Java ⟶ Maven.

Python ⟶ Flask.

Node

EntryPoint   Python.
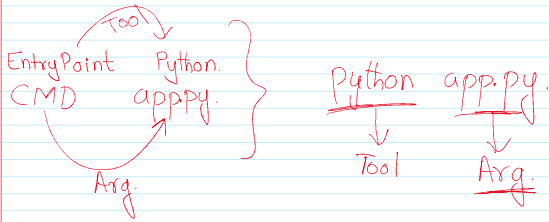CMD   apppy.
          Arg.

Python app.py.
  ↓   ↓
Tool   Arg.

Apach 2 & MongoDB

Entrypoint ["runBoth.sh"]

∘ run apache2
∘ run mongoDB

    ADD Name1 value1
    ADD Name2 value2

Entrypoint ["main.sh"]

① CMD / EntryPoint
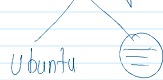
② EntryPoint
        CMD
        ADD

③ EntryPoint ["Tool1", "A1", "A3"]
                      Arg.

① Create DockerFile.
    from Ubuntu.
    CMD Sleep 10.
         Tool   Argument
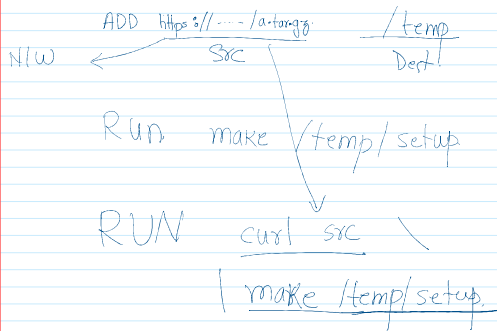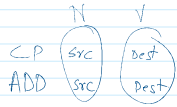
② Build
        from Dockerfile ⟶ Local Images
                              Ubuntu ⊜

③ Run

    Custom Docker Image ⟶ Container.

FROM ubuntu
ENTRYPOINT sleep
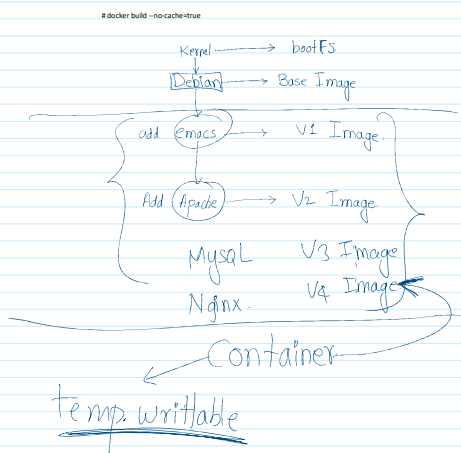
cat > Dockerfile
FROM ubuntu

ENTRYPOINT ["sleep"]
CMD ["10"]

docker build -t custom_sleep .

docker run custom_sleep
docker run custom_sleep 50

$ at now + 1 hour

$ sleep 60m && ls

N   V
CP   (Src)   (Dest)
ADD   (Src)   (Dest)

ADD https://----/a.tar.gz.   /temp
          Src   Dest!
N/W ←

RUN   make /temp/setup

RUN   curl src \
       | make /temp/setup.

curl vs Wget

docker build --no-cache=true

Kernel ————→ bootFS
Debian ————→ Base Image
add (emacs) ——→ V1 Image.
Add (Apache) ——→ V2 Image.
MysaL   V3 Image
Nginx.   V4 Image

Container

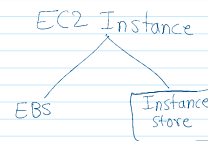temp. writable

① .dockerignore
        |
        ↓
System Hidden

② Immutable & ephemeral

    Container should be

③ Minimize the Number of Layers

    RUN   apt update && \
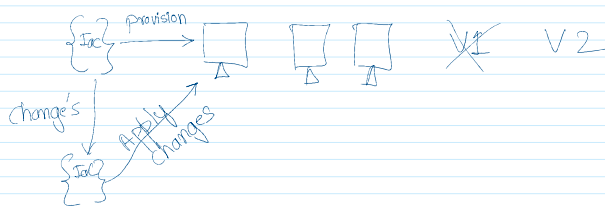   apt install jdk8 -y.

EC2 Instance
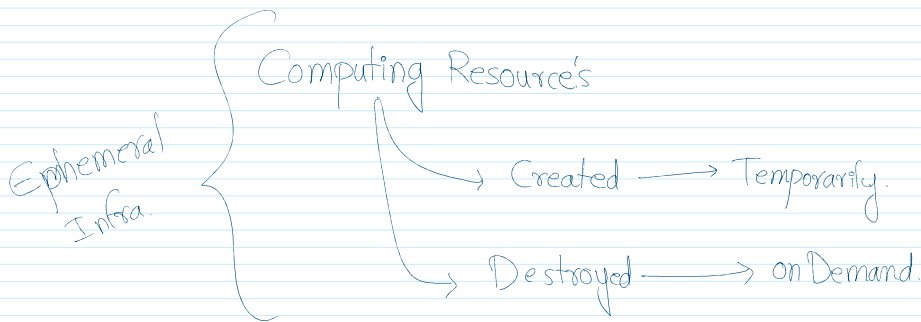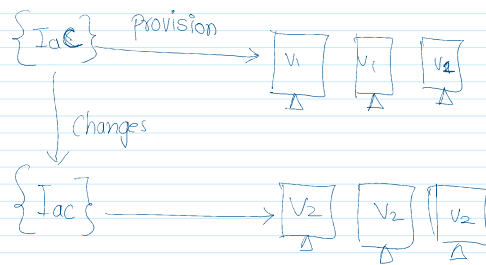EBS   Instance store

④ Avoid Unecessary Packages
- Reduces Complexity.
- Dependencies should need manage.
- File Sizes
- Number of Build time.

SRE

---

* Mutable Infrastructure
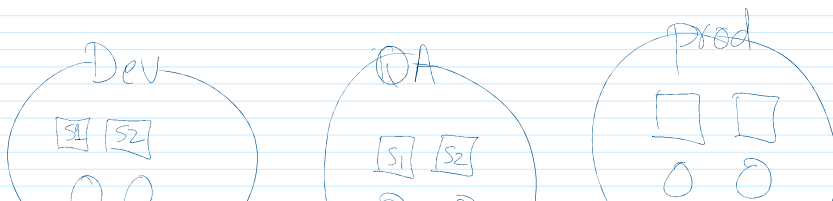

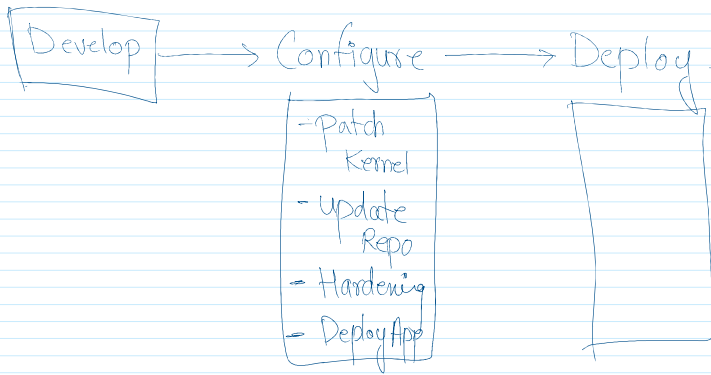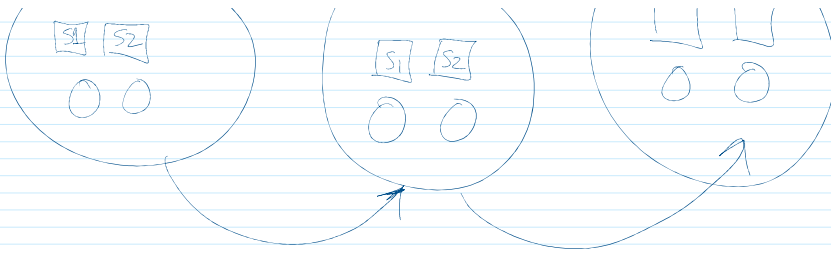* Immutable Infra

---

Ephemeral Infra. { Computing Resources

→ Created ──→ Temporarily.

→ Destroyed ──────→ on Demand.

Im Infra

S1 S2
○ ○

S1 S2
○ ○

⊔ ⊔
○ ○

Develop ⟶ Configure ⟶ Deploy

- Patch Kernel
- Update Repo
- Hardening
- Deploy App

Build Startegies of Docker Image

Dockerfile

Base + Init → Compile → Linting → testing → Release
C_1 C_2 C_3 C_4 C_5

FROM Ubuntu As C1
RUN apt install git -y

FROM JDK:8
COPY --from=C1 ---/target/a.jar /Webapps/
                      Src

Java Maven

↳ target/a.jar

```
FROM maven:3.5.2-jdk-9 AS build
COPY src /usr/src/app/src
COPY pom.xml /usr/src/app
RUN mvn -f /usr/src/app/pom.xml clean package

FROM openjdk:9
COPY --from=build /usr/src/app/target/flighttracker-1.0.0-SNAPSHOT.jar /usr/app/flighttracker-1.0.0-SNAPSHOT.jar
EXPOSE 8080
ENTRYPOINT ["java","-jar","/usr/app/flighttracker-1.0.0-SNAPSHOT.jar"]
```

S1

SRC

Jar

S2