📕

[공통] 구미 2반 D209 포팅 메뉴얼

I. 기술스택

1. 프로젝트 기술 스택
    a. 이슈관리 : Jira
    b. 형상관리 : Gitlab
    c. 커뮤니케이션 : Mattermost, Notion, Google Docs
    d. 개발환경(IDE)
        i. InteiliJ
        ii. Visual Studio Code
        iii. Android Studio
        iv. UI : Figma
2. Database : MySQL Workbench 8.0
3. Server : AWS EC2, AWS S3
4. FrontEnd
    a. React 18.2.0
    b. tailwind CSS
5. BackEnd
    a. Java JDK 17
    b. Spring Boot 3.1.1
    c. Lombok, Swagger3, Querydsl-Jpa
6. Mobile
    a.
7. CI/CD
    a. Jenkins
    b. Nginx

II. 빌드

1. 환경변수 형태

    application.yml

```
## 보안키는 뒤의 5자리를 삭제 후 기재 함

<JWT>
spring:
  profiles:
    group:
      "dev-profile": "jwt"
      "prod-profile": "jwt"
    include: jwt
```

```
<Kakao OAUTH>
security:
    oauth2:
      client:
        registration:
          kakao:
            client-id: 66c5ba77d82e4dbed66a1f8fc91
            redirect-uri: http://i9d209.p.ssafy.io/oauth/callback/kakao
            authorization-grant-type: authorization_code
            scope: account_email,profile_nickname
            client-name: Kakao
            client-authentication-method: POST
          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

<AWS S3>
cloud:
  aws:
    credential:
      accessKey: AKIAQO6CUPHGBSC
      secretKey: 7dSYdzxNSWWmzPFZ+CG1cYZZpPm7IHTfJNK
    s3:
      bucket: petmily-pjt-bucket  # s3 버킷 이름
    region:
      static: ap-northeast-2  # region
      auto: false
    stack:
      auto: false
```

- application-jwt.yml

```
jwt:
  secret: ZG9uZ2h1bi1zaGFycC1kYnJ1YJH3ZWItcHJvagVjdC11c2luZy1qd3Qtc2VjcmV0L32RvbmhshdW4tc3ByaW5nLWJvb3Qtand0LWJhY2stZW5kLWFuZC66qc

  access:
    expiration: 20000
    header: Authorization

  refresh:
    expiration: 90
    header: Authorization-refresh
```

- application.properties

```
server.port=8081

<My SQL>
spring.jackson.time-zone=Asia/Seoul
spring.h2.console.enabled=true
spring.datasource.url=jdbc:mysql://i9d209.p.ssafy.io:3306/petmilydb?useSSL=false&allowPublicKeyRetrieval=true
spring.datasource.username=root
spring.datasource.password=qwe123
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

<JPA>
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect
spring.jpa.properties.hibernate.show_sql=false
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.hibernate.ddl-auto=update
spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect

<S3 upload max size>
spring.servlet.multipart.max-file-size=200MB
spring.servlet.multipart.max-request-size=200MB
```

- email.properties

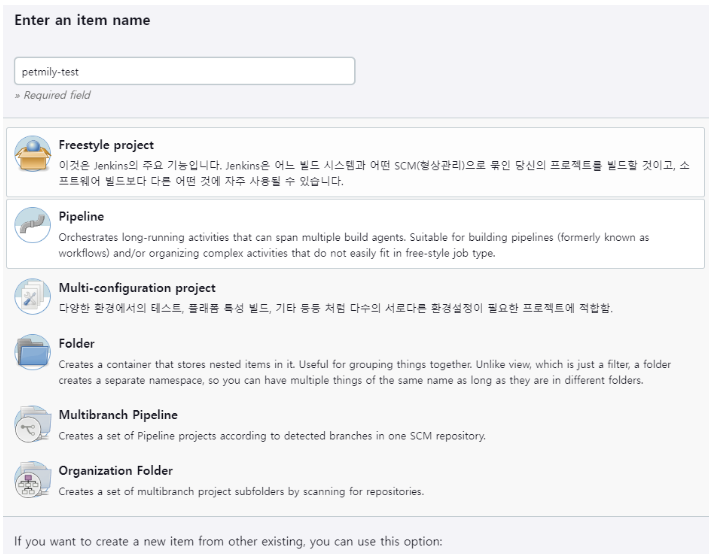
```
mail.smtp.auth=true
mail.smtp.starttls.required=true
mail.smtp.starttls.enable=true
mail.smtp.socketFactory.class=javax.net.ssl.SSLSocketFactory
mail.smtp.socketFactory.fallback=false
mail.smtp.port=465
mail.smtp.socketFactory.port=465
```

```
AdminMail.id = ryejinee@gmail.com
AdminMail.password = baluxdzsqnr
```

- firebase-spring

```
{
  "type": "service_account",
  "project_id": "petmily-2d449",
  "private_key_id": "41c8b28c425135cbf903c762335fae06332",
  "private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDc09GTmRJcjhI5\nDDAAuwU+pVG9uGXMI
  "client_email": "firebase-adminsdk-n5bdz@petmily-2d449.iam.gserviceaccount.com",
  "client_id": "115878982465747546709",
  "auth_uri": "https://accounts.google.com/o/oauth2/auth",
  "token_uri": "https://oauth2.googleapis.com/token",
  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",
  "client_x509_cert_url": "https://www.googleapis.com/robot/v1/metadata/x509/firebase-adminsdk-n5bdz%40petmily-2d449.iam.gservicea
  "universe_domain": "googleapis.com"
}
```

2. 빌드하기
   a. Front
      i. jenkins파이프라인 생성



Enter an item name

petmily-test

» 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.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
다양한 환경에서의 테스트, 플래폼 특성 빌드, 기타 등등 처럼 다수의 서로다른 환경설정이 필요한 프로젝트에 적합함.

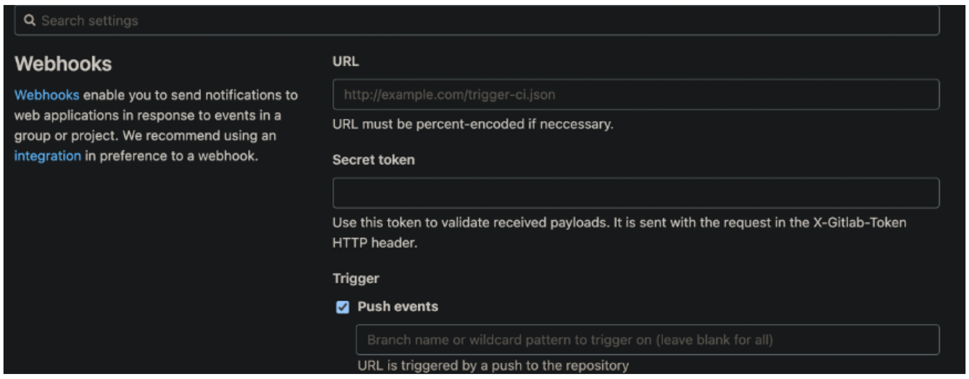
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

- webhooks 설정



Search settings

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

http://example.com/trigger-ci.json

URL must be percent-encoded if neccessary.

Secret token

Use this token to validate received payloads. It is sent with the request in the X-Gitlab-Token HTTP header.

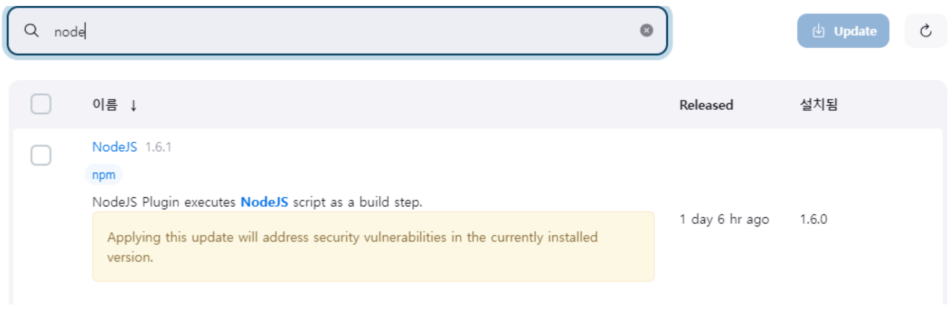
Trigger

✓ Push events

Branch name or wildcard pattern to trigger on (leave blank for all)

URL is triggered by a push to the repository

- jenkins에 node.js 추가
  - jenkins 관리 → plugins



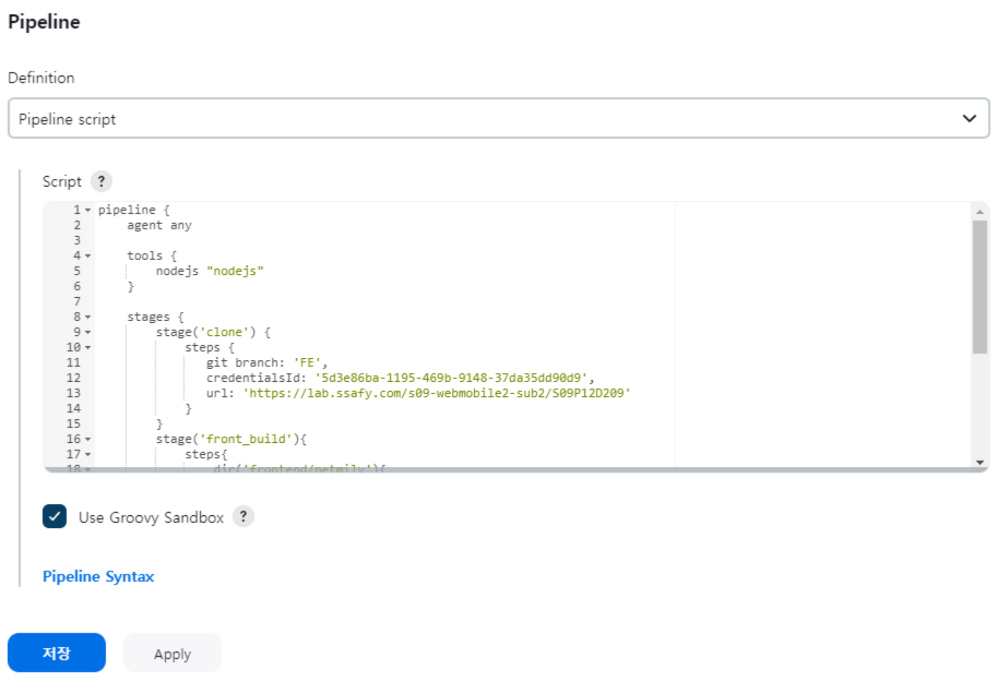
node

Update

	이름 ↓	Released	설치됨
☐	NodeJS 1.6.1 npm NodeJS Plugin executes **NodeJS** script as a build step. Applying this update will address security vulnerabilities in the currently installed version.	1 day 6 hr ago	1.6.0

ii. jenkins 파이프라인 구성

- 스크립트 생성



Pipeline

Definition

Pipeline script

Script ?

```
pipeline {
    agent any

    tools {
        nodejs "nodejs"
    }

    stages {
        stage('clone') {
            steps {
               git branch: 'FE',
               credentialsId: '5d3e86ba-1195-469b-9148-37da35dd90d9',
               url: 'https://lab.ssafy.com/s09-webmobile2-sub2/S09P12D209'
            }
        }
        stage('front_build'){
            steps{
```

☑ Use Groovy Sandbox ?

Pipeline Syntax

저장 Apply

```
pipeline {
    agent any

    tools {
        nodejs "nodejs"
    }

    stages {
        stage('clone') {
            steps {
               git branch: 'FE',
               credentialsId: '5d3e86ba-1195-469b-9148-37da35dd90d9',
               url: 'https://lab.ssafy.com/s09-webmobile2-sub2/S09P12D209'
            }
        }
        stage('front_build'){
            steps{
                dir('frontend/petmily'){
                    sh 'npm install'
                    sh 'CI=false npm run build'
                }
            }
        }
         stage('deploy'){
            steps{
                sh 'sudo docker-compose up -d --build'
            }
        }
    }
}
```

iii. nginx 설정(frontend/nginx/default.conf)

```
server {
        listen 80 default_server;
        listen [::]:80 default_server;

        server_name i9d209.p.ssafy.io;

        location / {
                root /usr/share/nginx/html;
                index  index.html;
                try_files $uri $uri/ /index.html;
        }

        location /api/ {
                proxy_pass http://i9d209.p.ssafy.io:8081/;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-NginX-Proxy true;
                client_max_body_size 100m;
                proxy_connect_timeout 300s;
                proxy_read_timeout 600s;
                proxy_send_timeout 600s;
        }
}
```

iv. nginx 설치 및 nginx + react 도커파일 생성

- EC2 인스턴스에 접속해서 nginx 설치

```
FROM nginx:alpine
RUN rm -rf /etc/nginx/conf.d/default.conf
COPY /nginx/default.conf /etc/nginx/conf.d/default.conf

RUN rm -rf /usr/share/nginx/html/*
COPY /build /usr/share/nginx/html

EXPOSE 80
ENTRYPOINT ["nginx", "-g", "daemon off;"]
```

```
sudo yum install nginx
```

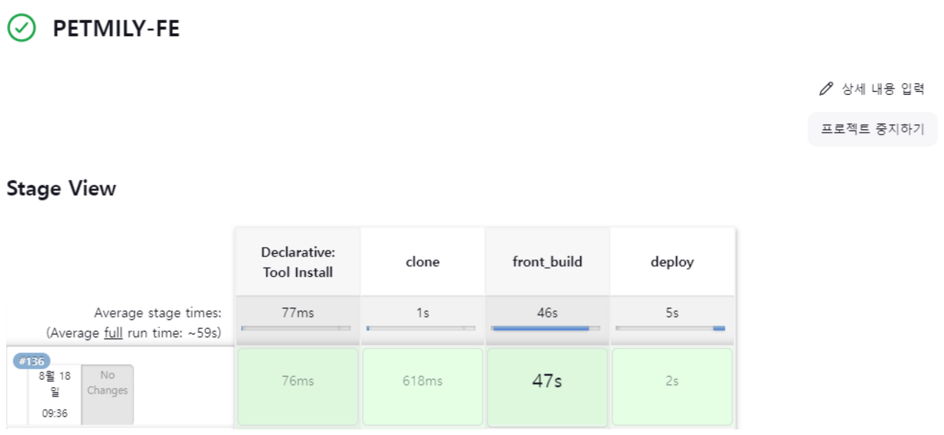
- nginx + react 도커파일 생성

```
FROM nginx:alpine
RUN rm -rf /etc/nginx/conf.d/default.conf
COPY /nginx/default.conf /etc/nginx/conf.d/default.conf

RUN rm -rf /usr/share/nginx/html/*
COPY /build /usr/share/nginx/html

EXPOSE 80
ENTRYPOINT ["nginx", "-g", "daemon off;"]
```

- docker compose 파일 설정(docker-compose.yml)

```
version: "3"
services:
  nginx:
    build:
      context: ./frontend/petmily
    ports:
      - 80:80
```

v. 결과


PETMILY-FE

Stage View

	Declarative: Tool Install	clone	front_build	deploy
Average stage times: (Average full run time: ~59s)	77ms	1s	46s	5s
#136 8월 18 일 09:36 No Changes	76ms	618ms	47s	2s

b. Back

i. Gradle 실행 (./gradlew build)

ii. Bootjar 실행 (java -jar (PJT).jar)

c. Mobile

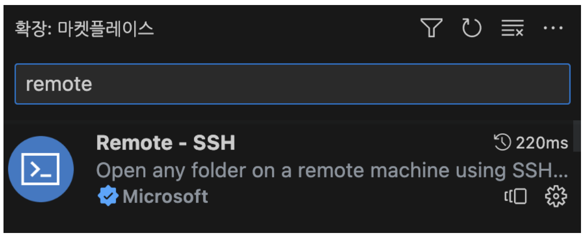
i. /app/release에서 petmily.apk 실행

3. 자동배포

a. EC2 ubuntu 연결

i. MobaXterm 설치

- https://mobaxterm.mobatek.net/download-home-edition.html(portable edition, installer edition 중 아무거나 설치)
  - 참고로 mac은 MobaXterm을 지원하지 않기 때문에 terminal에서 실행하거나 vscode에서 remote-ssh라는 확장 프로그램을 깔아서 사용

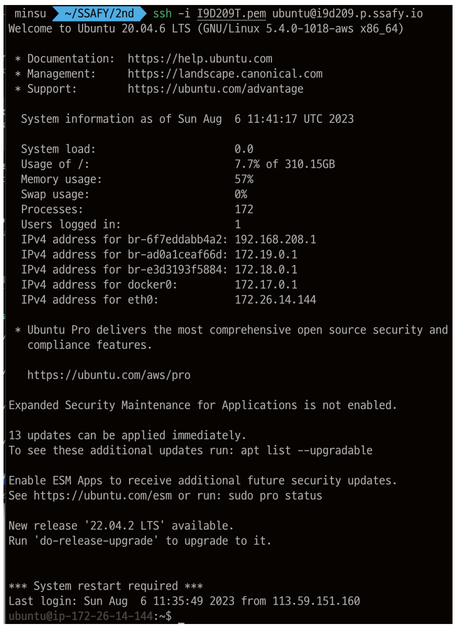


ii. 다운받은 pem파일이 있는 위치로 이동

`cd “pem파일 경로”`

ex) 만약에 pem파일이 “SSAFY”디렉토리 안의 “2nd” 디렉토리에 있다면? `cd SSAFY/2nd`

ii. `ssh -i I9D209T.pem ubuntu@i9d209.p.ssafy.io` 명령어 실행



```
 minsu  ~/SSAFY/2nd  ssh -i I9D209T.pem ubuntu@i9d209.p.ssafy.io
Welcome to Ubuntu 20.04.6 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 Sun Aug  6 11:41:17 UTC 2023

  System load:                     0.0
  Usage of /:                      7.7% of 310.15GB
  Memory usage:                    57%
  Swap usage:                      0%
  Processes:                       172
  Users logged in:                 1
  IPv4 address for br-6f7eddabb4a2: 192.168.208.1
  IPv4 address for br-ad0a1ceaf66d: 172.19.0.1
  IPv4 address for br-e3d3193f5884: 172.18.0.1
  IPv4 address for docker0:         172.17.0.1
  IPv4 address for eth0:            172.26.14.144

 * 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.

13 updates can be applied immediately.
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

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


*** System restart required ***
Last login: Sun Aug  6 11:35:49 2023 from 113.59.151.160
ubuntu@ip-172-26-14-144:~$ 
```

b. EC2 docker 설치

   i. 설치 명령어 및 실행

```
$ sudo apt update
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common
$ sudo wget -qO- https://get.docker.com/ | sh
```

```
$ sudo systemctl start docker
$ sudo systemctl enable docker
```

c. docker-compose 설치 (jenkins와 mysql이 docker container로 실행)

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

- docker-compose.yml

```
version: "3"
services:
  jenkins:
    image: jenkins/jenkins:jdk17
    container_name: petmily_jenkins
    user: root
    volumes:
      - /home/ubuntu/jenkins:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
    ports:
      - "2090:8080"

  db:
    image: mysql
    restart: always
    environment:
      MYSQL_ROOT_PASSWORD: qwe123
      TZ: Asia/Seoul
      MYSQL_DATABASE: petmilydb
      MYSQL_USER: root
      MYSQL_PASSWORD: qwe123
    container_name: petmilydb
    volumes:
      - /home/ubuntu/db:/var/lib/mysql
      - /home/ubuntu/my.cnf:/etc/mysql/my.cnf
    ports:
      - "3306:3306"
```

d. jenkins 설정

i. jenkins 내 docker 설치

```
# jenkins container 접속
docker exec -it jenkins /bin/bash

# Docker 설치
## - Old Version Remove
apt-get remove docker docker-engine docker.io containerd runc

## - Setup Repo
apt-get update
apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | gpg --dearmor -o /etc/apt/keyrings/docker.gpg
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/debia
  $(lsb_release -cs) stable" | tee /etc/apt/sources.list.d/docker.list > /dev/null

## - Install Docker Engine
apt-get update
apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

ii. jenkins 설정

http://i9d209.p.ssafy.io:2090 으로 jenkins 접속 가능

~~ID : admin // PW: qwe123!~~

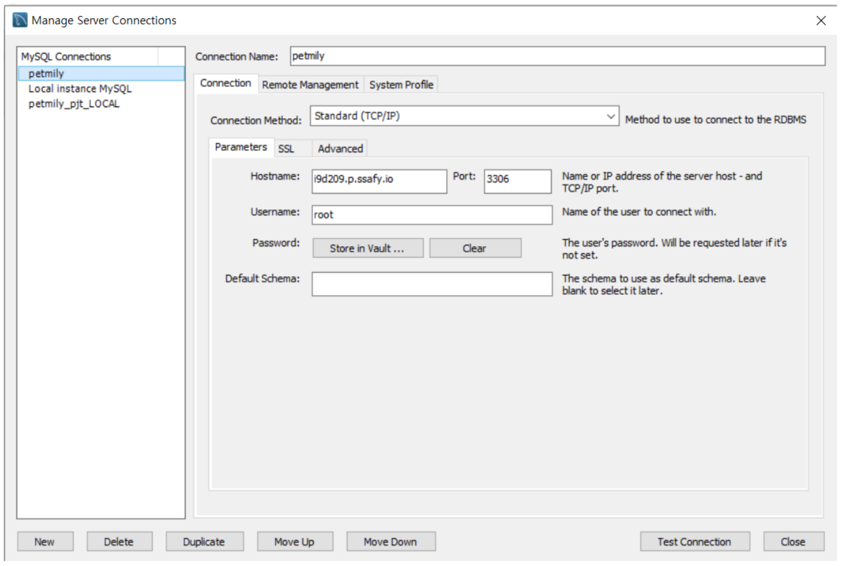
```
$ docker exec -it jenkins /bin/bash
$ cat /var/jenkins_home/secrets/initialAdminPassword
```

e. 파이프라인 설정

- pipeline

```
pipeline {

    agent any

    environment {
        imagename = "ryejin/petmily"
        registryCredential = 'docker_petmily'
        dockerImage = ''
    }

    stages {
        stage('git clone') {
            steps {
                 git branch: 'BE', credentialsId: 'gitlab_petmily', url: 'https://lab.ssafy.com/s09-webmobile2-sub2/S09P12D20
            }
        }

        stage('build') {
            steps {
                dir('back_end/petmily_pjt_temp') {
                    sh "chmod +x gradlew"
                    sh "./gradlew clean bootJar"
                }
            }
        }

         stage('docker-build'){
            steps {
                echo 'Build Docker'
                dir('back_end/petmily_pjt_temp'){
                    script {
                        sh "pwd"
                        dockerImage = docker.build imagename

                    }
                }
            }
        }

        stage('docker-push'){
            steps{
                echo 'Docker Delete and Push'
                sshagent(credentials: ['petmily-ec2-key']) {
                    sh '''
                        ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io "docker stop $(docker ps -aq --filter ancest
                        ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io "docker rm -f $(docker ps -aq --filter ances
                        ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io "docker rmi ryejin/petmily:1.0"

                    EXISTING_CONTAINER=$(ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io "docker ps -q --filter 'exp
                    if [ ! -z "$EXISTING_CONTAINER" ]; then
                        ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io "docker stop $EXISTING_CONTAINER"
                    fi
                    '''
                }
                script {
                    docker.withRegistry('', registryCredential) {
                        dockerImage.push("1.0")
                    }
                }
            }
        }
        stage('SSH-Server-EC2'){
            steps {
                echo 'SSH'

                sshagent(credentials: ['petmily-ec2-key']) {
                    sh 'ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io "who am i"'
                    sh "ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io 'docker pull ryejin/petmily:1.0'"
                    sh "ssh -o StrictHostKeyChecking=no ubuntu@i9d209.p.ssafy.io 'docker run -d -p 8081:8081 ryejin/petmily:1
                }
            }
        }

    }
}
```

4. DB 설정 및 계정
    a. My SQL workbench 8.0
        i. ~~Username : root // PW : qwe123~~



Manage Server Connections

MySQL Connections: petmily, Local instance MySQL, petmily_pjt_LOCAL

Connection Name: petmily

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

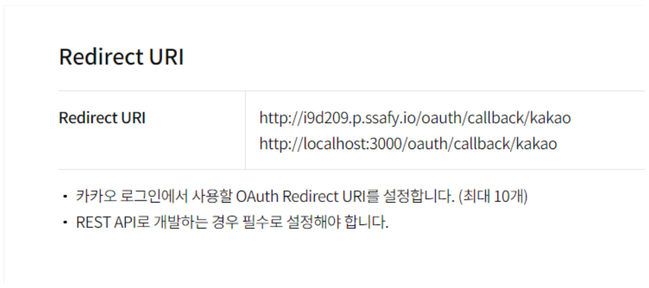
Hostname: i9d209.p.ssafy.io — Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: root — Name of the user to connect with.

Password: Store in Vault ... / Clear — The user's password. Will be requested later if it's not set.

Default Schema: — The schema to use as default schema. Leave blank to select it later.

5. 외부서비스
    a. 카카오 로그인 (https://developers.kakao.com/)
        i. 어플리케이션 등록 후 도메인 등록



Redirect URI

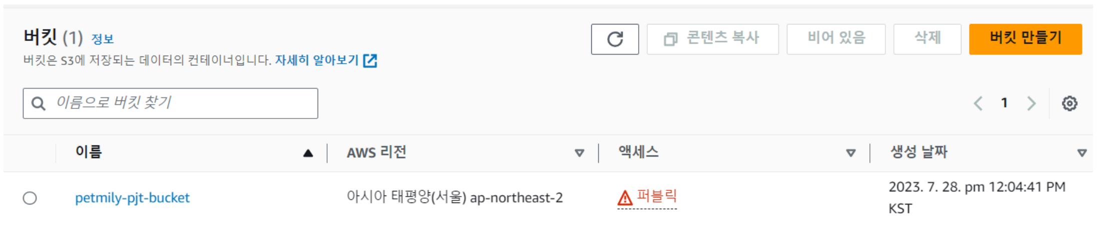
Redirect URI	http://i9d209.p.ssafy.io/oauth/callback/kakao http://localhost:3000/oauth/callback/kakao

- 카카오 로그인에서 사용할 OAuth Redirect URI를 설정합니다. (최대 10개)
- REST API로 개발하는 경우 필수로 설정해야 합니다.

ii. 인가 코드는 1. 환경변수 - application.yml 참고

b. AWS S3

i. 버킷 생성



버킷 (1) 정보

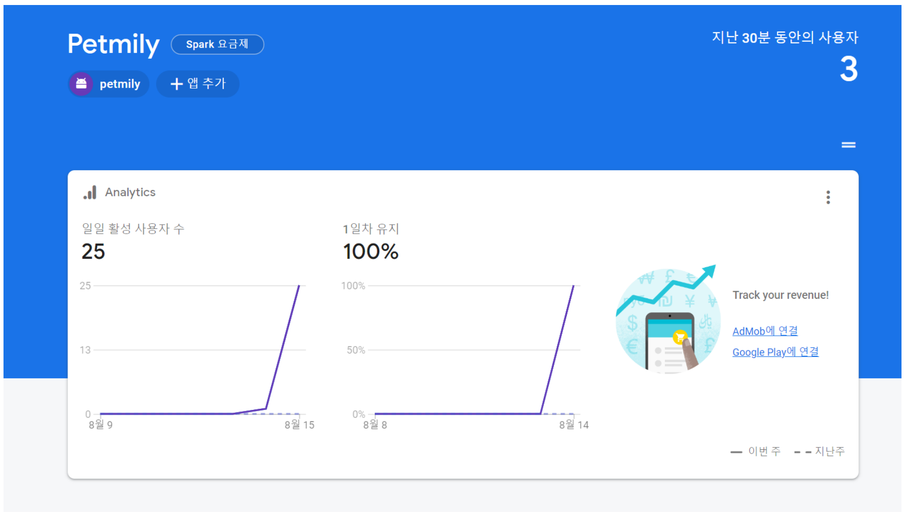
버킷은 S3에 저장되는 데이터의 컨테이너입니다. 자세히 알아보기

	이름	AWS 리전	액세스	생성 날짜
○	petmily-pjt-bucket	아시아 태평양(서울) ap-northeast-2	⚠ 퍼블릭	2023. 7. 28. pm 12:04:41 PM KST

ii. S3 Credential은 1. 환경변수 - application.yml 참고

c. Firebase

i. 프로젝트 생성



ii. seceret key 1. 환경변수 - firebase-spring 참고