



Cookscape 포팅 매뉴얼

목차

1. 개발 버전 및 환경
 2. 설치
 - 2.1 MySQL
 - 2.2 Nginx
 - 2.3 Docker
 - 2.4 Jenkins
 3. 배포
 - 3.1 Frontend
 - 3.2 Backend
4. 계정 정보

1. 개발 버전 및 환경

Frontend	Backend	Unity
<ul style="list-style-type: none">• React.js 18.2• Redux 8.0.5• Router 6.10.0• Visual Studio Code	<ul style="list-style-type: none">• Java 11• SpringBoot 2.7.9• Spring Data JPA• Swagger• IntelliJ IDEA	<ul style="list-style-type: none">• Unity 2021.3.19f1• Visual Studio 2022• Visual Studio Code
Server	CI/CD	Database
<ul style="list-style-type: none">• AWS ec2 (Ubuntu 20.04.6)• Nginx 1.18.0• Docker 23.0.3	<ul style="list-style-type: none">• Jenkins• GitLab	<ul style="list-style-type: none">• MySQL 8.0.32

2. 설치

2.1 MySQL

- MySQL 설치

```
# 패키지 인스톨러 업데이트
sudo apt update

# MySQL 설치
sudo apt install mysql-server
```

- MySQL 실행

```
mysql -u root -p
```

- 데이터베이스 생성

```
create database cookscape_db;
```

- 계정 생성 후 권한 부여

```
# 계정생성
create user 'cookscape_ssafy'@'%'identified by 'cookscape109';

# 권한부여
grant all on cookscape_db.* to 'cookscape_ssafy'@'%';
```

2.2 Nginx

- Nginx 설치

```
sudo apt update
sudo apt install nginx
```

- Nginx 방화벽 포트 허용

```
sudo ufw allow 'Nginx full'
```

- SSL 인증서 발급

```
sudo apt update
sudo apt upgrade -y
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d j8b109.p.ssafy.io
```

- nginx 설정 세팅 - SSL 적용

```
sudo vim /etc/nginx/conf.d/sites.conf
```

```
#sites.conf
server {
    server_name j8b109.p.ssafy.io;

    location / {
        proxy_pass http://j8b109.p.ssafy.io:3000;
    }

    location /api {
        proxy_pass https://j8b109.p.ssafy.io:9999;
    }

    location /test {
        proxy_pass https://j8b109.p.ssafy.io:8888;
    }

    location /jenkins {
```

```

        proxy_pass http://j8b109.p.ssafy.io:8080;
    }

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/j8b109.p.ssafy.io/fullchain.pem; # managed by Certbot
    ssl_certificate_key /etc/letsencrypt/live/j8b109.p.ssafy.io/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 ($host = j8b109.p.ssafy.io) {
        return 301 https://$host$request_uri;
    } # managed by Certbot

    listen 80;
    server_name j8b109.p.ssafy.io;
    return 404; # managed by Certbot
}

```

2.3 Docker

- Https를 통해 repository를 이용하기 위해서 package 설치

```

sudo apt-get update
sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release

```

- Docker의 Official GPG key 등록

```

sudo mkdir -m 0755 -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

```

- Repository setup

```

echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

```

- 도커 엔진 설치

```

sudo apt-get update
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

```

- 도커 컴포즈 플러그인 설치

```

sudo apt-get update
sudo apt-get install docker-compose-plugin

```

2.4 Jenkins

- docker-compose.yml 작성

```
version: '3'

services:
  jenkins:
    image: jenkins/jenkins:lts
    container_name: jenkins
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
      - /jenkins:/var/jenkins_home
    ports:
      - "8080:8080"
    privileged: true
    user: root
    environment:
      - TZ=Asia/Seoul
      - JENKINS_OPTS="--prefix=/jenkins"
```

- docker compose 실행

```
docker compose up -d
```

3. 배포

GitLab이 Push Event를 감지하여 Jenkins로 Webhook 이벤트 발생, Jenkins가 Jenkinsfile을 읽어 script 실행하여 자동 빌드 및 배포 진행

3.1 Frontend

- Jenkinsfile

```
pipeline {
  agent any
  environment{
    TARGET_HOST = "ubuntu@52.79.110.20"
    IMAGE_NAME = "front/develop"
    CONTAINER_NAME = "front_develop"
    VERSION = "1.0"
    PROJECT_PATH = "/home/ubuntu/front-develop"
  }
  stages {
    stage('Clone') {
      steps {
        git branch: 'fe_develop', credentialsId: 'cookscape', url: 'https://lab.ssafy.com/s08-metaverse-game-sub2/S08P22B109.g
      }
    }
    stage('Build') {
      steps {
        dir("../Frontend/cookscape"){
          sshagent (credentials: ['cookscape_ssh']){
            sh 'scp -o StrictHostKeyChecking=no -r ./ * ${TARGET_HOST}:${PROJECT_PATH}'
          }
        }
      }
    }
    stage('Test') {
      steps {
      }
    }
    stage('Stop process'){
      steps{
        sshagent (credentials: ['cookscape_ssh']){
          script{
            try{
              sh'''
                ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker rm -f ${CONTAINER_NAME}"
                ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker rmi ${IMAGE_NAME}:${VERSION}"
              '''
            } catch (e){
              echo 'Remove Container Failed : ${CONTAINER_NAME}'
            }
          }
        }
      }
    }
  }
}
```

```
}  
    }  
}  
  
stage('Deploy'){  
    steps(  
        sshagent(credentials: ['cookscope_ssh']){  
            sh'''  
ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker build --tag ${IMAGE_NAME}:${VERSION} ${PROJECT_PATH}"  
ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker run --name ${CONTAINER_NAME} -p 3000:3000 -d ${IMAGE_NAME}:$"  
'''  
        }  
    )  
}  
}
```

- Dockerfile

```
FROM node:16.0-alpine as build
WORKDIR /app

# 패키지 설치 단계
COPY ["./package.json", "."]
COPY ["./package-lock.json", "."]
RUN ["npm", "install"]

# 빌드 단계
COPY [".", "."]

EXPOSE 3000

CMD ["npm", "start"]
```

3.2 Backend

- Jenkinsfile

```

pipeline {
    agent any
    environment{
        TARGET_HOST = "ubuntu@52.79.110.20"
        IMAGE_NAME = "server/release"
        CONTAINER_NAME ="server_release"
        LAST_VERSION = "1.0"
        NEW_VERSION = "1.0"
        PROJECT_PATH = "/home/ubuntu/server-prod"
    }
    stages {
        stage('Clone') {
            steps {
                git branch: 'be_release', credentialsId: 'cookscape' ,url: 'https://lab.ssafty.com/s08-metaverse-game-sub2/S08P22B109.g
            }
        }
        stage('Build') {
            steps {
                dir("./Backend/Cookscape"){
                    sh 'chmod +x gradlew'
                    sh './gradlew clean build'
                }
            }
        }
        stage('Test') {
            steps {
            }
        }
        stage('Copy build file'){
            steps{
                dir("./Backend/Cookscape"){
                    sshagent (credentials: ['cookscape_ssh']){
                        sh 'scp ./deploy/Dockerfile ${TARGET_HOST}:${PROJECT_PATH}'
                        sh 'scp ./build/libs/Cookscape.jar ${TARGET_HOST}:${PROJECT_PATH}'
                    }
                }
            }
        }
    }
}

```

```

    }
    stage('Stop process'){
      steps{
        sshagent (credentials: ['cookscape_ssh']){
          script{
            try{
              sh'''
                ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker rm -f ${CONTAINER_NAME}"
                ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker rmi ${IMAGE_NAME}:${LAST_VERSION}"
                '''
            } catch (e){
              echo 'Remove Container Failed : ${CONTAINER_NAME}'
            }
          }
        }
      }
    }
  }
}
stage('Deploy'){
  steps{
    sshagent (credentials: ['cookscape_ssh']){
      sh'''
        ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker build --tag ${IMAGE_NAME}:${NEW_VERSION} --build-arg ARG_PRO
        ssh -o StrictHostKeyChecking=no ${TARGET_HOST} "docker run -v server-file:/file --name ${CONTAINER_NAME} -p 9999:99
        '''
      }
    }
  }
}
}
}
}

```

- Dockerfile

```

FROM openjdk:11.0.11-jre-slim

WORKDIR /jar

COPY ["Cookscape.jar", "."]

ARG ARG_PROFILE
ENV PROFILE=${ARG_PROFILE}

ENTRYPOINT ["java", "-Dspring.profiles.active=${PROFILE}", "-Duser.timezone=Asia/Seoul", "-jar", "Cookscape.jar"]

```

- SpringBoot 배포 설정파일

```

server:
  servlet:
    encoding:
      enabled: true
      force: true
      charset: UTF-8
    address: 0.0.0.0

  ssl:
    key-store: classpath:SSL/keystore.p12
    key-store-type: PKCS12
    key-store-password: cookscapeb109

  error:
    include-exception: false      # Response 에 Exception 표시 여부 (true | false)
    include-message: always      # Response 에 Exception Message 표시 여부 (never | always | on_param)
    include-stacktrace: on_param # Response 에 Stack Trace 표시 여부 (never | always | on_param)
  whitelabel:
    enabled: true                # 에러 발생 시 Spring 기본 에러 페이지 노출 여부 (true | false)

  http2:
    enabled: true

server:
  servlet:
    context-path: /api
    port: 9999

logging:
  level:

```

```

    org.hibernate.type.descriptor.sql.BasicBinder: trace

spring:

    mvc:
        pathmatch:
            matching-strategy: ant_path_matcher

    jpa:
        hibernate:
            ddl-auto: none
        properties:
            hibernate:
                format_sql: false
        generate-ddl: true
        database: mysql
        database-platform: org.hibernate.dialect.MySQL8Dialect
        defer-datasource-initialization: true
        show-sql: false
        open-in-view: false

    datasource:
        driver-class-name: com.mysql.cj.jdbc.Driver
        url: jdbc:mysql://j8b109.p.ssafy.io:3306/cookscape_db?useUnicode=true&characterEncoding=utf8&serverTimezone=Asia/Seoul&zeroDateTim
        username: cookscape_ssafy
        password: cookscape109

    servlet:
        multipart:
            enabled: true
            max-file-size: 500MB
            max-request-size: 500MB
            location: ./file/temp

    file:
        app:
            url: /v1/file/app/**
        #    windowPath: file:/file/app/window/
        windowPath: ../file/app/window/

    swagger:
        enable: true

    jwt:
        header: Authorization
        secret: d1659096d8c5bd34c2e96fdb235c123afccf23450af0870d95813d820c0dd44661507fe41d5a12c5b796299d260f314d9a5bdca31050fc00a2a4b44c9d31
        validity-second:
            access: 172800
            refresh: 172800

```

4. 계정정보

	MySQL	Jenkins	Springboot SSL
아이디	cookscape_ssafy	cookscape_ssafy	cookscape_ssafy
비밀번호	cookscape109	cookscape109	cookscape109
DB명	cookscape_db		