포팅매뉴얼

- JVM
  - java version "11.0.14" 2022-01-18 LTS
- 웹 서버
  - AWS EC2 (Ubuntu 20.04 LTS)
    - MobaXterm 접속 후 Session→SSh (Remote host : ubuntu@i7c101.p.ssafy.io / use private key에 pem키 첨부
- Intellij (Ultimate 22.1.3)
- Nginx (/etc/nginx/sites-available/myapp.conf)

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

        root /home/ubuntu/build;

        index index.html index.htm index.nginx-debian.html;

        server_name _;

        location / {
                try_files $uri $uri/ =404;
        }
}

server {
        root /home/ubuntu/build;

        index index.html index.htm index.nginx-debian.html;
        server_name www.awa24.site awa24.site; # managed by Certbot

        location / {
                # First attempt to serve request as file, then
                # as directory, then fall back to displaying a 404.
                try_files $uri $uri/ =404;
        }

        listen [::]:443 ssl ipv6only=on; # managed by Certbot
        listen 443 ssl; # managed by Certbot
        ssl_certificate /etc/letsencrypt/live/awa24.site/fullchain.pem; # managed by Certbot
        ssl_certificate_key /etc/letsencrypt/live/awa24.site/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 {
 #location /api{
#       include /etc/nginx/proxy_params;
        #proxy_pass http://i7c101.p.ssafy.io:8081/api/;
        #proxy_pass https://i7c101.p.ssafy.io:8081/api/;
#       proxy_pass https://awa24.site:8081/api/;
#       proxy_redirect off;
#       charset utf-8;


#       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;
#  }
#}

# 80 포트로 접근시 443 포트로 리다이렉트 시켜주는 설정
server {
           return 301 https://$host$request_uri;
         #  managed by Certbot


           listen 80;
           server_name awa24.site;
           return 404; # managed by Certbot
   }
```

- Backend Build
  - Git clone 이후 `./gradlew bootJar` 명령어 실행 시 build/libs/폴더 내 jar파일 생성
  - resource/application.yml

```
spring:
  datasource:
    url: jdbc:mariadb://i7c101.p.ssafy.io/AwA
    username: AwA
    password: tpwlslrnldudnj!#%
    driver-class-name: org.mariadb.jdbc.Driver
  security:
    oauth2:
      client:
        registration:
          google:
            client-id: 654989514571-ji9os6ujroj61ajmf3mviched2h7i3jh.apps.googleusercontent.com
            client-secret: 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
            scope:
              - profile
              - email
          naver:
            client-id: 3IlTyXUB0h2gwt4O1Lhv
            client-secret: 5ariPPNgVF
            redirect-uri: '{baseUrl}/{action}/oauth2/code/{registrationId}'
            authorization-grant-type: authorization_code
            client-name: Naver
            scope:
```

```
              - name
              - email
          kakao:
            authorization-grant-type: authorization_code
            client-id: 6aedbfc39ab1738bda8e5bbf952587ed
            client-secret: zVzwaD3ISM71jWvqc2uasloliMlbjHf5
            redirect-uri: "{baseUrl}/{action}/oauth2/code/{registrationId}"
            client-name: Kakao
            client-authentication-method: POST
            scope:
              - profile_nickname
              - account_email
        provider:
          naver:
            authorization-uri: https://nid.naver.com/oauth2.0/authorize
            token-uri: https://nid.naver.com/oauth2.0/token
            user-info-uri: https://openapi.naver.com/v1/nid/me
            user-name-attribute: response
          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

    port:
  mvc:
    pathmatch:
      matching-strategy: ant_path_matcher


  jpa:
    hibernate:
      ddl-auto: update
      properties:
        hibernate:
          #show_sql: true
          format_sql: true
    #generate-ddl: true

  mail:
    host: smtp.gmail.com
    port: 587
    username: kimsejin159@gmail.com
    password: wcoonqgxuqzfloqg
    properties:
      mail.smtp.auth: true
      mail.smtp.starttls.enable: true


logging:
  level:
    org.hibernate.sql: debug
    org.hibernate.type: trace

server:
  port: 8081
  ssl:
    key-store: classpath:keystore.p12
    key-store-type: PKCS12
    key-store-password: tpwlslrnldudnj!#%
  http2:
    enabled: true
```

- **resource/keystore.p12 파일 필요**

> https://s3-us-west-2.amazonaws.com/secure.notion-static.com/497921bf-2e76-454e-bc59-6f8b221a5c0d/Untitled.p12

- DB 접속 정보
  - Heidi
    - 네트워크 유형 : MariaDb or MySQL (TCP/IP)
    - Library : libmariadb.dll
    - 호스트명 / IP : i7c101.p.ssafy.io
    - 사용자 : AwA
    - 암호 : tpwlslrnldudnj!#%
    - 포트 : 3306

▼ CI/CD

> ec2 인스턴스 서버에 Jenkins를 설치하고 해당 서버에서 빌드하여 배포하였음

▼ Jenkins 설치 (in ubuntu)

> 💡 아래와 같은 순서로 하지 않으면 jenkins 설치 X

`sudo apt-get update`

jdk 설치

`sudo apt-get install openjdk-11-jdk`

gradle 설치 (Spring Boot 에서 사용한 gradle 과 같은 버전으로)

`sudo add-apt-repository ppa:cwchien/gradle`

`sudo apt-get install gradle-7.4.2`

`sudo apt-get install gradle-{원하는 버전}`

git 설치

```
sudo apt-get install git
```

젠킨스 저장소 key 다운로드

```
sudo wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -
```

source.list.d에 jenkins.list추가

```
echo deb http://pkg.jenkins.io/debian-stable binary/ | sudo tee
/etc/apt/sources.list.d/jenkins.list
```

key등록

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys FCEF32E745F2C3D5
```

```
sudo apt-get update
```

젠킨스 설치

```
sudo apt-get install jenkins
```

젠킨스 서버 포트 번호 변경 원할 시

```
sudo vi /etc/default/jenkins에서 HTTP_PORT=8080을 원하는 포트로 변경
```

젠킨스 상태확인(중지/시작)

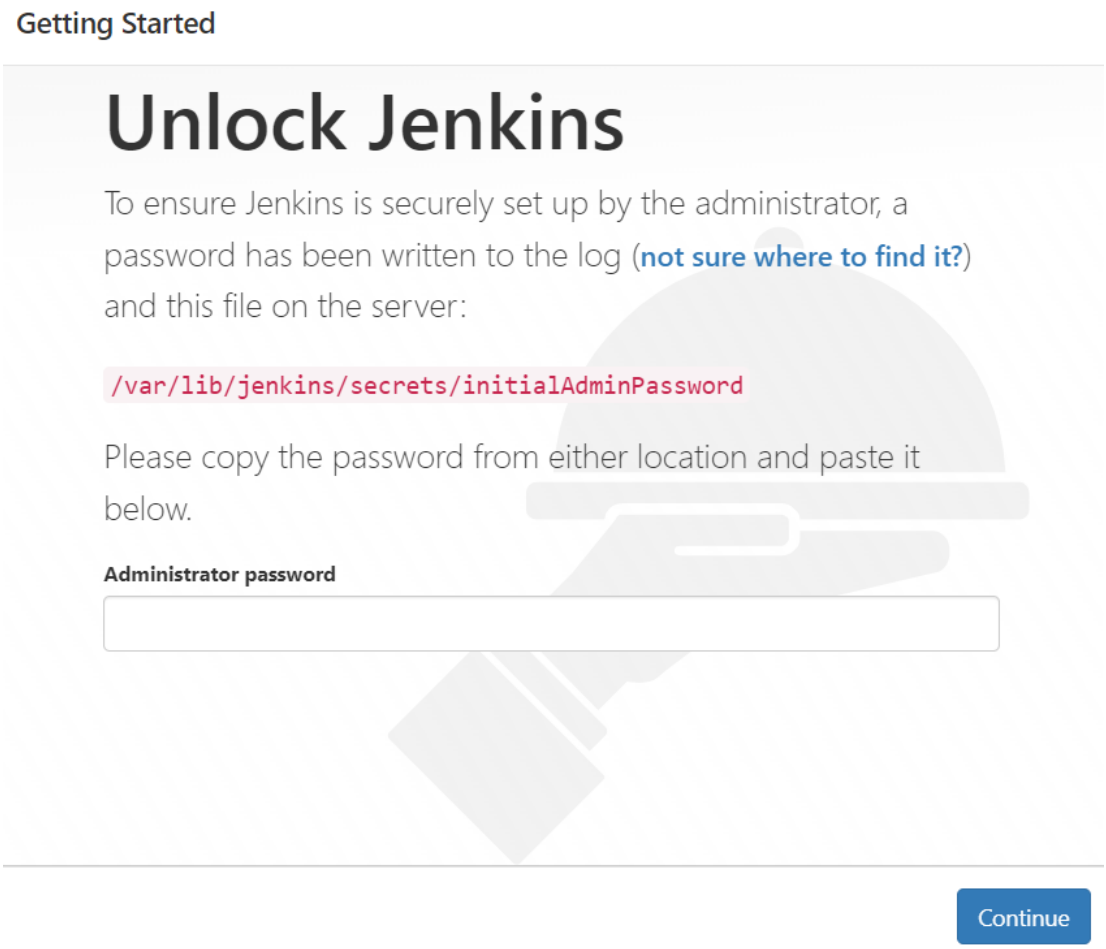
```
sudo systemctl status(stop/start) jenkins
```

젠킨스 비밀번호 확인

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

젠킨스 접속(ec2인스턴스주소:포트번호)

http://ec2-xxxxxxxxxxxxxxxx2.compute.amazonaws.com:8080



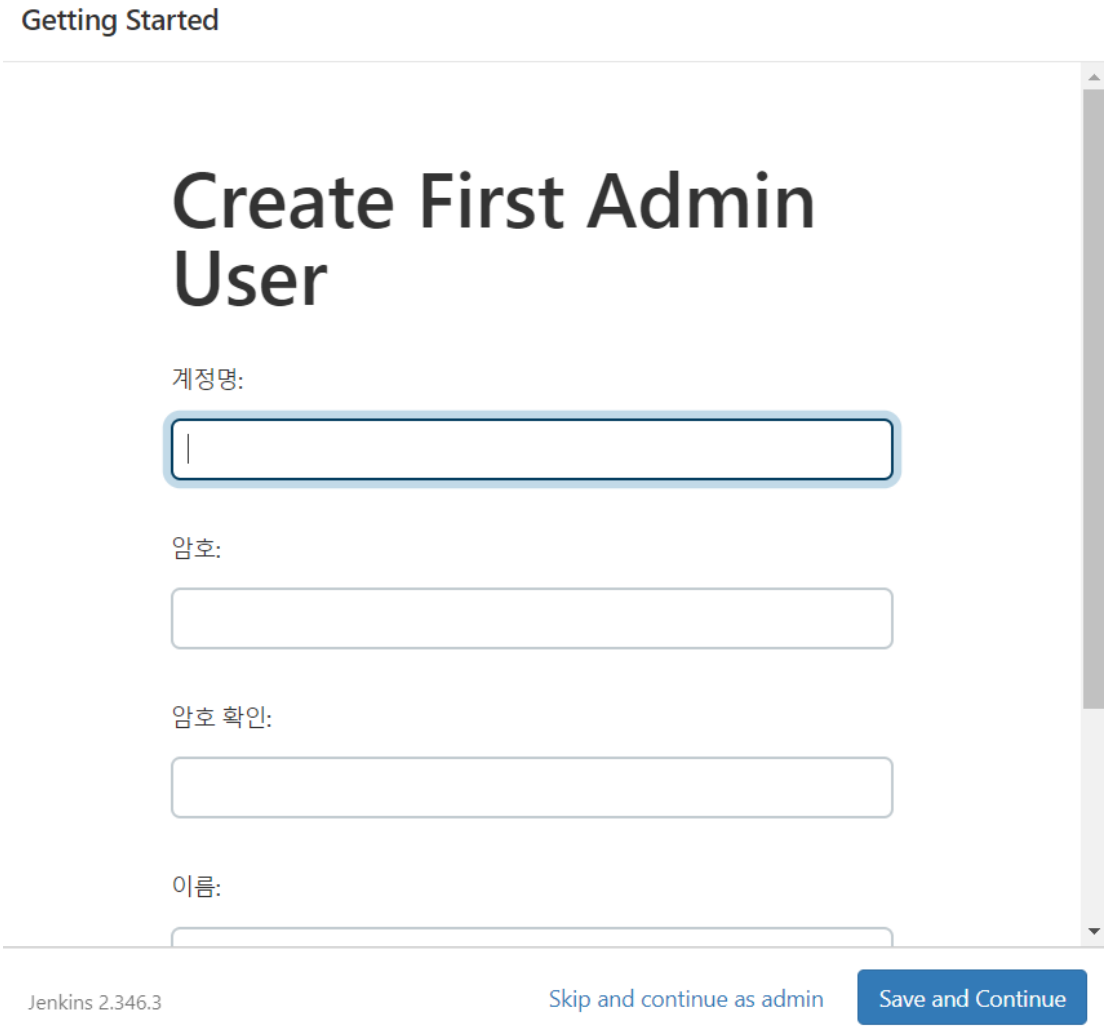
password는

> `sudo cat /var/lib/jenkins/secrets/initialAdminPassword`

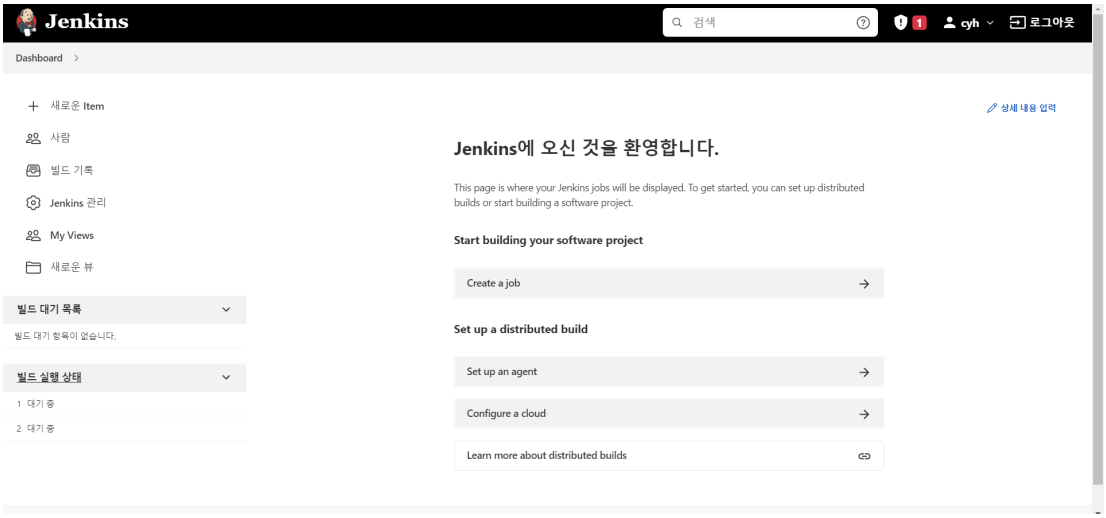
에서 확인한 비밀번호 입력

추천 플러그인 설치

계정명(ID), 비밀번호, 이름, 이메일 입력


Getting Started
Create First Admin User
계정명:
암호:
암호 확인:
이름:
Jenkins 2.346.3
Skip and continue as admin
Save and Continue

설치, 로그인 완료


Jenkins
검색
cyh
로그아웃
Dashboard
새로운 Item
사람
빌드 기록
Jenkins 관리
My Views
새로운 뷰
빌드 대기 목록
빌드 대기 항목이 없습니다.
빌드 실행 상태
1 대기 중
2 대기 중
상세 내용 입력
Jenkins에 오신 것을 환영합니다.
This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.
Start building your software project
Create a job
Set up a distributed build
Set up an agent
Configure a cloud
Learn more about distributed builds

▼ Jenkins, Gitlab 연결

Jenkins 프로젝트 생성

- 새로운 item 클릭, 프로젝트명 입력, Freestyle project 생성

Jenkins Plugin 설치 및 Global Tool Configuration 설정

- Jenkins 관리 → 플러그인 관리 → 설치 가능에서 gitlab 검색 후 설치
- Jenkins 관리 → Glodbal Tool Configuration → Add Gradle 클릭 → 이름, 버전 선택

생성된 프로젝트 설정

- 소스 코드 관리 (Git 선택)

  Repository URL ：연동하고자하는 gitlab repository 클론하여 입력

  Credentials → Add 선택

  - Kind (Username with password) 선택
  - Username : gitlab 아이디
  - Password : gitlab 패스워드
  - ID : Jenkins에서 보일 아이디
  - Description : Jenkins에서 보일 설명

  Branch Specifier : 입력한 branch에 있는 코드를 빌드

- 빌드 유발 (Build when a change is pushed to Gitlab ~ 선택)

  - 고급 선택 → Secret token 생성 (Generate)

  > 💡 GitLab Settings → Webhooks → URL (Build when a change is pushed to Gitlab ~ 선택했을 때 나오는 URL 입력) , Secret Token은 위에서 생성한 값 입력 → Add webhook 클릭

- Build (Execute shell 선택)

💡 Command에 아래 내용 입력

ls -al

REPOSITORY=/var/lib/jenkins/workspace/AwA/AwA
PROJECT_NAME=AwA

cd $REPOSITORY/$PROJECT_NAME

sudo git checkout Back

sudo chmod +x gradlew

sudo ./gradlew clean build

cd $REPOSITORY
sudo cp $REPOSITORY/$PROJECT_NAME/build/libs/*.jar $REPOSITORY/
JAR_NAME=$(ls -tr $REPOSITORY | grep jar | tail -n 1)

sudo nohup java -jar $REPOSITORY/$JAR_NAME 2>&1 &

끝! 깃랩 특정 branch에 push 하면 알아서 스프링부트 프로젝트가 빌드되고 백그라운드로 배포된다!