CI/CD

시간 설정(KST)

```
sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime
```

sudo ln -sf /usr/share/zoneinfo/Asia/Seoul /etc/localtime

I. Docker

- 공식 홈페이지 통해 설치

https://docs.docker.com/engine/install/ubuntu/

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/k
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

# To install the latest version, run:
sudo apt-get install docker-ce docker-ce-cli containerd.io doc
```

```
# Docker User setting
sudo usermod -aG docker $USER
```

II. Jenkins

https://taeyeon47.tistory.com/35

```
sudo docker run -d 9090:8080 -v /home/ubuntu/jenkins-data:/var
```

- 젠킨스 설정을 mount해서 설정 유지(/home/ubuntu/jenkins-data) → chmod 777 설정
- 9090번 포트 사용
- timezone Asia/Seoul로 설정
- jdk 17 사용

가. Jenkins Dashboard

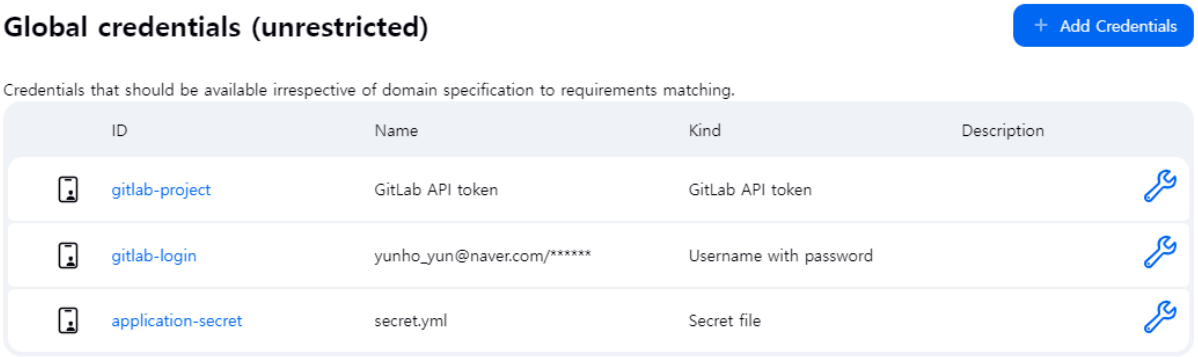
기본 URL: http://i12a405.p.ssafy.io:9090/

- **id 및 pw는 구글드라이브 내 secret 참고**
- **SSL 인증서 적용 후 URL 변경("III-3. Jenkins nginx 세팅" 참고)**

나. Jenkins Credentials 세팅

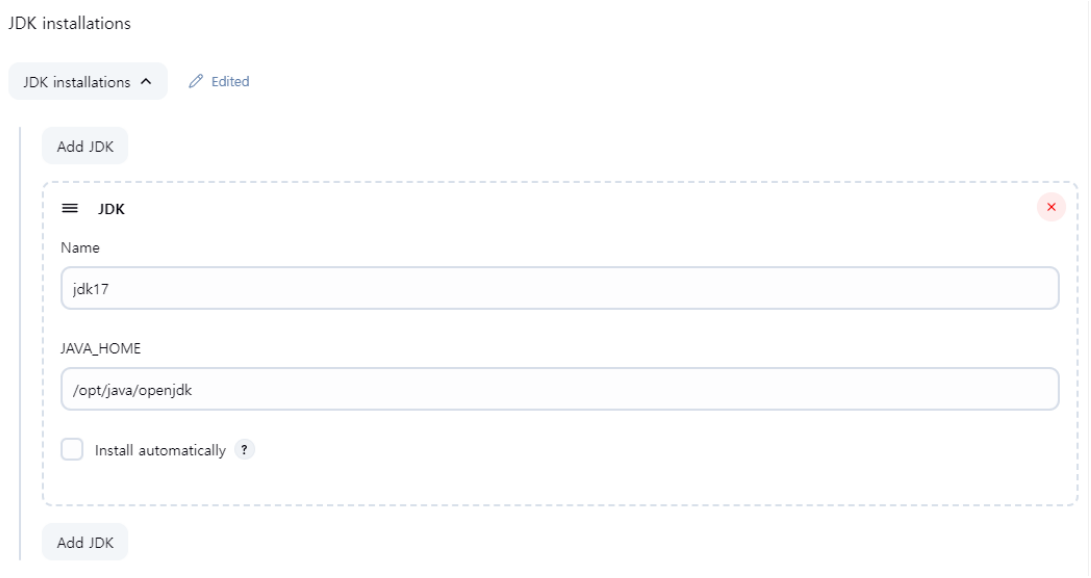
System → Domain (Global) → Add credentials

- Gitlab Project token 등록(Kind: GitLab API tokens)
- Gitlab Personal API token 등록(Kind: Username with password)
- secret.yml 파일 등록(Kind: Secret file) // application-secret.yml

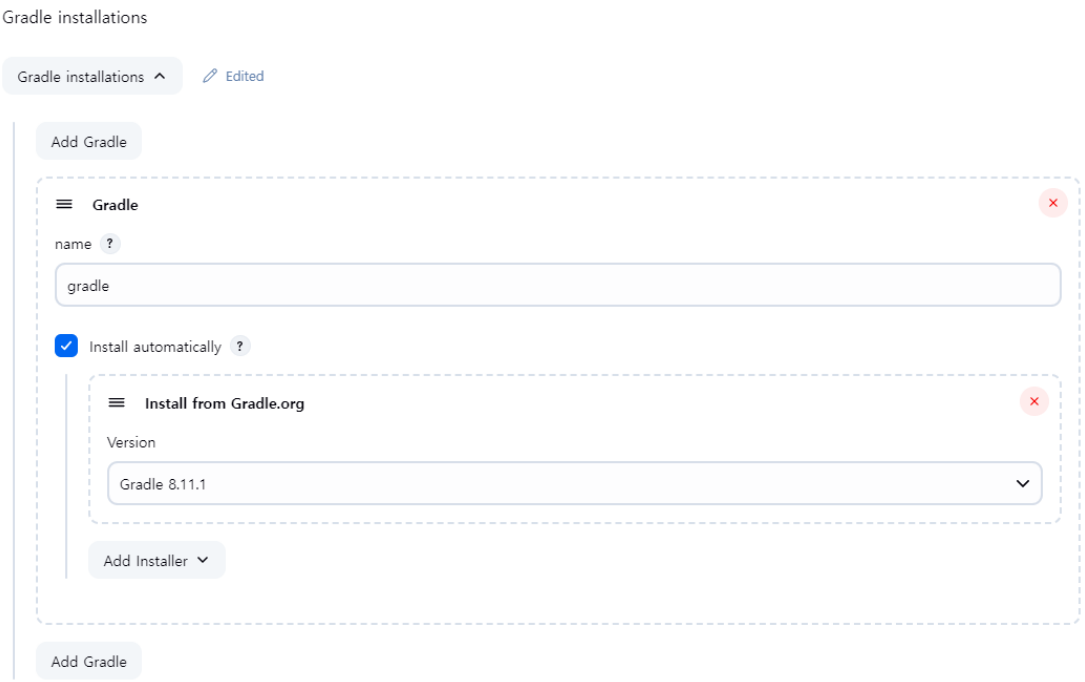


다. Jenkins Tools 세팅

- **JDK installations**
  - docker로 Jenkins를 설치했으면 openjdk17로 깔려있음

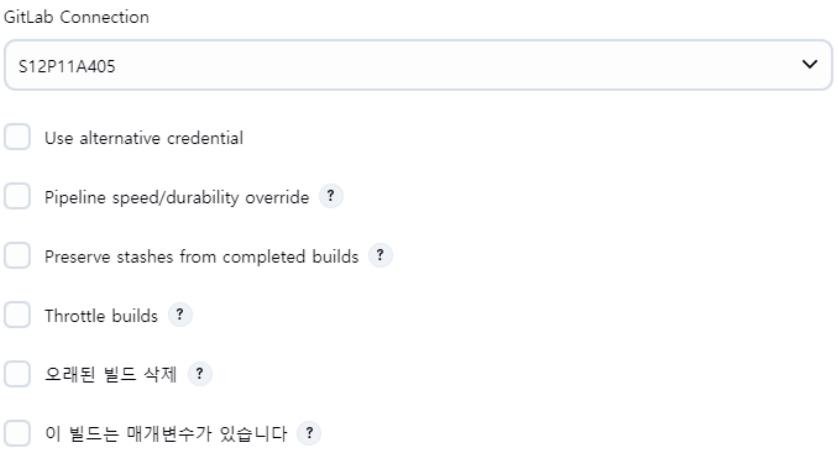


- **Gradle installations**
  - Gradle 프로젝트의 버전에 맞게 설정



라. Jenkins item 생성

- **Gitlab Connection 등록**

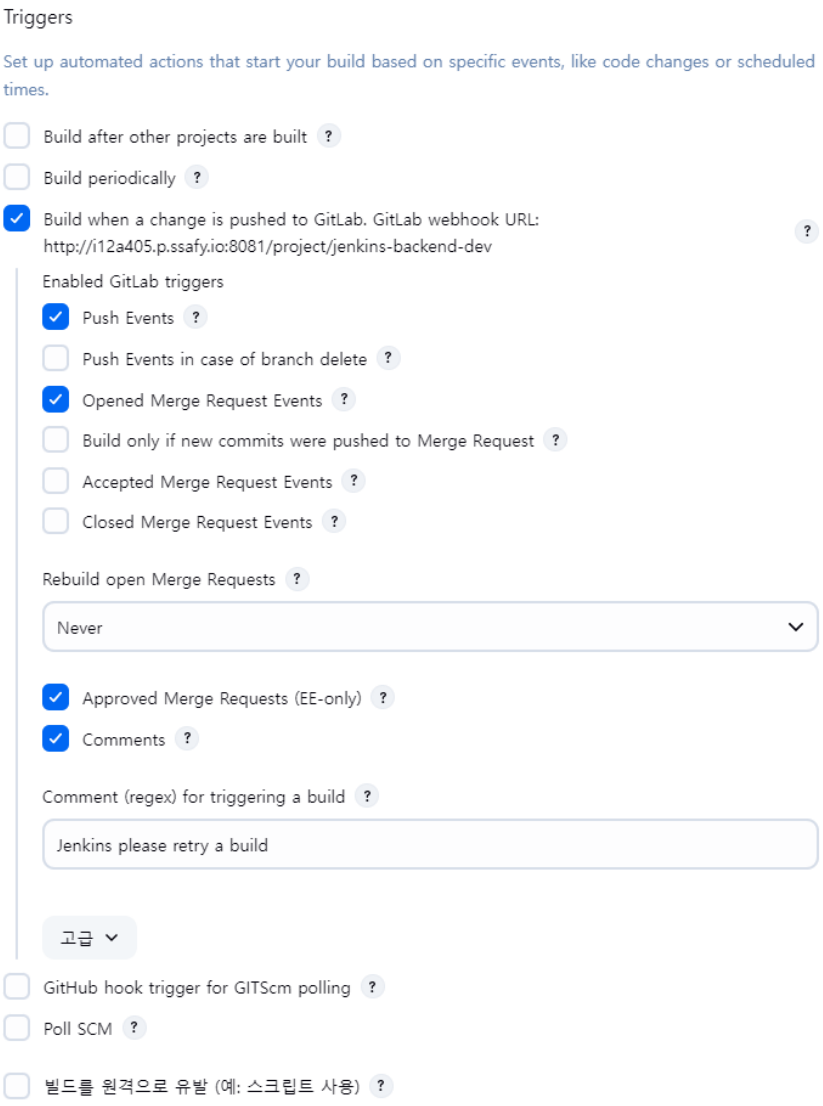


- **고급 버튼 클릭해서 gitlab webhook에 등록할 token 발급**

- **Trigger 세팅**

  SSL 인증서 설정 이후 다시 세팅 필요(URL이 https로 변경)

  - **Gitlab에 Push, Merge Request 시 build되도록 설정**



- **Pipeline 작성**

  https://taeyeon47.tistory.com/38

  - **환경변수 유의**

    https://velog.io/@tank3a/AWS%EC%97%90-Jenkins%EB%A5%BC-%EA%B5%AC%EC%B6%95%ED%95%B4%EB%B3%B4%EC%9E%90-5-Spring-Boot%EC%9D%98-%ED%99%98%EA%B2%BD%EB%B3%80%EC%88%98-%ED%8C%8C%EC%9D%BC-%EC%B6%94%EA%B0%80

  - **build 등 단계별 state를 gitlab에 동기화 — 확인 필요**

    https://stackoverflow.com/questions/48932616/how-to-change-in-jenkins-pipelinegitlab-status-in-cases-tests-failing-or-succeed

https://docs.gitlab.com/ee/api/commits.html#post-the-build-status-to-a-commit

https://github.com/jenkinsci/gitlab-plugin?tab=readme-ov-file#readme

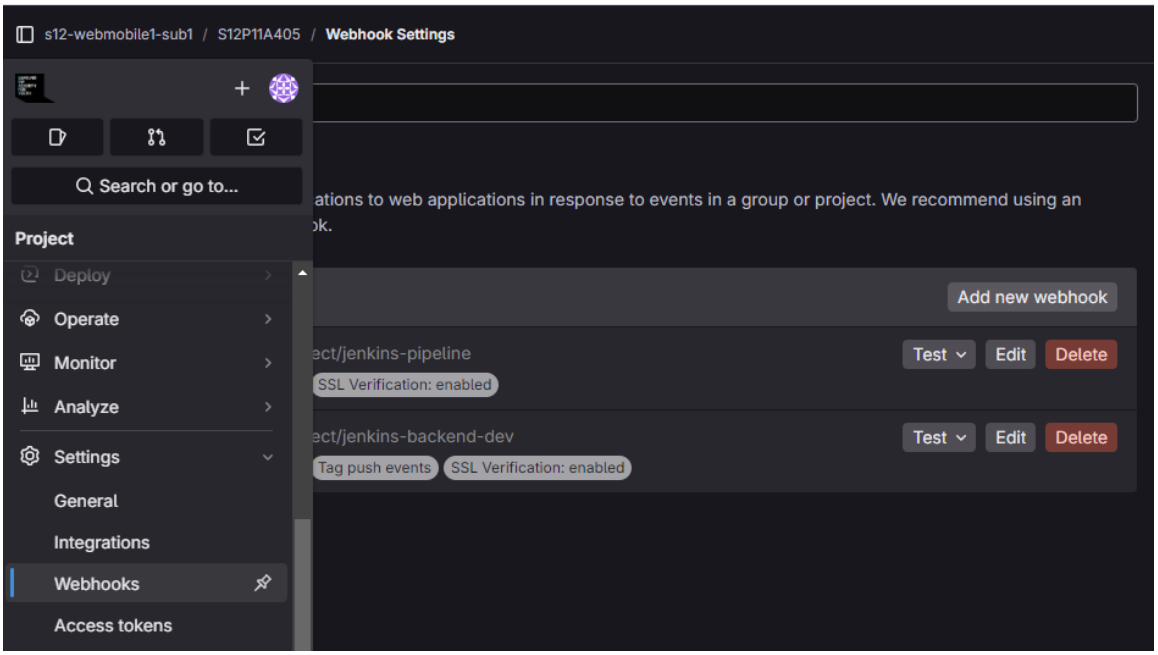
```
pipeline{
    agent any
    options {
      gitLabConnection('S12P11A405')
    }
    tools {
        gradle 'gradle'
        jdk 'jdk17'
    }
    environment {
        JAVA_HOME = "tool jdk17"
    }
    stages{
        stage('gitclone'){
            steps{
                git branch: 'dev',
                credentialsId: 'gitlab-login',
                url: "https://lab.ssafy.com/s12-webmobile1-
            }
        }
        stage('Add env'){
            steps{
                withCredentials([file(credentialsId: 'appli
                    script {
                        def filePath = '/var/jenkins_home/w
                        def fileExists = sh(script: "test -

                        if (fileExists == 'true') {
                            sh "rm ${filePath}"
                        }

                        sh "cp ${secret} ${filePath}"
                    }
                }
```

```
            }
        }
        stage('Clean and Build'){
            steps {
                updateGitlabCommitStatus name: 'build', sta
                dir('/var/jenkins_home/workspace/jenkins-ba
                    sh 'pwd'
                    sh 'ls -al'
                    sh 'gradle wrapper'
                    sh 'chmod +x ./gradlew'
                    sh 'chmod +x ./gradlew.bat'
                    sh 'java --version'
                    sh './gradlew clean test build '
                }
            }
            post {
                success {
                  updateGitlabCommitStatus name: 'build', s
                }
                failure {
                  updateGitlabCommitStatus name: 'build', s
                }
            }
        }
    }
}
```

마. Gitlab-Jenkins Webhook 세팅

https://velog.io/@suhongkim98/jenkins-gitlab-%EC%97%B0%EB%8F%99-%EB%B0%8F-webhook-%EC%84%A4%EC%A0%95%ED%95%98%EA%B8%B0

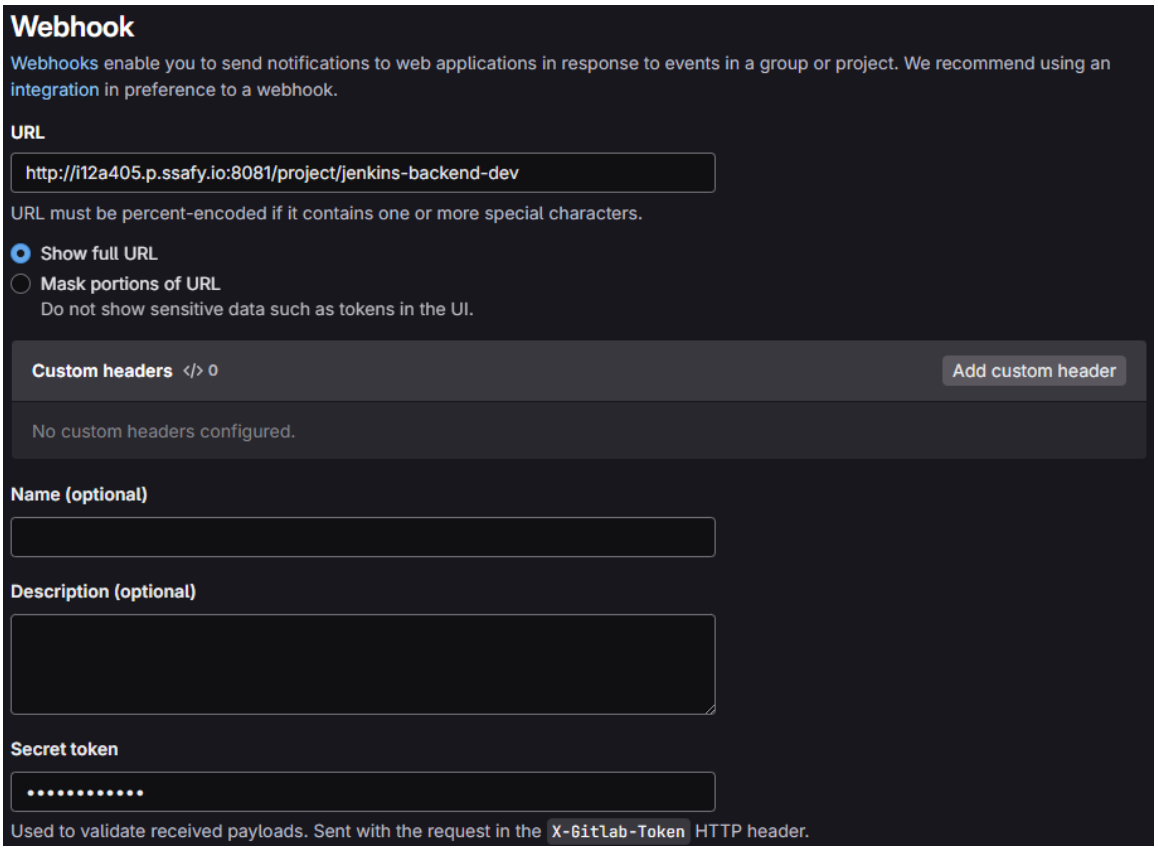
- **Gitlab 프로젝트의 Webhook 페이지 접속**



- **Webhook URL 설정**

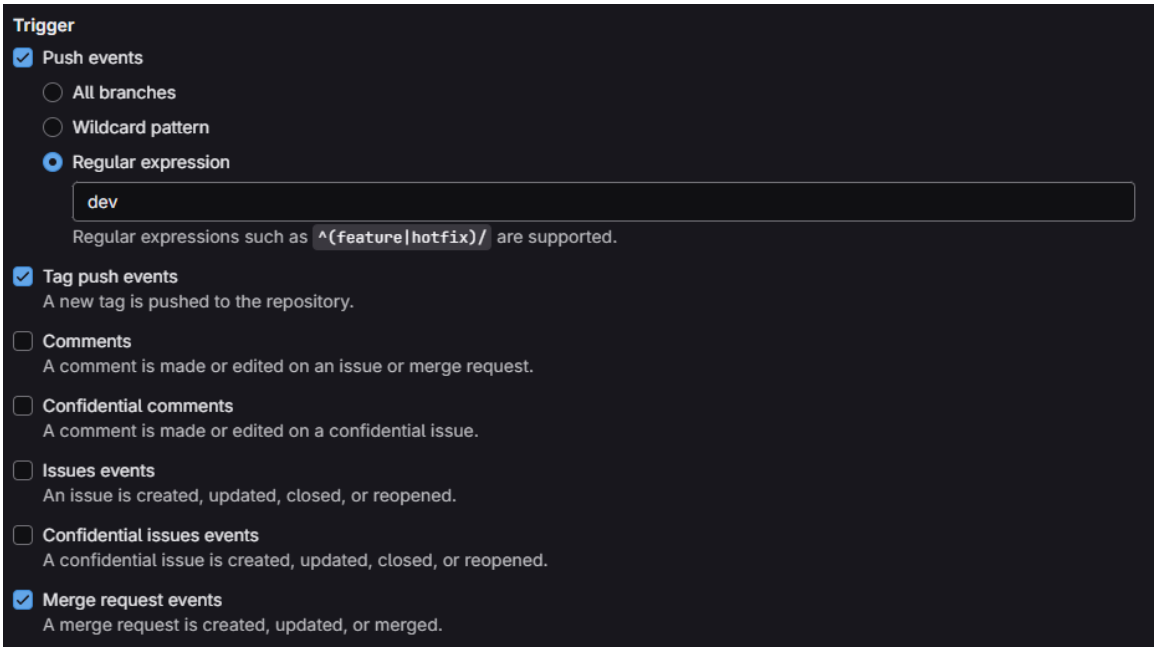
  SSL 인증서 설정 이후 다시 세팅 필요(URL이 https로 변경)

  - Jenkins에서 조회되는 URL과 token 입력



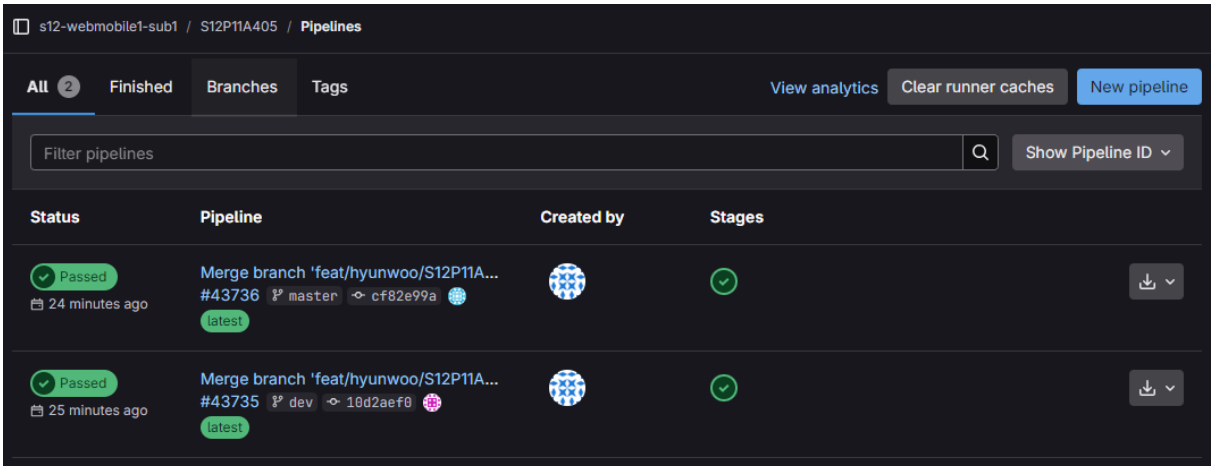
- **Trigger 구성**

- Push와 Merge Request 체크
- Regular expression으로 브랜치 명시

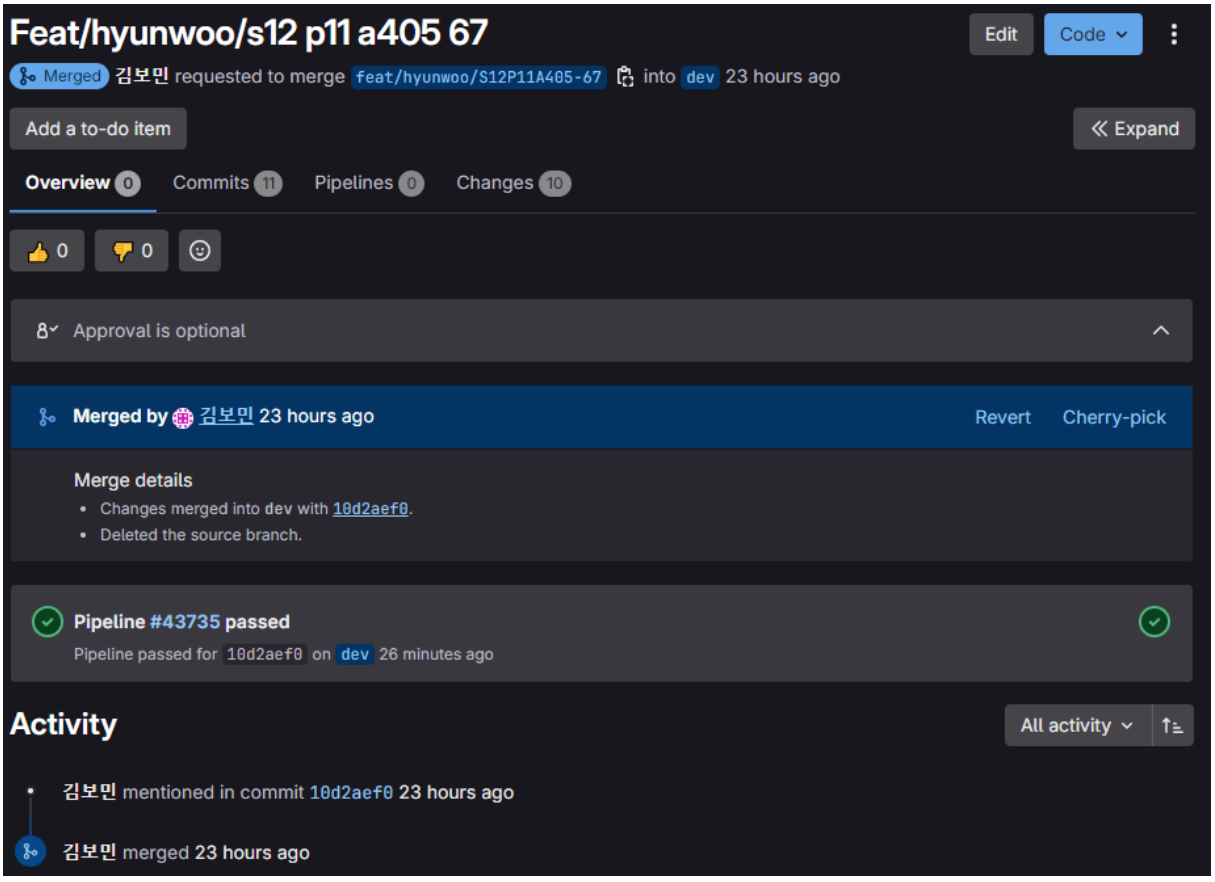


바. 빌드 테스트 확인

- **프로젝트 Build → Pipeline에서 확인**



- **Merge Request 페이지에서 확인**



III. Nginx

가. 설치

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install nginx
sudo service nginx start
sudo service nginx status
```

나. SSL 인증서 설정

https://chaechaeros.tistory.com/234

https://certbot.eff.org/instructions?ws=nginx&os=snap

```
# 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-get remove certbot

# certbot 설치
sudo snap install --classic certbot

# certbot 명령을 로컬에서 실행할 수 있도록 snap의 certbot 파일을 로컬의
ln -s /snap/bin/certbot /usr/bin/certbot
```

다. Jenkins nginx 세팅

- Jenkins 관리 → System에서 URL 변경

- 아래 nginx 설정 후 기존 URL에서 아래 URL로 접속 경로 변경
  - https://i12a405.p.ssafy.io/jenkins/
  - http://i12a405.p.ssafy.io:9090/jenkins → 로그인 불가
- nginx 설정
  - conf 파일 작성
  - `sudo nginx -t` (문법 체크)
  - `sudo service nginx restart`

```
#/etc/nginx/conf.d/jenkins.conf
server {
    listen 80;
    server_name i12a405.p.ssafy.io;

    # HTTP -> HTTPS 리다이렉션
    location / {
        return 301 https://$host/jenkins;
    }

    # 9090 포트로 직접 접근할 때 Jenkins로 포워딩
    location /jenkins {
        proxy_pass http://127.0.0.1:9090;  # 외부 9090 포트를
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwa
        proxy_set_header X-Forwarded-Proto https;
    }
}

server {
    listen 443 ssl;
    server_name i12a405.p.ssafy.io;

    ssl_certificate /etc/letsencrypt/live/i12a405.p.ssafy.i
    ssl_certificate_key /etc/letsencrypt/live/i12a405.p.ssa

    # HTTPS -> Jenkins로 포워딩
    location /jenkins {
        proxy_pass http://127.0.0.1:9090;  # 외부 9090 포트를
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwa
        proxy_set_header X-Forwarded-Proto https;
    }
}
```

IV. Deployment

//TODO

Redis

```
# /redis/conf/redis.conf
bind 0.0.0.0 # 외부 접속 허용
port 6379 # 포트
requirepass YOUR_PASSWORD # 접속 비밀번호
maxmemory 2gb # 최대 메모리
appendonly yes # 백업
```

```
docker run -d -p 6379:6379 -v ./data:/data -v ./conf/redis.con
```

Trouble shooting

- **gradle과 jdk를 직접 설치했으나, 환경변수가 인식되지 않음**
  - `Please set the JAVA_HOME variable in your environment to match the location of your Java installation.`

  이미 openjdk가 설치되어 있어 LibrericaJDK를 설치하면 JDK 충돌됨 → openjdk를 쓰기로 결정

  ▼ **관련 로그**

```
23:54:45  Started by user 공통4반5팀
23:54:46  [Pipeline] Start of Pipeline
23:54:47  [Pipeline] node
23:54:47  Running on Jenkins in /var/jenkins_home/worksp
23:54:47  [Pipeline] {
23:54:47  [Pipeline] stage
23:54:47  [Pipeline] { (Declarative: Tool Install)
23:54:47  [Pipeline] tool
23:54:48  Unpacking https://services.gradle.org/distribu
```

```
23:54:53  [Pipeline] envVarsForTool
23:54:53  [Pipeline] tool
23:54:53  Unpacking https://download.bell-sw.com/java/17
23:54:55  [Pipeline] envVarsForTool
23:54:56  [Pipeline] }
23:54:56  [Pipeline] // stage
23:54:56  [Pipeline] withEnv
23:54:56  [Pipeline] {
23:54:56  [Pipeline] stage
23:54:56  [Pipeline] { (gitclone)
23:54:56  [Pipeline] tool
23:54:56  [Pipeline] envVarsForTool
23:54:56  [Pipeline] tool
23:54:56  [Pipeline] envVarsForTool
23:54:56  [Pipeline] withEnv
23:54:56  [Pipeline] {
23:54:56  [Pipeline] git
23:54:56  The recommended git tool is: NONE
23:54:56  using credential gitlab-login
23:54:56  Cloning the remote Git repository
23:54:56  Cloning repository https://lab.ssafy.com/s12-w
23:54:56   > git init /var/jenkins_home/workspace/jenkin
23:54:56  Fetching upstream changes from https://lab.ssa
23:54:56   > git --version # timeout=10
23:54:56   > git --version # 'git version 2.39.5'
23:54:56  using GIT_ASKPASS to set credentials
23:54:56   > git fetch --tags --force --progress -- http
23:54:57   > git config remote.origin.url https://lab.ss
23:54:57   > git config --add remote.origin.fetch +refs/
23:54:57  Avoid second fetch
23:54:57   > git rev-parse refs/remotes/origin/master^{c
23:54:57  Checking out Revision cf82e99a2bb5fac70afae703
23:54:57   > git config core.sparsecheckout # timeout=10
23:54:57   > git checkout -f cf82e99a2bb5fac70afae7034ee
23:54:57   > git branch -a -v --no-abbrev # timeout=10
23:54:57   > git checkout -b master cf82e99a2bb5fac70afa
23:54:57  Commit message: "Merge branch 'feat/hyunwoo/S1
23:54:57  First time build. Skipping changelog.
23:54:57  [Pipeline] }
```

```
23:54:57  [Pipeline] // withEnv
23:54:57  [Pipeline] }
23:54:57  [Pipeline] // stage
23:54:57  [Pipeline] stage
23:54:57  [Pipeline] { (Build Backend)
23:54:57  [Pipeline] tool
23:54:57  [Pipeline] envVarsForTool
23:54:57  [Pipeline] tool
23:54:57  [Pipeline] envVarsForTool
23:54:57  [Pipeline] withEnv
23:54:57  [Pipeline] {
23:54:57  [Pipeline] dir
23:54:57  Running in /var/jenkins_home/workspace/jenkins
23:54:57  [Pipeline] {
23:54:57  [Pipeline] sh
23:54:58  + pwd
23:54:58  /var/jenkins_home/workspace/jenkins-pipeline/b
23:54:58  [Pipeline] sh
23:54:58  + ls -al
23:54:58  total 52
23:54:58  drwxr-xr-x 4 jenkins jenkins 4096 Jan 26 14:54
23:54:58  drwxr-xr-x 6 jenkins jenkins 4096 Jan 26 14:54
23:54:58  -rw-r--r-- 1 jenkins jenkins   54 Jan 26 14:54
23:54:58  -rw-r--r-- 1 jenkins jenkins 1628 Jan 26 14:54
23:54:58  -rw-r--r-- 1 jenkins jenkins  360 Jan 26 14:54
23:54:58  -rw-r--r-- 1 jenkins jenkins 1522 Jan 26 14:54
23:54:58  drwxr-xr-x 3 jenkins jenkins 4096 Jan 26 14:54
23:54:58  -rw-r--r-- 1 jenkins jenkins 8762 Jan 26 14:54
23:54:58  -rw-r--r-- 1 jenkins jenkins 2966 Jan 26 14:54
23:54:58  -rw-r--r-- 1 jenkins jenkins   28 Jan 26 14:54
23:54:58  drwxr-xr-x 4 jenkins jenkins 4096 Jan 26 14:54
23:54:58  [Pipeline] sh
23:54:58  + chmod +x ./gradlew
23:54:58  [Pipeline] sh
23:54:58  + chmod +x ./gradlew.bat
23:54:58  [Pipeline] sh
23:54:59  + java --version
23:54:59  openjdk 17.0.13 2024-10-15
23:54:59  OpenJDK Runtime Environment Temurin-17.0.13+11
```

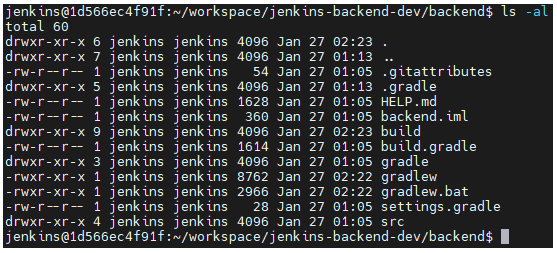
```
23:54:59  OpenJDK 64-Bit Server VM Temurin-17.0.13+11 (b
23:54:59  [Pipeline] sh
23:54:59  + ./gradlew clean build
23:54:59
23:54:59  ERROR: JAVA_HOME is set to an invalid director
23:54:59
23:54:59  Please set the JAVA_HOME variable in your envi
23:54:59  location of your Java installation.
23:54:59
23:54:59  [Pipeline] }
23:54:59  [Pipeline] // dir
23:54:59  [Pipeline] }
23:54:59  [Pipeline] // withEnv
23:54:59  [Pipeline] }
23:54:59  [Pipeline] // stage
23:54:59  [Pipeline] }
23:54:59  [Pipeline] // withEnv
23:54:59  [Pipeline] }
23:54:59  [Pipeline] // node
23:54:59  [Pipeline] End of Pipeline
23:54:59  ERROR: script returned exit code 1
23:55:00  Finished: FAILURE
```

- **GradleWrapperMain 클래스를 찾지 못함**

  `Error: Could not find or load main class org.gradle.wrapper.GradleWrapperMain Caused by: java.lang.ClassNotFoundException: org.gradle.wrapper.GradleWrapperMain`

  약 5시간 넘게 해결방법을 쉽게 찾지 못한 문제로, `.gitignore` 에 gradle 하위 파일을 설정해, Jenkins에서 git clone 방식으로 가져왔을 때 gradle-wrapper.jar 파일이 없어서 발생

  `.gitignore`에 gradle-wrapper.jar를 제외하는 것 외에 `gradle wrap`으로 직접 gradle-wrapper.jar을 생성하는 것이 가능하다 하여 시도해보았으나, **프로젝트 경로 상 gradle이 directory로 되어 있어 gradle 명령이 되지 않음(bash: ./gradle: Is a directory)**



gradle을 설치 후 재시도했으나 동일 증상...

결국 Jenkins 컨테이너 및 마운트 디렉토리를 모두 삭제 후 처음부터 gradle 툴 세팅 후 실행하여 해결함 + plugin 설치 후 **safe restart → restart 시 docker가 자동으로 띄워지지 않음 → 수동 restart(run) (`docker ps -a` 컨테이너 삭제 후)**

▼ **관련 로그**

```
00:22:54  Started by user공통4반5팀00:22:54  [Pipeline]
Start of Pipeline00:22:54  [Pipeline] node00:22:54  Ru
nning onJenkins in /var/jenkins_home/workspace/jenkins
-pipeline
00:22:54  [Pipeline] {00:22:54  [Pipeline] withEnv00:2
2:54  [Pipeline] {00:22:54  [Pipeline] stage00:22:54
[Pipeline] { (Declarative: Tool Install)00:22:54  [Pip
eline] tool00:22:54  [Pipeline] envVarsForTool00:22:54
[Pipeline] tool00:22:54  [Pipeline] envVarsForTool00:2
2:54  [Pipeline] }
00:22:54  [Pipeline] // stage
00:22:54  [Pipeline] withEnv00:22:54  [Pipeline] {00:2
2:54  [Pipeline] stage00:22:54  [Pipeline] { (gitclon
e)00:22:54  [Pipeline] tool00:22:54  [Pipeline] envVar
sForTool00:22:55  [Pipeline] tool00:22:55  [Pipeline]
envVarsForTool00:22:55  [Pipeline] withEnv00:22:55  [P
ipeline] {00:22:55  [Pipeline] git00:22:55  The recomm
ended git tool is: NONE
00:22:55  using credential gitlab-login
00:22:55   > git rev-parse --resolve-git-dir /var/jenk
ins_home/workspace/jenkins-pipeline/.git # timeout=10
00:22:55  Fetching changes from the remote Git reposit
ory
00:22:55   > git config remote.origin.urlhttps://lab.s
safy.com/s12-webmobile1-sub1/S12P11A405.git # timeout=
10
```

```
00:22:55  Fetching upstream changes fromhttps://lab.ss
afy.com/s12-webmobile1-sub1/S12P11A405.git00:22:55   >
git --version # timeout=10
00:22:55   > git --version # 'git version 2.39.5'
00:22:55  using GIT_ASKPASS to set credentials
00:22:55   > git fetch --tags --force --progress --htt
ps://lab.ssafy.com/s12-webmobile1-sub1/S12P11A405.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
00:22:55   > git rev-parse refs/remotes/origin/master^
{commit} # timeout=10
00:22:55  Checking out Revision cf82e99a2bb5fac70afae7
034eebc63baed6304d (refs/remotes/origin/master)
00:22:55   > git config core.sparsecheckout # timeout=
10
00:22:55   > git checkout -f cf82e99a2bb5fac70afae7034
eebc63baed6304d # timeout=10
00:22:55   > git branch -a -v --no-abbrev # timeout=10
00:22:55   > git branch -D master # timeout=10
00:22:55   > git checkout -b master cf82e99a2bb5fac70a
fae7034eebc63baed6304d # timeout=10
00:22:55  Commit message: "Merge branch 'feat/hyunwoo/
S12P11A405-24' into 'master'"
00:22:55   > git rev-list --no-walk cf82e99a2bb5fac70a
fae7034eebc63baed6304d # timeout=10
00:22:55  [Pipeline] }
00:22:55  [Pipeline] // withEnv
00:22:55  [Pipeline] }
00:22:55  [Pipeline] // stage
00:22:55  [Pipeline] stage00:22:55  [Pipeline] { (Buil
d Backend)00:22:55  [Pipeline] tool00:22:55  [Pipelin
e] envVarsForTool00:22:55  [Pipeline] tool00:22:55  [P
ipeline] envVarsForTool00:22:55  [Pipeline] withEnv00:
22:55  [Pipeline] {00:22:56  [Pipeline] dir00:22:56  R
unning in /var/jenkins_home/workspace/jenkins-pipelin
e/backend
00:22:56  [Pipeline] {00:22:56  [Pipeline] sh00:22:56
+ pwd
00:22:56  /var/jenkins_home/workspace/jenkins-pipelin
e/backend
```

```
00:22:56  [Pipeline] sh00:22:56  + ls -al
00:22:56  total 52
00:22:56  drwxr-xr-x 4 jenkins jenkins 4096 Jan 26 15:
22 .
00:22:56  drwxr-xr-x 6 jenkins jenkins 4096 Jan 26 14:
54 ..
00:22:56  -rw-r--r-- 1 jenkins jenkins   54 Jan 26 14:
54 .gitattributes
00:22:56  -rw-r--r-- 1 jenkins jenkins 1628 Jan 26 14:
54 HELP.md
00:22:56  -rw-r--r-- 1 jenkins jenkins  360 Jan 26 14:
54 backend.iml
00:22:56  -rw-r--r-- 1 jenkins jenkins 1522 Jan 26 14:
54 build.gradle
00:22:56  drwxr-xr-x 3 jenkins jenkins 4096 Jan 26 14:
54 gradle
00:22:56  -rw-r--r-- 1 jenkins jenkins 8762 Jan 26 15:
22 gradlew
00:22:56  -rw-r--r-- 1 jenkins jenkins 2966 Jan 26 15:
22 gradlew.bat
00:22:56  -rw-r--r-- 1 jenkins jenkins   28 Jan 26 14:
54 settings.gradle
00:22:56  drwxr-xr-x 4 jenkins jenkins 4096 Jan 26 14:
54 src
00:22:56  [Pipeline] sh00:22:56  + chmod +x ./gradlew
00:22:56  [Pipeline] sh00:22:57  + chmod +x ./gradlew.
bat
00:22:57  [Pipeline] sh00:22:57  + java --version
00:22:57  openjdk 17.0.13 2024-10-15
00:22:57  OpenJDK Runtime Environment Temurin-17.0.13+
11 (build 17.0.13+11)
00:22:57  OpenJDK 64-Bit Server VM Temurin-17.0.13+11
(build 17.0.13+11, mixed mode)
00:22:57  [Pipeline] sh00:22:57  + ./gradlew clean bui
ld
00:22:57  Error: Could not find or load main class or
g.gradle.wrapper.GradleWrapperMain
00:22:57  Caused by: java.lang.ClassNotFoundException:
org.gradle.wrapper.GradleWrapperMain
```

```
00:22:57  [Pipeline] }
00:22:57  [Pipeline] // dir
00:22:57  [Pipeline] }
00:22:57  [Pipeline] // withEnv
00:22:58  [Pipeline] }
00:22:58  [Pipeline] // stage
00:22:58  [Pipeline] }
00:22:58  [Pipeline] // withEnv
00:22:58  [Pipeline] }
00:22:58  [Pipeline] // withEnv
00:22:58  [Pipeline] }
00:22:58  [Pipeline] // node
00:22:58  [Pipeline] End of Pipeline
00:22:58  ERROR: script returned exit code 1
00:22:58  Finished: FAILURE
```

▼ 참고 링크

https://velog.io/@jh5253/%EC%98%A4%EB%A5%98-%EA%B8%B0%EB%B3%B8-%ED%81%B4%EB%9E%98%EC%8A%A4-org.gradle.wrapper.GradleWrapperMain%EC%9D%84%EB%A5%BC-%EC%B0%BE%EA%B1%B0%EB%82%98-%EB%A1%9C%EB%93%9C%ED%95%A0-%EC%88%98-%EC%97%86%EC%8A%B5%EB%8B%88%EB%8B%A4

https://androidhuman.tistory.com/537

https://plugins.jenkins.io/saferestart/)