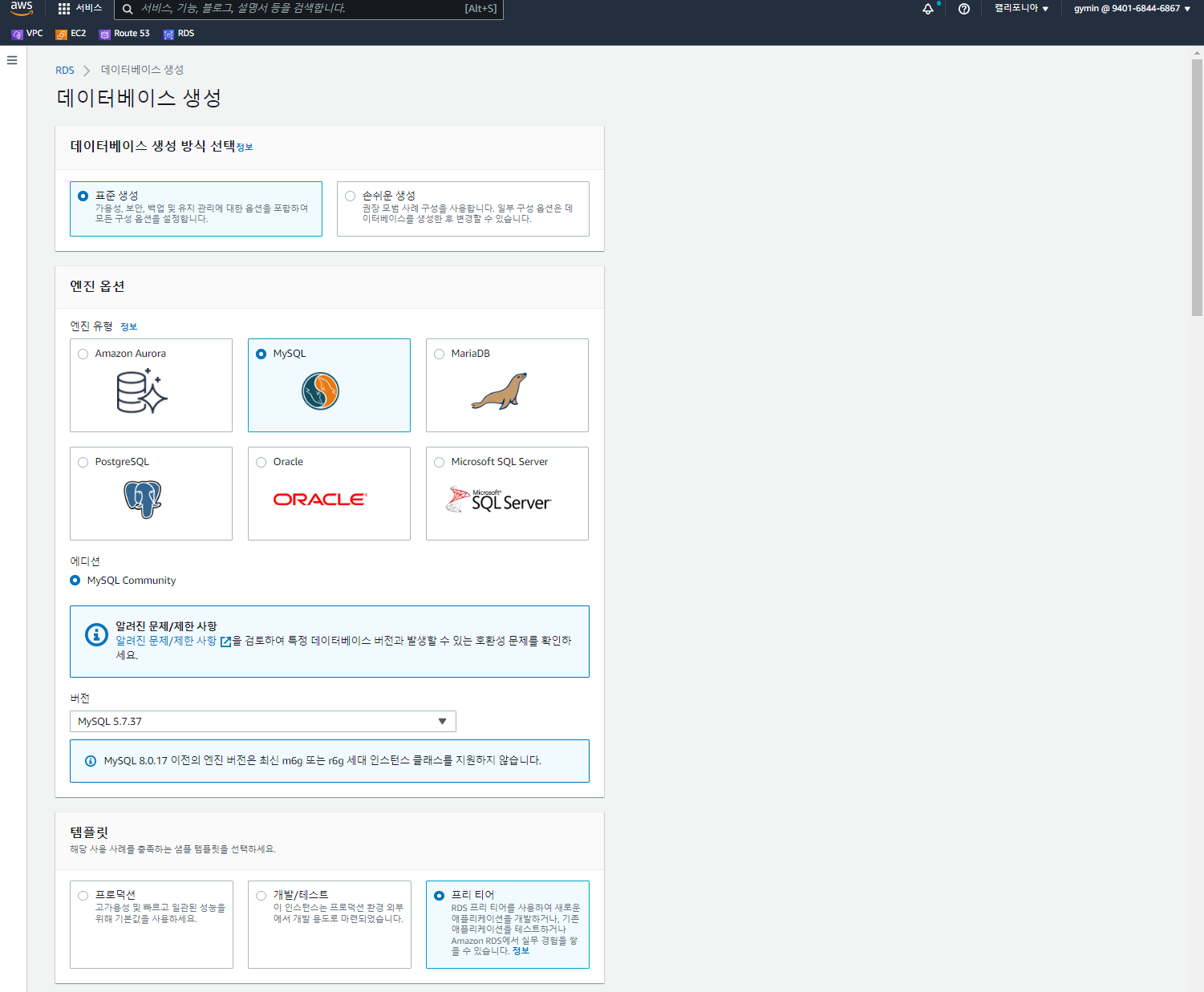
# AWS 2tier 구성하기

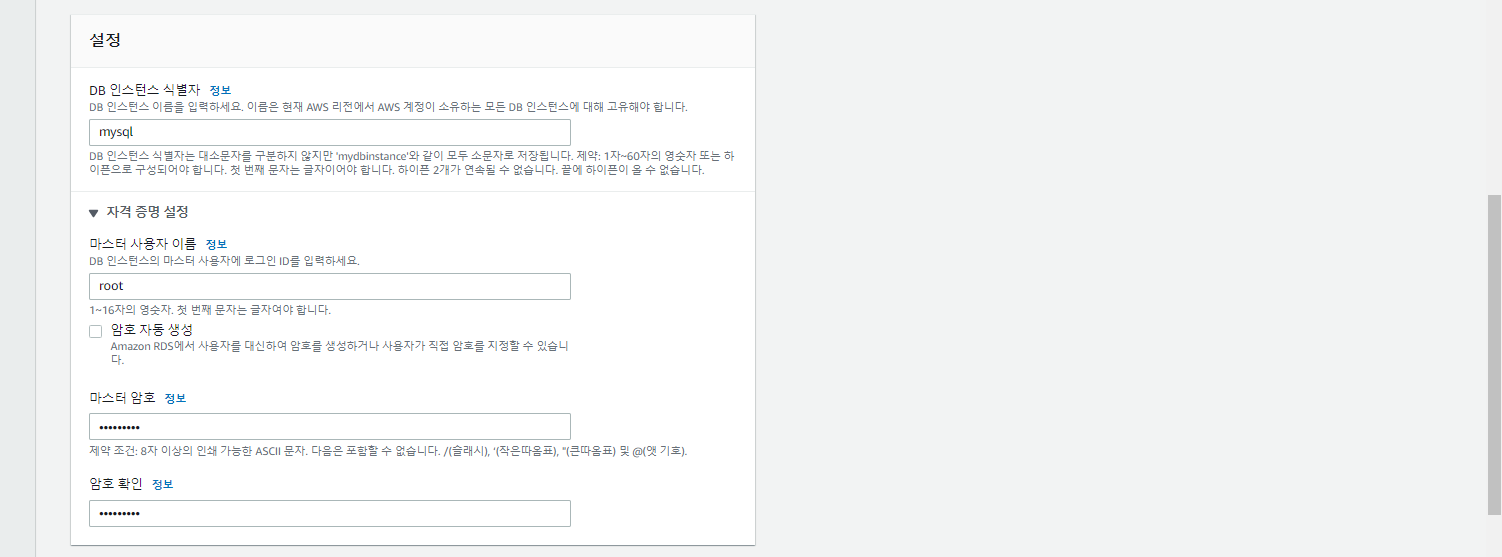
Web 단을 생략하고 바로 was로 접속하여 CICD-test 창이 뜨도록한다

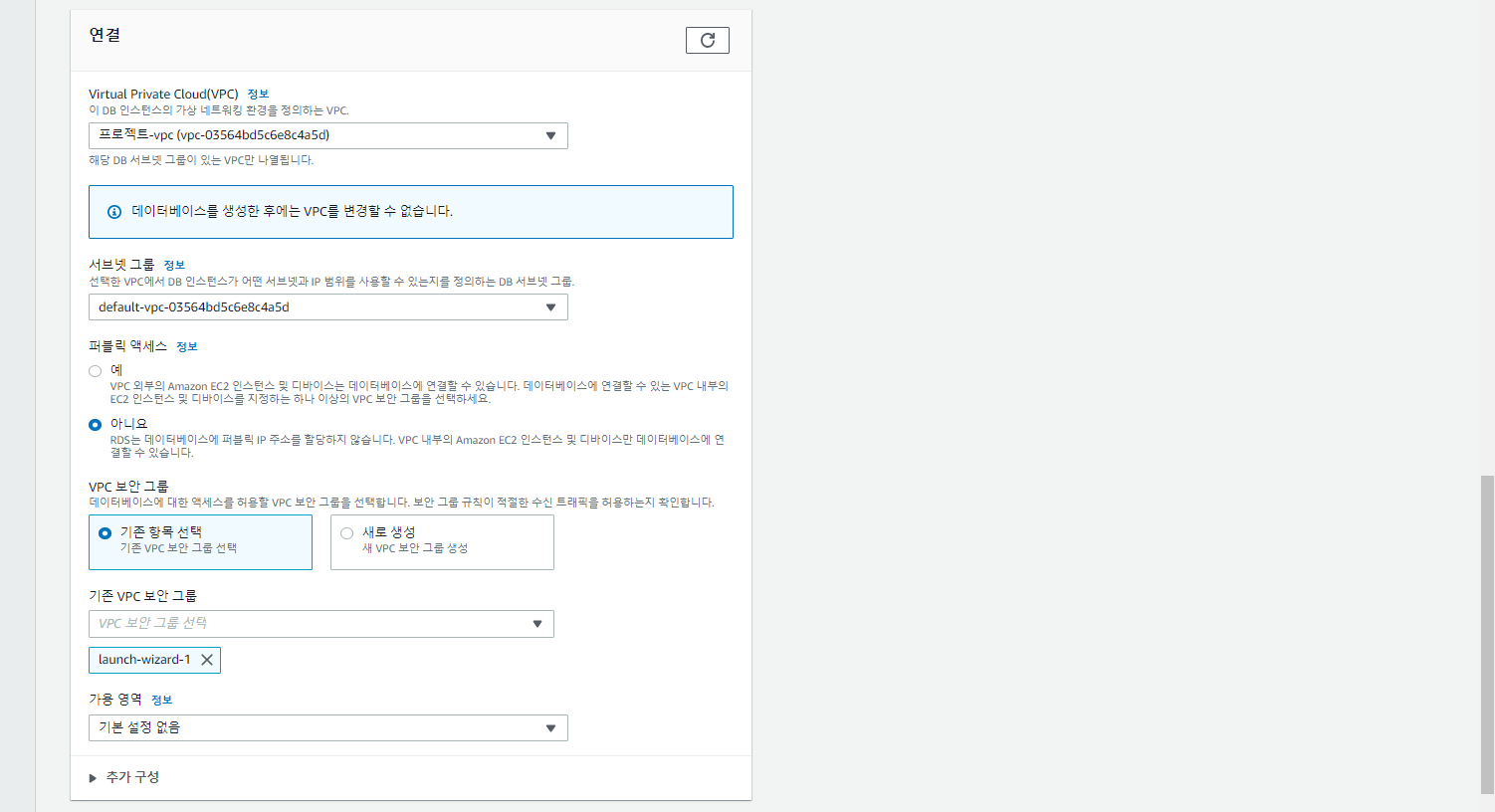
WAS 서버 접속시 LB를 통해 접속하도록 설정한다

DB는 aws RDS를 사용하며 mysql 5.7을 사용한다

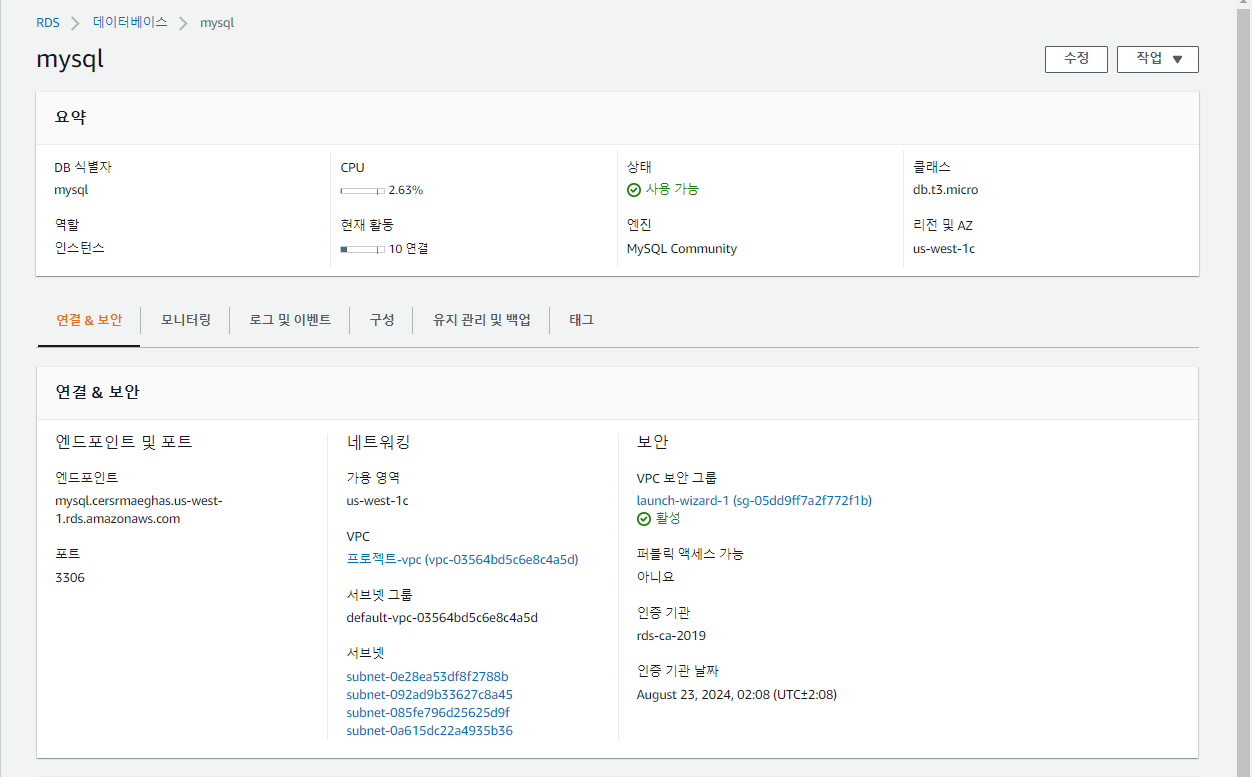
## RDS 만들기







* 나머지는 그대로 두고 생성



엔드포인트를 기억해둔다

mysql.cersrmaeghas.us-west-1.rds.amazonaws.com

## Redis 서버 생성

Redis용 VM을 하나 생성하고 접속한다

* AMI : amazon linux
* 보안그룹 : all-tcp

다음 명령어 입력

|  |
| --- |
| sudo su -  sudo amazon-linux-extras install -y epel  yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm  yum install -y redis  vi /etc/redis.conf  /bid  bind 0.0.0.0 # 추가  # bind 127.0.0.1 # 주석처리  /requirepass foobared  requirepass frodo5020!! # 추가, 패스워드 입력  # requirepass foobared # 주석처리  systemctl start redis |

## DB 생성

RDS와 같은 VPC에서 VM을 하나 만들어 RDS에 접속한다 (redis vm 사용 가능)

|  |
| --- |
| mysql -h mysql.cersrmaeghas.us-west-1.rds.amazonaws.com -u root -p  CREATE DATABASE boot\_board default CHARACTER SET UTF8;  CREATE TABLE `tbl\_board` (  `board\_idx` int NOT NULL AUTO\_INCREMENT,  `title` varchar(50) NOT NULL,  `contents` text,  `hit\_count` int DEFAULT '0',  `created\_datetime` date DEFAULT NULL,  `creator\_id` varchar(50) NOT NULL,  `updated\_datetime` date DEFAULT NULL,  `updater\_id` varchar(50) DEFAULT NULL,  `deleted\_yn` varchar(5) DEFAULT NULL,  PRIMARY KEY (`board\_idx`)  );  CREATE TABLE `tbl\_mail` (  `mail\_idx` int NOT NULL AUTO\_INCREMENT,  `receiver` varchar(5) NOT NULL,  `sender` varchar(5) NOT NULL,  `title` varchar(50) NOT NULL,  `contents` text,  `read\_yn` varchar(5) NOT NULL,  `deleted\_yn` varchar(5) NOT NULL,  `mail\_date` date NOT NULL,  PRIMARY KEY (`mail\_idx`)  );  CREATE TABLE `tbl\_users` (  `user\_name` varchar(50) NOT NULL,  `passwd` varchar(50) NOT NULL,  `enabled` tinyint DEFAULT '1',  `first\_name` varchar(50) NOT NULL,  `last\_name` varchar(50) NOT NULL,  `regdate` datetime DEFAULT NULL,  PRIMARY KEY (`user\_name`),  UNIQUE KEY `user\_name\_UNIQUE` (`user\_name`)  );  CREATE TABLE `tbl\_users\_roles` (  `user\_name` varchar(50) NOT NULL,  `role\_name` varchar(50) NOT NULL,  PRIMARY KEY (`user\_name`),  CONSTRAINT `FK\_users\_roles` FOREIGN KEY (`user\_name`) REFERENCES `tbl\_users` (`user\_name`)  ); |

## WAS 서버 생성

1. JDK, tomcat 설치

<https://gammistory.tistory.com/191> 참고

2. was file 생성

|  |
| --- |
| server.port=8080  spring.main.allow-bean-definition-overriding=true  spring.datasource.hikari.driver-class-name=com.mysql.cj.jdbc.Driver  spring.datasource.hikari.jdbc-url=jdbc:mysql://mysql.cersrmaeghas.us-west-1.rds.amazonaws.com:3306/boot\_board?serverTimezone=UTC  spring.datasource.hikari.username=root  spring.datasource.hikari.password=rlarudals  spring.datasource.hikari.connection-test-query=SELECT 1  #root  mybatis.configuration.map-underscore-to-camel-case=true  spring.session.store-type=redis  spring.redis.host=10.0.7.198  spring.redis.password=frodo5020!!  spring.redis.port=6379  spring.session.redis.flush-mode=on\_save  spring.session.redis.namespace=spring:session  spring.thymeleaf.cache=false  spring.thymeleaf.enabled=true  spring.thymeleaf.prefix=classpath:/templates/  spring.thymeleaf.suffix=.html |

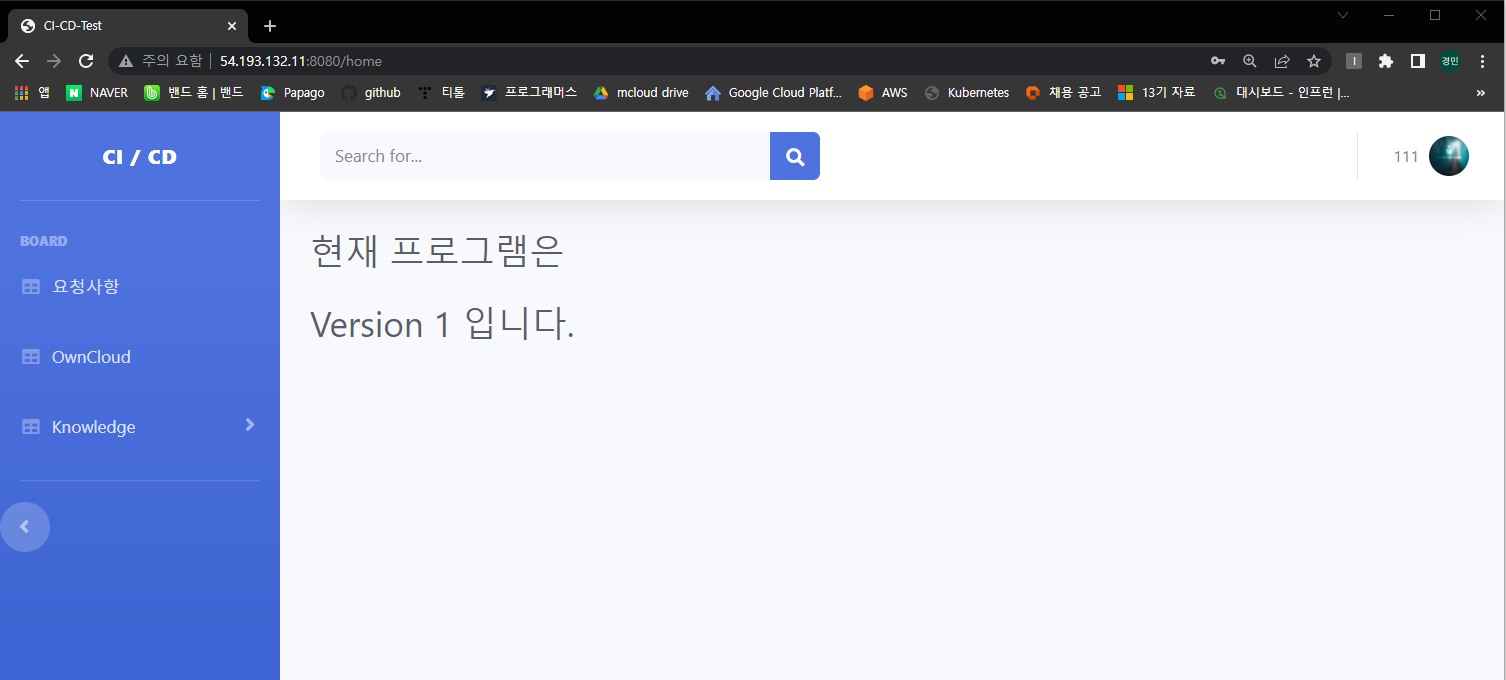
3. 빌드하기

|  |
| --- |
| gradle build |

4. 빌드한 파일을 tomcat server로 옮기기

|  |
| --- |
| cp /home/ec2-user/my-spring-board-0.0.1-SNAPSHOT.war /opt/tomcat/apache-tomcat-9.0.62/webapps/ROOT.war  systemctl restart tomcat |

Tomcat VM 접속하기

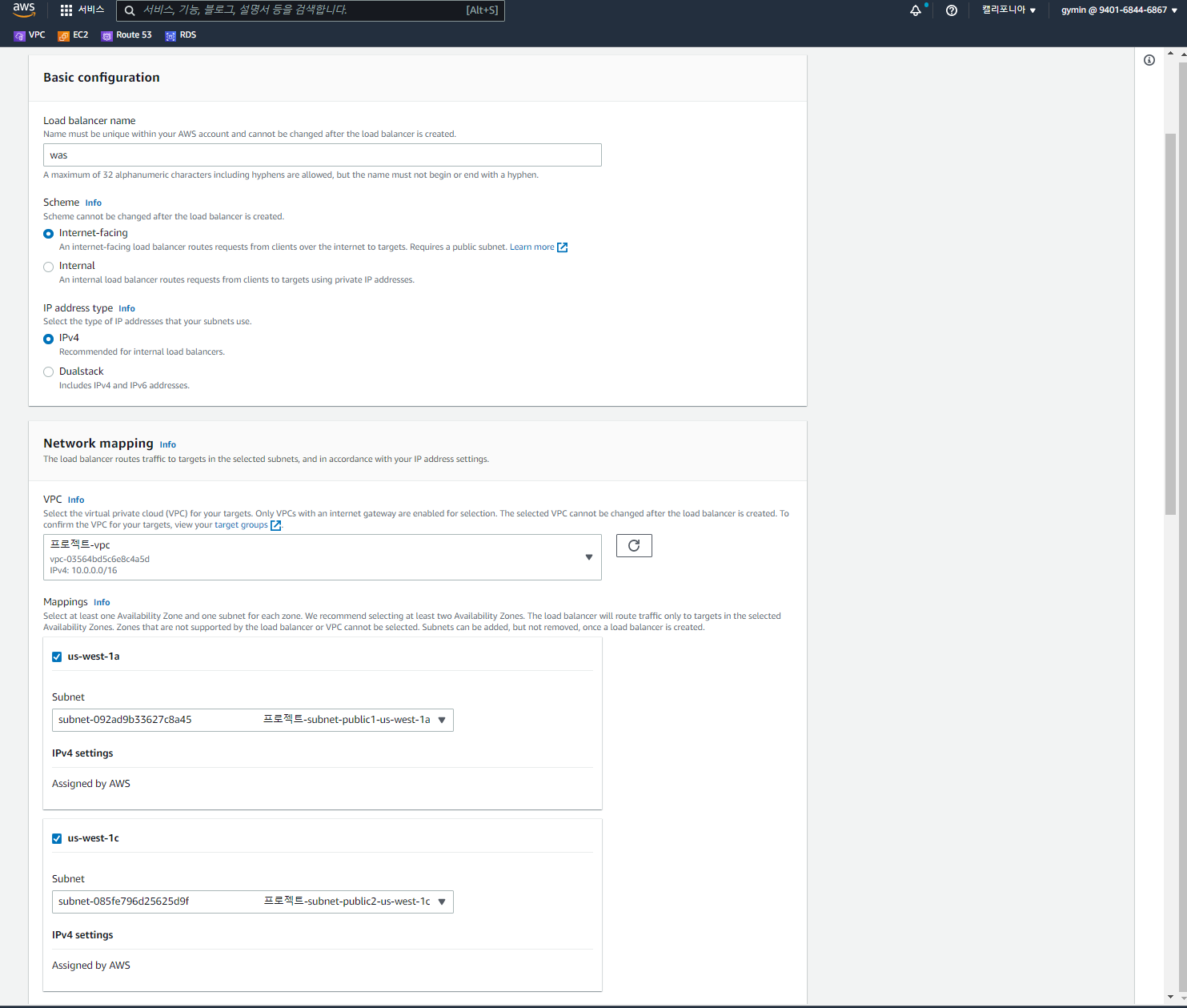


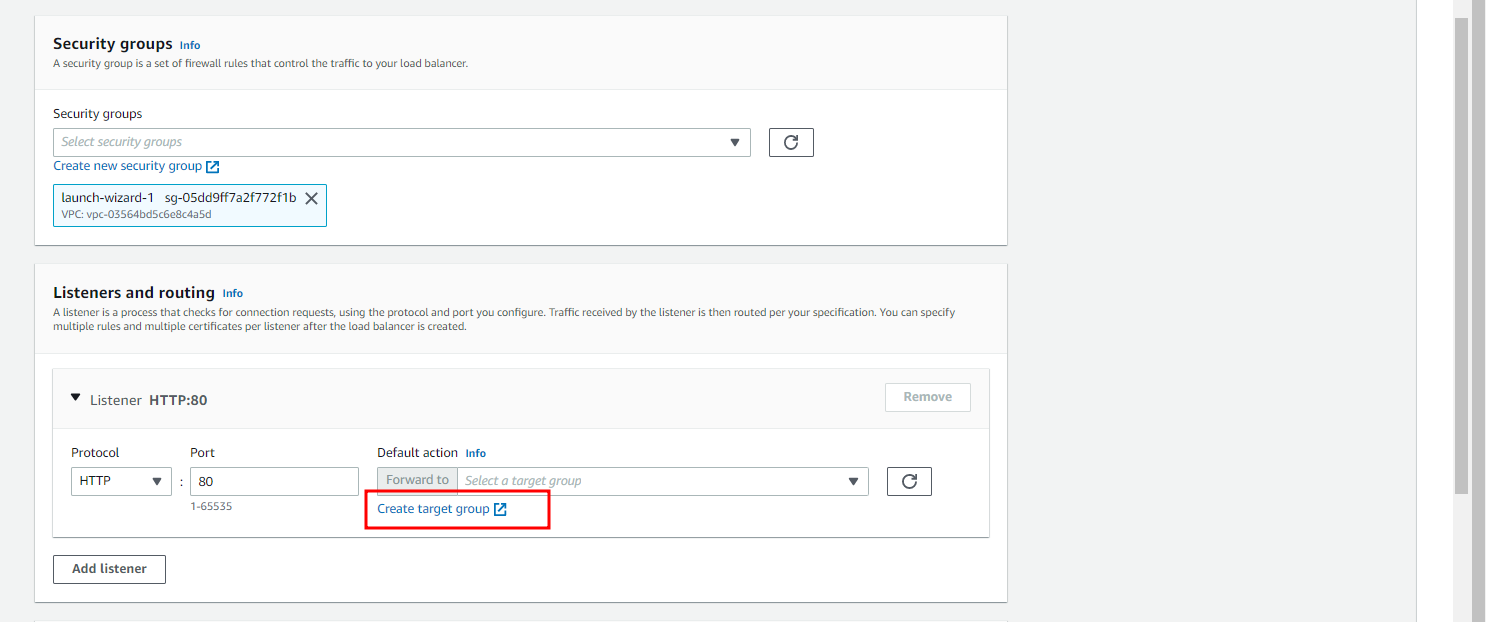
로그인도 잘 됨

## Load Balancer

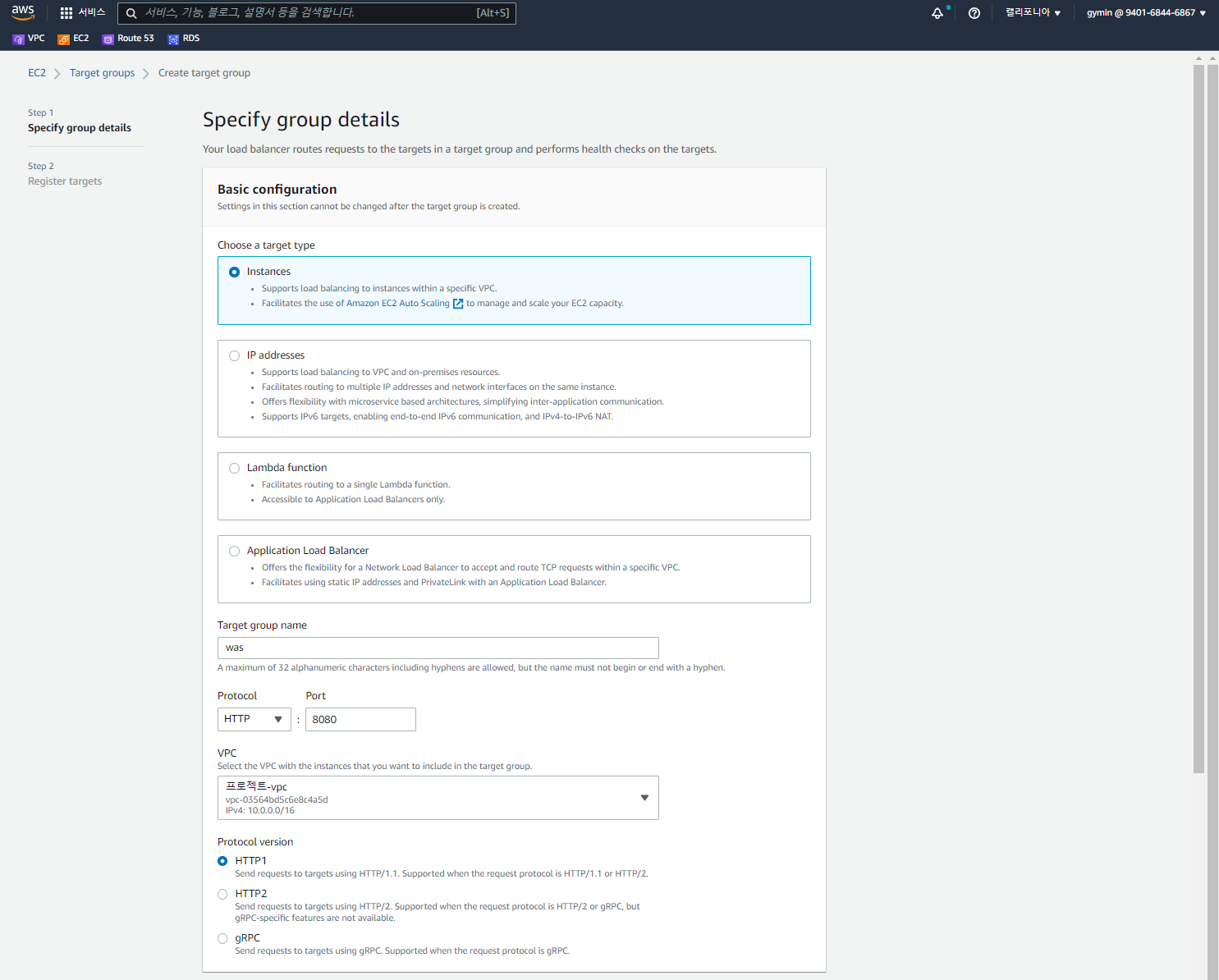
tomcat (port 8080) LB

Application Load Balancer 로 생성

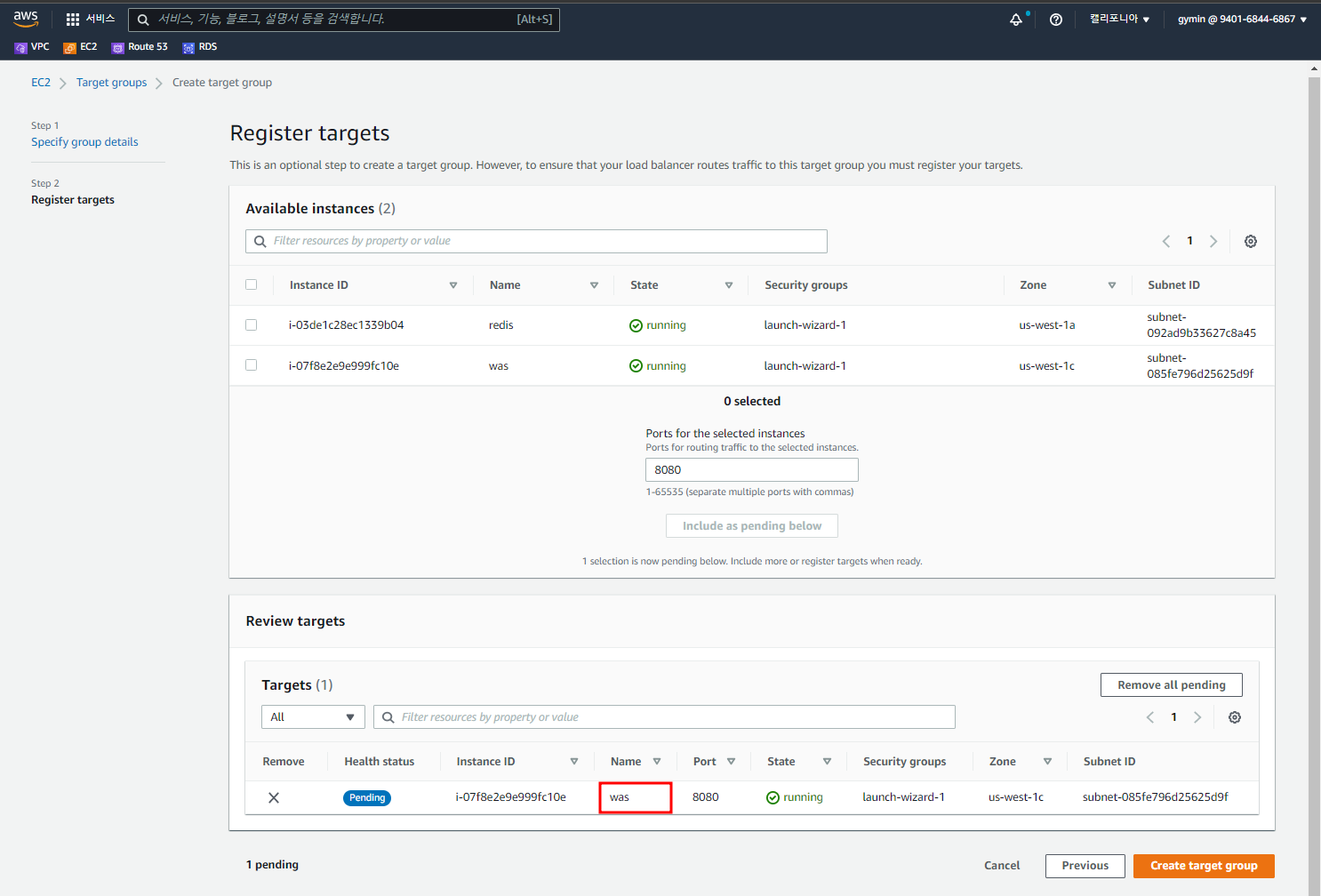




대상 그룹

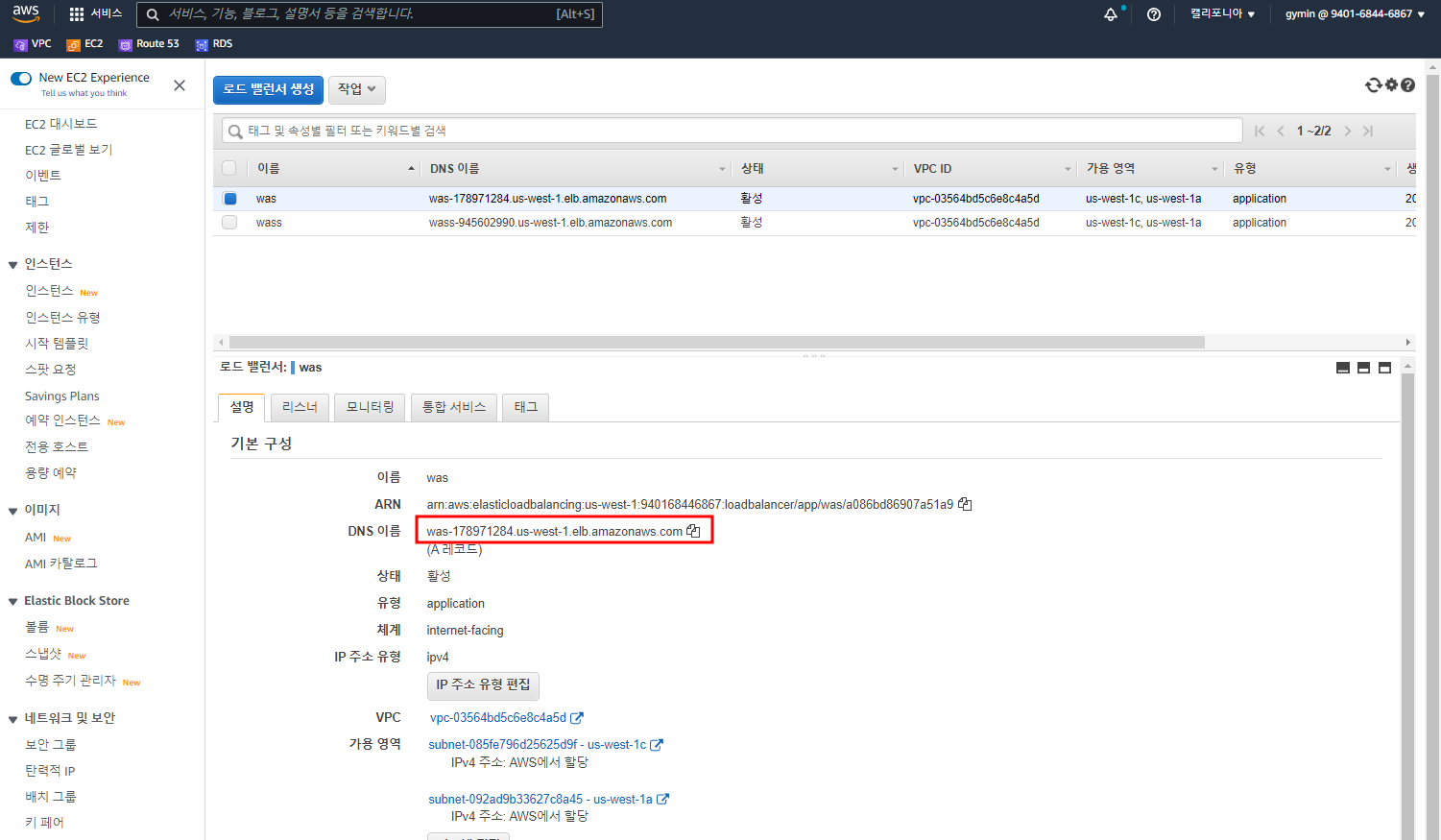


Next

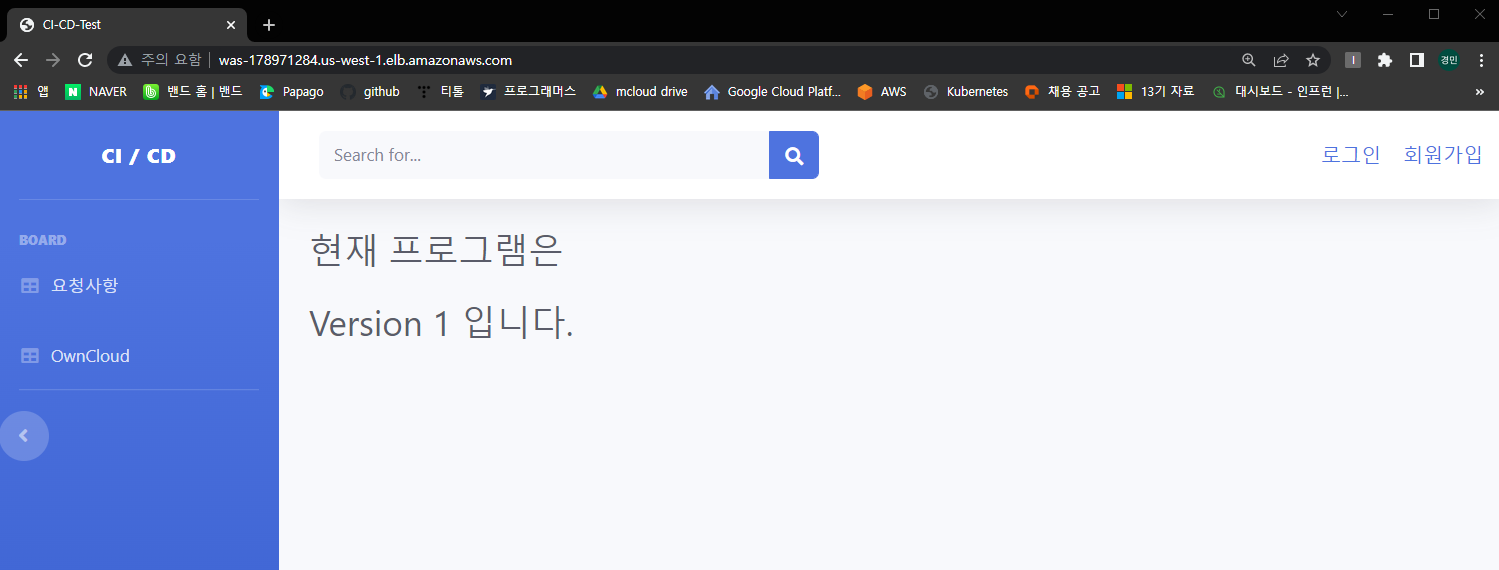


LB설정으로 돌아가서 방금 만든 대상그룹 선택 후 생성 확인

## LB 접속확인하기



접속 확인



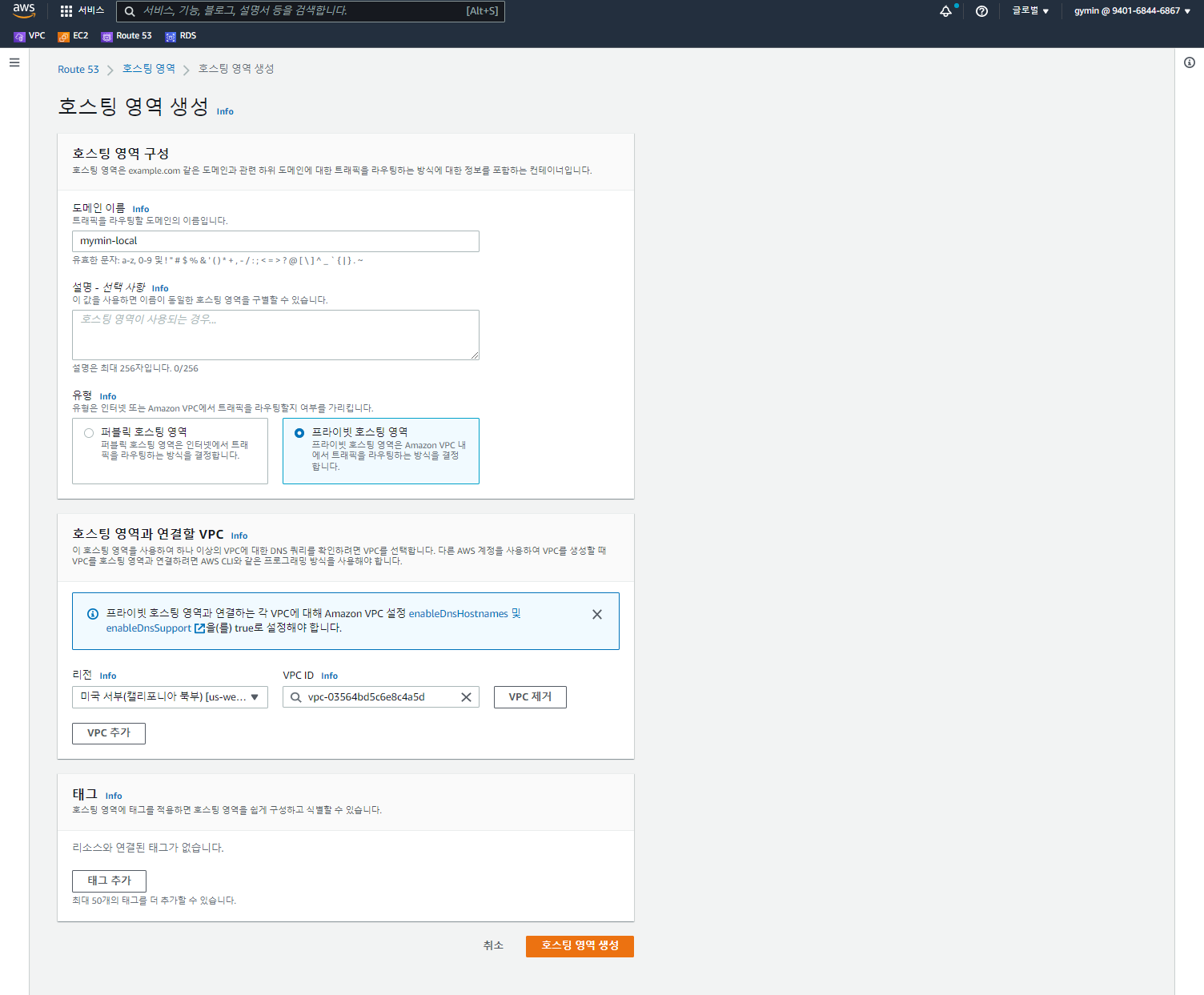
# 부록

## Private DNS 사용하기

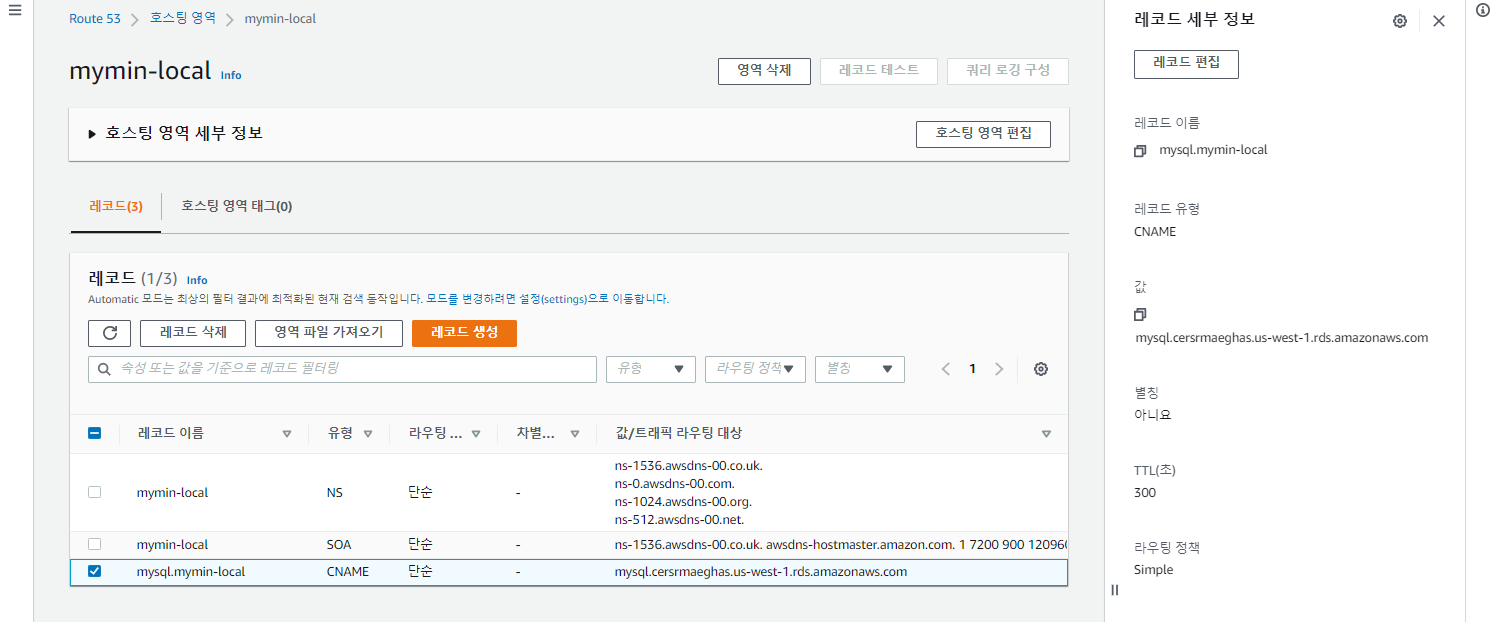
Redis conf 에서 ip나 유동적인 주소를 사용하는 것은 바람직하지 않다

Aws route53에서 VPC내에서만 사용할 수 있는 도메인을 설정할 수 있다

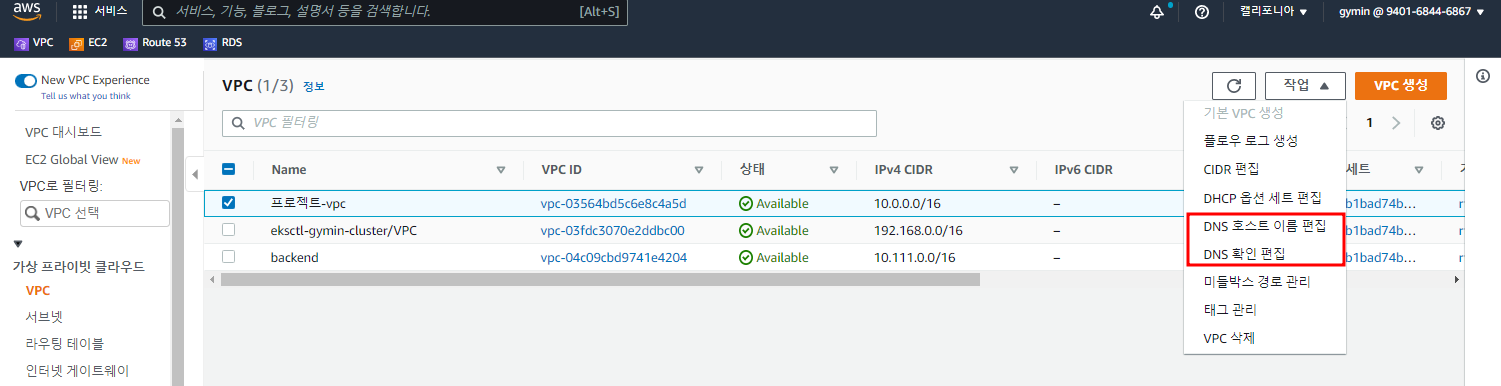
1. Route53 메뉴 > 생성



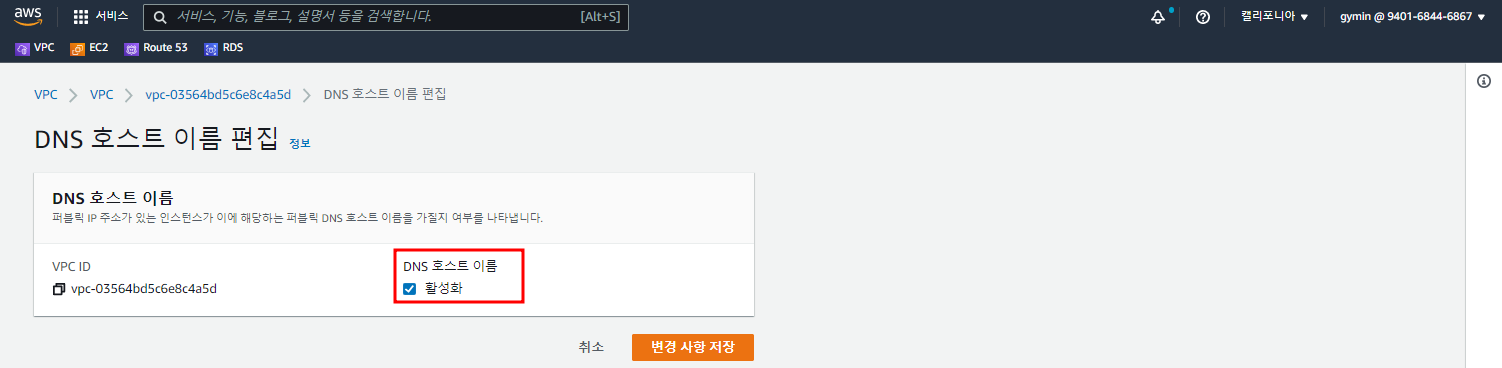
생성한 호스팅 영역에서 레코드를 생성한다

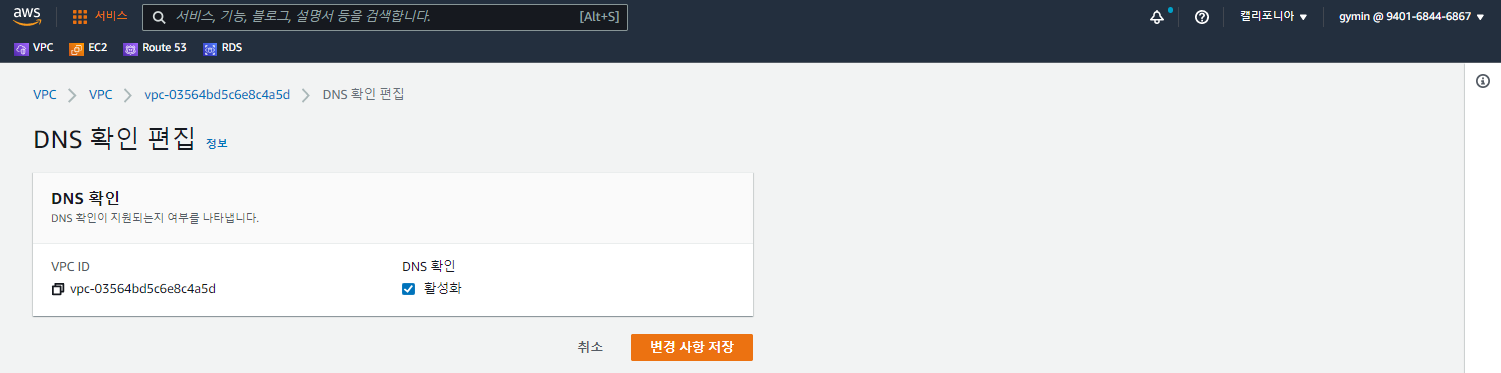


2. VPC 작업 편집



두 메뉴에 들어가 DNS 호스트 이름을 활성화 한다





3. 접속 확인

