

## 목록

개발환경_및_프로젝트_기술_스택.....	1
Server_Port_테이블.....	3
백엔드_빌드_방법.....	5
프론트엔드_빌드_방법.....	6
Jenkins_files.....	8



# 개발환경 및 프로젝트 기술 스택

## 이슈 관리

- Jira

## 형상 관리

- Gitlab

## UI/UX

- Figma

## Communication

- Mattermost
- Notion

## DB

- MySQL 8.0.32

## OS

- Windows 10
- Ubuntu 20.04 LTS

## IDE

- Visual Studio Code 1.76.2
- IntelliJ Ultimate 2022.3.2

## 빅데이터

- HDFS 3.0.0+cdh6.3.2
- Hive 2.1.1+cdh6.3.2
- Hue 4.2.0+cdh6.3.2
- Sqoop 1.4.7+cdh6.3.2
- YARN 3.0.0+cdh6.3.2
- ZooKeeper 3.4.5+cdh6.3.2

## Front-End

- node -v 18.12.1
- vite
- react -v 18.2.0
- tailwind -v 3.2.7
- chartjs -v 4.2.1
- jspdf -v 2.5.1
- xlsx 0.18.5
- xslx 1.0.0

## Back-End

- Java-1.8.0-OpenJDK
- Spring Boot 2.7.9-SNAPSHOT
- Gradle 7.6
- JPA : latest
- Spring Cloud : latest

## Server

- Amazon EC2

- react-google-charts -v 4.0.0

## 기타 편의 툴

- Postman
- Sonarqube (SSAFY) 8.9.9

- Docker 23.0.1
- R3
- Jenkins
  - Jenkins 2.375.1
  - openjdk 11.0.17+8
- Ansible
  - ansible core 2.12.10
  - python 3.8.10
- certbot 0.40.0



# Server Port 테이블

## Jenkins Server (Local)

Service(Container name)	IP address	Port	contents
jenkins	172.17.0.0/16 (dynamic)	50000:50000 8080:8080	docker container
node-spring	172.17.0.0/16 (dynamic)	30022:22	docker container

## Ansible Server

Service(Container name)	IP address	Port	contents
ansible	EC2 Public IP	-	host

## AWS SSAFY Server

Service(Container name)	IP address (docker network)	Port	contents
<b>rabbitmq</b>	172.18.0.0/16 (dynamic)	5672, 15672	docker container
<b>mysql</b>	172.18.0.0/16 (dynamic)	3307	docker container
<b>nginx</b>	172.18.0.0/16 (dynamic)	443(HTTPS)	docker container
<a href="#">config-service</a>	172.18.0.0/16 (dynamic)	8888	docker container
<a href="#">discovery-service</a>	172.18.0.0/16 (dynamic)	8761	docker container
<a href="#">apigateway-service</a>	172.18.0.0/16 (dynamic)	8000	docker container
<a href="#">user-service</a>	172.18.0.0/16 (dynamic)	8081	docker container

Service(Container name)	IP address (docker network)	Port	contents
<a href="#">news-service</a>	172.18.0.0/16 (dynamic)	8082	docker container
<a href="#">trade-service</a>	172.18.0.0/16 (dynamic)	8085	docker container



# 백엔드 빌드 방법

# MYSQL Container 실행

```
docker run -d -p 3307:3306 -e MYSQL_ROOT_PASSWORD=1234 --name cnt-mysql mysql
```

# RabbitMQ Container 실행

```
docker run -d --name rabbitmq -p 5672:5672 -p 15672:15672 --restart=unless-stopped rabbitmq:management
```

+) Local의 Profile은 dev, EC2의 Profile은 prod로 설정

1. 프로젝트를 클론 받는다.
2. 받은 프로젝트의 back폴더로 이동한다.

- ssafycnt-apigateway-service
- ssafycnt-config-service
- ssafycnt-discovery-service
- ssafycnt-news-service
- ssafycnt-trade-service
- ssafycnt-user-service

3. 각 파일을 순서대로 실행한다.
  - a. ssafycnt-config-service
  - b. ssafycnt-discovery-service
  - c. ssafycnt-apigateway-service
  - d. ssafycnt-user-service
  - e. ssafycnt-trade-service
  - f. ssafycnt-news-service



# 프론트엔드 빌드 방법

1. `git clone {프로젝트}` # 프로젝트를 클론 받는다.
2. `cd S08P22B308/front/` # 받은 프로젝트의 front 폴더로 이동한다.
3. `npm install` # 패키지 다운로드
4. `npm run dev` # 프로젝트 실행

```
MINGW64:/c/Users/SSAFY/Desktop/새 폴더/S08P22B308/front
SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Desktop/새 폴더
$ git clone https://lab.ssafy.com/s08-bigdata-dist-sub2/S08P22B308.git
Cloning into 'S08P22B308'...
remote: Enumerating objects: 5502, done.
remote: Counting objects: 100% (102/102), done.
remote: Compressing objects: 100% (90/90), done.
remote: Total 5502 (delta 24), reused 12 (delta 0), pack-reused 5400
Receiving objects: 100% (5502/5502), 18.19 MiB | 8.60 MiB/s, done.
Resolving deltas: 100% (1992/1992), done.

SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Desktop/새 폴더
$ cd S08P22B308/

SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Desktop/새 폴더 /S08P22B308 (main)
$ cd front/

SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Desktop/새 폴더 /S08P22B308/front (main)
$ npm install

added 429 packages, and audited 430 packages in 10s

96 packages are looking for funding
  run `npm fund` for details

5 high severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Desktop/새 폴더 /S08P22B308/front (main)
$ npm run dev

> front@0.0.0 dev
> vite

VITE v4.1.4 ready in 873 ms
+ Local:   http://localhost:5173/
+ Network: use --host to expose
```



# Jenkins files

- Jenkins → Ansible → EC2 순서로 배포가 이루어집니다.
- Gitlab에서 소스 코드가 변경되고 Push가 되면 빌드를 시작합니다.
- Jenkins의 Multibranch Pipeline이 Gitlab을 1분마다 확인하면서 변경사항을 체크합니다.
- Multibranch Pipeline은 Gitlab의 DevOps 폴더에 있습니다.

## FrontEnd CI/CD Pipeline

SSAFYCNT-React-CI

```
pipeline {
  agent any
  tools{
    nodejs 'node18'
  }

  parameters{
    string(name: 'BRANCH_NAME')
  }

  environment {
    // GIT_BUSINESS_CD = "master"
    GIT_CREDENTIAL_ID = 'gitlab-personal-token'
    GITLAB_URL = 'https://lab.ssafty.com/s08-bigdata-dist-sub2/S08P22B308.git'
  }

  stages {
    stage('git checkout'){
      steps{
        git branch: "${BRANCH_NAME}", credentialsId: "${env.GIT_CREDENTIAL_ID}", url: "${env.GITLAB_URL}"
        sh "chmod +x -R ${env.WORKSPACE}"
      }
    }

    // stage('Install Dependencies') {
    //   steps {
    //     dir('front'){
    //       sh 'npm i'
    //     }
    //   }
    // }

    stage('React Build') {
      steps {
        script {
          script {
            sh 'npm --prefix ./front install --network-timeout=100000'
            sh 'npm --prefix ./front run build'
          }
        }
      }
    }

    stage('Build to Remote Server with dist') {
      steps{
        sshPublisher(
          publishers: [
            sshPublisherDesc(
              configName: 'ansible-server-aws',
              transfers: [
                sshTransfer(
                  cleanRemote: false,
                  excludes: '',
                  execCommand: '',
                  execTimeout: 300000,
                  flatten: false,
                  makeEmptyDirs: false,
                  noDefaultExcludes: false,
                  patternSeparator: '[, ]+',
                  remoteDirectorySDF: false,
                  sourceFiles: 'front/dist/**/*',
                  removePrefix: 'front/dist',
                  remoteDirectory: 'ssafycnt/dist'
                )
              ],
              usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false
            )
          ]
        )
      }
    }

    stage('Build to Remote Server with assets') {
      steps{
        sshPublisher(
```



```

    },
    usePromotionTimestamp: false, useWorkspaceInPromotion: false, verbose: false)))
  }
}
}
}
}

```

## BackEnd CI/CD Pipeline

### SSAFYCNT-Spring-CI

```

pipeline {
  agent any
  tools{
    gradle 'gradle-7.6'
  }

  parameters{
    string(name: 'BRANCH_NAME')
  }

  environment {
    // GIT_BUSINESS_CD = "master"
    GIT_CREDENTIAL_ID = 'gitlab-personal-token'
    GITLAB_URL = 'https://lab.ssafy.com/s08-bigdata-dist-sub2/S08P22B308.git'
  }

  stages {
    stage('Gitlab clone'){
      steps{
        git branch: "${BRANCH_NAME}", credentialsId: "${env.GIT_CREDENTIAL_ID}", url: "${env.GITLAB_URL}"
        sh "chmod +x -R ${env.WORKSPACE}"
      }
    }

    stage('Build') {
      parallel{
        stage('Config build'){
          when{ expression{ "${BRANCH_NAME}" ==~ /(main|back_build)/}}
          steps{
            dir('back/ssafycnt-config-service'){
              sh '''
                gradle clean build --exclude-task test
              '''
            }
          }
        }
        stage('Discovery build'){
          when{ expression{ "${BRANCH_NAME}" ==~ /(main|back_build)/}}
          steps{
            dir('back/ssafycnt-discovery-service'){
              sh '''
                gradle clean build --exclude-task test
              '''
            }
          }
        }
        stage('Apigateway build'){
          when{ expression{ "${BRANCH_NAME}" ==~ /(main|back_build)/}}
          steps{
            dir('back/ssafycnt-apigateway-service'){
              sh '''
                gradle clean build --exclude-task test
              '''
            }
          }
        }
        stage('User build'){
          when{ expression{ "${BRANCH_NAME}" ==~ /(main|back_build|ssafycnt-user-service)/}}
          steps{
            dir('back/ssafycnt-user-service'){
              sh '''
                gradle clean build --exclude-task test
              '''
            }
          }
        }
        stage('News build'){

```



```

when{ expression{ "${BRANCH_NAME}" ==- /(main|back_build)/}}
steps{
    sshPublisher(publishers: [
        sshPublisherDesc(
            configName: 'ansible-server-aws',
            transfers: [
                sshTransfer(
                    cleanRemote: false, excludes: '',
                    execCommand: 'ansible-playbook -i inventory/hosts ssafycnt/create-ssafycnt-back-containers-playboo
            ]
        )
    ]
}
stage('Create news containers'){
    when{ expression{ "${BRANCH_NAME}" ==- /(ssafycnt-news-service)/}}
    steps{
        sshPublisher(publishers: [
            sshPublisherDesc(
                configName: 'ansible-server-aws',
                transfers: [
                    sshTransfer(
                        cleanRemote: false, excludes: '',
                        execCommand: 'ansible-playbook -i inventory/hosts ssafycnt/create-ssafycnt-back-containers-playboo
                ]
            )
        ]
    }
}
stage('Create user containers'){
    when{ expression{ "${BRANCH_NAME}" ==- /(ssafycnt-user-service)/}}
    steps{
        sshPublisher(publishers: [
            sshPublisherDesc(
                configName: 'ansible-server-aws',
                transfers: [
                    sshTransfer(
                        cleanRemote: false, excludes: '',
                        execCommand: 'ansible-playbook -i inventory/hosts ssafycnt/create-ssafycnt-back-containers-playboo
                ]
            )
        ]
    }
}
stage('Create trade containers'){
    when{ expression{ "${BRANCH_NAME}" ==- /(ssafycnt-trade-service)/}}
    steps{
        sshPublisher(publishers: [
            sshPublisherDesc(
                configName: 'ansible-server-aws',
                transfers: [
                    sshTransfer(
                        cleanRemote: false, excludes: '',
                        execCommand: 'ansible-playbook -i inventory/hosts ssafycnt/create-ssafycnt-back-containers-playboo
                ]
            )
        ]
    }
}
}
}
}

```