



포팅메뉴얼

Dev Env

1. git lab 소스 클론 후 빌드 및 배포할 수 있도록 정리
 - CICD - 준님
 - 하둠 - 민아님
 - 백엔드
 - 프론트엔드
2. 프로젝트에 사용한 외부 서비스
 - S3 - 소정님
 - 카카오oauth2 - 소정님
3. DB 덤프파일 최신본
4. 시연 시나리오

1. git lab 소스 클론 후 빌드 및 배포할 수 있도록 정리

CICD - 준님

하둠 - 민아님

백엔드

- 버전
 - IDE : IntelliJ
 - 언어 : Java(OpenJDK 11)
 - DataBase : MySQL 8
 - Framework
 - Spring Boot
 - Gradle
 - Application.yml
 - JPA
 - Spring Security

- 설정파일
 - ▼ build.gradle

```
plugins {  
    id 'org.springframework.boot' version '2.7.3'
```

```

        id 'io.spring.dependency-management' version '1.0.13.RELEASE'
        id 'java'
    }

    group = 'com.sarac'
    version = '0.0.1-SNAPSHOT'
    sourceCompatibility = '11'

    configurations {
        compileOnly {
            extendsFrom annotationProcessor
        }
    }

    jar {
        enabled = false
    }

    repositories {
        mavenCentral()
    }

    dependencies {
        implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
        implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'
        implementation 'io.jsonwebtoken:jjwt:0.9.1'
        implementation 'org.springframework.boot:spring-boot-starter-web'
        implementation 'com.google.code.gson:gson:2.9.0'
        implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
        implementation group: 'org.javassist', name: 'javassist', version: '3.15.0-GA'
        implementation group: 'org.mindrot', name: 'jbcrypt', version: '0.4'
        implementation group: 'com.googlecode.json-simple', name: 'json-simple', version: '1.1.1'
        implementation group: 'io.springfox', name: 'springfox-boot-starter', version: '3.0.0'

        compileOnly 'org.projectlombok:lombok'
        runtimeOnly 'mysql:mysql-connector-java'
        annotationProcessor 'org.projectlombok:lombok'
        testImplementation 'org.springframework.boot:spring-boot-starter-test'
        testImplementation 'org.springframework.security:spring-security-test'
        implementation group: 'org.springframework.boot', name: 'spring-boot-starter-security', version: '2.7.1'
        implementation group: 'org.springframework.boot', name: 'spring-boot-starter-oauth2-client', version: '2.7.2'
    }

    tasks.named('test') {
        useJUnitPlatform()
    }
}

```

▼ application.yml

```

#JPA
spring:
  datasource:
    url: jdbc:mysql://[서버주소]:3306/a505?allowPublicKeyRetrieval=true&useSSL=false&characterEncoding=UTF-8&serverTimezone=UTC
    username: [mysql 아이디]
    password: [mysql 비밀번호]
    driver-class-name: com.mysql.cj.jdbc.Driver
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL8Dialect
    show_sql: true
    hibernate:
      ddl-auto: update
      naming:
        strategy: org.hibernate.cfg.ImprovedNamingStrategy

    properties:
      hibernate:
        format_sql: true
        enable_lazy_load_no_trans: true
#      dialect: org.hibernate.dialect.PostgreSQLDialect
#      default_batch_fetch_size: 100

#oauth2
security:
  oauth2:
    client:
      registration:
        kakao:
          client-id: [client-id]
          redirect-uri: http://[서버주소]:8080/login/oauth2/code/kakao
          client-authentication-method: POST

```

```

        authorization-grant-type: authorization_code
        scope: profile_nickname, profile_image, account_email
        client-name: Kakao
    provider:
        kakao:
            authorization_uri: https://kauth.kakao.com/oauth/authorize
            token_uri: https://kauth.kakao.com/oauth/token
            user-info-uri: https://kapi.kakao.com/v2/user/me
            user_name_attribute: id

#이미지
servlet:
    multipart:
        max-file-size: 10MB
        max-request-size: 10MB
#카카오 어드민키
kakao:
    admin : [카카오 어드민 키]
    redirect : http://[서버주소]:8081

#jwt
jwt:
    secret: zoqxlsthwjdwkdwkdaos

#s3
cloud:
    aws:
        credentials:
            access-key: [S3 access-key]
            secret-key: [S3 secret-key]
        s3:
            bucket: sarac-a505
        region:
            static: ap-northeast-2 # 리전 정보
        stack:
            auto: false

logging:
    level:
        com:
            amazonaws:
                util:
                    EC2MetadataUtils: error

#알라딘
aladin:
    ttbkey: [알라딘 ttbkey]

```

- 설정파일

- ▼ build.gradle

```

plugins {
    id 'org.springframework.boot' version '2.7.3'
    id 'io.spring.dependency-management' version '1.0.13.RELEASE'
    id 'java'
}

group = 'com.sarac'
version = '0.0.1-SNAPSHOT'
sourceCompatibility = '11'

configurations {
    compileOnly {
        extendsFrom annotationProcessor
    }
}

jar {
    enabled = false
}

repositories {
    mavenCentral()
}

dependencies {
    implementation 'org.springframework.boot:spring-boot-starter-thymeleaf'
    implementation 'org.springframework.cloud:spring-cloud-starter-aws:2.2.6.RELEASE'
    implementation 'io.jsonwebtoken:jjwt:0.9.1'
    implementation 'org.springframework.boot:spring-boot-starter-web'
}

```

```

implementation 'com.google.code.gson:gson:2.9.0'
implementation 'org.springframework.boot:spring-boot-starter-data-jpa'
implementation group: 'org.javassist', name: 'javassist', version: '3.15.0-GA'
implementation group: 'org.mindrot', name: 'jbcrypt', version: '0.4'
implementation group: 'com.googlecode.json-simple', name: 'json-simple', version: '1.1.1'
implementation group: 'io.springfox', name: 'springfox-boot-starter', version: '3.0.0'

compileOnly 'org.projectlombok:lombok'
runtimeOnly 'mysql:mysql-connector-java'
annotationProcessor 'org.projectlombok:lombok'
testImplementation 'org.springframework.boot:spring-boot-starter-test'
testImplementation 'org.springframework.security:spring-security-test'
implementation group: 'org.springframework.boot', name: 'spring-boot-starter-security', version: '2.7.1'
implementation group: 'org.springframework.boot', name: 'spring-boot-starter-oauth2-client', version: '2.7.2'
}

tasks.named('test') {
    useJUnitPlatform()
}

```

▼ application.yml

```

#JPA
spring:
  datasource:
    url: jdbc:mysql://[서버주소]:3306/a505?allowPublicKeyRetrieval=true&useSSL=false&characterEncoding=UTF-8&serverTimezone=UTC
    username: [mysql 아이디]
    password: [mysql 비밀번호]
    driver-class-name: com.mysql.cj.jdbc.Driver
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher
  jpa:
    database: mysql
    database-platform: org.hibernate.dialect.MySQL8Dialect
    show_sql: true
    hibernate:
      ddl-auto: update
      naming:
        strategy: org.hibernate.cfg.ImprovedNamingStrategy

  properties:
    hibernate:
      format_sql: true
      enable_lazy_load_no_trans: true
#    dialect: org.hibernate.dialect.PostgreSQLDialect
#    default_batch_fetch_size: 100

#oauth2
security:
  oauth2:
    client:
      registration:
        kakao:
          client-id: [client-id]
          redirect-uri: http://[서버주소]:8080/login/oauth2/code/kakao
          client-authentication-method: POST
          authorization-grant-type: authorization_code
          scope: profile_nickname, profile_image, account_email
          client-name: Kakao
      provider:
        kakao:
          authorization_uri: https://kauth.kakao.com/oauth/authorize
          token_uri: https://kauth.kakao.com/oauth/token
          user-info-uri: https://kapi.kakao.com/v2/user/me
          user_name_attribute: id

#이미지
servlet:
  multipart:
    max-file-size: 10MB
    max-request-size: 10MB
#카카오 어드민키
kakao:
  admin : [카카오 어드민 키]
  redirect : http://[서버주소]:8081

#jwt
jwt:
  secret: zoqx1sthwjdwkdWkdaos

#s3
cloud:

```

```

aws:
  credentials:
    access-key: [S3 access-key]
    secret-key: [S3 secret-key]
  s3:
    bucket: sarac-a505
  region:
    static: ap-northeast-2 # 리전 정보
  stack:
    auto: false

logging:
  level:
    com:
      amazonaws:
        util:
          EC2MetadataUtils: error

#알라딘
aladin:
  ttbkey: [알라딘 ttbkey]

```

프론트엔드

- 버전
 - IDE : Visual Studio Code
 - Framework
 - Vue 2 LTS
 - router
 - vuetify
 - bootstrap vue
 - babel
 - webpack
 - vuetify bootstrap vue
 - Nodejs(16.17.0 LTS)

• 설정파일

▼ .env

```

VUE_APP_API_URL = https://sarac.tk/

VUE_APP_API_URL_KAKAO = https://sarac.tk/oauth2/authorization/kakao

```

▼ .env.development

```

VUE_APP_API_URL = "http://localhost:8080"

VUE_APP_API_URL_KAKAO = http://localhost:8080/oauth2/authorization/kakao

```

▼ package.json

```

{
  "name": "sarac-frontend",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint",
    "start": "webpack-dev-server --inline --content-base . --history-api-fallback"
  },

```

```

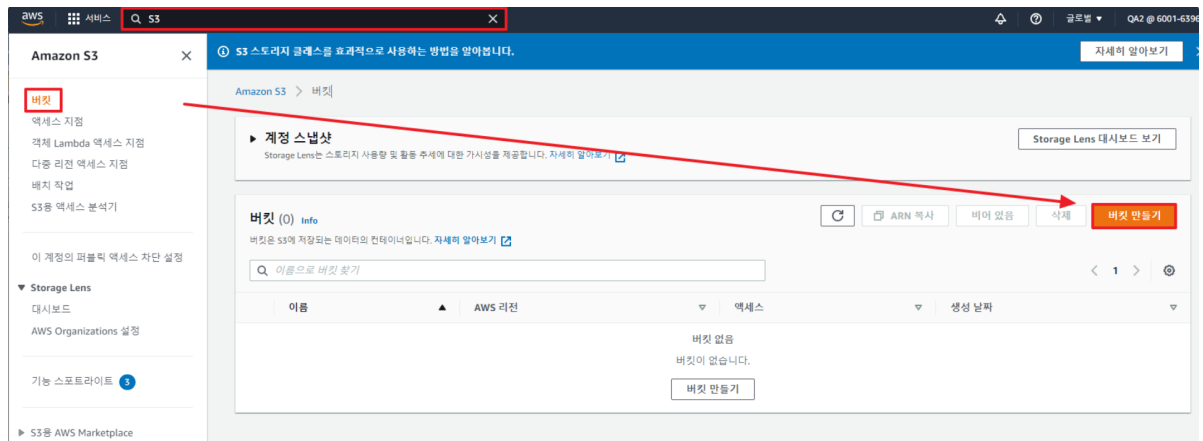
"dependencies": {
  "axios": "^0.27.2",
  "bootstrap": "^4.6.1",
  "bootstrap-vue": "^2.22.0",
  "core-js": "^3.8.3",
  "vue": "^2.7.10",
  "vue-barcode-reader": "^1.0.2",
  "vue-d3-cloud": "^0.2.0",
  "vue-router": "^3.5.1",
  "vuetify": "^2.6.0",
  "vuex": "^3.6.2"
},
"devDependencies": {
  "@babel/core": "^7.12.16",
  "@babel/eslint-parser": "^7.12.16",
  "@vue/cli-plugin-babel": "~5.0.0",
  "@vue/cli-plugin-eslint": "~5.0.0",
  "@vue/cli-plugin-router": "~5.0.0",
  "@vue/cli-plugin-vuex": "~5.0.0",
  "@vue/cli-service": "~5.0.0",
  "eslint": "^7.32.0",
  "eslint-plugin-vue": "^8.0.3",
  "sass": "~1.32.0",
  "sass-loader": "^10.0.0",
  "vue-cli-plugin-vuetify": "~2.5.5",
  "vue-template-compiler": "^2.6.14",
  "vuetify-loader": "^1.7.0"
},
"eslintConfig": {
  "root": true,
  "env": {
    "node": true
  },
  "extends": [
    "plugin:vue/essential",
    "eslint:recommended"
  ],
  "parserOptions": {
    "parser": "@babel/eslint-parser"
  },
  "rules": {
    "vue/multi-word-component-names": 0
  }
},
"browserslist": [
  "> 1%",
  "last 2 versions",
  "not dead"
]
}

```

- 실행순서
 - Frontend의 sarac폴더 에서 vsCode로 열기
 - 터미널에 npm i 입력
 - 터미널에 npm run serve 실행

2. 프로젝트에 사용한 외부 서비스

S3 - 소정님



aws s3에 버킷 생성하기

일반 구성

버킷 이름 버킷 이름은 고유값

tistory-test-bucket-01

버킷 이름은 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

객체 소유권 [Info](#) 다른 AWS 계정에서도 소유권을 갖거나 접속제어 가능유무

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 비활성화됨(권장)

이 버킷의 모든 객체는 이 계정이 소유합니다. 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 **[ACL 비활성화를 선택]해서 내가 접속한 계정에서만**
버킷 소유자 적용 **소유하도록 한다**



모든 퍼블릭 액세스 차단을 비활성화하면 이 버킷과 그 안에 포함된 객체가 퍼블릭 상태가 될 수 있습니다. 정적 웹 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.

버킷 버전 관리- 비활성화 / 서버측 암호화 -비활성화

설정 후 버킷 생성 누르면 버킷 생성 완료

외부에서 사용하기 위해 버킷 정책 생성

버킷 정책 생성기 주소 - <http://awspolicygen.s3.amazonaws.com/policygen.html>

위 주소 들어가서 정책 생성하기

Step 1: Select Policy Type

A Policy is a container for permissions. The different types of policies you can create are an IAM Policy, an S3 Bucket Policy, an SNS Topic Policy, a VPC Endpoint Policy, and an SQS Queue Policy.

Select Type of Policy **S3 Bucket Policy** 정책 타입 선택 → 버킷 정책

Step 2: Add Statement(s)

A statement is the formal description of a single permission. See a description of elements that you can use in statements.

Effect **Allow** 정책 적용 허용

Deny

Principal ***** 버킷 정책이 적용될 대상 → 전체(*)

Use a comma to separate multiple values.

AWS Service **Amazon S3** All Services (**)

Use multiple statements to add permissions for more than one service.

Actions **2 Action(s) Selected** 버킷에 수행할 액션 (GetObject, PutObject) 선택

Amazon Resource Name (ARN) **arn:aws:s3:::testory-test-b** 버킷에 어떤 리소스에 적용할지 → 버킷이름 선택, 리소스 전체(*)

ARN should follow the following form: `arn:aws:s3:::{BucketName}/{Key}`. Use a comma to separate multiple values.

`arn:aws:s3:::버킷이름/*`

Add Conditions (Optional)

Add Statement

정책 생성후 만들어진 JSON 복사 후

버킷 (1) Info

버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기

이름으로 버킷 찾기

이름	AWS 리전	액세스	생성 날짜
tistory-test-bucket-01	아시아 태평양(서울) ap-northeast-2	객체를 퍼블릭으로 설정할 수 있음	2022. 4. 8. pm 8:58:23 PM KST

tistory-test-bucket-01 Info

객체 | 속성 | 권한 | 지표 | 관리 | 액세스 지정

권한 개요

액세스
객체를 퍼블릭으로 설정할 수 있음

퍼블릭 액세스 차단(버킷 설정)

퍼블릭 액세스는 ACL(액세스 제어 목록), 버킷 정책, 액세스 지정 정책 또는 모두를 통해 버킷 및 객체에 부여됩니다. 모든 S3 버킷 및 객체에 대한 퍼블릭 액세스가 차단되었는지 확인하려면 [모든 퍼블릭 액세스 차단]을 활성화합니다. 이 설정은 이 버킷 및 해당 액세스 지정에만 적용됩니다. AWS에서는 [모든 퍼블릭 액세스 차단]을 활성화하도록 권장하지만, 이 설정을 적용하기 전에 퍼블릭 액세스가 없어도 애플리케이션이 올바르게 작동하는지 확인합니다. 버킷 또는 내부 객체에 어느 정도 수준의 퍼블릭 액세스가 필요한 경우 특정 스트리지 사용 사례에 맞게 개별 설정을 사용자 지정할 수 있습니다. 자세히 알아보기

편집

모든 퍼블릭 액세스 차단
⚠ 비활성
▶ 이 버킷의 개별 퍼블릭 액세스 차단 설정

버킷 정책

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

표시할 정책이 없습니다.

편집 삭제 복사

버킷 정책

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

정책 예제

버킷 ARN
arn:aws:s3:::tistory-test-bucket-01

정책

```

1 {
2   "Version": "2012-10-17",
3   "Id": "Policy1649421058532",
4   "Statement": [
5     {
6       "Sid": "Stmt1649420985040",
7       "Effect": "Allow",
8       "Principal": "*",
9       "Action": [
10        "s3:GetObject",
11        "s3:PutObject"
12      ],
13       "Resource": "arn:aws:s3:::tistory-test-bucket-01/*"
14     }
15   ]
16 }

```

문 편집
Stmt1649420985040

1. 작업 추가
서비스 선택
서비스 필터링

포함됨
S3

사용 가능
AMP
API Gateway
API Gateway V2
Access Analyzer
Account

참고사이트 : <https://inpa.tistory.com/entry/AWS-S3-버킷-생성-사용법-실전-구축>


카카오oauth2 - 소정님

전체 애플리케이션 (4)

애플리케이션 이름


+

애플리케이션 추가하기




사락

ID 797120 OWNER Web



놀멍쉬멍

ID 775049 OWNER Web



제주여행

카카오 개발자에 애플리케이션 추가

내 애플리케이션 > 앱 설정 > 플랫폼

iOS

iOS 플랫폼 등록

Web

삭제

수정

사이트 도메인	http://localhost:8080
---------	-----------------------

- 카카오 로그인 사용 시 Redirect URI를 등록해야 합니다. [등록하러 가기](#)

내애플리케이션>앱설정>플랫폼에 사이트 도메인 설정하고 Redirect URI 등록하러가기

포팅메뉴얼

10

활성화 설정

상태 ON

카카오 로그인 API를 활용하면 사용자들이 번거로운 회원 가입 절차 대신, 카카오톡으로 서비스를 시작할 수 있습니다.
상태가 OFF일 때도 카카오 로그인 설정 항목을 변경하고 서버에 저장할 수 있습니다.
상태가 ON일 때만 실제 서비스에서 카카오 로그인 화면이 연결됩니다.

Redirect URI

카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다.
여러개의 URI를 줄바꿈으로 추가해주세요. (최대 10개)
REST API로 개발하는 경우 필수로 설정해야 합니다.
예시: (O) <https://example.com/oauth> (X) <https://www.example.com/oauth>

활성화ON 후 리다이렉트 URI 등록(시큐리티 사용시 <http://이름/login/oauth2/code/kakao> 형식)

개인정보

항목 이름	ID	상태
닉네임	profile_nickname	● 사용 안함 설정
프로필 사진	profile_image	● 사용 안함 설정
카카오계정(이메일)	account_email	● 사용 안함 설정
성별	gender	● 사용 안함 설정
연령대	age_range	● 사용 안함 설정
생일	birthday	● 사용 안함 설정

필요한 정보 상태 변경

3. DB 덤프파일 최신본

exec / DB 덤프파일 폴더에 13개 SQL파일

4. 시연 시나리오

exec 폴더에 PDF파일