



# 포팅 메뉴얼

[개발 환경](#)

[앱 빌드 메뉴얼](#)

[서버 배포 메뉴얼](#)

[Spring Server 수동 배포](#)

[credential-config.yaml](#)

[EC2 세팅](#)

[서버 시간 변경](#)

[도커 설치](#)

[데이터베이스](#)

[JAVA](#)

[Nginx](#)

[certbot](#)

[젠킨스](#)

[AI 포팅, 학습 메뉴얼](#)

[AWS S3 Bucket](#)

## 개발 환경

- Windows (10, 11)
- 안드로이드
  - Android Studio 2021.3.1
  - 테스트 기기 : Galaxy S 10, Galaxy Z Flip 4
- AI
  - Python
  - Pytorch
  - GPU서버
- 데이터베이스
  - MySQL 5.7
- AWS Ubuntu 20.04 LTS
- 백엔드
  - IntelliJ 2022.1
  - openJDK 11
  - Spring Boot 2.7.5
  - Gradle 7.5.1
- 인프라
  - Jenkins 2.361.2
  - Docker 20.10.12
  - Nginx 1.18.0

## 앱 빌드 메뉴얼

## 서버 배포 메뉴얼

### Spring Server 수동 배포

```
git clone https://lab.ssafy.com/s07-final/S07P31A304.git

cd intaGralServer
# credential-config.yaml을 복사
sudo cp /var/lib/jenkins/workspace/credential-config.yaml ./src/main/resources
# 빌드
chmod +x gradlew
./gradlew build
# 실행
java -jar ./build/libs/intaGral-0.0.1-SNAPSHOT.jar
```

## credential-config.yaml

```
spring:
  datasource:
    driver-class-name: com.mysql.cj.jdbc.Driver
    url: jdbc:mysql://k7a304.p.ssafy.io:3306/integral?characterEncoding=UTF-8&serverTimezone=UTC
    hikari:
      username: a304
      password: integral304

  jwt:
    secret: aW50YWdyYWw=
    # unit is ms. 15 * 24 * 60 * 60 * 1000 = 15days
    expiration: 1296000000

  cloud:
    aws:
      region:
        static: ap-northeast-2
      stack:
        auto: false
      credentials:
        access-key: AKIAUCIWIW7JX0XURU5P
        secret-key: eF0KcZ+QrVVfMNS9Nv/WpimemmqSJVXQaeH6ZhcR
      s3:
        bucket: integral-file-upload-bucket

---
spring:
  config:
    activate:
      on-profile: prod

  datasource:
    url: jdbc:mysql://localhost:3306/integral?characterEncoding=UTF-8&serverTimezone=UTC
```

위 Yaml 파일을 ec2의 `/var/lib/jenkins/workspace/` 로 복사

- spring 서버 빌드시 필요

## EC2 세팅

### 서버 시간 변경

```
# 서버 시간 확인
date

# 서버 시간 변경
sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

### 도커 설치

```
# apt update
sudo apt update

# docker 설치
sudo apt install docker.io
```

## 데이터베이스

```
# 도커 mysql 5.7 이미지 pull
sudo docker pull mysql:5.7

# 도커 MySQL 설치
sudo docker run -d -p 3306:3306 -v ~/mysql:/var/lib/mysql -e MYSQL_ROOT_PASSWORD='banatag304' --name mysql5.7 mysql:5.7 --character-se

# ~/mysql로 dump 파일 이동

# 도커 mysql 접속
sudo docker exec -it mysql5.7 mysql -u root -p

# mysql root 이름 변경 X
```

```

update user set user='a304' where user='root';

# 여기서 sql 실행
# bash에서 dump 실행

# 테이블 생성 X
CREATE DATABASE moweb DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;

# 계정 생성
create user 'a304'@localhost identified by 'integral304';
create user 'a304'@'%' identified by 'integral304';

# schema 생성 X
create database ;

# moweb 데이터베이스 권한 부여
grant all privileges on moweb.* to 'user_a507'@localhost identified by '5moweB0!@7';
grant all privileges on moweb.* to 'user_a507'@'%' identified by '5moweB0!@7';

grant select, update, delete, insert on integral.* to 'a304'@localhost identified by 'integral304';
grant select, update, delete, insert on integral.* to 'a304'@'%' identified by 'integral304';

# 디비 변경사항 메모리에 반영
flush privileges;

# 권한 확인
show grants for 'user_a507'@localhost;
show grants for 'user_a507'@'%;

# 시간 설정
set time_zone='Asia/Seoul';
set global time_zone='Asia/Seoul';

```

## JAVA

```

# apt-get update
sudo apt-get update
sudo apt-get upgrade

# java 11 jdk 설치
sudo apt-get install openjdk-11-jre openjdk-11-jdk

# java 버전 확인
java -version

# 환경 변수 적용
vim ~/.bashrc

# ~/.bashrc
export JAVA_HOME=$(dirname $(dirname $(readlink -f $(which java))))
export PATH=
$PATH
:$JAVA_HOME/bin

# 변경사항 적용
source ~/.bashrc

```

## Nginx

```

sudo apt update

sudo apt install nginx

/etc/nginx/sites-available/default 수정
server_name    k7a304.p.ssafy.io

sudo systemctl start nginx

```

### certbot 적용 후

```

sudo vim /etc/nginx/sites-available/default

# 443 포트
server {
    # reverse proxy 설정
    # Backend API
    location /api {
        proxy_pass http://127.0.0.1:8080/api;
    }
}

```

```

    }
    ....
}

# 80 포트
server {

    if ($host = k7a304.p.ssafy.io) {
        # http request method를 보장하기 위해 308로 redirect
        return 308 https://$host$request_uri;
    }

    listen 80 default_server;
    listen [::]:80 default_server;

    server_name k7a304.p.ssafy.io;
    return 404; # managed by Certbot

}

```

## 파일 업로드 용량 제한 증가

```

sudo vim /etc/nginx/nginx.conf

# client_max_body_size 용량; 을 추가
http {

    ##
    # Basic Settings
    ##

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;
    # server_tokens off;

    # max file size
    # default 1M
    client_max_body_size 10M;
    ...
}

```

## certbot

```

sudo apt update

sudo apt install certbot python3-certbot-nginx

sudo certbot --nginx

```

```

Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): 이메일@domain.com

```

```

Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
-----
(A)gree/(C)ancel: A

```

```
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
```

```
- - - - -
(Y)es/(N)o: N
```

```
Which names would you like to activate HTTPS for?
```

```
- - - - -
1: k7a304.p.ssafy.io
```

```
Select the appropriate numbers separated by commas and/or spaces, or leave input
blank to select all options shown (Enter 'c' to cancel): 1
```

```
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
```

```
- - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
```

```
- - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
```

## 젠킨스

```
# jenkins key 파일 업데이트
sudo wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee /etc/apt/sources.list.d/jenkins.list'

# jenkins 설치
sudo apt-get install jenkins

# jenkins 실행
sudo systemctl daemon-reload
sudo systemctl start jenkins
sudo systemctl status jenkins

# jenkins 초기 비밀번호 확인
cat var/lib/jenkins/secrets/initialAdminPassword

# suggested plugin install 선택
# gitlab 검색 → 다 설치
# nodejs 검색 → 설치
```

```
# 포트수정
# jenkins config 열기
sudo vim /usr/lib/systemd/system/jenkins.service

# Port to listen on for HTTP requests. Set to -1 to disable.
# To be able to listen on privileged ports (port numbers less than 1024),
# add the CAP_NET_BIND_SERVICE capability to the AmbientCapabilities
# directive below.
Environment="JENKINS_PORT=8090"
```

```
# jenkins build excute shell

cd integralServer
sudo cp /var/lib/jenkins/workspace/credential-config.yaml ./src/main/resources
chmod +x gradlew
./gradlew build --stacktrace
docker login -u "docker아이디" -p "docker패스워드" docker.io
docker stop A304_Backend && docker rm A304_Backend;
docker rmi "docker아이디"/integral_springboot:latest;
```

```
docker build -t "docker아이디"/integral_springboot:latest .
docker push "docker아이디"/integral_springboot:latest
docker run -d -p 8080:8080 --name A304_Backend --restart=always "docker아이디"/integral_springboot:latest
```

## AI 포팅, 학습 메뉴얼

1. integralModel/yolov5 디렉토리로 이동
2. anaconda 가상환경 (선택)
3. 파이썬 패키지 설치 ( `pip install -r requirements.txt` 로 간단히 설치 )

- 파이썬 3.9 이상
- clearml
- mlflow
- docker
- fiftyone
- pycocotools

4.

mlflow 시작

`mlflow server --backend-store-uri sqlite:///mlflow.db --default-artifact-root ./artifacts --host 0.0.0.0 --port 8304 &`

학습 시작

`python train.py --img 640 --batch 64 --epochs 100 --data coco.yaml --weights yolov5m.pt`

테스트

`python3 detect.py --source ../datasets/coco2017/images/room1/ --weights ./runs/train/exp65/weights/best.pt`

clearml 데이터 싱크 ( 데이터셋 위치에서 )

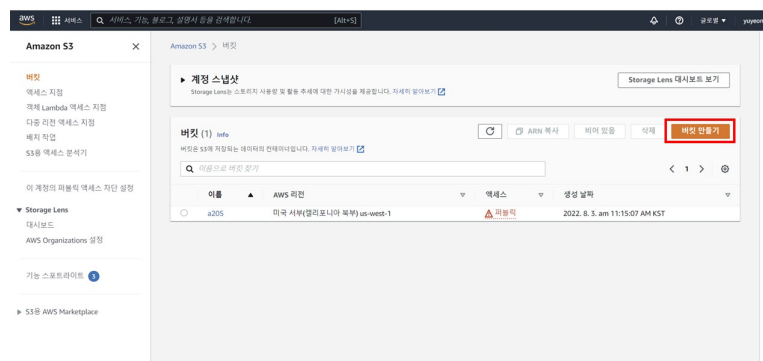
`clearml-data sync --project a304 --name coco128 --folder .`

clearml 학습

`python coco128train.py --img 640 --batch 16 --epochs 50 --data clearml:///7747ad4c50064a3fa864564d7ffd7ea0 --weights yolov5m.pt --cache`

## AWS S3 Bucket

1. AWS 계정 생성
2. AWS S3 검색해서 S3 Management Console 접속 → 버킷 만들기



3. S3 버킷 설정 후 생성

Amazon S3 > 버킷 > 버킷 만들기

### 버킷 만들기

버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기

**일반 구성**

버킷 이름

이름은 전체적으로 고유해야 하며 공백 또는 대문자를 포함할 수 없습니다. 버킷 이름 규칙 참조

AWS 리전

미국 서부(캘리포니아 북부) us-west-1

기본 버킷에서 설정 복사 - 선택 사항

다들 구성의 버킷 설정한 복사합니다.

버킷 선택

**객체 소유권**

다른 AWS 계정에서 이 버킷에 작성한 객체의 소유권 및 액세스 제어 목록(ACL)의 사용을 제어합니다. 객체 소유권은 객체에 대한 액세스를 지정할 수 있는 사용자를 결정합니다.

☐ ACL 비활성화(권장)  
 이 버킷의 모든 객체는 이 계정의 소유입니다. 이 버킷과 그 객체에 대한 액세스는 정책을 통해서만 지정됩니다.

☒ ACL 활성화됨  
 이 버킷의 객체는 다른 AWS 계정에서 소유할 수 있습니다. 이 버킷 및 객체에 대한 액세스는 ACL을 사용하여 지정할 수 있습니다.

객체 소유권

☒ 버킷 소유자 선택  
 이 버킷에 작성된 새 객체가 bucket-owner-full-control 인접 ACL을 지정하는 경우 새 객체는 버킷 소유자가 소유합니다. 그렇지 않은 경우 객체 라인이 소유합니다.

☐ 객체 라인  
 객체 라인은 객체 소유자로 유지됩니다.

☒ 새 객체에 대해서만 객체 소유권을 적용하려면 버킷 정책이 객체 접근으로 bucket-owner-full-control 인접 ACL을 요구하도록 지정해야 합니다. 자세히 알아보기

### 이 버킷의 퍼블릭 액세스 차단 설정

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지정 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단 설정을 활성화합니다. 이 설정은 이 버킷 및 객체 설정에 적용됩니다. AWS에서는 모든 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 이 버킷 또는 내부 객체에 대한 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개별 설정을 사용자 지정할 수 있습니다. 자세히 알아보기

- ☐ **로론 퍼블릭 액세스 차단**  
이 설정을 활성화하면 아래 4개의 설정을 모두 활성화한 것과 같습니다. 다음 설정 각각은 서로 독립적입니다.
  - ☐ **새 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단**  
S3은 새로 추가된 버킷 또는 객체에 적용되는 퍼블릭 액세스 권한을 차단하며, 기존 버킷 및 객체에 대한 새 퍼블릭 액세스 ACL 생성을 금지합니다. 이 설정은 ACL을 사용하여 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 권한을 변경하지 않습니다.
  - ☐ **임의의 ACL(액세스 제어 목록)을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단**  
S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 모든 ACL을 무시합니다.
  - ☐ **새 퍼블릭 버킷 또는 액세스 지정 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단**  
S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 새 버킷 및 액세스 지정 정책을 차단합니다. 이 설정은 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 정책을 변경하지 않습니다.
  - ☐ **임의의 퍼블릭 버킷 또는 액세스 지정 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 및 교차 계정 액세스 차단**  
S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 정책을 사용하는 버킷 또는 액세스 지정에 대한 퍼블릭 및 교차 계정 액세스를 무시합니다.

**모든 퍼블릭 액세스 차단을 비활성화하면 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있습니다.**  
정책 및 사이트 호스팅과 같은 구체적으로 확인된 사용 사례에서 퍼블릭 액세스가 필요한 경우가 아니면 모든 퍼블릭 액세스 차단을 활성화하는 것이 좋습니다.

☐ 현재 설정으로 인해 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있음을 알고 있습니다.

#### 4. 생성한 버킷 클릭 → 권한 설정으로 들어가 버킷 정책 생성

Amazon S3 > 버킷 > a205

**a205**

버킷 액세스 제어

객체 | **권한** | 정책 | 객체 | 액세스 지정

관련 개요

객체

**퍼블릭 액세스 차단(버킷 설정)**  
퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지정 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단을 활성화합니다. 이 설정은 이 버킷 및 객체 설정에 적용됩니다. AWS에서는 모든 퍼블릭 액세스 차단을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 이 버킷 또는 내부 객체에 대한 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스토리지 사용 사례에 맞게 아래 개별 설정을 사용자 지정할 수 있습니다. 자세히 알아보기

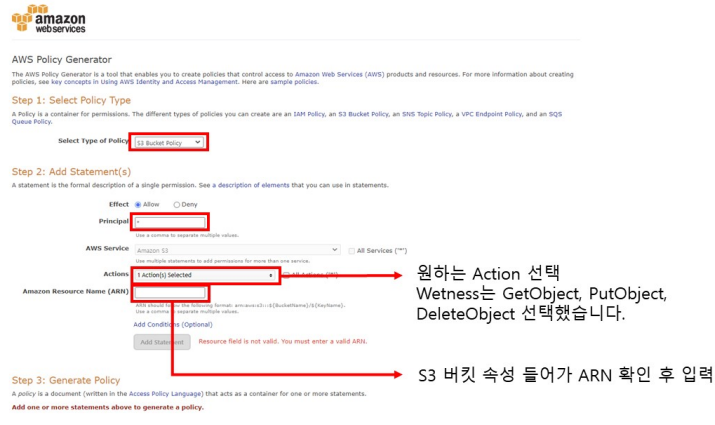
☐ 로론 퍼블릭 액세스 차단  
☒ 임의의 객체 퍼블릭 액세스 차단 설정

**버킷 정책**  
Amazon S3은 버킷 정책에 지정된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이 소유한 객체를 사용할 수 없습니다. 자세히 알아보기

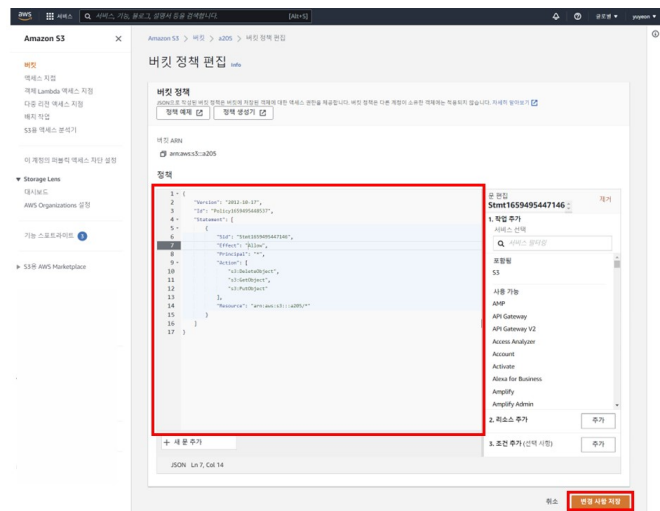
편집  삭제

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::a205/*"
    }
  ]
}
  
```



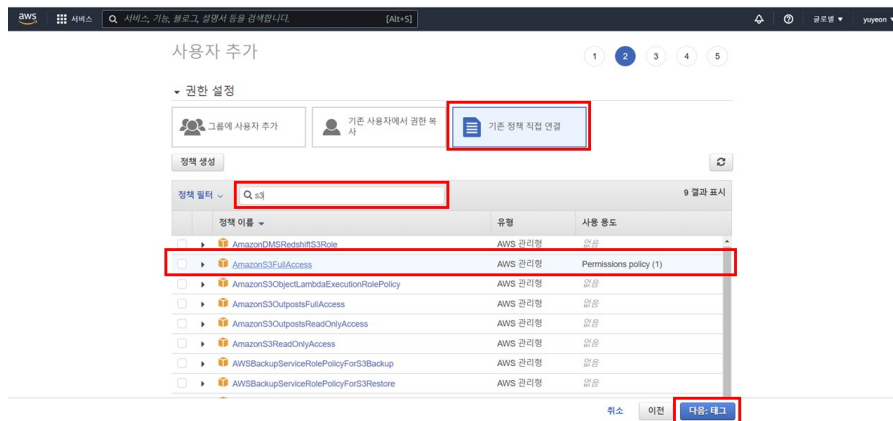
##### 5. 생성된 JSON을 정책에 입력



##### 6. IAM 검색해서 IAM 콘솔로 들어가, 사용자 추가







7. 생성한 직후, csv 다운로드 클릭해 AccessKey, SecretKey를 저장

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): 이메일@domain.com
```