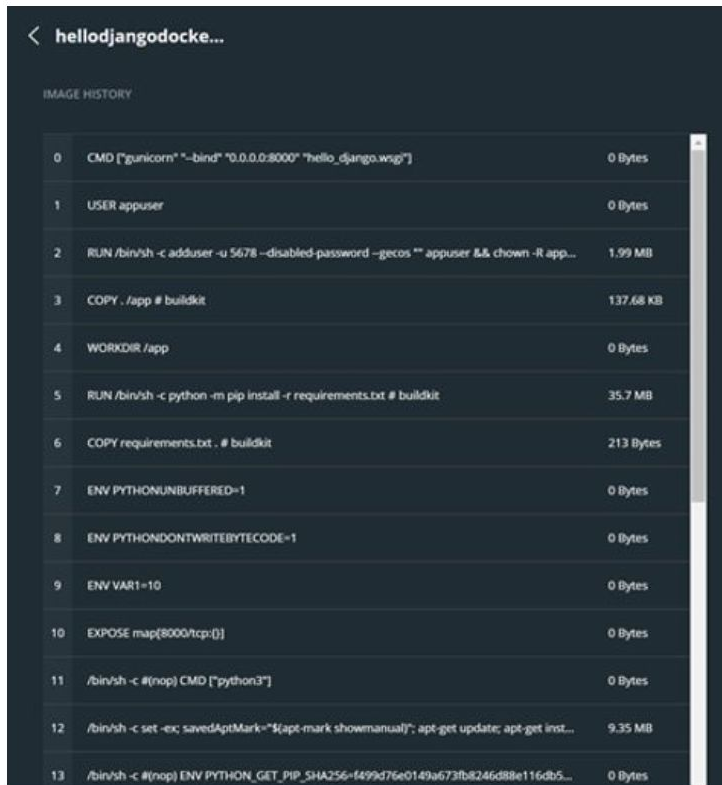
Docker images

Read-only templates

Provide instructions for a Docker image

Collections of immutable layers

Can be built on top of other images



The screenshot shows the Docker image history for a container named 'hellodjangodocke...'. The history is displayed as a table with columns for the layer index, the command used to create the layer, and the size of the layer. The layers are listed from 0 to 13, showing the build process from the base image to the final application.

INDEX	CMD	SIZE
0	CMD ["gunicorn" "--bind" "0.0.0.0:8000" "hello_django.wsg"]	0 Bytes
1	USER appuser	0 Bytes
2	RUN /bin/sh -c adduser -u 5678 --disabled-password --gecos "" appuser && chown -R app...	1.99 MB
3	COPY . /app # buildkit	137.68 KB
4	WORKDIR /app	0 Bytes
5	RUN /bin/sh -c python -m pip install -r requirements.txt # buildkit	35.7 MB
6	COPY requirements.txt . # buildkit	213 Bytes
7	ENV PYTHONUNBUFFERED=1	0 Bytes
8	ENV PYTHONDONTWRITEBYTECODE=1	0 Bytes
9	ENV VAR1=10	0 Bytes
10	EXPOSE map[8000/tcp:[]]	0 Bytes
11	/bin/sh -c #(nop) CMD ["python3"]	0 Bytes
12	/bin/sh -c set -ex; savedAptMark="\$(apt-mark showmanual)"; apt-get update; apt-get inst...	9.35 MB
13	/bin/sh -c #(nop) ENV PYTHON_GET_PIP_SHA256=f499d76e0149a673fb8246d88e116db5...	0 Bytes

Docker containers

Created from Docker images

Running instances of the Docker images

Contain everything needed

Docker Compose

Tool for running multi-container
Docker applications

Uses a YAML file (with extension .yaml)
to define services and dependencies

YAML - data serialization format
commonly used for configuration files

Named docker-compose.yaml

```
version: '3.7'
services:
  web:
    build:
      context: ..
      dockerfile: df.dev.Dockerfile
    environment:
      - DB_URI=postgres://postgres:postgres@db/idea_box
    command: bash -c "flask migrate && flask run -p 5000 -h 0.0.0.0"
    ports:
      - 5000:5000
    links:
      - db
    volumes:
      - ../:/code
  db:
    image: postgres:10.1
    environment:
      POSTGRES_DB: idea_box
```