포팅 매뉴얼

1. 빌드 및 배포

1-1. 실행 환경

프론트엔드

- Visual Studio Code: 1.84.1
    - Node.js: 18.17.0

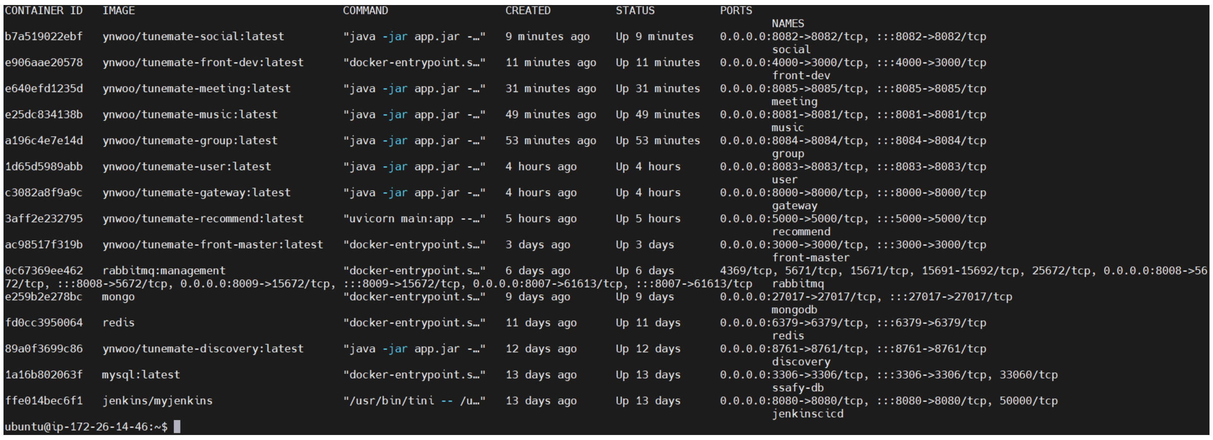
백엔드

- IntelliJ IDEA: 2023.1.3 (Ultimate Edition)
    - JVM: Azul Zulu version 17.0.2
- PyCharm: 2023.2.4 (Community Edition)
    - Python: 3.11.4

서버

- MobaXterm: Personal Edition v23.2 Build 5082

1-2. 배포 환경



```
CONTAINER ID   IMAGE                                COMMAND                  CREATED          STATUS           PORTS
                                                                                                                      NAMES
b7a519022ebf   ynwoo/tunemate-social:latest         "java -jar app.jar -…"   9 minutes ago    Up 9 minutes     0.0.0.0:8082->8082/tcp, :::8082->8082/tcp
                                                                                                                      social
e906aae20578   ynwoo/tunemate-front-dev:latest      "docker-entrypoint.s…"   11 minutes ago   Up 11 minutes    0.0.0.0:4000->3000/tcp, :::4000->3000/tcp
                                                                                                                      front-dev
e640efd1235d   ynwoo/tunemate-meeting:latest        "java -jar app.jar -…"   31 minutes ago   Up 31 minutes    0.0.0.0:8085->8085/tcp, :::8085->8085/tcp
                                                                                                                      meeting
e25dc834138b   ynwoo/tunemate-music:latest          "java -jar app.jar -…"   49 minutes ago   Up 49 minutes    0.0.0.0:8081->8081/tcp, :::8081->8081/tcp
                                                                                                                      music
a196c4e7e14d   ynwoo/tunemate-group:latest          "java -jar app.jar -…"   53 minutes ago   Up 53 minutes    0.0.0.0:8084->8084/tcp, :::8084->8084/tcp
                                                                                                                      group
1d65d5989abb   ynwoo/tunemate-user:latest           "java -jar app.jar -…"   4 hours ago      Up 4 hours       0.0.0.0:8083->8083/tcp, :::8083->8083/tcp
                                                                                                                      user
c3082a8f9a9c   ynwoo/tunemate-gateway:latest        "java -jar app.jar -…"   4 hours ago      Up 4 hours       0.0.0.0:8000->8000/tcp, :::8000->8000/tcp
                                                                                                                      gateway
3aff2e232795   ynwoo/tunemate-recommend:latest      "uvicorn main:app --…"   5 hours ago      Up 5 hours       0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
                                                                                                                      recommend
ac98517f319b   ynwoo/tunemate-front-master:latest   "docker-entrypoint.s…"   3 days ago       Up 3 days        0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
                                                                                                                      front-master
0c67369ee462   rabbitmq:management                  "docker-entrypoint.s…"   6 days ago       Up 6 days        4369/tcp, 5671/tcp, 15671/tcp, 15691-15692/tcp, 25672/tcp, 0.0.0.0:8008->56
72/tcp, :::8008->5672/tcp, 0.0.0.0:8009->15672/tcp, :::8009->15672/tcp, 0.0.0.0:8007->61613/tcp, :::8007->61613/tcp   rabbitmq
e259b2e278bc   mongo                                "docker-entrypoint.s…"   9 days ago       Up 9 days        0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
                                                                                                                      mongodb
fd0cc3950064   redis                                "docker-entrypoint.s…"   11 days ago      Up 11 days       0.0.0.0:6379->6379/tcp, :::6379->6379/tcp
                                                                                                                      redis
89a0f3699c86   ynwoo/tunemate-discovery:latest      "java -jar app.jar -…"   12 days ago      Up 12 days       0.0.0.0:8761->8761/tcp, :::8761->8761/tcp
                                                                                                                      discovery
1a16b802063f   mysql:latest                         "docker-entrypoint.s…"   13 days ago      Up 13 days       0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp
                                                                                                                      ssafy-db
ffe014bec6f1   jenkins/myjenkins                    "/usr/bin/tini -- /u…"   13 days ago      Up 13 days       0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp
                                                                                                                      jenkinscicd
ubuntu@ip-172-26-14-46:~$
```

서버

- Server: AWS EC2 Ubuntu 20.04.6 LTS

- Docker: 24.0.7
  - Jenkins: 2.429
    - SSH Agent
    - Docker
    - Docker Commons
    - Docker Pipeline
    - Docker API
    - Generic Webhook Trigger
    - GitLab
    - GitLab API
    - GitLab Authentication
- Nginx: nginx/1.18.0 (Ubuntu)

DB

- MySQL: 8.2.0
- Redis: 7.2.2
- MongoDB: 7.0.2

Message Queue

- RabbitMQ: 3.12.8

1-3. 서버 구축

초기 설정

```
sudo apt-get update // 인스턴스에 설치된 패키지 목록을 최신화
sudo apt-get upgrade // 인스턴스에 설치된 패키지들을 최신 버전으로 업그레이드
```

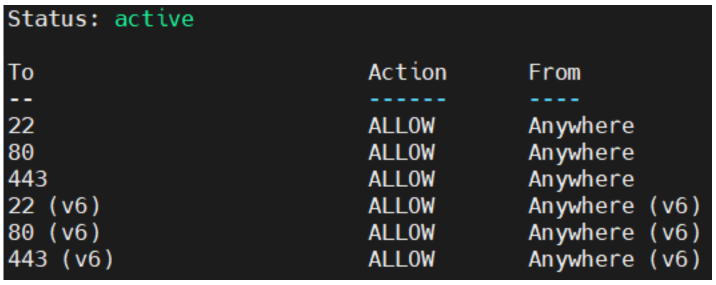
```
sudo timedatectl set-timezone Asia/Seoul
```

Nginx 설치

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install nginx
```

방화벽 설정

- `sudo ufw status`



```
Status: active

To                         Action      From
--                         ------      ----
22                         ALLOW       Anywhere
80                         ALLOW       Anywhere
443                        ALLOW       Anywhere
22 (v6)                    ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)
443 (v6)                   ALLOW       Anywhere (v6)
```

HTTPS, SSL 설정

- Certbot

```
# snap을 이용하여 core 설치 -> snap을 최신 버전으로 유지하기 위해 설치
$ sudo snap install core

# core를 refresh 해준다.
$ sudo snap refresh core

# 기존에 잘못된 certbot이 설치되어있을 수도 있으니 삭제 해준다.
$ sudo apt remove certbot

# certbot 설치
$ sudo snap install --classic certbot

# certbot 명령을 로컬에서 실행할 수 있도록 snap의 certbot 파일을 로컬의 cerbot과 링크(연결) 시켜준다. -s 옵션은 심볼릭링크를 하겠다는 것.
$ ln -s /snap/bin/certbot /usr/bin/certbot
```

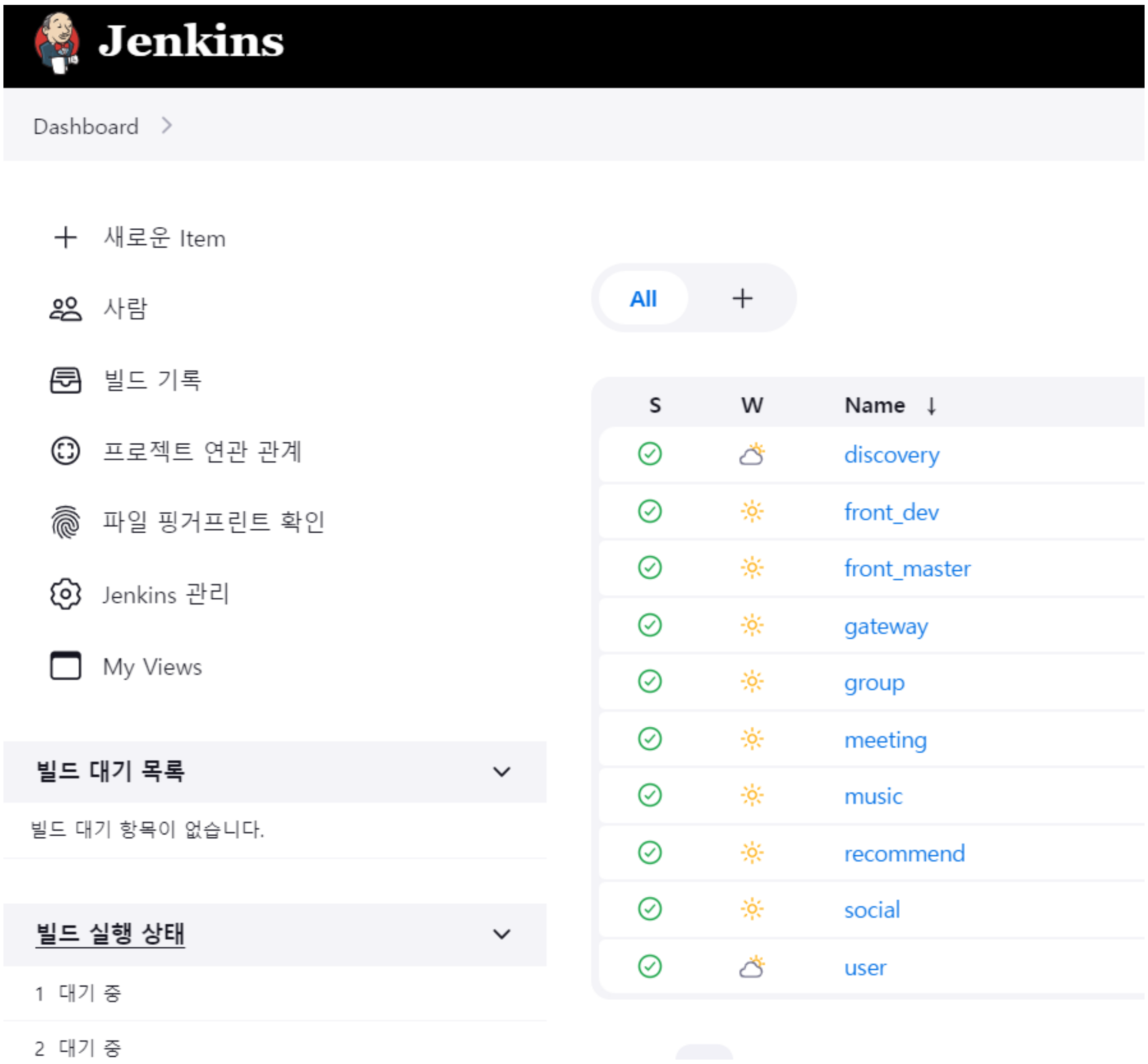
```
sudo certbot --nginx
```

도커 설치

```
sudo apt-get -y install apt-transport-https ca-certificates curl gnupg-agent software-properties-common

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 $(lsb_release -cs) stable"

sudo apt-get update && sudo apt-get install docker-ce docker-ce-cli containerd.io


sudo apt install jq

DCVERSION=$(curl --silent https://api.github.com/repos/docker/compose/releases/latest | jq .name -r)

DCDESTINATION=/usr/bin/docker-compose

sudo curl -L https://github.com/docker/compose/releases/download/${DCVERSION}/docker-compose-$(uname -s)-$(uname -m) -o $DCDESTINATION

sudo chmod 755 $DCDESTINATION

docker-compose -v
```

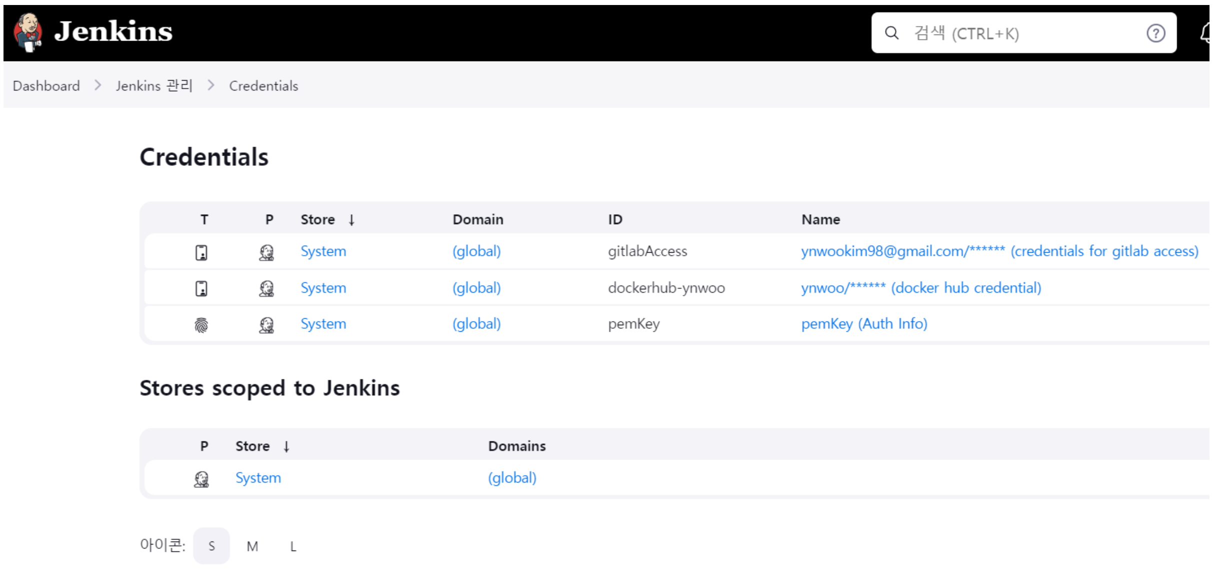
젠킨스 컨테이너 생성

- 대시보드



S	W	Name ↓
✓	⛅	discovery
✓	☀	front_dev
✓	☀	front_master
✓	☀	gateway
✓	☀	group
✓	☀	meeting
✓	☀	music
✓	☀	recommend
✓	☀	social
✓	⛅	user

- Credentials 생성



Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	gitlabAccess	ynwookim98@gmail.com/****** (credentials for gitlab access)
		System	(global)	dockerhub-ynwoo	ynwoo/****** (docker hub credential)
		System	(global)	pemKey	pemKey (Auth Info)

Stores scoped to Jenkins

P	Store ↓	Domains
	System	(global)

Gitlab Webhook 연결

- Gitlab 레포지토리로 이동

- 설정 → Webhook
- URL에 Jenkins의 Gitlab Webhook URL을 입력
- Secret token에 Jenkins에서 생성한 Secret token을 입력

1-4. 설정 파일

Nginx

- `/etc/nginx/conf.d/default.conf`

```
upstream frontend {
    server 0.0.0.0:3000;
}

upstream backend {
    server 0.0.0.0:8000;
}

server {
    server_name 54.180.143.51 k9a603.p.ssafy.io;

    location / {
        return 301 $scheme://tunemate.co.kr$request_uri;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/tunemate.co.kr/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/tunemate.co.kr/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 {
  server_name tunemate.co.kr www.tunemate.co.kr;

  location / {
    proxy_pass http://frontend;
    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;
  }

  location /api {
    proxy_pass http://backend;
    proxy_http_version 1.1;
          proxy_set_header Upgrade $http_upgrade;
          proxy_set_header Connection "upgrade";
    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;
  }

  location /webjars {
    proxy_pass http://backend;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
  }

  location /v3 {
                proxy_pass http://backend;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
        }

  location /user-service/v3 {
                proxy_pass http://backend;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
        }

  location /social-service/v3 {
                proxy_pass http://backend;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
        }

  location /music-service/v3 {
                proxy_pass http://backend;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
        }
  location /group-service/v3 {
                proxy_pass http://backend;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
        }
  location /meeting-service/v3 {
                proxy_pass http://backend;
                proxy_http_version 1.1;
                proxy_set_header Upgrade $http_upgrade;
                proxy_set_header Connection "upgrade";
                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;
        }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/tunemate.co.kr/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/tunemate.co.kr/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 ($host = www.tunemate.co.kr) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


    if ($host = tunemate.co.kr) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


  listen 80;
  server_name tunemate.co.kr www.tunemate.co.kr;
    return 404; # managed by Certbot




}

server {
    if ($host = k9a603.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot


    listen 80;
    server_name 54.180.143.51 k9a603.p.ssafy.io;
    return 404; # managed by Certbot


}
```

Jenkins 프론트엔드 배포 파이프라인

- develop 브랜치

```
pipeline {
    agent any
    environment {
        VERSION = "latest"
        DOCKERHUB_REPOSITORY_FRONT = "ynwoo/tunemate-front-dev"
        DOCKERHUB_CREDENTIAL = credentials('dockerhub-ynwoo')
        CONTAINER_NAME_FRONT = "front-dev"
        SSH_CONNECTION = "ubuntu@k9a603.p.ssafy.io"
        PORT_FRONT = "4000"
    }
    stages {
        stage('Git Clone') {
            steps {
                sh "#### Git Clone Start ####"
                git branch: 'develop', credentialsId: 'gitlabAccess', url: 'https://lab.ssafy.com/s09-final/S09P31A603.git'
                sh "#### Git Clone Success ####"
                sh "cp /var/jenkins_home/.env /var/jenkins_home/workspace/front_dev/FE/"
            }
        }

        stage("Build Images") {
            steps {
                sh "docker compose build"
            }

        }
        stage('Push Images'){
            steps {
                sh "echo $DOCKERHUB_CREDENTIAL_PSW | docker login -u $DOCKERHUB_CREDENTIAL_USR --password-stdin"
                sh "docker push $DOCKERHUB_REPOSITORY_FRONT:$VERSION"
            }
        }
        stage('Deploy Frontend Server') {
            steps {
                sshagent(credentials: ['pemKey']) {
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker rm -f $CONTAINER_NAME_FRONT'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker rmi -f $DOCKERHUB_REPOSITORY_FRONT:$VERSION'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker pull $DOCKERHUB_REPOSITORY_FRONT:$VERSION'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'echo y | docker image prune'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker images'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker run -d --name $CONTAINER_NAME_FRONT -p $PORT_FRONT
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker ps'"
                }
            }
        }
    }
}
```

- master 브랜치

```
pipeline {
    agent any
    environment {
        VERSION = "latest"
        DOCKERHUB_REPOSITORY_FRONT = "ynwoo/tunemate-front-master"
        DOCKERHUB_CREDENTIAL = credentials('dockerhub-ynwoo')
        CONTAINER_NAME_FRONT = "front-master"
        SSH_CONNECTION = "ubuntu@k9a603.p.ssafy.io"
        PORT_FRONT = "3000"
    }
    stages {
        stage('Git Clone') {
            steps {
                sh "#### Git Clone Start ####"
                git branch: 'master', credentialsId: 'gitlabAccess', url: 'https://lab.ssafy.com/s09-final/S09P31A603.git'
                sh "#### Git Clone Success ####"
                sh "cp /var/jenkins_home/.env /var/jenkins_home/workspace/front_master/FE/"
            }
        }

        stage("Build Images") {
            steps {
                sh "docker compose build"
            }

        }
        stage('Push Images'){
            steps {
                sh "echo $DOCKERHUB_CREDENTIAL_PSW | docker login -u $DOCKERHUB_CREDENTIAL_USR --password-stdin"
                sh "docker images"
                sh "docker push $DOCKERHUB_REPOSITORY_FRONT:$VERSION"
            }
        }
        stage('Deploy Frontend Server') {
            steps {
                sshagent(credentials: ['pemKey']) {
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker rm -f $CONTAINER_NAME_FRONT'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker rmi -f $DOCKERHUB_REPOSITORY_FRONT:$VERSION'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker pull $DOCKERHUB_REPOSITORY_FRONT:$VERSION'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'echo y | docker image prune'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker images'"
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker run -d --name $CONTAINER_NAME_FRONT -p $PORT_FRONT
                    sh "ssh -o StrictHostKeyChecking=no $SSH_CONNECTION 'docker ps'"
                }
            }
        }
    }
}
```

1-5. 환경 변수

프론트엔드

`/home/ubuntu/config/fe/.env`

```
NEXT_PUBLIC_SPOTIFY_CLIENT_ID=30fd5d15cb8a40dcbexxxxxxxxxxxxxxx
```

백엔드

`/home/ubuntu/config/.env`

```
# Discovery
eureka-server-port=8761

# Gateway
gateway-server-port=8000

eureka-service-url=http://k9a603.p.ssafy.io:8761/eureka

jwt-private-key=tunemate-jwt-private-key-for-access-token
```

`/home/ubuntu/config/group.env`

```
spring-application-name=group-service

datasource-password=kim**************
datasource-url=jdbc:mysql://k9A603.p.ssafy.io:3306/GROUPSERVICE
datasource-username=root

eureka-service-url=http://k9A603.p.ssafy.io:8761/eureka

openapi-service-version=1.0.0
openapi-service-url=https://tunemate.co.kr/api/v1/group-service
```

`/home/ubuntu/config/meeting.env`

```
DATABASE_USERNAME=root
DATABASE_PASSWORD=kim****************
```

```
DATABASE_MEETING_URL=jdbc:mysql://k9A603.p.ssafy.io:3306/MEETING?serverTimezone=UTC&useUnicode=true&characterEncoding=utf8
EUREKA_SERVICE_URL=http://k9a603.p.ssafy.io:8761/eureka
HOSTNAME=k9A603.p.ssafy.io
OPENAPI_SERVICE_VERSION=1.0.0
OPENAPI_SERVICE_URL=https://tunemate.co.kr/api/v1/meeting-service
```

`/home/ubuntu/config/music.env`

```
DATABASE_USERNAME=root
DATABASE_PASSWORD=kim*********
DATABASE_MUSIC_URL=jdbc:mysql://k9A603.p.ssafy.io:3306/MUSIC?serverTimezone=UTC&useUnicode=true&characterEncoding=utf8
EUREKA_SERVICE_URL=http://k9a603.p.ssafy.io:8761/eureka
HOSTNAME=k9A603.p.ssafy.io
OPENAPI_SERVICE_VERSION=1.0.0
OPENAPI_SERVICE_URL=https://tunemate.co.kr/api/v1/music-service
```

`/home/ubuntu/config/recommend.env`

```
DATABASE_URL=k9A603.p.ssafy.io
DATABASE_USERNAME=root
DATABASE_PASSWORD=kim********
EUREKA_SERVER=http://k9a603.p.ssafy.io:8761/eureka
APP_NAME=recommendation-service
INSTANCE_PORT=5000
INSTANCE_HOST=k9A603.p.ssafy.io
DATABASE_PORT=3306
```

`/home/ubuntu/config/social.env`

```
DATABASE_SOCIAL_URL=jdbc:mysql://k9A603.p.ssafy.io:3306/SOCIAL?serverTimezone=UTC&useUnicode=true&characterEncoding=utf8
DATABASE_USERNAME=root
DATABASE_PASSWORD=kim**********

EUREKA_SERVICE_URL=http://k9a603.p.ssafy.io:8761/eureka

SPRING_APP_NAME=social-service
OPENAPI_SERVICE_VERSION=1.0.0
OPENAPI_SERVICE_URL=https://tunemate.co.kr/api/v1/social-service
MONGODB_URI=mongodb+srv://S09P31A603:VkAd*******@ssafy.ngivl.mongodb.net/S09P31A603?authSource=admin

HOSTNAME=tunemate.co.kr

RABBITMQ_PORT=8008
RABBITMQ_USERNAME=a603
RABBITMQ_PASSWORD=kim*******
RABBITMQ_PORT_TWO=8007
```

`/home/ubuntu/config/user.env`

```
client-id=30fd5d15cb*****************************
client-secret=cf03d3*****************************
datasource-password=kim********
datasource-url=jdbc:mysql://k9A603.p.ssafy.io:3306/USERSERVICE
datasource-username=root
eureka-service-url=http://k9A603.p.ssafy.io:8761/eureka
jwt-private-key=tunemate-jwt-private-key-for-access-token
redirect-uri=https://tunemate.co.kr/api/v1/user-service/login/oauth2/code/spotify
HOST_NAME=k9A603.p.ssafy.io
HOSTNAME=k9A603.p.ssafy.io
spring-application-name=user-service
redis-url=k9a603.p.ssafy.io
redis-port=6379
redis-password=kim************
openapi-service-version=1.0.0
openapi-service-url=https://tunemate.co.kr/api/v1/user-service
authentication-success-redirect-url=https://tunemate.co.kr/
cookie-domain=tunemate.co.kr
```

2. 외부 서비스

2-1. Spotify API

https://developer.spotify.com/

- Create App으로 앱 생성
- Settings의 **Redirect URIs**
  `https://tunemate.co.kr/api/v1/user-service/login/oauth2/code/spotify`
- **User Management에서 계정 등록**
- **Client ID 및 Client secret 키 사용**