3주차 진행 보고서

2021.03.2~ 2021.03.17 B71116장혜리

이전 계획 및 수행 내용

# 서버 구축 및 DB 추가

## 수행 내용

1. AWS EC2에 톰캣 서버 구축, Spring프로젝트 생성 및 연동
2. AWS RDB에 데이터베이스 생성
   * 기존 NoSQL 서버에는 채팅 정보만 저장
   * 나머지 데이터는 RDB에 저장

## 변경사유

안드로이드에서는 보안상의 이유로 외부라이브러리와의 직접 연동이 불가능하기 때문에 웹을 거쳐서 DB에 접속해야 합니다. 다시 말해 별도의 웹 서버를 필요로 합니다. 하지만 구글에서 제공하는 NoSQL 데이터베이스인 Cloud Firestore와 파일 저장소인 Cloud Storage를 사용하면 따로 웹 서버를 구축하지 않고 데이터와 파일을 주고받을 수 있습니다. 이러한 점 때문에 지난학기에는 시간 절약을 위해 위 두가지 서비스를 프로토타입 제작에 사용하였습니다. 하지만 개발을 이어 나가면서 아래와 같은 이유로 데이터베이스와 파일 저장소 변경을 고려할 수밖에 없었습니다.

### Cloud Firestore의 데이터 중복 문제

NoSQL DB는 데이터가 여러 컬렉션에 중복되어 있기 때문에 수정 시 모든 컬렉션에서 수정해야하는 단점이 있다. 이는 데이터가 자주 변경(수정)되는 우리 애플리케이션에 적합하지 않은 데이터베이스이다.

### Cloud Storage의 느린 속도

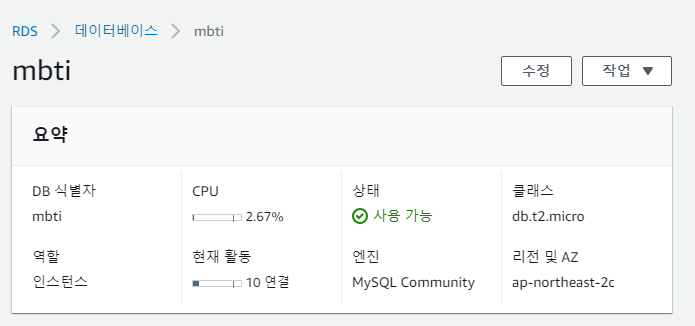
구글에서 제공하는 파일 저장소인 firebase Storage는 업로드/다운로드 속도가 매우 느리다. 이미지를 압축해서 올려도 클라이언트에 반영되기까지 수 초가량 소요됩니다.

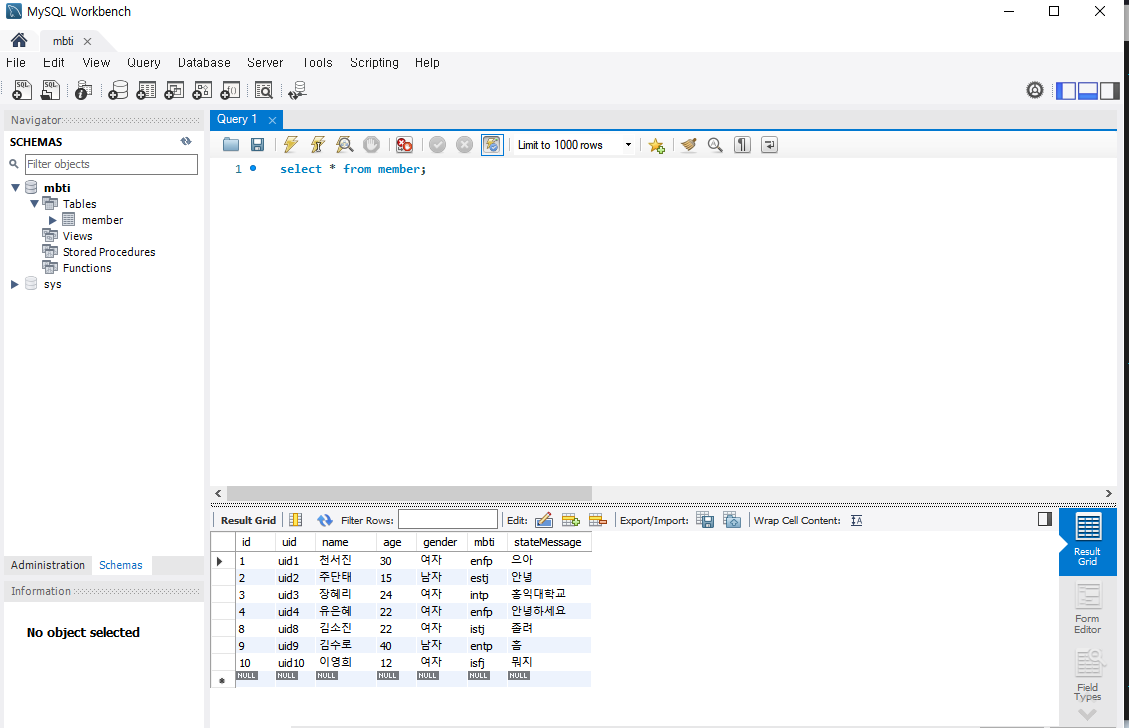
## 결론

때문에 방학동안 AWS EC2를 이용하여 새롭게 서버를 구축하였습니다. 해당 서버는 이미지 파일 저장 및 DB접근을 위한 웹서버로 활용할 예정입니다.



또한 AWS RDB를 이용하여 MySQL 데이터베이스를 새로 구축하였습니다. 수정이 없고 읽기 빈도가 잦은 소셜 로그인 및 채팅 정보 저장에는 기존에 사용하던 NoSQL DB를, 그 외 데이터는 새로 구축한 RDB에 저장할 예정입니다.





## 참조

[스프링 부트와 AWS로 혼자 구현하는 웹 서비스](https://velog.io/@swchoi0329/series/%EC%8A%A4%ED%94%84%EB%A7%81-%EB%B6%80%ED%8A%B8%EC%99%80-AWS%EB%A1%9C-%ED%98%BC%EC%9E%90-%EA%B5%AC%ED%98%84%ED%95%98%EB%8A%94-%EC%9B%B9-%EC%84%9C%EB%B9%84%EC%8A%A4)

[SQL vs NoSQL (MySQL vs. MongoDB)](https://siyoon210.tistory.com/130)

[SQL? NoSQL?](https://brunch.co.kr/@kooslab/181)

[스프링 핵심 원리 - 기본편 - 인프런 | 강의](https://www.inflearn.com/course/%EC%8A%A4%ED%94%84%EB%A7%81-%ED%95%B5%EC%8B%AC-%EC%9B%90%EB%A6%AC-%EA%B8%B0%EB%B3%B8%ED%8E%B8)

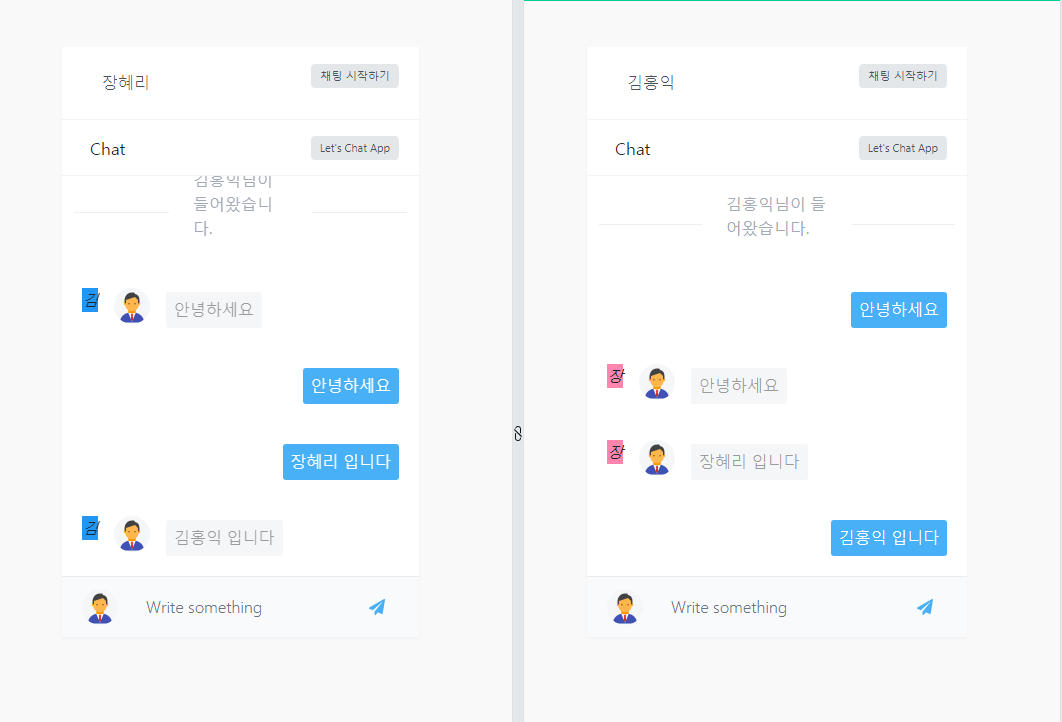
# 채팅 구현

## 수행 내용

1. Spring 프레임워크인 WebSocket을 활용하여 유저간 채팅 기능 구현

## WebSocket을 활용한 유저 간 채팅 기능 구현

현재 유저간 메시지를 주고받는 단순 채팅 기능만 완성한 상태입니다. [MBTI chat](http://52.78.50.239:8080/chat) 에서 동작을 확인하실 수 있습니다.



인터페이스 구현 시 오픈 소스 프레임 워크인 부트스트랩을 사용하였습니다. 채팅 구현에 이용한 템플릿의 링크는 다음과 같습니다. [Bootstrap 4 Simple chat application](https://bbbootstrap.com/snippets/simple-chat-application-57631463)

## 참조

[SpringBoot - 스프링부트에서 채팅프로그램(소켓통신) 만들기-1(단순 채팅, 메시지 보내기) :: 개발 흔적 남기기](https://myhappyman.tistory.com/102?category=873296)

[[Spring]Springboot + websocket 채팅[1]](https://ratseno.tistory.com/71?category=773803)

[Bootstrap 4 Simple chat application](https://bbbootstrap.com/snippets/simple-chat-application-57631463)

# 테스팅 자동화

* 단위테스트/통합테스트 도입
  + 테스트 라이브러리: spring-boot-starter-test
* 배포 및 테스팅 자동화를 위한 Travis CI 환경 설정

## 단위테스트/통합테스트 도입

테스트 라이브러리로 spring-boot-starter-test을 사용하였습니다. 해당 라이브러리를 통해 사용하는 기능은 다음과 같습니다.

### spring-boot-starter-test

junit: 단위 테스트 프레임워크

assertj: 테스트 코드 가독성을 돕는 라이브러리

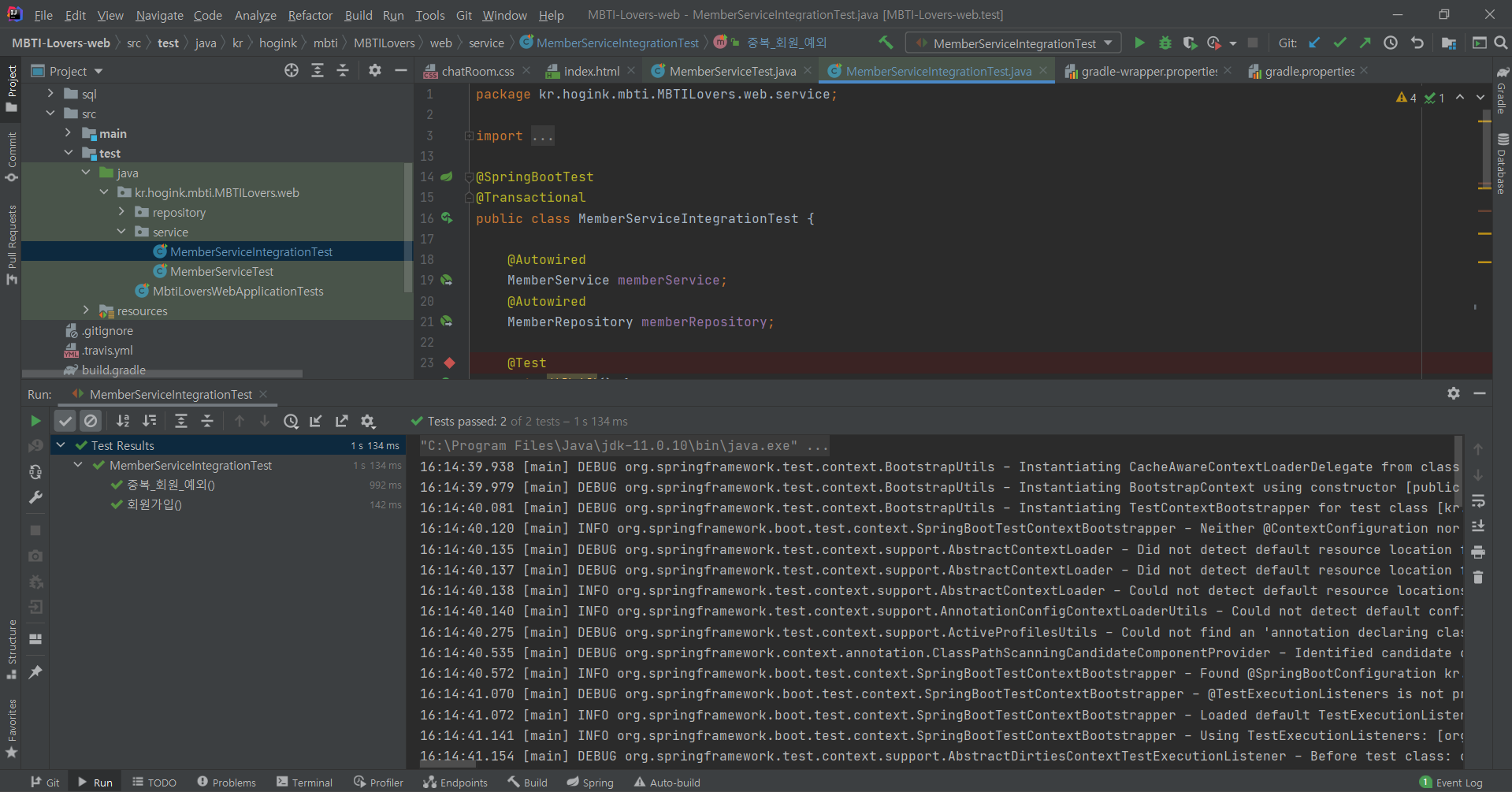
spring-test: 스프링 통합 테스트

현재 회원가입 및 조회에만 테스트가 적용되어 있습니다.

### 단위 테스트 예시

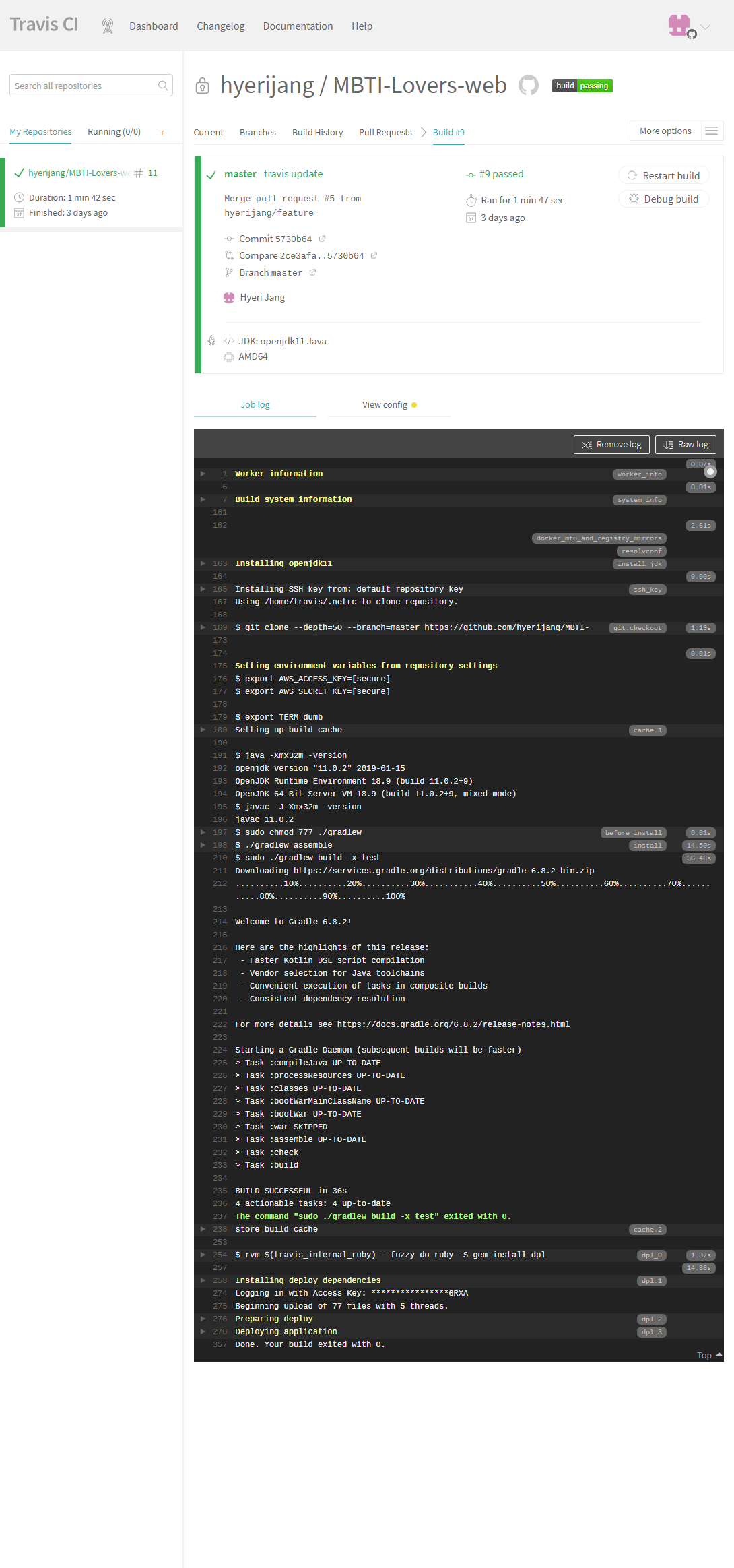
### 

### 통합테스트 예시



## 배포 및 테스팅 자동화를 위한 Travis CI 환경 설정

Travis CI와 AWS S3, CodeDeploy를 사용하여 배포 및 테스팅을 자동화하였습니다. master 브랜치에 커밋이 push 되면 자동으로 서버에 반영됩니다.



## 참조

[스프링 핵심 원리 - 기본편 - 인프런 | 강의](https://www.inflearn.com/course/%EC%8A%A4%ED%94%84%EB%A7%81-%ED%95%B5%EC%8B%AC-%EC%9B%90%EB%A6%AC-%EA%B8%B0%EB%B3%B8%ED%8E%B8)

[스프링 부트와 AWS로 혼자 구현하는 웹 서비스](https://velog.io/@swchoi0329/series/%EC%8A%A4%ED%94%84%EB%A7%81-%EB%B6%80%ED%8A%B8%EC%99%80-AWS%EB%A1%9C-%ED%98%BC%EC%9E%90-%EA%B5%AC%ED%98%84%ED%95%98%EB%8A%94-%EC%9B%B9-%EC%84%9C%EB%B9%84%EC%8A%A4)

다음 면담까지 계획

* 서버 구축 및 DB 연결
  + 안드로이드 프로젝트에 mySQL DB 연결 및 소스 코드 수정
* 채팅 구현
  + 채팅방 구현
  + 채팅 기록 DB에 저장
* 무중단 배포를 위한 NGINX 환경 설정