

Docker부터 Jenkins까지

2024.01.24 광주 실습코치 서정빈



코치 세션의 내용은 모두 정답인 방법이 아닙니다.

실습코치들의 경험을 바탕으로

교육생분들의 더 원활한 프로젝트를 위해 제공해드립니다.



정신을 차려보니 어느새 3주차?!



1월

일	월	화	수	목	금	토
	1 신정	2 방학	3	4	5	6
			공통 PJT 부트캠프			싸피레이스 결선
7	8	9	10	11	12 평가1	13
	공통 PJT 1주차 (Sub-PJT I)					
14	15	16	17	18	19	20
	공통 PJT 2주차 (Sub-PJT II)					
21	22	23	24	25	26 평가2	27
	공통 PJT 3주차 (Sub-PJT II)					
28	29	30	31			
	공통 4주차 (Sub-PJT III)					



서버와는 많이 친해지셨나요?

서버와 첫 만남



서버와 첫 만남



Services ▾

Search for services, features, marketplace products, and docs [Alt+S]

New EC2 Experience
Tell us what you think

EC2 Dashboard **New**

Events

Tags

Limits

▼ Instances

Instances **New**

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Scheduled Instances

Capacity Reservations

▼ Images

AMIs

Instances (1) **Info**

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Ubuntu-GUI-...		Running		Initializing	No alarms	

Select an instance above

© 2008 - 2021, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. [Privacy Policy](#) [Terms of Use](#) [Cookie preferences](#)

서버와 첫 만남



```
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1018-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management:   https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

System information as of Fri Jan 19 02:55:15 UTC 2024

System load:                Processes:
Usage of /:                  Users logged in:
Memory usage:               IPv4 address for docker0:
Swap usage:                 IPv4 address for eth0:

* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

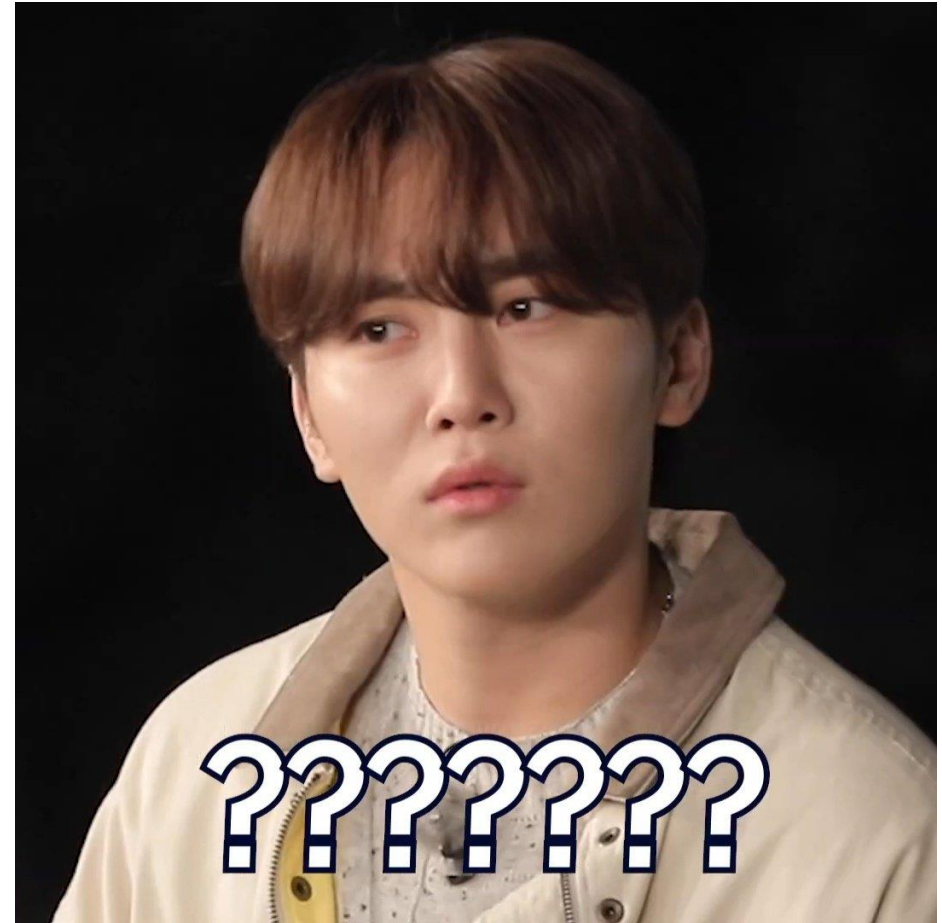
Expanded Security Maintenance for Applications is not enabled.

50 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

3 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

*** System restart required ***
Last login: Fri Jan  5 03:13:48 2024 from
ubuntu@           :~$
```



서버와 첫 만남



서버 배포 설정 과정



EC2 설정

NginX&SSL

Docker 설정

배포

Jenkins 설정

서버 배포 설정 과정



EC2 설정

NginX&SSL

Docker 설정

배포

Jenkins 설정



✓ 터미널 프로그램 사용추천

- MobaXterm
- Putty
- Termius
- PowerShell

✓ `ssh -i {pem키}.pem ubuntu@i10{팀코드}.p.ssafy.io`



✓ 터미널 프로그램 사용추천

- MobaXterm
- Putty
- Termius
- PowerShell

✓ **ssh -i {pem키}.pem ubuntu@i10{팀코드}.p.ssafy.io**



✓ 서버 시간 변경

- **`sudo timedatectl set-timezone Asia/Seoul`**
- `date` 입력 후 현재 시간이 맞게 나오는지 확인



✓ 서버 시간 변경

➤ `sudo timedatectl set-timezone Asia/Seoul`

➤ **date** 입력 후 현재 시간이 맞게 나오는지 확인



✓미러 서버 변경

➤ **`sudo vi /etc/apt/sources.list`**

➤ `:%s/kr.archive.ubuntu.com/mirror.kakao.com/`



✓미러 서버 변경

➤ `sudo vi /etc/apt/sources.list`

➤ `:%s/kr.archive.ubuntu.com/mirror.kakao.com/`



✓ 패키지 업데이트 및 업그레이드

➤ **sudo apt update**

➤ **sudo apt upgrade**



✓ 패키지 업데이트 및 업그레이드

➤ **sudo apt update**

```
Err:11 http://ppa.launchpad.net/certbot/certbot/ubuntu focal Release  
404 Not Found [IP: http://ppa.launchpad.net/certbot/certbot/ubuntu]  
E: The repository 'http://ppa.launchpad.net/certbot/certbot/ubuntu focal Release' does not have a Release file.  
N: Updating from such a repository can't be done securely, and is therefore disabled by default.  
N: See apt-secure(8) manpage for repository creation and user configuration details.
```



✓ 패키지 업데이트 및 업그레이드

➤ **sudo add-apt-repository --remove ppa:certbot/certbot**

```
ubuntu@ . :~$ sudo add-apt-repository --remove ppa:certbot/certbot
The PPA has been DEPRECATED.

To get up to date instructions on how to get certbot for your systems, please see https://certbot.eff.org/docs/install.html.
More info: https://launchpad.net/~certbot/+archive/ubuntu/certbot
Press [ENTER] to continue or Ctrl-c to cancel removing it.
```



✓ 패키지 업데이트 및 업그레이드

➤ `sudo apt update`

➤ **`sudo apt upgrade`**



✓ 가상 메모리 할당

- **free -h**를 입력하여 현재 메모리 용량 확인
- 보통 RAM의 절반정도 크기를 할당([참고링크](#))



✓ 가상 메모리 할당

- free -h를 입력하여 현재 메모리 용량 확인
- 보통 RAM의 절반정도 크기를 할당([참고링크](#))

서버 배포 설정 과정



EC2 설정

NginX&SSL

Docker 설정

배포

Jenkins 설정



1. 방화벽 확인

➤ **sudo ufw status**

2. 방화벽 허용

➤ **sudo ufw allow {포트번호}**

포트번호	역할
22	ssh
80	http
443	https
8989	gerrit

방화벽 설정



1. 방화벽 확인

➤ `sudo ufw status`

2. 방화벽 허용

➤ `sudo ufw allow {포트번호}`

포트번호	역할
22	ssh
80	http
443	https
8989	gerrit



NGINX

- ✓ 오픈소스 웹 서버 프로그램
- ✓ 외에도 Apache, IIS, LiteSpeed 등이 있음.



1. NginX 설치

➤ **`sudo apt install nginx -y`**

2. NginX 상태 확인

➤ `sudo systemctl status nginx`



1. NginX 설치

➤ `sudo apt install nginx -y`

2. NginX 상태 확인

➤ `sudo systemctl status nginx`



```
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since
     Docs: man:nginx(8)
  Main PID:      (nginx)
    Tasks: 3 (limit: 4710)
   Memory: 7.5M
    CGroup: /system.slice/nginx.service
            └─ nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
               └─ nginx: worker process
                  └─ nginx: worker process
```

2. NginX 상태 확인

➤ **sudo systemctl status nginx**



3. NginX 환경설정

➤ **`sudo vi /etc/nginx/site-available/{파일명}.conf`**



✓ 보안 프로토콜

✓ Cerbot을 이용한 letsencrypt 발급



1. letsencrypt 설치

➤ **sudo apt-get install letsencrypt**

2. Certbot 설치

➤ **sudo apt-get install certbot python3-certbot-nginx**



3. Certbot Nginx 연결

➤ **sudo certbot --nginx**

- 이메일 입력
- 약관 동의 - **Y**
- 이메일 수신동의
- 도메인 입력 - **i10{팀코드}.p.ssafy.io**
- http 입력시 리다이렉트 여부 - **2**



3. NginX 환경설정

➤ **`sudo vi /etc/nginx/site-available/{파일명}.conf`**

SSL 등록 후 NginX 확인



```
listen [::]:443 ssl ipv6only=on; # managed by Certbot
listen 443 ssl; # managed by Certbot
ssl_certificate /etc/letsencrypt/live/ /fullchain.pem; # managed by Certbot
ssl_certificate_key /etc/letsencrypt/live/ /privkey.pem; # managed by Certbot
include /etc/letsencrypt/options-ssl-nginx.conf; # managed by Certbot
ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem; # managed by Certbot

}
server {
    if ($host = ) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80 ;
    listen [::]:80 ;
    server_name
    return 404; # managed by Certbot

}
```

서버 배포 설정 과정



EC2 설정

NginX&SSL

Docker 설정

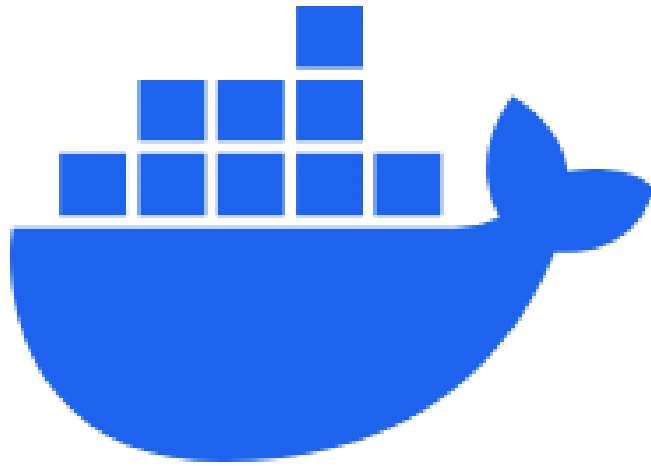
배포

Jenkins 설정



제 PC에서는 됩니다.





docker

- ✓ 컨테이너를 관리할 수 있는 오픈 소스 플랫폼
- ✓ 동일한 실행 환경 보장



✓ 공식문서 확인하기([참고링크](#))

1. Set up Docker's apt repository.

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor --output /etc/apt/keyrings/docker.gpg --no-tty
sudo chmod a+r /etc/apt/keyrings/docker.gpg

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >/dev/null
sudo apt-get update
```

2. Install the Docker packages.

Latest Specific version

To install the latest version, run:

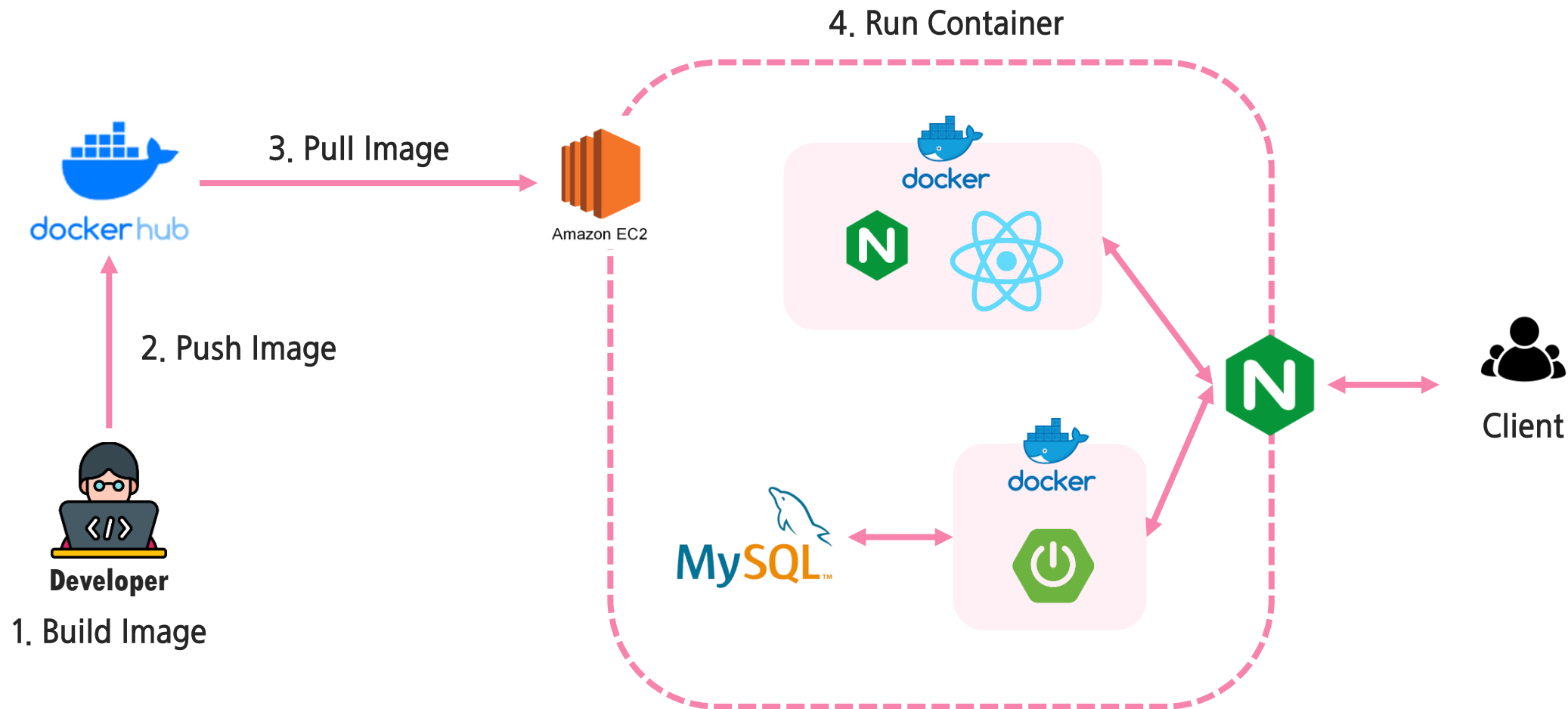
```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin dockershim
```

3. Verify that the Docker Engine installation is successful by running the hello-world image.

```
$ sudo docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits.

서버 배포 흐름





- ✓ Docker Image를 모아두는 저장소
- ✓ push, pull을 통해 이미지 업로드 및 불러오기 가능

Dockerfile 작성하기(참고링크)



- **FROM** 베이스 이미지
- **WORKDIR** 작업 디렉토리
- **RUN** 빌드 명령 실행
- **COPY** 복사
- **EXPOSE** 포트 할당
- **CMD** 컨테이너 생성 및 실행

Dockerfile

> Dockerfile

```
1 # nginx 이미지를 사용합니다. 뒤에 tag가 없으면 latest 를 사용합니다.
2 FROM nginx
3
4 # work dir 고정
5 WORKDIR /app
6
7 # work dir 에 build 폴더 생성 /app/build
8 RUN mkdir ./build
9
10 # host pc의 현재경로의 build 폴더를 workdir 의 build 폴더로 복사
11 ADD ./build ./build
12
13 # nginx 의 default.conf 를 삭제
14 RUN rm /etc/nginx/conf.d/default.conf
15
16 # host pc 의 nginx.conf 를 아래 경로에 복사
17 COPY ./nginx.conf /etc/nginx/conf.d
18
19 # 3000 포트 오픈
20 EXPOSE 3000
21
22 # container 실행 시 자동으로 실행할 command. nginx 시작함
23 CMD ["nginx", "-g", "daemon off;"]
```

Docker Image 만들기



1. 프로젝트 빌드하기

2. Image 빌드하기

➤ `docker build -t {도커허브 계정}/{레포지토리}:{태그} .`

3. DockerHub push 하기

➤ `docker push {도커허브 계정}/{레포지토리}:{태그}`

Docker Image 만들기



1. 프로젝트 빌드하기

2. Image 빌드하기

➤ **docker build -t {도커허브 계정}/{레포지토리}:{태그} .**

3. DockerHub push 하기

➤ **docker push {도커허브 계정}/{레포지토리}:{태그}**

Docker Image 만들기



1. 프로젝트 빌드하기

2. Image 빌드하기

➤ `docker build -t {도커허브 계정}/{레포지토리}:{태그} .`

3. DockerHub push 하기

➤ `docker push {도커허브 계정}/{레포지토리}:{태그}`

Docker Image 배포하기



1. DockerHub pull 받기

➤ **sudo docker pull {도커허브 계정}/{레포지토리}:{태그}**

2. 컨테이너 실행(참고링크)

➤ **sudo docker run -d --rm -p {호스트포트}:{컨테이너포트}**
--name {컨테이너이름} {이미지ID}

Docker Image 배포하기



1. DockerHub pull 받기

➤ `docker pull {도커허브 계정}/{레포지토리}:{태그}`

2. 컨테이너 실행([참고링크](#))

➤ `sudo docker run -d --rm -p {호스트포트}:{컨테이너포트}`
`--name {컨테이너이름} {이미지ID}`

Docker 알아 두면 좋은 명령어



1. Docker Image 파일 조회

➤ **sudo docker images**

2. Docker Container 조회

➤ **sudo docker ps**



3. Docker Image 파일 삭제

- **sudo docker rmi {이미지ID}**

4. Docker Container 중단 / 삭제

- **sudo docker stop {컨테이너ID}**
- **sudo docker rm {컨테이너ID}**

서버 배포 설정 과정



EC2 설정

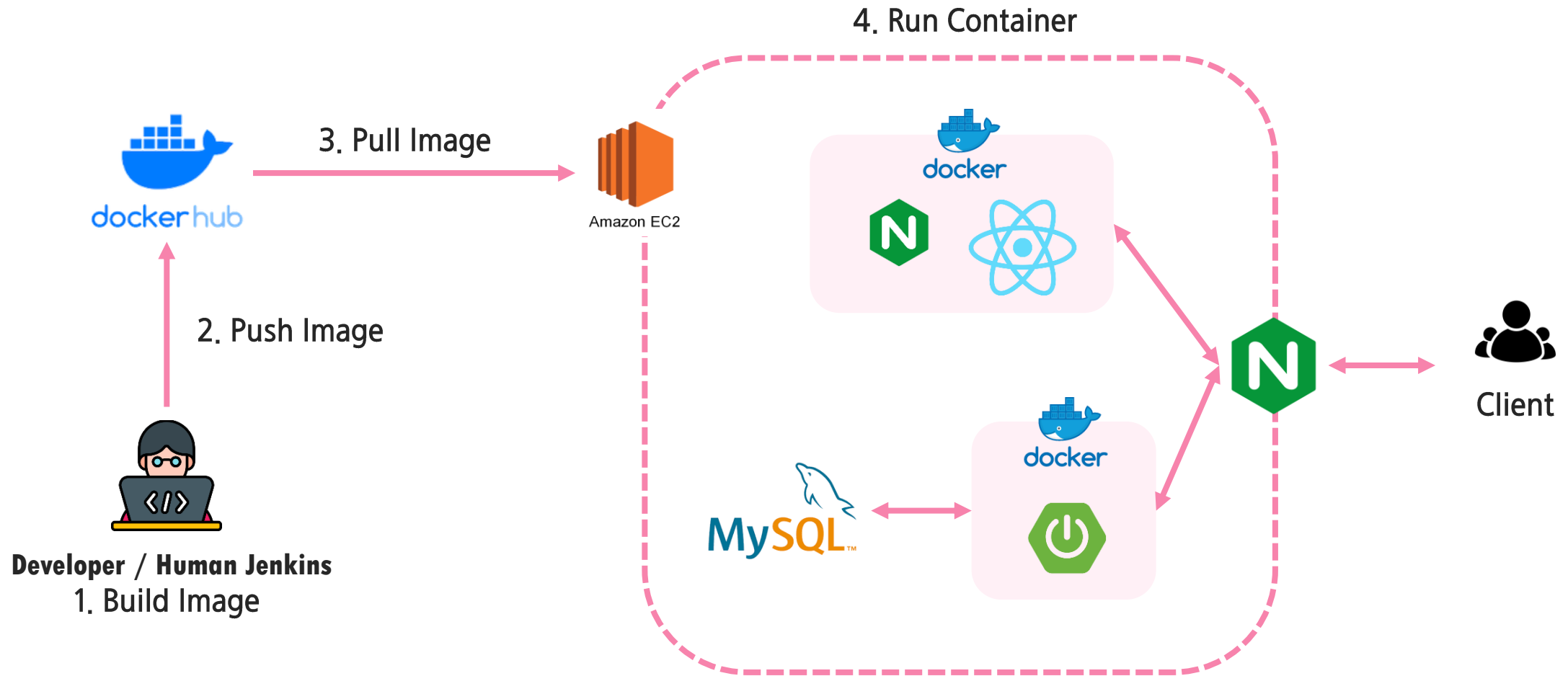
NginX&SSL

Docker 설정

배포

Jenkins 설정

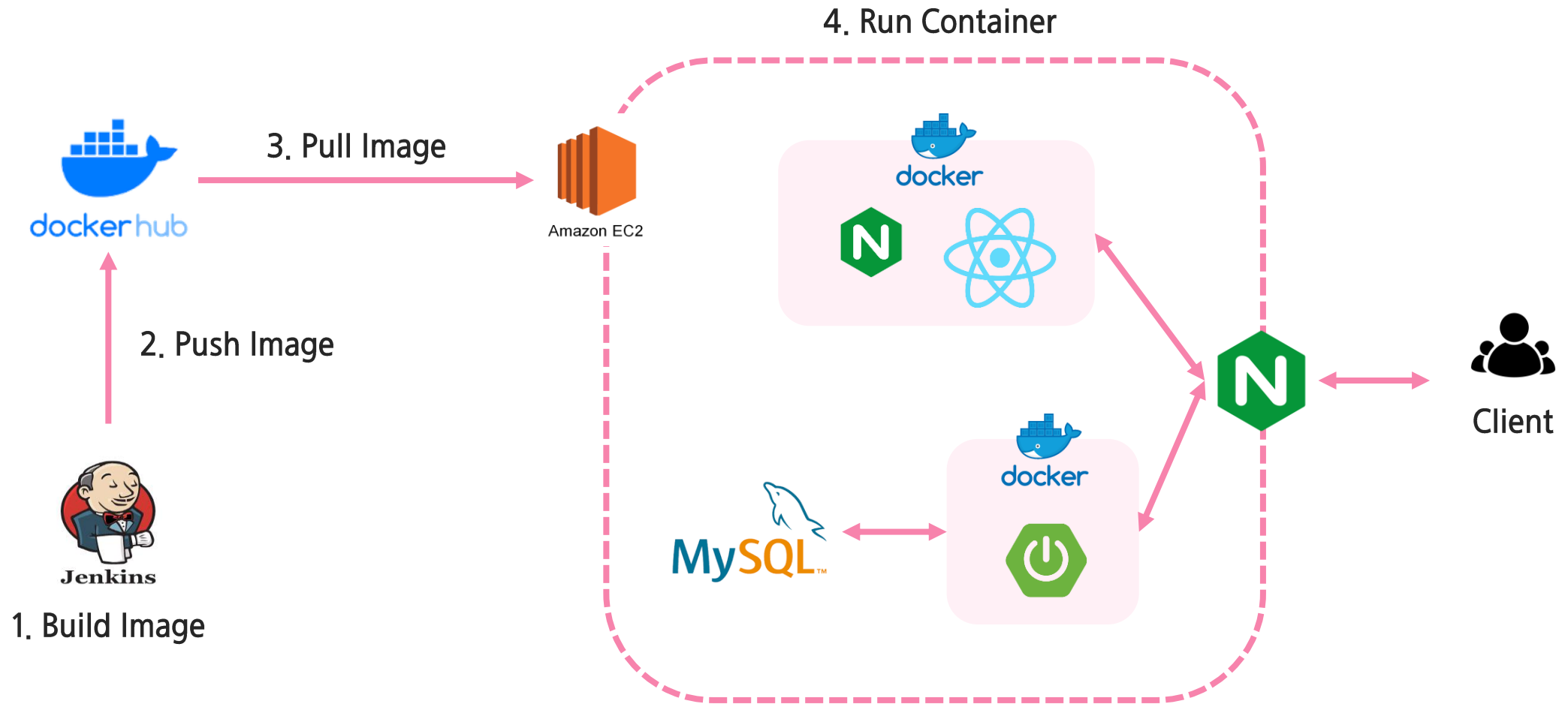
서버 배포 흐름



인간 Jenkins



서버 배포 흐름





Jenkins

- ✓ 오픈 소스 지속적 통합 및 배포(CI/CD) 도구
- ✓ 빌드, 테스트, 배포 등을 자동화 및 관리



✓ SSAFY 가이드 확인하기(SSAFY Git -> Help -> 매뉴얼)

[CI/CD] jenkins 설치 가이드



1.패키지 매니저 설치 방식과 2.Docker 설치 방식 2가지 방식이 있으니 원하는 방식을 참고하여 진행해 주시기 바랍니다.

목차

1. linux 기본 방식(패키지매니저) 설치 시

- 1). jenkins 설치
- 2). 환경 설정 변경(중요)
- 3). 주요 명령어

2. docker 방식 설치 시

- 1). jenkins container 생성 및 구동
- 2). 환경 설정 변경(중요)
- 3). 주요 명령어

※ jenkins 초기 설정 진행(일반적인 과정과 동일)

※ 설치 이후 플러그인 추가 설치에 이상 있는 경우

방법 1. 웹에서 변경

방법 2. 서버 내 환경 설정 파일 수정

본 가이드는 제공된 EC2 서버에서 jenkins 설치 또는 jenkins 플러그인 설치 시 미러 사이트와의 통신 불안정으로 정상 설치가 어려운 경우를 위해 제공합니다.

Ubuntu의 패키지 매니저인 apt 방식으로 jenkins를 설치하는 경우와, docker를 사용하여 jenkins container를 구동하는 경우 각각 설정에 있어 약간의 차이가 있으므로 둘 중 원하는 항목을 참고하시기 바랍니다.

Freestyle vs. Pipeline



1. Freestyle 장점

- 학습 진입장벽 낮음

2. Freestyle 단점

- 파이프라인 커스텀 불가
- 병렬처리 미지원

Freestyle vs. Pipeline



1. Pipeline 장점

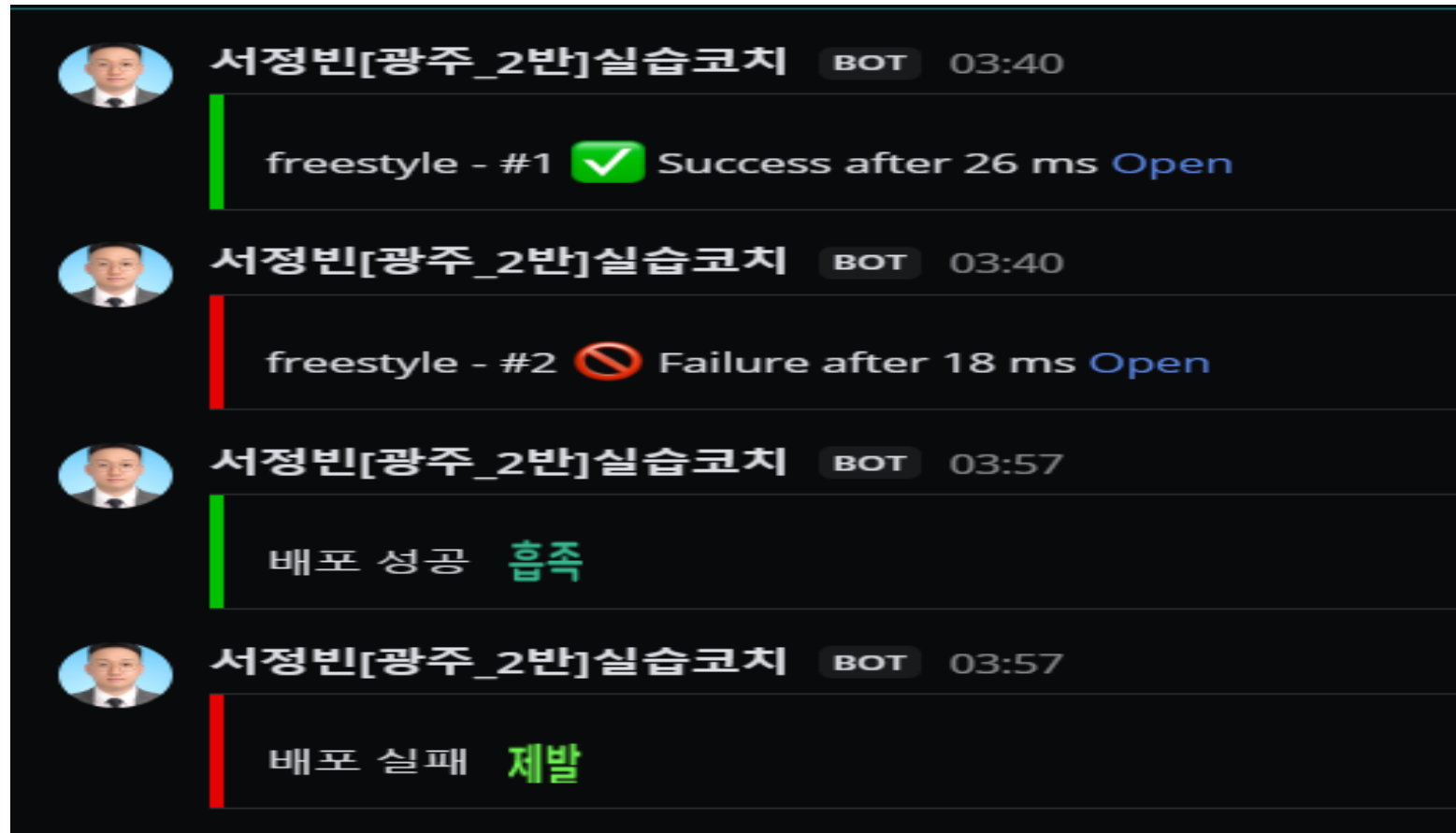
- 파이프라인 커스텀 가능
- 병렬처리 지원

2. Pipeline 단점

- 학습진입 장벽 비교적 높음



✓ Mattermost Notification 플러그인 설치



서버 배포 설정 과정



EC2 설정

NginX&SSL

Docker 설정

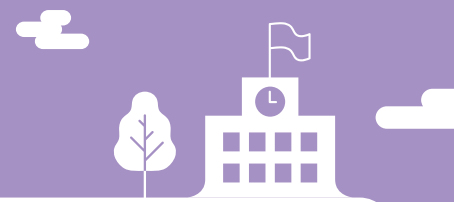
배포

Jenkins 설정



누구나 접근할 수 있다. ⇨ 누구나 해킹할 수 있다.

- ✓ 서버의 Port 번호가 Well-Known 값으로 설정 되어있다. (3306, 6379, 27017)
- ✓ 계정 아이디가 root로 설정되어 있다. (모든 권한 Open)
- ✓ 계정 비밀번호가 1234, ssafy, root, asdf, 1q2w3e4r!으로 설정되어 있다.

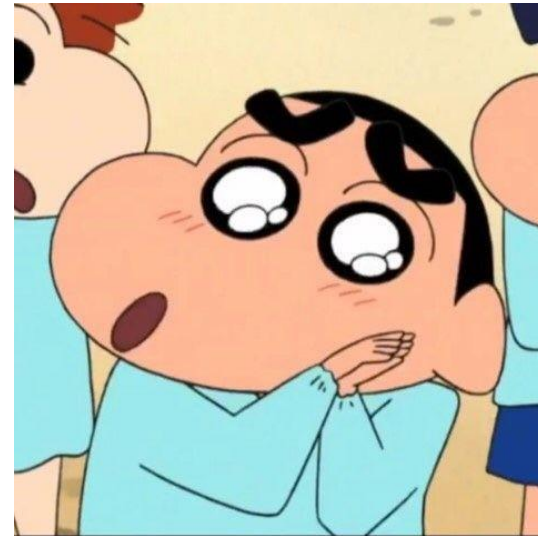


- ✓ 그대로 따라 하시는 것은 권장하지 않습니다
- ✓ 인프라에서 에러가 나는 이유는 정말 다양합니다

똑같이 했는데 안돼요!
뿌앵!



~~에서 ~~설정을 하려고 했는데,
0x00 에러가 납니다!



감사합니다.

궁금한 사항은 각 반 실습코치 또는 저에게 MM 주세요!

