개발 환경

- Java 17.0.10
- Spring 6.1.3
- Spring Boot 3.2.2
- Gradle 8.5
- Spring Security Spring Boot 종속
- MySQL 8.0.30
- mongoDB 7.0.5
- IntelliJ 2023.3.2
- JSON WebToken jjwt api 0.11.5
- Spring Cloud AWS Starter 3.1.0
- AWS Java SDK S3 1.12.651
- JAVA 17.0.10 설치

https://www.oracle.com/java/technologies/javase/jdk17-archive-downloads.html

• JAVA 환경변수 설정 방법

컴퓨터 설정 - 시스템 - 정보 - 고급 시스템 설정

시스템 속성의 고급 탭 - 환경변수

시스템 변수에 변수 : JAVA_HOME 값 : java 설치된 폴더 경로 저장

JAVA_HOME C:\Program Files\Java\jdk-17

변수: Path 에 %JAVA_HOME%\bin 추가

• IntelliJ 2023.3.2 설치

jetbrains 공식 사이트에서 다운로드 후 진행

https://www.jetbrains.com/ko-kr/idea/

MySQL 설정

• MySQL 설치

MySQL :: Download MySQL Installer (Archived Versions)

Please note that these are old versions. New releases will have recent bug fixes and features! To download the latest release of MySQL Installer, please visit

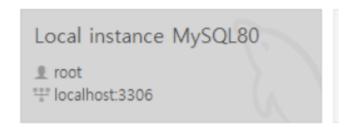
https://downloads.mysql.com/archives/installer/

MySQL 8.0.30로 버전 지정, 하단의 mysql-installer-community-8.0.30.0.msi을 통한 설치

기본 설정으로 설치, root 사용자에 대한 비밀번호 설정, 사용자 추가 없이 설치 진행

• Mysal 사용자 추가 및 DB 권한 부여

MySQL Connections ⊕ ®



• create database DB명; 으로 사용할 DB 생성



Navigator

MANAGEMENT

Server Status

Client Connections

Users and Privileges

Status and System Variables

Data Export

▲ Data Import/Restore

INSTANCE 🕄

Startup / Shutdown

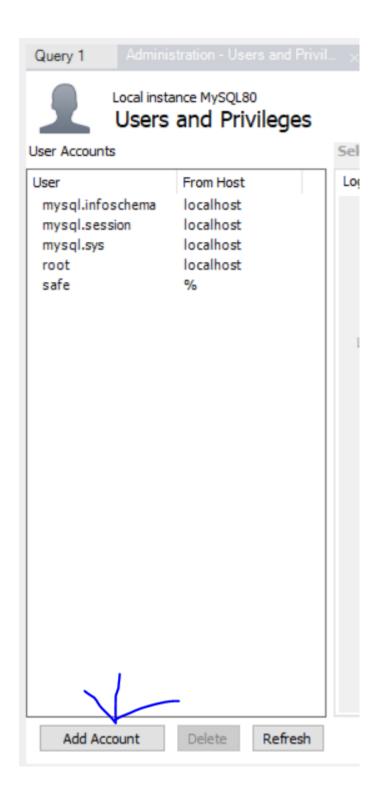
Server Logs

Ø Options File

PERFORMANCE

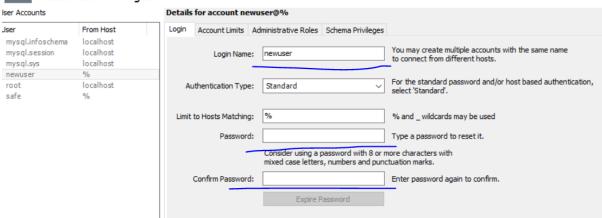
Dashboard

Performance Reports





Local instance MySQL80 Users and Privileges

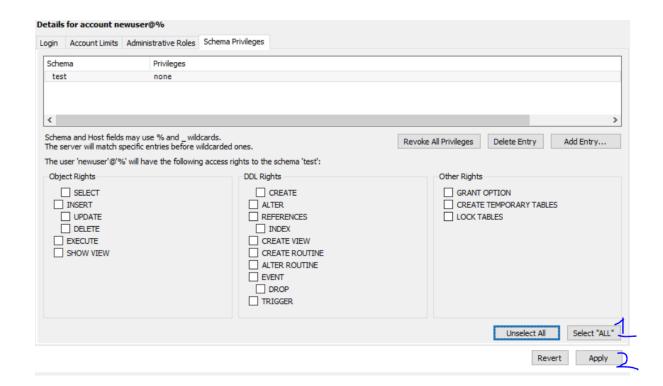






New Schema Privilege Definition

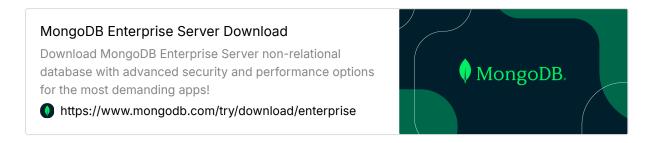
Select the Schema for which the user 'newuser' will have the privileges you want to define. Schema All Schema (%) This rule w This rule w Schemas matching pattern: You may u Escape the Select a sp Selected schema: information_schema



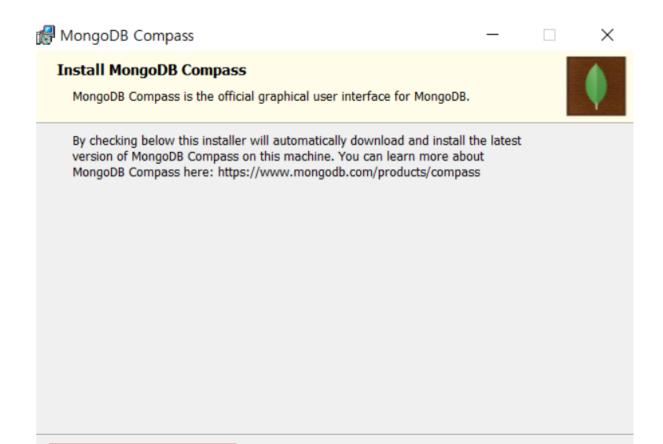
• 이후 생성한 회원으로 접속하면 지정한 DB만 사용 가능.

MongoDB 설정

- MongoDB 7.0.5, windows 버전, msi 로 다운로드.
- 다운로드 누르면 개인 정보 입력하는 창 나옴. 입력하고 submit 하면 다운로드 시작.



• 설치 도중 MongoDB Compass 설치하는지 물어봄. 체크 하고 넘어가기.



Back

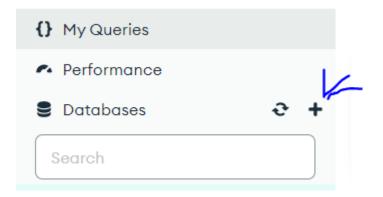
Next

Cancel

• msi 실행 하고 전부 기본 설정으로 설치

✓ Install MongoDB Compass

- 설치 끝나면 시스템 환경 변수 설정 창 열기.
- 시스템 변수 → path 누르고 편집 버튼 클릭.
- 새로 만들기 버튼 누르고 몽고db 설치한 경로에 있는 bin 폴더의 주소 입력.
 ex) C:\Program Files\MongoDB\Server\7.0\bin
- MongoDB Compass 실행해서 connect 누르기.
- <u>localhost:27017</u> 로 이동하면 It looks like you are trying to access MongoDB over HTTP on the native driver port. 라는 문구가 나올 것.
- MongoDB Compass 에서 DB 생성.



Create Database

×

Database Name

checkList

Collection Name

checkLists

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. Learn More

Additional preferences (e.g. Custom collation, Capped, Clustered collections)



• Mongosh 설치를 위해 해당 링크로 이동.

MongoDB Shell Download

The MongoDB Shell is a modern command-line experience, full with features to make it easier to work with your database. Free download. Try now!

https://www.mongodb.com/try/download/shell



- 설치 기준 버전은 2.1.1 인데 버전은 상관X, windows x64(10+), 패키지는 zip 형태로 다운로드.
- 압축 풀고 폴더에 들어가고 bin 안에 있는 mongosh.exe 파일 실행. 해당 파일로 몽고 db에 CLI 형태로 명령어 사용 가능.
- 파일 실행하고 엔터 누르면 Enterprise test> 이런 형태로 입력 대기중 일거임.
- use admin 입력해서 admin이라는 이름의 DB 사용.
- db.createUser({ user: 'safe', pwd: 'safe102', roles: [{ "role": 'readWrite', "db": "checkList" }] }) 사용자 추가하는 명령어 입력. safe나 safe102 대신 원하는 걸로 생성해도 됨.
- 이러면 mongodb 설정과 <u>application.properties</u>에 입력할 몽고db 사용자 생성 완료.

S3 버킷 설정

AWS root 계정 생성

- 버킷 버킷 만들기
- 버킷명은 고유값으로 생성
- 리전은 EC2 서버와 동일하게
- 객체 소유권 설정 [타 계정도 소유권이나 접속제어 가능 유무]
 - ACL 비활성화[내가 접속한 계정에서만 소유]모든 퍼블릭 엑세스 차단 해제, 버킷 버전 관리 비활성화
 - 。 기본 암호화 SSE-E3, 버킷 키 활성화

IAM 사용자 생성 권한 정책 - AmazonS3FullAccess 사용자 생성

IAM Identity Center(AWS Single Sign-On)을 통한 사용자 생성

사용자 생성 - 생성 시 MFA 인증 추가 필요

SSO login 진행 - Command line or programmatic access

o Acces Key, Secret Key 발급

설정

application.properties

```
spring.datasource.driver-class-name=com.mysgl.cj.jdbc.Driv
spring.datasource.url=jdbc:mysql://[URL]:3306/[DB Name]?se
spring.datasource.username=[username]
spring.datasource.password=[userpassword]
spring.data.mongodb.host=[hostname(localhost)]
spring.data.mongodb.port=27017
spring.data.mongodb.authentication-database=[사용자 정보가 저
spring.data.mongodb.database=[DB Name]
spring.data.mongodb.username=[username]
spring.data.mongodb.password=[userpassword]
spring.jpa.hibernate.ddl-auto=update
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect
spring.servlet.multipart.max-file-size=5MB
spring.servlet.multipart.max-request-size=20MB
spring.servlet.multipart.enabled=true
spring.profiles.include=oauth
#AWS auth
cloud.aws.credentials.accessKey=[S3 Bucket Access Key]
cloud.aws.credentials.secretKey=[S3 Bucket Secret Key]
cloud.aws.region.static=ap-northeast-2
cloud.aws.stack.auto=false
cloud.aws.s3.bucket=[S3 Bucket Name]
```

```
#CLOVA OCR
naver.ocr.apiUrl=[CLOVA OCR API URL]
naver.ocr.secretKey=[CLOVA OCR API Secret Key]
# google login
google.auth.url=https://oauth2.googleapis.com
google.login.url=https://accounts.google.com
google.redirect.uri=https://[Domain Name]/api/google/login.
google.client.id=[Google Client ID]
google.secret=[Google Secret Key]
google.auth.scope=profile,email,openid
# openVidu
OPENVIDU_URL: https://[Domain Name or ip]:4443
OPENVIDU_SECRET: [Secret Key]
# ssl server
server.port = [port]
server.address = [서버 주소]
server.ssl.key-store-type=[인증서 store 타입]
server.ssl.key-store=[인증서 store 위치]
server.ssl.key-store-password=[인증서 store password]
server.ssl.key-password=[인증서 password]
server.http.enabled=true
```

jwt.yml

```
secret-key: [JWT Secret Key]
expiration-hours: 3
issuer: [issuer Name]
```

외부 서비스

- CLOVA OCR
 - https://guide.ncloud-docs.com/docs/ko/clovaocr-overview

- S3
 - https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html
- Google Login
 - https://cloud.google.com/identity-platform/docs/web/google?hl=ko