55.15Jun. Containers(Python Flask, Java Spring)

```
# syntax=docker/dockerfile:1.4
FROM --platform=$BUILDPLATFORM python:3.10-alpine AS builder

WORKDIR /app

COPY requirements.txt /app
RUN --mount=type=cache,target=/root/.cache/pip \
    pip3 install -r requirements.txt

COPY . /app

ENTRYPOINT ["python3"]
CMD ["app.py"]

FROM builder as dev-envs

RUN <<EOF
apk update
apk add git
EOF

RUN <<EOF
addgroup -S docker
adduser -S --shell /bin/bash --ingroup docker vscode
EOF
# install Docker tools (cli, buildx, compose)
COPY --from=gloursdocker/docker / /
```
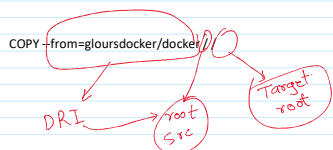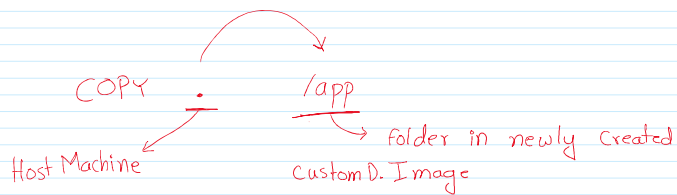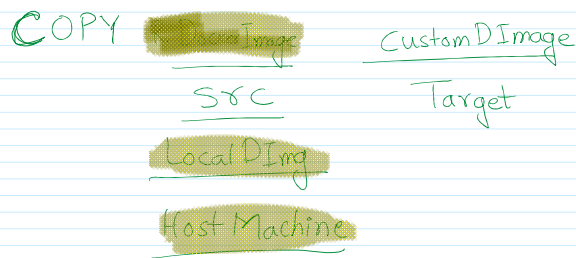
```
RUN --mount=type=cache,target=/root/.cache/pip \
    pip3 install -r requirements.txt

RUN mkdir abc
```

Programming: Java (Javac & Java) + JDBC

Database Server: Postgresql Server.
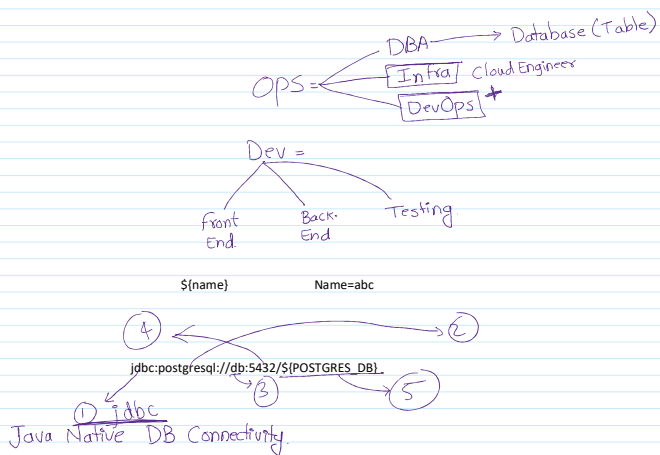Build Tool: Apache Maven

Web Framework: Spring Boot
    ◦ JPA ◦ Spring Boot Test.
    ◦ Web ◦

Containers: - Docker
    - Dockerfile
    - Docker Compose.
    - Docker Volume.

DB Framework: ◦ Hibernate.
      - SQL Dialect

DBA ────→ Database (Table)
OPS ←── [Intra] Cloud Engineer
   [DevOps] ✦

Dev =
 Front Back. Testing
 End. End

${name}   Name=abc

④ ←──────────→ ②

jdbc:postgresql://db:5432/${POSTGRES_DB}

① jdbc ──→ ③ ⑤
Java Native DB Connectivity.

② postgresql = Database Server.

③ db = Database Server Name
  → localhost
  → public IP
  → private IP
  Domain/sub domain

④ 5432 ──→ PgSQL port

  3306 ──→ MySQL port

      ──→ ✷
MS SQL TCP = 1433
Server TCP = 1434
ports TCP = 4022
   TCP = 135
   , UND 1434

Server ports { TCP = 135
              UDP = 1434

⑤ DBName

Database
   ╱   ╲
Storage   In-Memory
                ↘
                  SQLite, h2

Syntax:-
   ConnectMethod : DBServerName :// DBServerIP : PORT / DBName

                                        jdbc: h2 : file : C :/ abc/ db1

Example
   jdbc : mysql :// 53.22.12.11 : 3306 / ABC
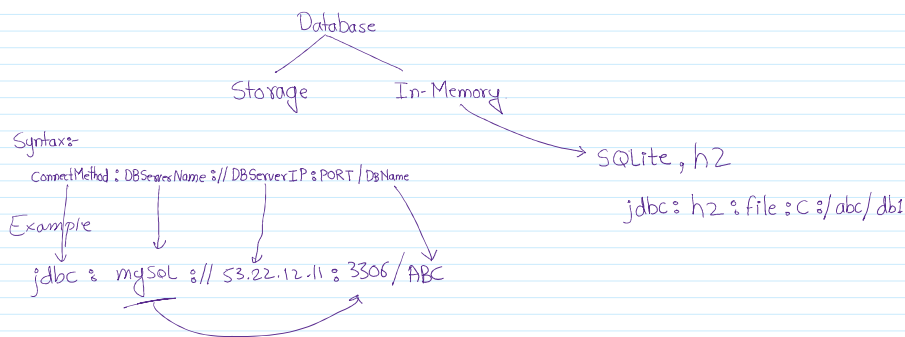
Vendors
        ╱→ MySQL
   OS ─┤
        ╲→ postgreSQL

         ╱→ Azure SQL
   MS ──┤
         ╲→ MS SQL Server

spring.jpa.hibernate.ddl-auto=none

   1] none - No change

   2] update - Change structure of Tables in database

   3] Create - Create every time, not drop on close ──→ Garbage Collector.

   4] create-drop ──────── drop on close   ✗

Docker Compose

   • Services
        ✓ backend: ────────→ 8080
        ✓ db:
                          ↘
   5432 ←──────────   Docker
                            Object
   • Volumes: ↗
        db-data: ────────
   • secrets: ↗
        ═══
   • networks: ↗

Socket based Comm

○ networks: