



## **NONING 포팅 메뉴얼**

A202 다섯논쟁이

# 목차

I. 개요.....	2
1. 프로젝트 개요.....	2
2. 프로젝트 사용 도구.....	2
3. 개발 환경.....	2
4. 외부 서비스.....	3
II. 빌드.....	7
1. 환경변수 형태.....	7
2. 빌드하기.....	9
3. 배포하기.....	9

# I. 개요

## 1. 프로젝트 개요

‘애인이 이성친구의 찿뽕 떼어주는거 가능?’ 찿뽕 논쟁에 대한 당신의 생각은 어떤가요? 다른 사람들의 생각은 궁금하지 않나요? 이걸 고를까 저걸 고를까? 고민되는 다양한 논쟁들에 2022 년의 핫한 트렌드 ShortFun 을 결합한 논잉을 만나보세요.

논잉은 쉽고 빠르게 즐기는 다양한 논쟁거리들과 다른 사람들의 의견을 볼 수 있는 분석페이지, 같은 논쟁에 대해 얘기를 나눌 수 있는 실시간 음성 채팅 등 다양한 기능을 가지고 있는 ‘본격 편가르기 논쟁 SNS’입니다.

논잉과 함께 재미있는 논쟁을 시작해보세요. 논쟁을 멈추지 마! 논잉.

## 2. 프로젝트 사용 도구

이슈 관리 : JIRA

형상 관리 : Gitlab

커뮤니케이션 : Notion, Mattermost

디자인 : Figma

UCC : 모바비

CI/CD : Jenkins, Docker

## 3. 개발 환경

Frontend – React Native

Visual Studio Code IDE 1.69.0

Android Studio Chipmunk : 2021.2.1 Patch 2

react-native : ^0.69.2

AVD

Android SDK : 30

Android : 11.0 Google APIs

Model : Pixel 5 API 30

## Backend – Spring

IntelliJ IDEA : 2021.3.1

IntelliJ Runtime: 11.0.13+7-b1751.21 amd64

VM : OpenJDK 64-Bit Server VM by JetBrains s.r.o

JDK : 11.0.15

## Backend – Node

Node.js : 14.20.0

Python : 3.9

## 4. 외부 서비스

### Amazon Web Service S3 (Simple Storage Service)

- 1) AWS 가입하기
- 2) S3 버킷 생성

#### A. 버킷 만들기 선택



#### B. 버킷 이름 작성 및 AWS리전 서울 선택

버킷 이름

noning

버킷 이름은 전역에서 고유해야 하며 공백 또는 대문자를 포함할 수 없습니다. [버킷 이름 지정 규칙 참조](#)

AWS 리전

아시아 태평양(서울) ap-northeast-2

기존 버킷에서 설정 복사 - 선택 사항  
다음 구성의 버킷 설정만 복사됩니다.

버킷 선택

## C. 퍼블릭 액세스 차단 풀어주기

### 이 버킷의 퍼블릭 액세스 차단 설정

퍼블릭 액세스는 ACL(엑세스 제어 목록), 버킷 정책, 액세스 지점 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 이 버킷 및 해당 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 모든 퍼블릭 액세스 차단을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지점에만 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](#)

#### ☐ 모든 퍼블릭 액세스 차단

이 설정을 활성화하면 아래 4개의 설정을 모두 활성화한 것과 같습니다. 다음 설정 각각은 서로 독립적입니다.

##### ☐ 새 ACL(엑세스 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(엑세스 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 지점 정책을 통해 부여된 버킷 및 객체에 대한 퍼블릭 액세스 차단

S3은 버킷 및 객체에 대한 퍼블릭 액세스를 부여하는 새 버킷 및 액세스 지점 정책을 차단합니다. 이 설정은 S3 리소스에 대한 퍼블릭 액세스를 허용하는 기존 정책을 변경하지 않습니다.

##### ☐ 임의의 퍼블릭 버킷 또는 액세스 지점 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 지점에 대한 퍼블릭 및 교차 계정 액세스를 무시합니다.

## D. 버킷 만들기

## E. 버킷 – 권한에서 버킷 정책 설정

### 버킷 정책

JSON으로 작성된 버킷 정책은 버킷에 저장된 객체에 대한 액세스 권한을 제공합니다. 버킷 정책은 다른 계정이 소유한 객체에는 적용되지 않습니다. [자세히 알아보기](#)

[편집](#)[삭제](#)

```
{
  "Version": "2012-10-17",
  "Id": "Policy1659851181014",
  "Statement": [
    {
      "Sid": "Stmnt1659851171151",
      "Effect": "Allow",
      "Principal": "*",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::noning/*"
    }
  ]
}
```

[복사](#)

### 3) IAM 추가

#### 사용자 추가

1 2 3 4 5

#### 사용자 세부 정보 설정

동일한 액세스 유형 및 권한을 사용하여 한 번에 여러 사용자를 추가할 수 있습니다. [자세히 알아보기](#)

사용자 이름\*

이름이 "noning"인 사용자가 이미 존재합니다.

[+ 다른 사용자 추가](#)

#### AWS 액세스 유형 선택

이러한 사용자가 주로 AWS에 액세스하는 방법을 선택합니다. 프로그래밍 방식의 액세스만 선택하면 사용자가 위임된 역할을 사용하여 콘솔에 액세스하는 것을 방지할 수 없습니다. 액세스 키와 자동 생성된 암호가 마지막 단계에서 제공됩니다. [자세히 알아보기](#)

- AWS 자격 증명 유형 선택\***
- ☒ **액세스 키 – 프로그래밍 방식 액세스**  
AWS API, CLI, SDK 및 기타 개발 도구에 대해 **액세스 키 ID** 및 **비밀 액세스 키**를 생성합니다.
- ☐ **암호 – AWS 관리 콘솔 액세스**  
사용자가 AWS Management Console에 로그인할 수 있도록 허용하는 **비밀번호**를 생성합니다.

### A. IAM 사용자 추가 – 액세스 키 체크

#### 사용자 추가

1 2 3 4 5

#### ▼ 권한 설정

그룹에 사용자 추가

기존 사용자에서 권한 복사

기존 정책 직접 연결

정책 생성

↺

정책 필터

1 결과 표시

정책 이름	유형	사용 용도
<input type="checkbox"/> AmazonS3FullAccess	AWS 관리형	Permissions policy (1)

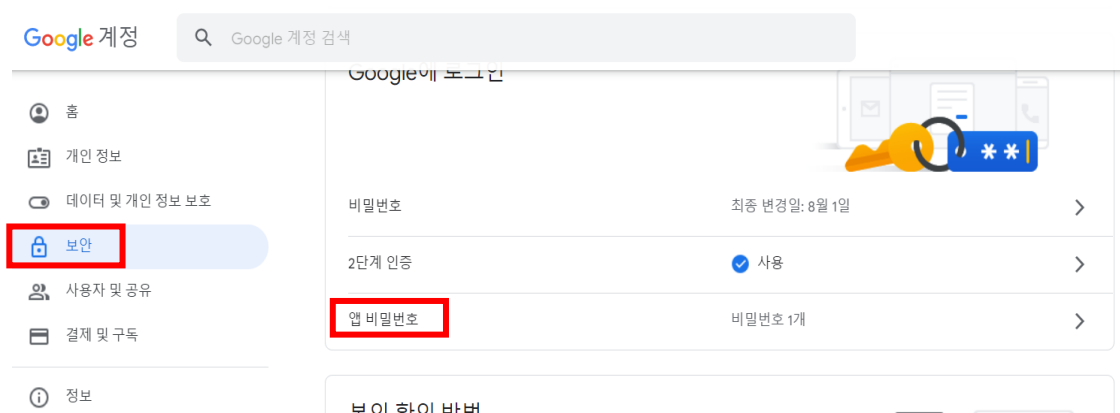
### B. S3FullAccess 정책 연결

### C. 생성된 사용자 csv파일 받은 후 Access Key, Secret key 확인

## Google Gmail SMTP

### 1) Google 메일 앱 비밀번호 발급

#### A. Google 계정 설정 - 2단계 인증 설정



#### B. Google 계정 설정 - 보안 - 앱 비밀번호

##### ← 앱 비밀번호

앱 비밀번호를 사용하면 2단계 인증을 지원하지 않는 기기의 앱에서 Google 계정에 로그인할 수 있습니다. 비밀번호를 한 번만 입력하면 기억할 필요가 없습니다. [자세히 알아보기](#)

앱 비밀번호

이름	생성됨	최종 사용일
Windows 컴퓨터의 메일	8월 2일	오후 1:42

앱 비밀번호를 생성할 앱 및 기기를 선택하세요.

메일

Windows 컴퓨터

생성

#### C. 메일 , Windows 컴퓨터 비밀번호 생성

#### D. Windows 컴퓨터용 앱 비밀번호 확인

## 2) POP/IMAP 설정

### A. Gmail – 모든 설정

#### 설정

기본설정 라벨 받은편지함 계정 및 가져오기 필터 및 차단된 주소 **전달 및 POP/IMAP** 부가기능 채팅 및 Meet 고급 오프라인 테마

---

**전달:**  
[자세히 알아보기](#)

전달 주소 추가

도움말: 필터를 만들면 메일 중 일부만 전달할 수도 있습니다.

---

**POP 다운로드:**  
[자세히 알아보기](#)

1 상태: 모든 메일에 대해 POP가 사용 설정되어 있습니다.

☒ 이미 다운로드된 메일을 포함하여 모든 메일에 POP를 활성화 하기

☐ 지금부터 수신되는 메일에만 POP를 사용하기

☐ POP 사용 안함

2. POP로 메시지를 여는 경우 Gmail 사본을 받은편지함에 보관하기 ▼

3. 이메일 클라이언트 구성 (예: Outlook, Eudora, Netscape Mail)  
[설정 방법](#)

---

**IMAP 액세스:**  
(IMAP를 사용하여 다른 클라이언트에서 Gmail에 액세스)  
[자세히 알아보기](#)

상태: IMAP를 사용할 수 있습니다.

☒ IMAP 사용

☐ IMAP 사용 안함

IMAP에서 메일을 삭제된 것으로 표시하는 경우:

☒ 자동 삭제 사용 - 서버를 즉시 업데이트(기본값)

☐ 자동 삭제 사용 안함 - 클라이언트가 서버를 업데이트할 때까지 대기

메일이 삭제된 것으로 표시되고 마지막으로 표시된 IMAP 폴더에서 삭제된 경우:

☒ 메일 보관(기본값)

☐ 메일을 휴지통으로 이동

☐ 메일을 즉시 완전삭제

### B. 전달 및 POP/IMAP 설정

## II. 빌드

### 1. 환경변수 형태

application.properties

# Maria DB

spring.datasource.driverClassName=org.mariadb.jdbc.Driver

spring.datasource.url : DB 주소

spring.datasource.username : 유저 이름

spring.datasource.password : 유저 비밀번호

# JPA

spring.jpa.hibernate.ddl-auto : update (변경되는 값만 적용 \*운영 DB 에 적용 X)



spring.jpa.show-sql : true (쿼리 로그 확인)

#### # JWT

JWT.SECRET : 시크릿 키

#### # Redis

spring.redis.host : 호스트 주소

spring.redis.password : 비밀번호

spring.redis.port : 포트

#### #SMTP

spring.mail.host : 주소 (구글 SMTP : smtp.gmail.com)

spring.mail.port : 포트 (구글 SMTP : 587)

spring.mail.username : 구글 ID

spring.mail.password : 메일 접속 비밀번호

spring.mail.properties.mail.smtp.auth : auth 명령 사용여부 ( true )

spring.mail.properties.mail.smtp.starttls.enable : TLS-protection 사용여부 ( true )

#### #WEBHOOK

notification.mattermost.enabled : 사용여부 ( true )

notification.mattermost.webhook-url : Webhook 주소

notification.mattermost.pretext : attachment 의 상단에 나오는 텍스트

notification.mattermost.author-name : attachment 의 상단에 나오는 이름

#### #S3 Bucket

cloud.aws.credentials.accessKey : 버킷 접근키

cloud.aws.credentials.secretKey : 버킷 비밀키

cloud.aws.stack.auto : 스택이름 자동 감지여부 ( false )

cloud.aws.s3.bucket : 버킷 이름

cloud.aws.region.static : 버킷 지역 (ap-northeast-2)

#### #multipart 사이즈 설정

spring.servlet.multipart.max-file-size=20MB

spring.servlet.multipart.max-request-size=20MB

## 2. 빌드하기

Frontend

npm i

npx react-native run-android

Backend – Spring ( port : 8888 )

Gradle 빌드

Bootjar

Boostrun

Backend – Node (port 3000, 8400-8880)

npm i

npm start

## 3. 배포하기

### Nginx

Nginx 설정

```
server {  
    listen 80;  
    listen [::]:80;  
  
    server_name i7a202.p.ssafy.io;  
    rewrite ^(.*)https://i7a202.p.ssafy.io$1 permanent;  
}  
  
server {  
    listen 443 ssl;  
    listen [::]:443;  
  
    server_name i7a202.p.ssafy.io;  
  
    # SSL configuration  
    ssl_certificate /etc/letsencrypt/live/i7a202.p.ssafy.io/fullchain.pem;  
    ssl_certificate_key /etc/letsencrypt/live/i7a202.p.ssafy.io/privkey.pem;
```

```
        location /api {
            proxy_pass http://i7a202.p.ssafy.io:8888;
        }
    }
```

Nginx 재실행

```
sudo service nginx restart
```

## Docker 설치

필요 패키지 설치

```
sudo apt install apt-transport-https ca-certificates curl software-properties-common
```

레포지토리 추가

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64]
https://download.docker.com/linux/ubuntu bionic stable"
```

Docker 설치

```
sudo apt install docker-ce
```

Docker compose 설치

```
sudo curl -L https://github.com/docker/compose/releases/download/v2.1.0/docker-
compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
```

권한 설정

```
sudo chmod +x /usr/local/bin/docker-compose
```

## Redis 설치

Redis 이미지 받아오기

```
docker image pull redis
```

Redis 8379 포트로 실행 및 비밀번호 설정

```
docker run -d --name noning_redis -p -8379:6379 redis --requirepass "비밀번호"
```

## Jenkins 설치

## Docker-compose.yml 파일 작성

```
version: "3"
services:
  jenkins:
    container_name: jenkins-compose
    image: jenkins/jenkins:lts-jdk11
    build:
      context: jenkins-dockerfile
      dockerfile: Dockerfile
    user: root
    ports:
      - "9090:8080"
    volumes:
      - /home/noning/compose/jenkins:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
      - /usr/bin/docker:/usr/bin/docker
      - /var/www/i7a202.p.ssafy.io/noning:/var/www/i7a202.p.ssafy.io/noning
    privileged: true
```









## Docker-compose 실행

```
docker-compose up -d
```

## 모든 Docker container 확인

```
docker ps -a
```

## Gitlab Plugin 설치

<b>Generic Webhook Trigger Plugin</b> 1.84 Can receive any HTTP request, extract any values from JSON or XML and trigger a job with those values available as variables. Works with GitHub, GitLab, Bitbucket, Jira and many more. <a href="#">Report an issue with this plugin</a>		
<b>Gitlab API Plugin</b> 5.0.1-78.v47a_45b_9f78b_7 This plugin provides <b>GitLab API</b> for other plugins. <a href="#">Report an issue with this plugin</a>		
<b>GitLab Authentication plugin</b> 1.16 This is the an authentication plugin using gitlab OAuth. <a href="#">Report an issue with this plugin</a> <div><b>This plugin is up for adoption!</b> We are looking for new maintainers. Visit our <a href="#">Adopt a Plugin</a> initiative for more information.</div>		
<b>GitLab Plugin</b> 1.5.35 This plugin allows <b>GitLab</b> to trigger Jenkins builds and display their results in the GitLab UI. <a href="#">Report an issue with this plugin</a>		

## Docker Plugin 설치

**Docker API Plugin** 3.2.13-37.vf3411c9828b9

This plugin provides [docker-java](#) API for other plugins.

[Report an issue with this plugin](#)

**This plugin is up for adoption!** We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.



**Docker Commons Plugin** 1.19

Provides the common shared functionality for various Docker-related plugins.

[Report an issue with this plugin](#)



**Docker Pipeline** 1.29

Build and use Docker containers from pipelines.

[Report an issue with this plugin](#)



**Docker plugin** 1.2.9

This plugin integrates Jenkins with [Docker](#)

[Report an issue with this plugin](#)



## Gradle 설정

### Gradle

Gradle installations

List of Gradle installations on this system

Add Gradle

Gradle  
name ?

gradle

☒ Install automatically ?

≡ Install from Gradle.org

Version

Gradle 7.5

Add Installer ▾

## Git Account 설정

### New credentials

Kind

Username with password

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Username ?

아이디

☐ Treat username as secret ?

Password ?

\*\*\*\*\*


## Backend - Spring 빌드 (w. Jenkins)

### 아이템 생성

Enter an item name

noning\_back

» Required field

 **Freestyle project**

이것은 Jenkins의 주요 기능입니다. Jenkins은 어느 빌드 시스템과 어떤 SCM(형상관리)으로 묶인 당신의 프로젝트를 빌드할 것이고, 소프트웨어 빌드보다 다른 어떤 것에 자주 사용될 수 있습니다.

## Git 설정

Git ?

Repositories ?

Repository URL ?

https://lab.ssafy.com/s07-webmobile2-sub2/S07P12A202.git

Credentials ?

+

Add

고급...

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/BE\_develop

## 빌드 유발 설정

**빌드 유발**

☐ 빌드를 원격으로 유발 (예: 스크립트 사용) ?

☐ Build after other projects are built ?

☐ Build periodically ?

☒ Build when a change is pushed to GitLab. GitLab webhook URL: <http://i7a202.p.ssafy.io:9090/project/noning> ?

Enabled GitLab triggers

☒ Push Events

Secret token ?

Generate

Secret Token 복사하기

## Graddle 로 빌드

**Build**


≡ **Invoke Gradle script** ?

☒ Invoke Gradle ?

Gradle Version

☐ Use Gradle Wrapper ?

Tasks ?

 고급...

## Excute Shell 설정

```
cd ./backend
docker build -t noning:latest .

docker stop noning_back
docker rm noning_back
docker rmi $(docker images -f "dangling=true" -q)
docker run --name noning_back -d -p 8888:8888 noning:latest
```

## Git WebHook 설정

### Webhooks

[Webhooks](#) enable you to send notifications to web applications in response to events in a group or project. We recommend using an [integration](#) in preference to a webhook.

#### URL

URL must be percent-encoded if it contains one or more special characters.

#### Secret token

Used to validate received payloads. Sent with the request in the `X-GitLab-Token` HTTP header.

#### Trigger

☒ Push events

Push to the repository.

## Backend – Node 빌드 (w. Jenkins)

아이템 생성, Git 설정, 빌드 유발, Git WebHook 은 Spring 과 동일 (branch 만 변경)  
Excute Shell 설정

```
cd ./chatserver
docker build -t noningchat:latest .
docker stop noning_chat
docker rm noning_chat
if [ $(docker images -f dangling=true -q) ]
then
docker rmi $(docker images -f "dangling=true" -q)
else
echo ""
fi
docker run --name noning_chat -d -p 3000:3000 -v
/var/www/i7a202.p.ssafy.io/noning:/var/www/i7a202.p.ssafy.io/noning
noningchat:latest
```

## Frontend apk 빌드

APK release 파일 생성

```
cd android
./gradlew assembleRelease
```

APK 파일 위치

```
/android/app/build/outputs/apk/release
```