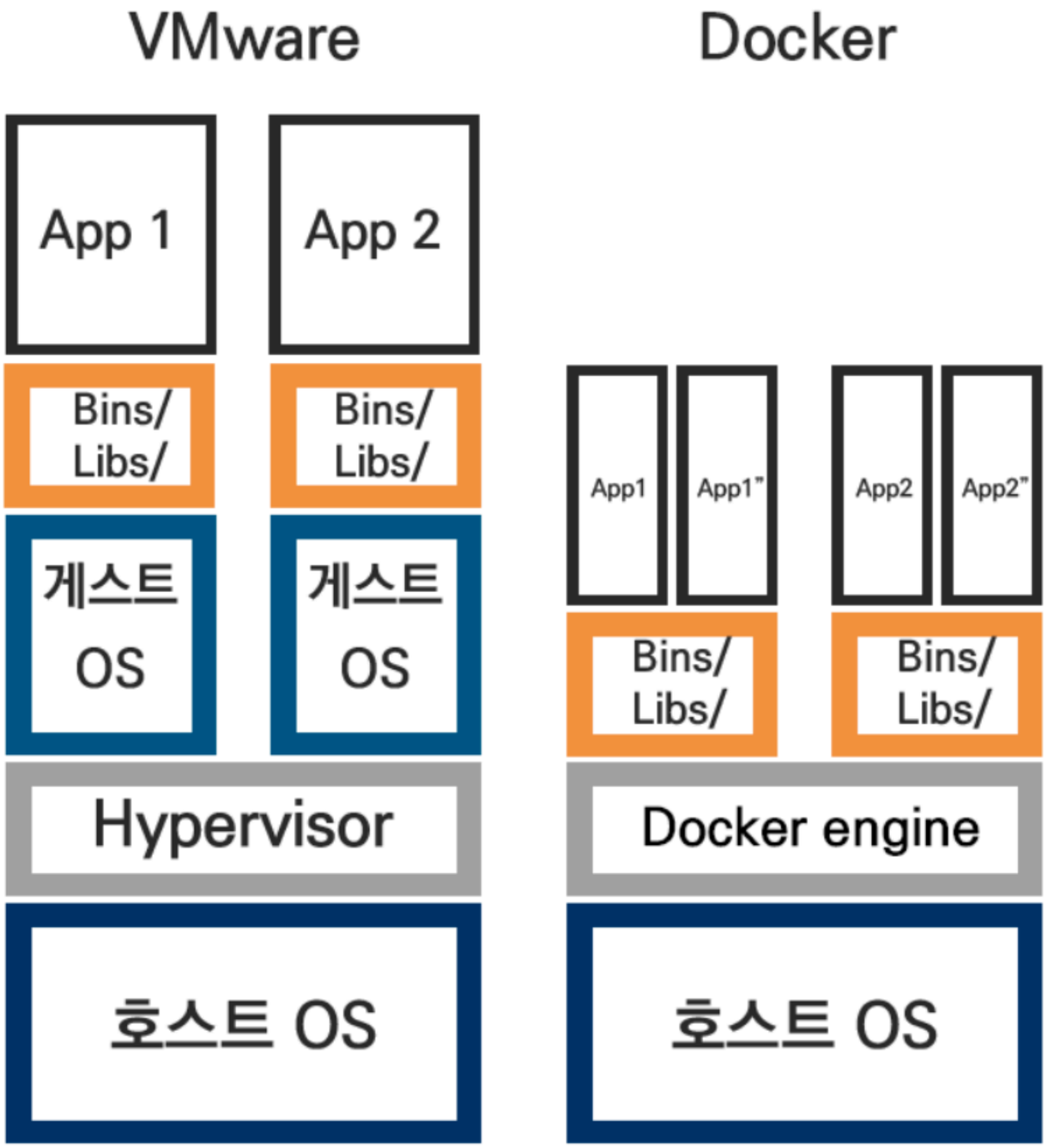
Docker

강희원 && 이재규

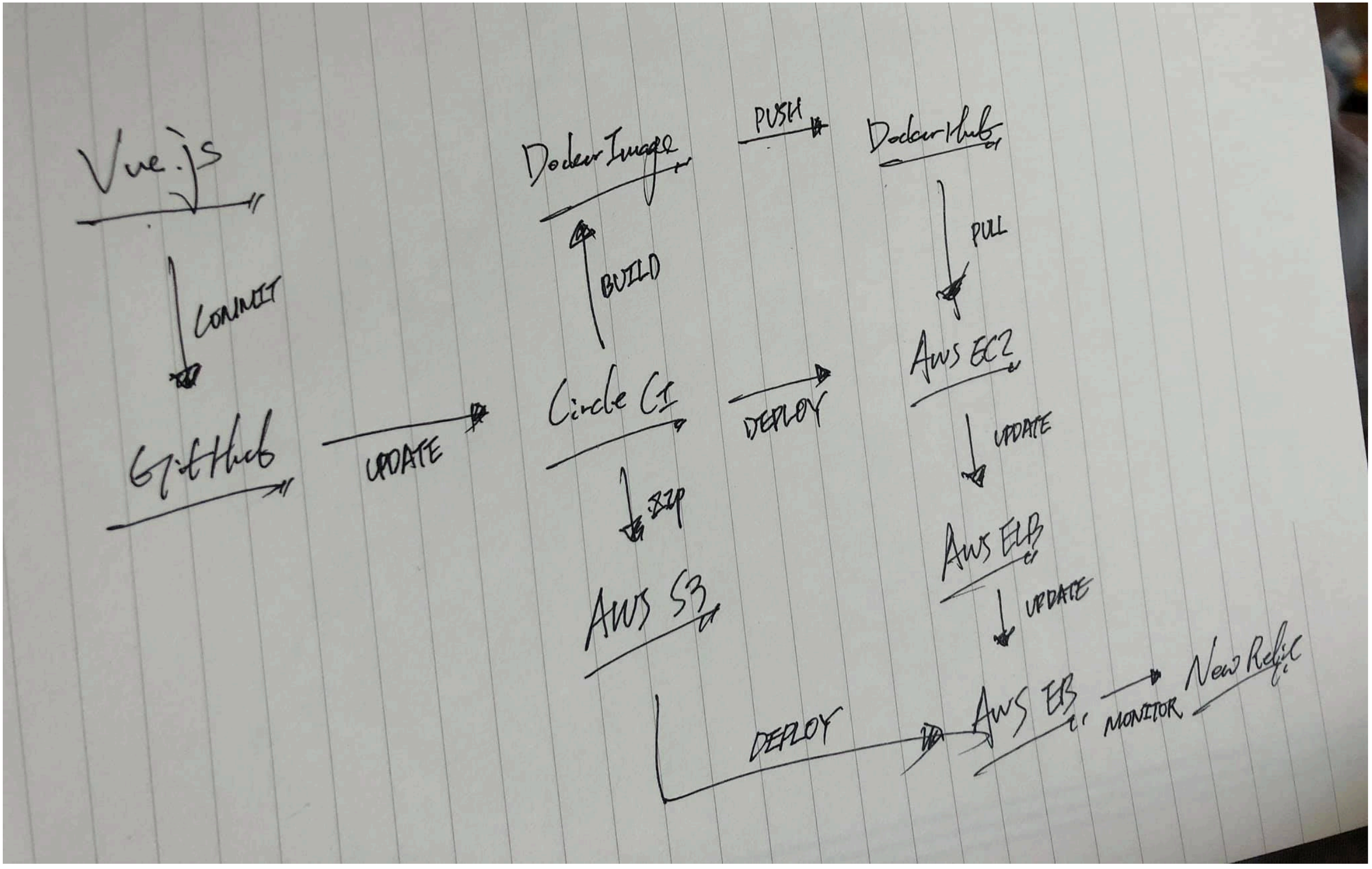
도커 구조

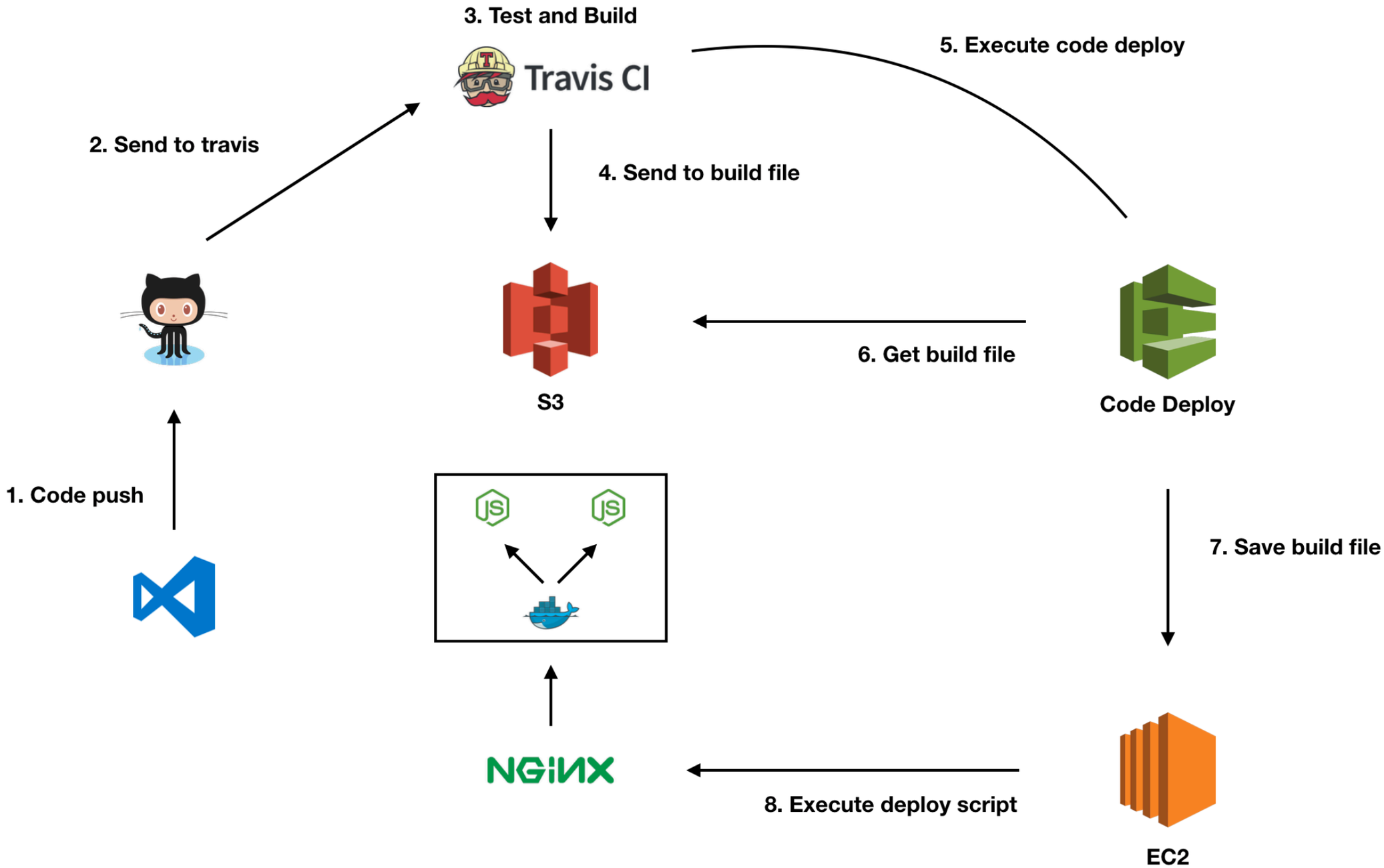

VMware
Docker
App 1
App 2
Bins/
Libs/
Bins/
Libs/
게스트
OS
게스트
OS
Hypervisor
호스트 OS
App1
App1"
App2
App2"
Bins/
Libs/
Bins/
Libs/
Docker engine
호스트 OS

도커 장점

- 배포환경과 테스트환경, 개발환경을 분리할 수 있다
  - 배포환경의 경우 EC2-RDS-~~~~
  - 테스트환경의 경우 EC2안에 몽땅
  - 개발환경은 로컬에 몽땅

도커 인프라 구축 사례


Vue.js
COMMIT
GitHub
UPDATE
Docker Image
BUILD
Circle CI
ZIP
AWS S3
PUSH
Docker Hub
PULL
DEPLOY
AWS EC2
UPDATE
AWS ELB
UPDATE
DEPLOY
AWS EB
MONITOR
New Relic


3. Test and Build
Travis CI
5. Execute code deploy
2. Send to travis
4. Send to build file
S3
Code Deploy
6. Get build file
1. Code push
7. Save build file
NGINX
8. Execute deploy script
EC2

도커 실습

- Dockerfile을 활용해 노드를 띄워보자

- https://github.com/kang-heewon/RM_Memo.git

- https://rm-memo.herokuapp.com

rm-memo.herokuapp.com

RM-MEMO

익명 (작성일 - 2019-03-06)

아아 메모 테스트

익명 (작성일 - 2019-03-23)

오랫만에 들어와보네

메모 작성

Write a comment...

댓글 작성

메모장 띄워보기

- git clone https://github.com/kang-heewon/RM_Memo
- git reset --hard f5f1484401b0981ae992c5a9942c0fd7587a1678
- npm install
- npm start

도커로 띄워보자

- git clone https://github.com/kang-heewon/RM_Memo
- git reset --hard f5f1484401b0981ae992c5a9942c0fd7587a1678
- touch Dockerfile
- vi Dockerfile

Dockerfile

```
FROM node:11.12
ADD ./ /var/www
WORKDIR /var/www
RUN npm install
CMD npm start
```

Dockerfile 실습

- docker build -t mynote ./

```
kang-heewon@ganghuiwons-MacBook-Pro ~/Project/Toyproject/RM_Memo master • docker build -t app .
Sending build context to Docker daemon    2.3MB
Step 1/5 : FROM node:11.12
 ---> 2698faaff1ee
Step 2/5 : ADD ./ /var/www
 ---> 79b752e2cde1
Step 3/5 : WORKDIR /var/www
 ---> Running in 841578220a1f
Removing intermediate container 841578220a1f
 ---> 3a53cdf8e0bc
Step 4/5 : RUN npm install
 ---> Running in aa4e5cb2d6f1

> node-sass@4.11.0 install /var/www/node_modules/node-sass
> node scripts/install.js


Downloading binary from https://github.com/sass/node-sass/releases/download/v4.11.0/linux-x64-67_binding.node
Download complete
Binary saved to /var/www/node_modules/node-sass/vendor/linux-x64-67/binding.node
Caching binary to /root/.npm/node-sass/4.11.0/linux-x64-67_binding.node

> node-sass@4.11.0 postinstall /var/www/node_modules/node-sass
> node scripts/build.js

Binary found at /var/www/node_modules/node-sass/vendor/linux-x64-67/binding.node
Testing binary
Binary is fine
npm WARN rm_memo@0.1.0 No description
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_modules/chokidar/node_modules/fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.7: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"})

added 2189 packages from 832 contributors and audited 37061 packages in 43.392s
found 63 low severity vulnerabilities
  run `npm audit fix` to fix them, or `npm audit` for details
Removing intermediate container aa4e5cb2d6f1
 ---> ff690a1c6a30
Step 5/5 : CMD npm start
 ---> Running in 6b7e2a371ad6
Removing intermediate container 6b7e2a371ad6
 ---> 4861d7eef1ee
Successfully built 4861d7eef1ee
Successfully tagged app:latest
```

Dockerfile 실습

- docker run -p 3000:8080 app

```
kang-heewon@ganghuiwons-MacBook-Pro ~/Project/Toyproject/RM_Memo master • docker run -p 3000:8080 app

> rm_memo@0.1.0 start /var/www
> react-scripts build && node app.js

Creating an optimized production build...
Compiled with warnings.

./src/components/MemoInfo.js
  Line 9:  Useless constructor  no-useless-constructor

Search for the keywords to learn more about each warning.
To ignore, add // eslint-disable-next-line to the line before.

File sizes after gzip:

  50.52 KB (+2.1 KB)  build/static/js/2.bb2f0842.chunk.js
  1.35 KB (+1 B)      build/static/js/main.8cf08e88.chunk.js
  979 B (-1 B)        build/static/css/main.18d1c015.chunk.css
  762 B               build/static/js/runtime~main.a8a9905a.js
  530 B (-2 B)        build/static/css/2.ee73a6db.chunk.css

The project was built assuming it is hosted at the server root.
You can control this with the homepage field in your package.json.
For example, add this to build it for GitHub Pages:

  "homepage" : "http://myname.github.io/myapp",

The build folder is ready to be deployed.
You may serve it with a static server:

  npm install -g serve
  serve -s build

Find out more about deployment here:

  https://bit.ly/CRA-deploy

App listening on port 8080!
DataBase is connected
```

Dockerfile 실습

- docker exec -it *container_id or names*

```
kang-heewon@ganghuiwons-MacBook-Pro ~ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                    NAMES
3ea4df3d54ce        app                 "/bin/sh -c 'npm sta…"   10 minutes ago      Up 10 minutes       0.0.0.0:3000->8080/tcp   eloquent_goldbe
rg
kang-heewon@ganghuiwons-MacBook-Pro ~ docker exec -it eloquent_goldberg /bin/bash
root@3ea4df3d54ce:/var/www#
```

- 탈출은 exit

Dockerfile 실습

- docker run -d -p 3000:8080 app

참고자료

- https://slides.com/byunkyuhyun/ec2-docker-docker-compose/
- https://velog.io/@jeff0720/Travis-CI-AWS-CodeDeploy-Docker-로-배포-자동화-및-무중단-배포-환경-구축하기