제목

문서 기록

|  |  |  |
| --- | --- | --- |
| 작성자 | 날짜 | 설명 |
| 임승하 | 2022-02-07 | Ver1.0 : 처음 작성 |
| 임승하 | 2022-02-08 | Ver2.0: 추가 및 수정 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

AWS

[VPC만들기 6](#_Toc95397897)

[설정 7](#_Toc95397898)

[Subnet 생성 9](#_Toc95397899)

[Web Subnet 생성 10](#_Toc95397900)

[API(WAS) Subnet 생성 11](#_Toc95397901)

[DB1 subnet 생성 12](#_Toc95397902)

[DB2 subnet 생성 13](#_Toc95397903)

[인터넷 게이트웨이 생성 14](#_Toc95397904)

[VPC와 연결 15](#_Toc95397905)

[라우팅 편집 16](#_Toc95397906)

[A. 서브넷 연결 17](#_Toc95397907)

[NACL 생성 18](#_Toc95397908)

[DB Nacl 생성 19](#_Toc95397909)

[서브넷 연결 20](#_Toc95397910)

[인바운드 / 아웃바운드 규칙 변경 21](#_Toc95397911)

[Web NACL 인 바운드 규칙 21](#_Toc95397912)

[API 인바운드 / 아웃바운드 규칙 22](#_Toc95397913)

[DB 인바운드 / 아웃바운드 규칙 22](#_Toc95397914)

[보안그룹 생성 23](#_Toc95397915)

[DB 보안그룹 생성 24](#_Toc95397916)

[Web 보안그룹 25](#_Toc95397917)

[EC2 인스턴스 생성 26](#_Toc95397918)

[프리티어 선택 26](#_Toc95397919)

[프리티어 인스턴스 선택 26](#_Toc95397920)

[인스턴스 세부 정보 구성 27](#_Toc95397921)

[스토리지 추가 27](#_Toc95397922)

[테그 추가 27](#_Toc95397923)

[보안그룹 선택 28](#_Toc95397924)

[생성완료 28](#_Toc95397925)

[키페어 생성 29](#_Toc95397926)

[탄력적 IP주소 할당 30](#_Toc95397927)

[탄력적 IP주소 연결 31](#_Toc95397928)

[IP주소와 EC2인스턴스 연결 33](#_Toc95397929)

[퍼블릭 IP 확인 34](#_Toc95397930)

[EC2 인스턴스 연결 35](#_Toc95397931)

[cmd로접속하기 36](#_Toc95397932)

[Putty로 연결 37](#_Toc95397933)

[실행 37](#_Toc95397934)

[등록 37](#_Toc95397935)

[Save private key 38](#_Toc95397936)

[putty실행 39](#_Toc95397937)

[Login as값 입력 40](#_Toc95397938)

[완성 40](#_Toc95397939)

[nginx설치완료 41](#_Toc95397940)

[RDS(서브 넷 그룹 생성) 42](#_Toc95397941)

[정보 입력 43](#_Toc95397942)

[생성 완료 44](#_Toc95397943)

[RDS(DB생성) 45](#_Toc95397944)

[엔진옵션 45](#_Toc95397945)

[버전/템플릿 설정 46](#_Toc95397946)

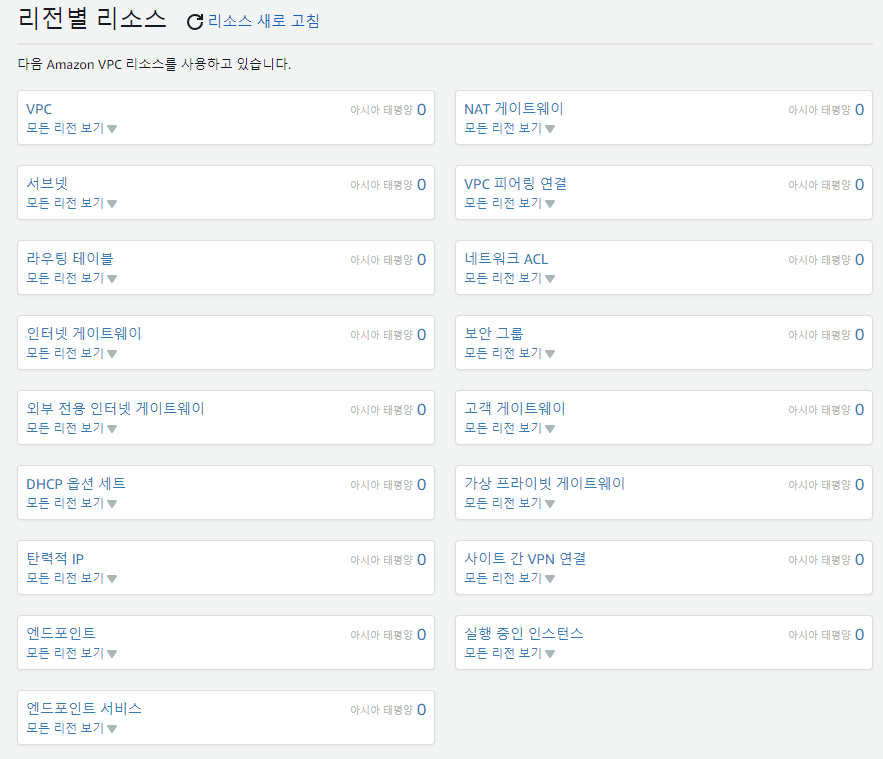
[설정 46](#_Toc95397947)

[연결 48](#_Toc95397948)

[추가 구성 49](#_Toc95397949)

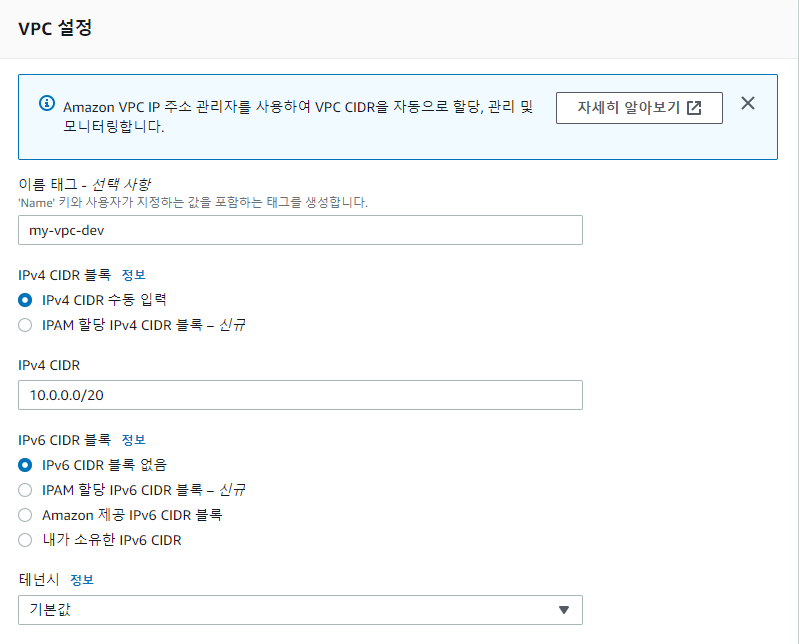
[생성 완료 51](#_Toc95397950)

# VPC만들기

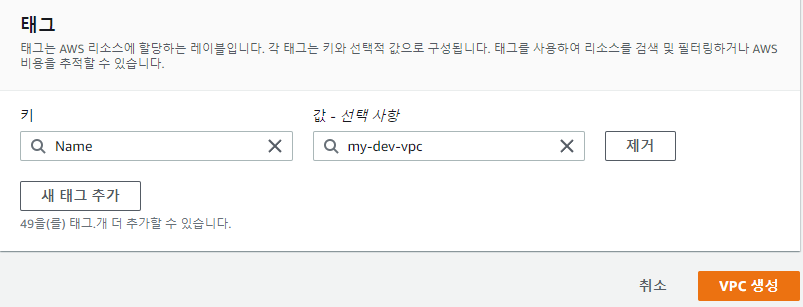


모든 리소스가 0임을 확인한다.

## 설정



사용 목적에 맞게 이름을 작성 후 사용하고자 하는 IP주소의 수를 파악 후 CIDR계산기를 이용하여 IPv4 CIDR 값을 넣어준다.

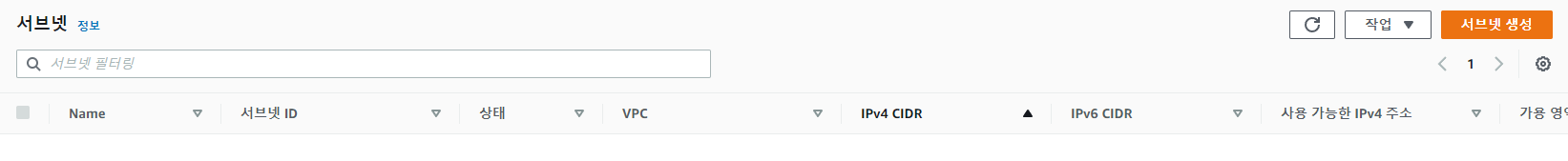


이후 태그 Name을 확인 후 VPC 생성 버튼을 눌러 VPC를 생성한다.

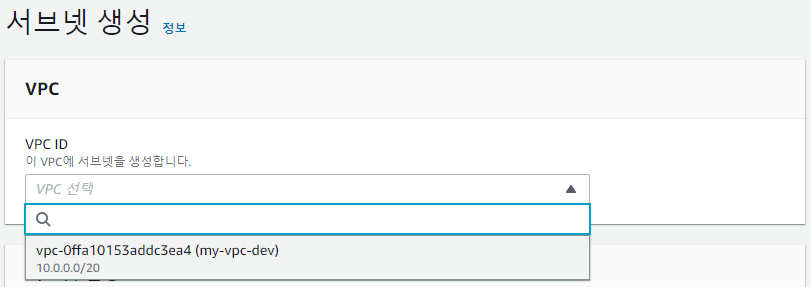


그럼 다음과 같이 VPC가 생성된 것을 확인할 수 있다.

# Subnet 생성



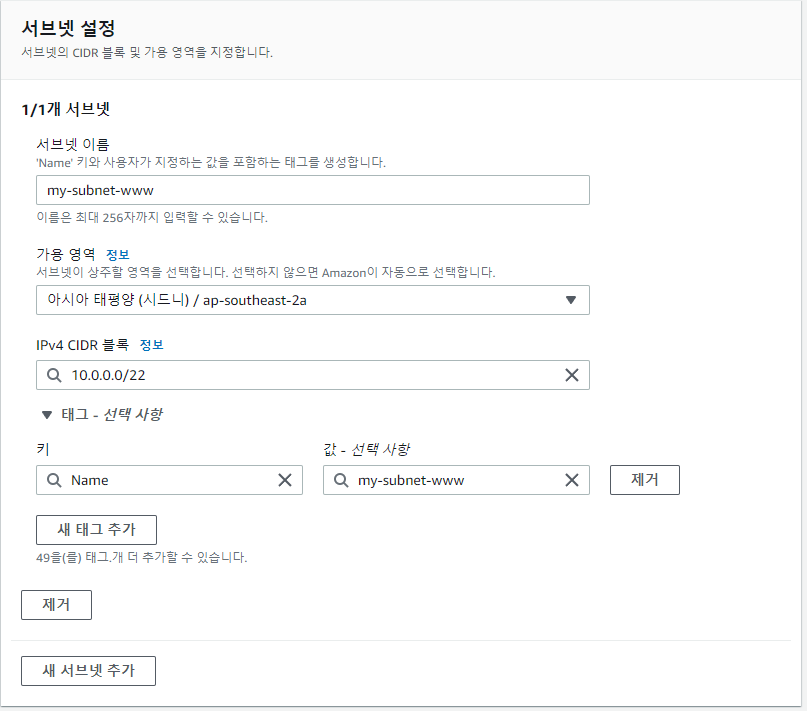
서브 넷 생성 버튼을 클릭한다.



서브넷을 설치하고 싶은 VPC를 선택한다.

선택하면 아래와 같은 화면으로 변경된다.

## Web Subnet 생성



다음과 같은 설정에서

이름/가용영역/CIDR값/태그 값을 입력한다.

사설 아이피 주소 중에 서버의 경우 10.0.0.0부터 시작되며 처음 VPC를 만들 때

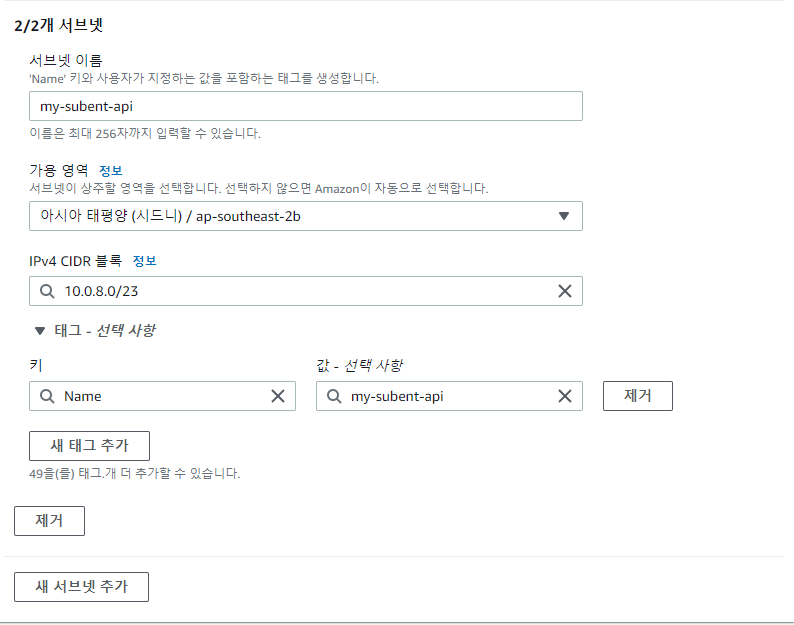
10.0.0.0/20(여유분 4000개) 중 Web 서버의 경우 1000개의 ip주소가 필요한 것으로 생각하고 할당한다.

범위 10.0.0.0 - 10.0.3.255의 경우 CIDR 표기법으로 10.0.0.0/22로 나타낼 수 있다



이 버튼을 눌러 나머지 API/DB 서브넷을 생성하자.

## API(WAS) Subnet 생성



마찬가지로 이름/ 가용역역 / CIDR 값을 입력해준다.

앞의 Web 서버의 경우 10.0.0.0/22 1000개를 할당했지만 2000개까지의 여유분을 줘서

api서버의 경우 10.0.8.0 - 10.0.9.255 범위에서 예상 500개의 ip주소를 사용하므로 CIDR계산기를 통해 계산해준다,

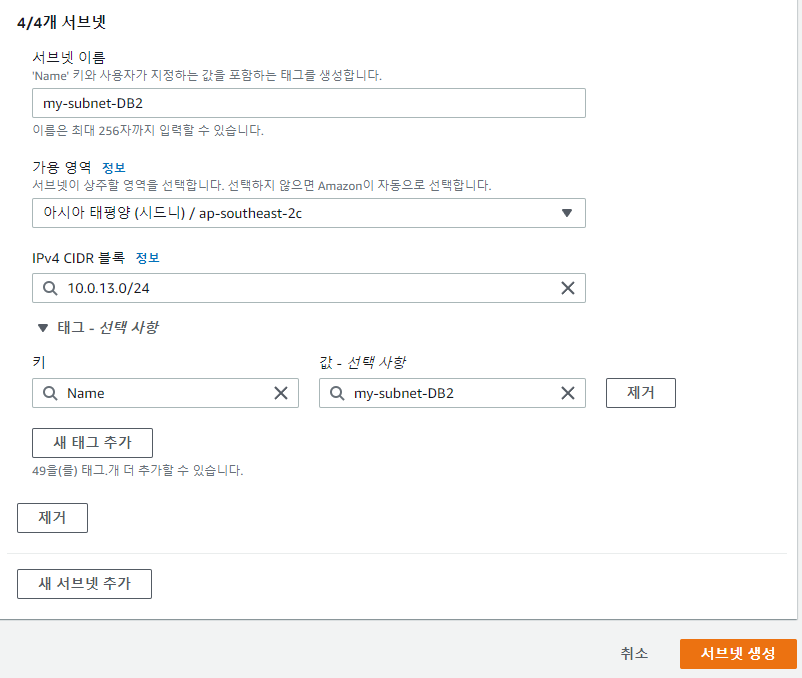
## DB1 subnet 생성



데이터 베이스의 경우 IP주소 사용은 별로 하지 않지만 여유분이 많으므로 많이 설정해 준다.

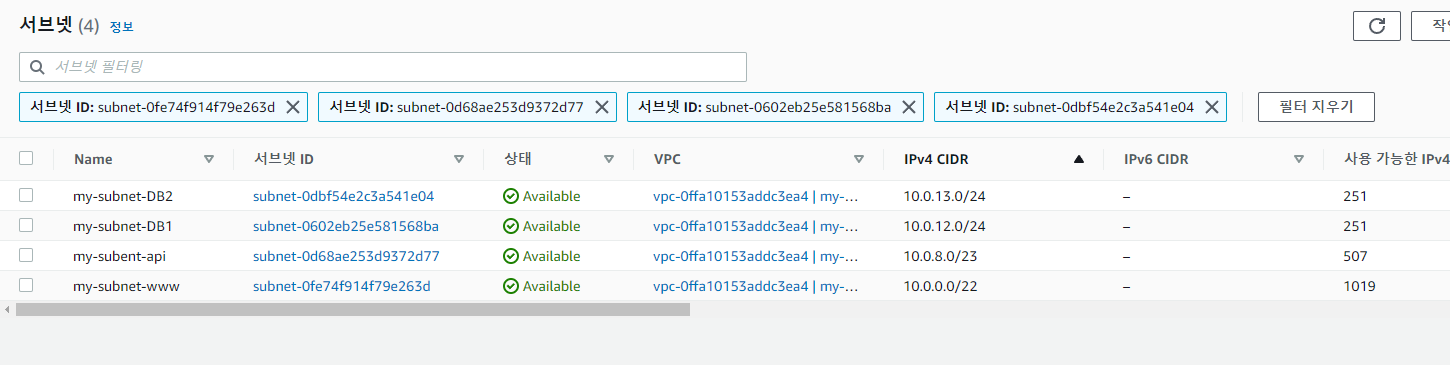
이때 DB1과 DB2의 가용영역이 겹치지 않도록 주의하자

## DB2 subnet 생성



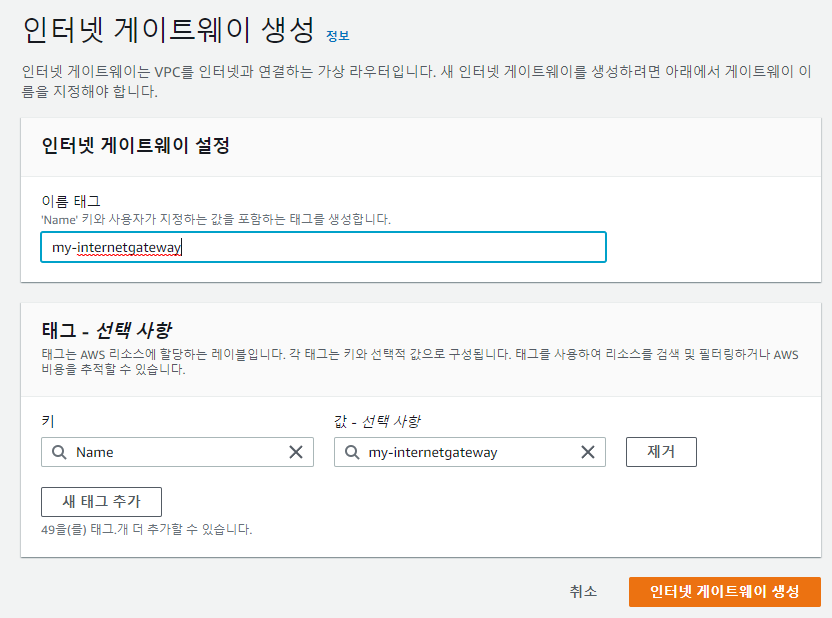
마찬가지로

서브넷 생성 완료



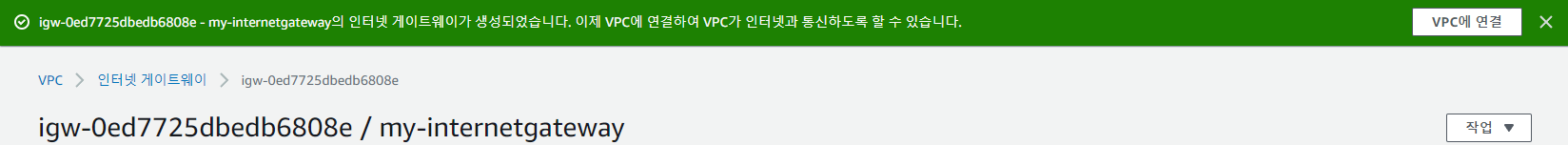
서브넷이 정상적으로 만들어 질 경우 다음과 같은 화면이 나오며, CIDR계산을 잘못 할 경우 서브넷이 생성되지 않는 점을 주의하자.

# 인터넷 게이트웨이 생성

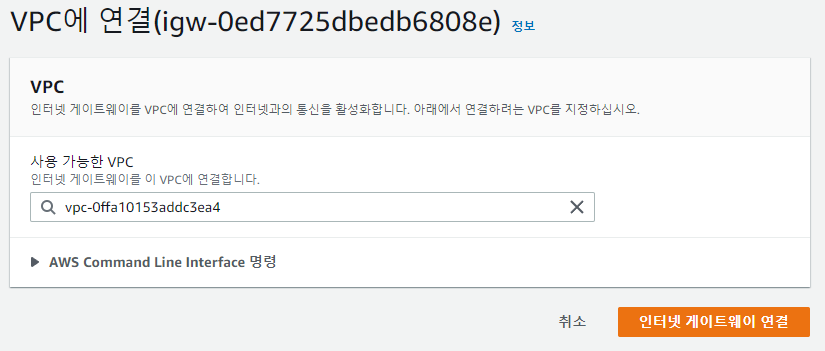


* 이름을 정해주고 인터넷 게이트웨이 생성 버튼을 누른다.

## VPC와 연결

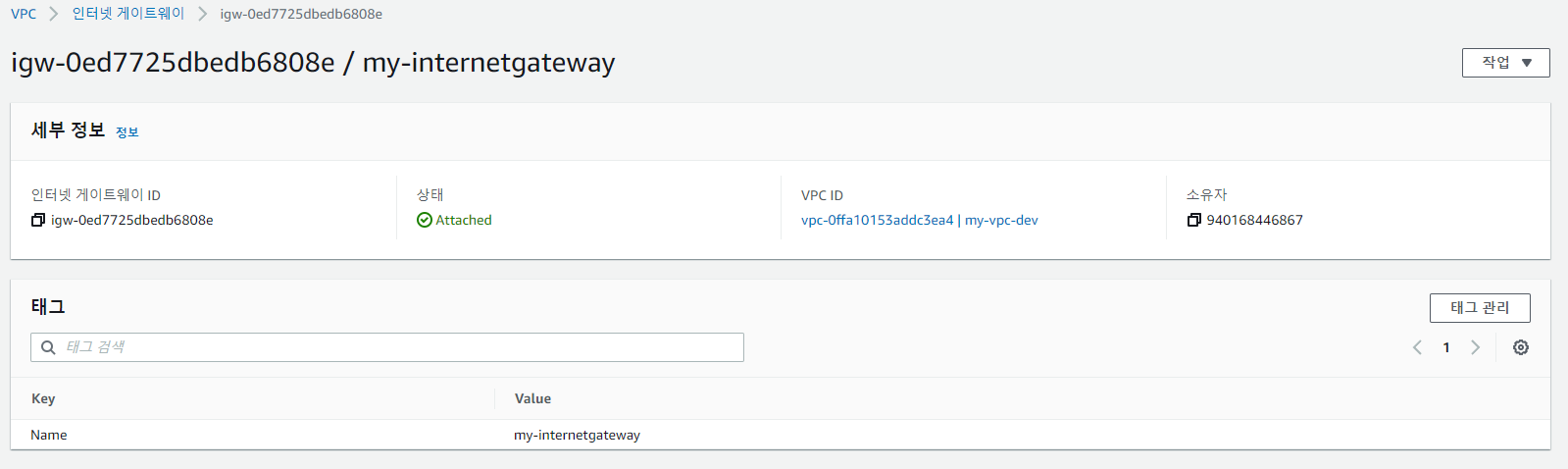


* 생성 시 VPC연결 이라는 문구가 뜨는데 이때 연결을 클릭해준다.

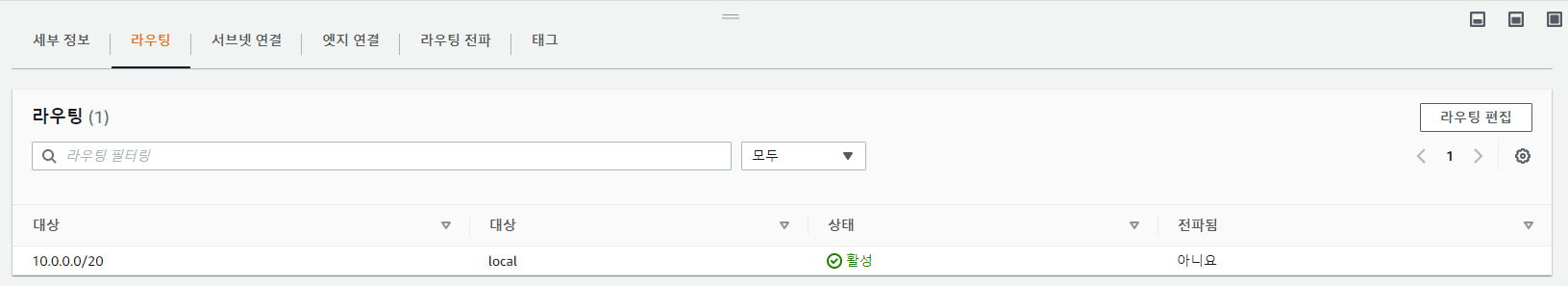


연결하고자 하는 VPC에 생성한 Gateway를 연결하자

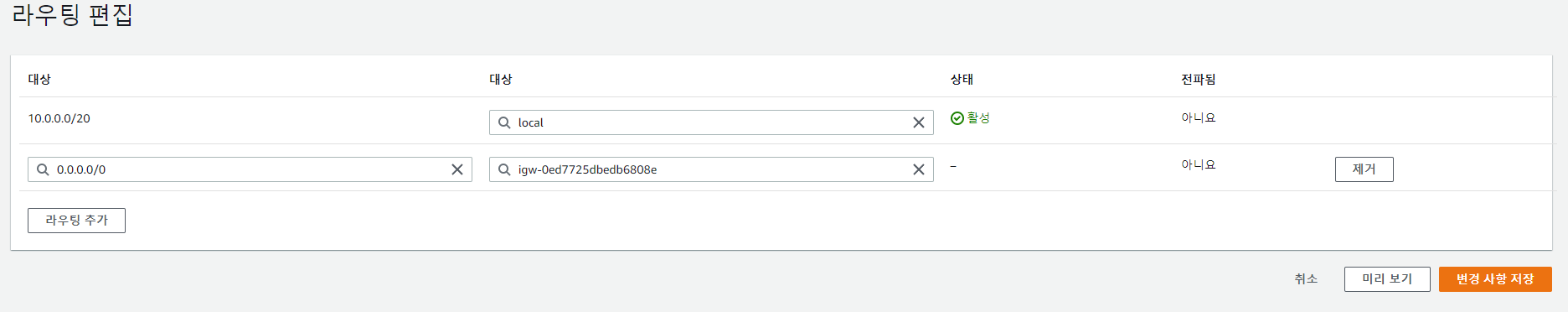
인터넷 게이트웨이 생성 완료



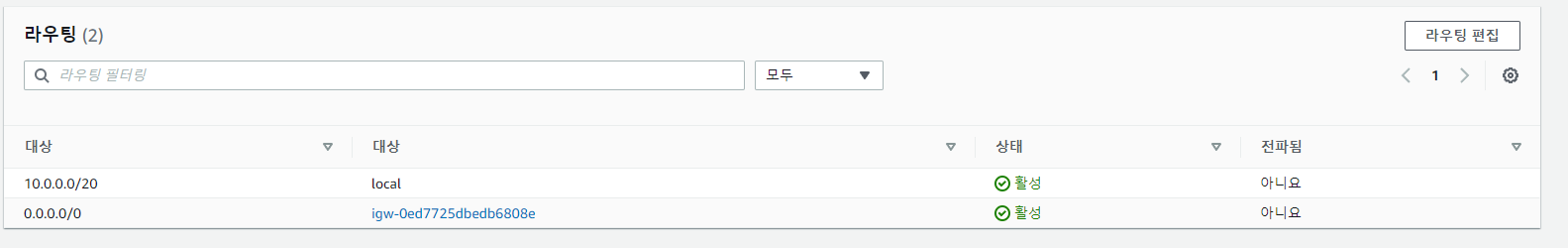
## 라우팅 편집



라우팅 편집을 선택한다.

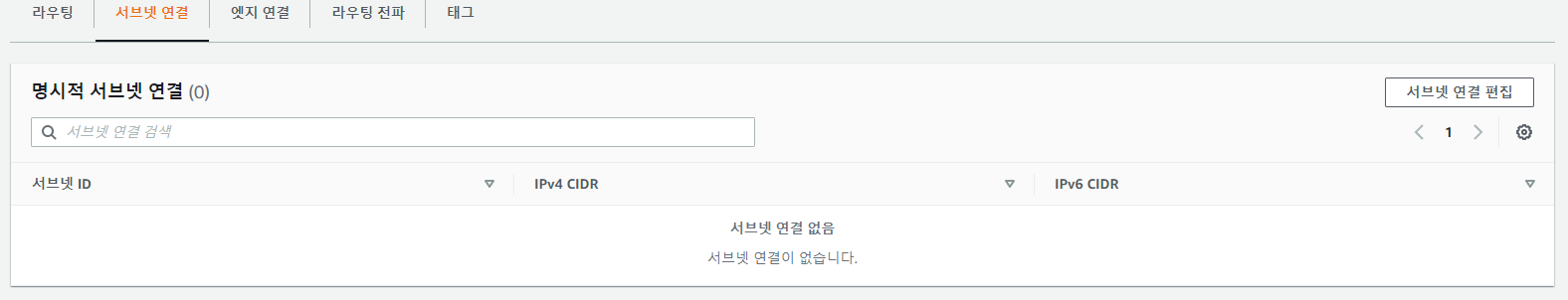


0.0.0.0/0 은 인터넷이며 편집창에서 추가해준다.



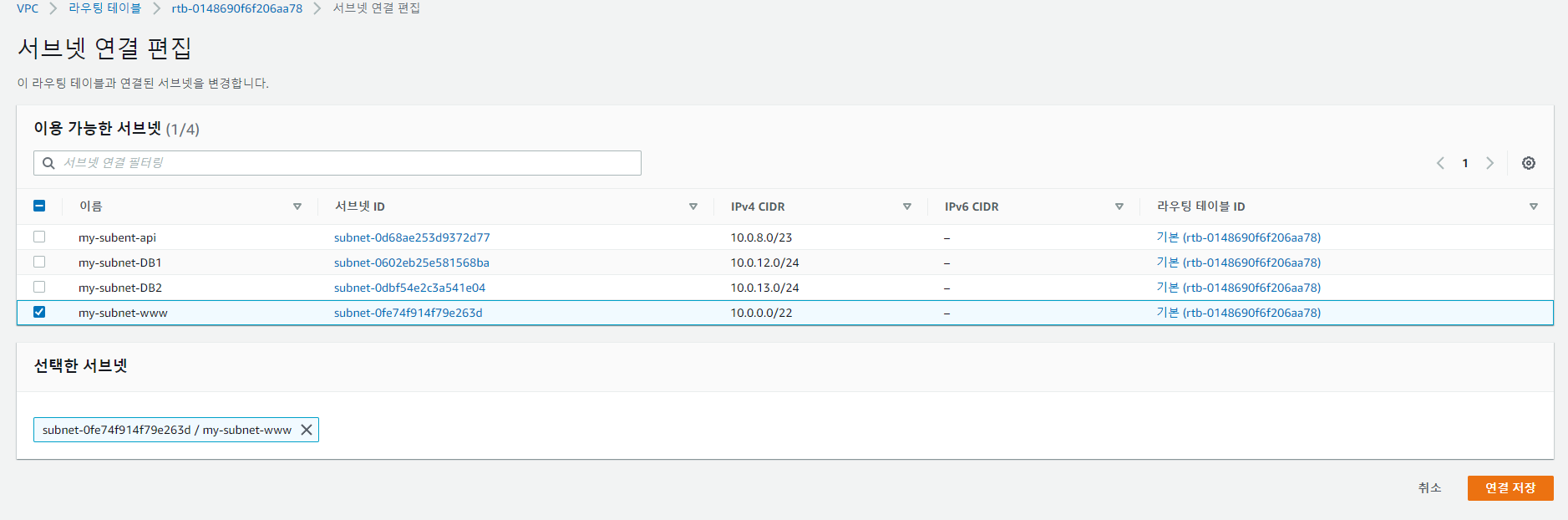
추가하면 다음과 같은 창이 나온다.

## 서브넷 연결



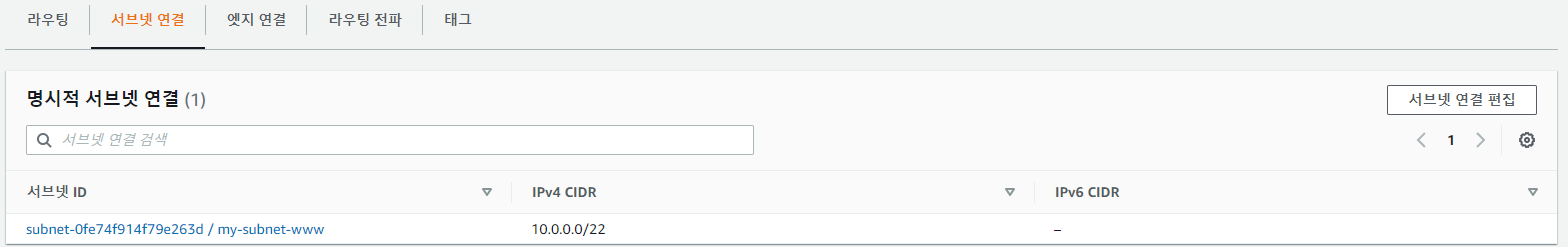
서브넷 연결 편집을 클릭한다.

(Private Subnet -> Public Subnet으로 변경하는 과정)

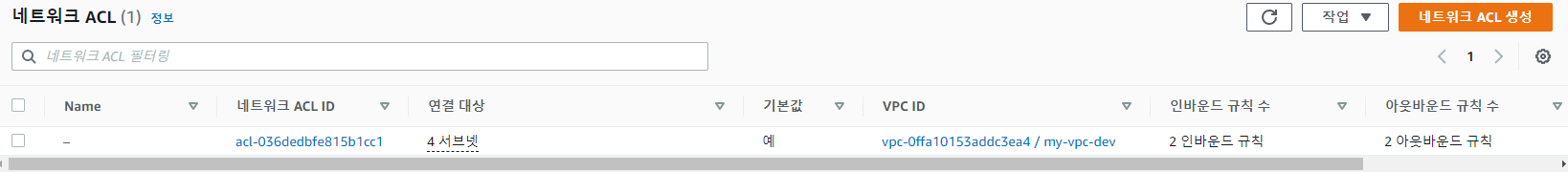


WWW(Web) Subnet- 과 라우팅을 연결한다.

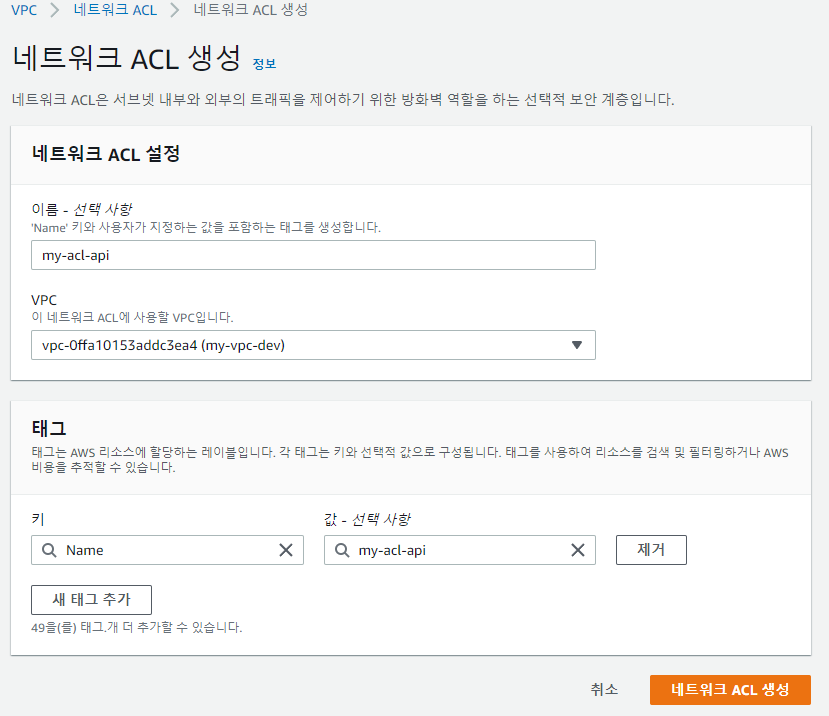
연결 확인



# NACL 생성

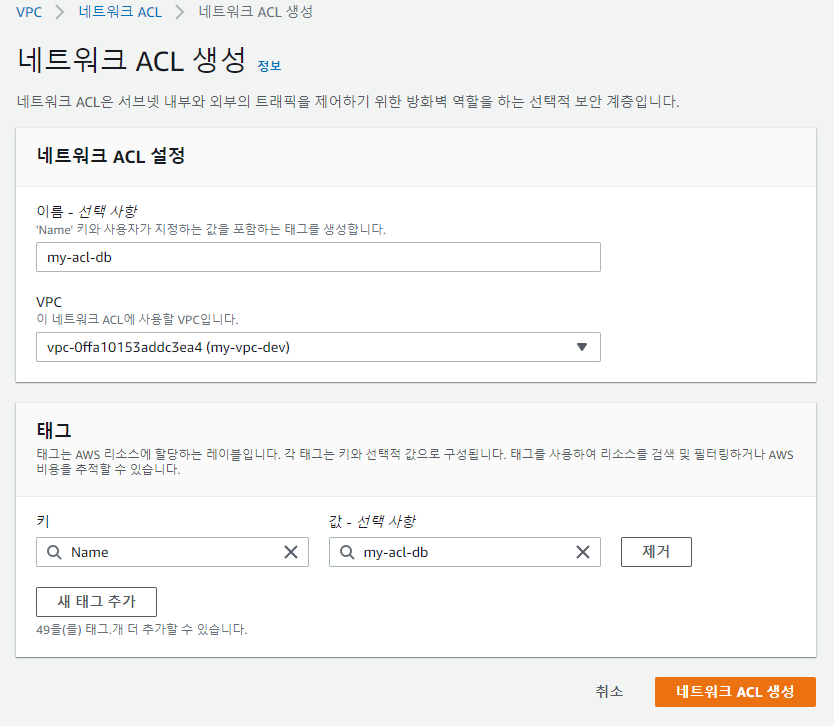
 처음 생성하지 않아도 기본값으로 하나 만들어져 있다. 이 NACL은 Web과 연결하자.

API(Was) Nacl 생성



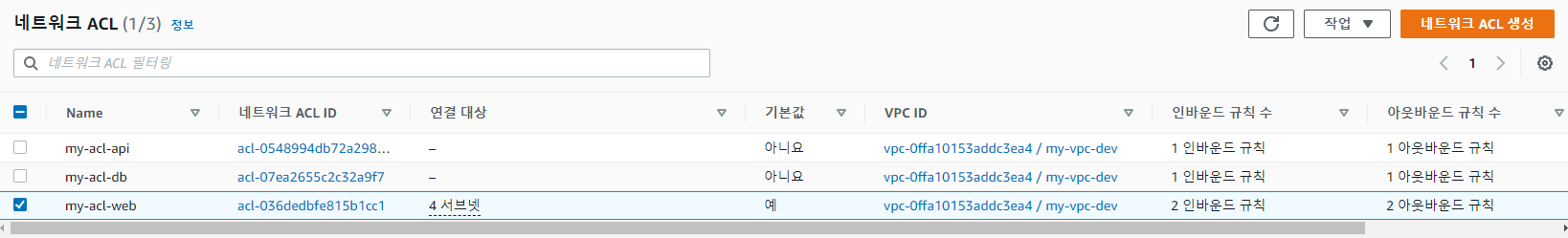
이름 / 연결 VPC를 선택하고 생성한다.

## DB Nacl 생성

­­­­

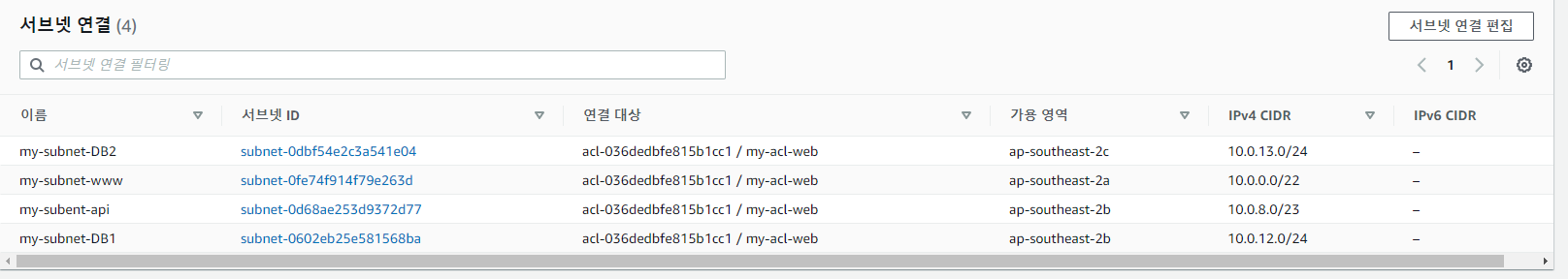
마찬가지로 이름과 VPC를 선택 후 생성한다.

Web NACL의 경우 기본생성 된 NACL 이용 후 태그 변경



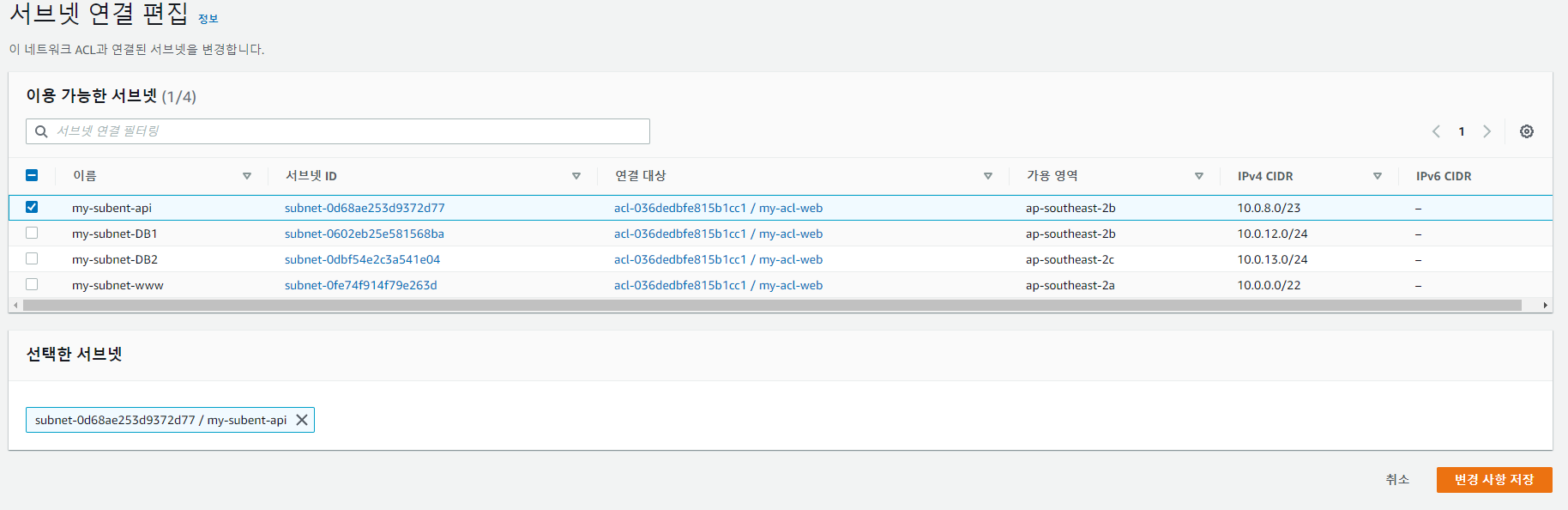
구별하기 쉽게 태그명을 변경한다

## 서브넷 연결



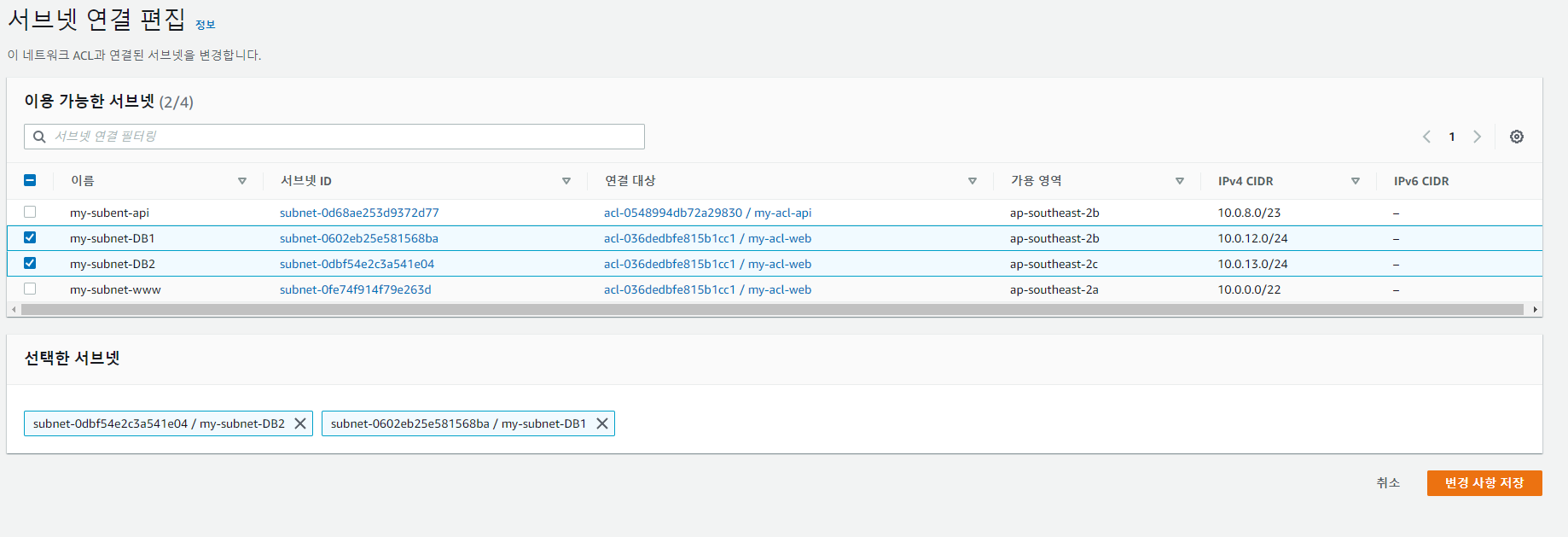
NACL을 API/DB/Web에 연결을 해주어야 한다.

API 서브넷 연결



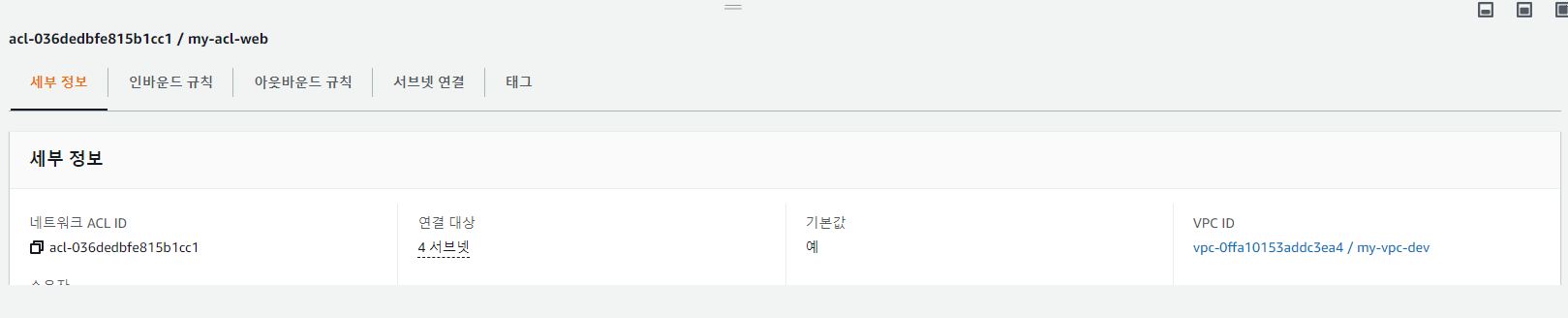
서브넷 연결 편집을 클릭후 API 태그명을 클릭 후 저장한다.

DB 서브넷 연결

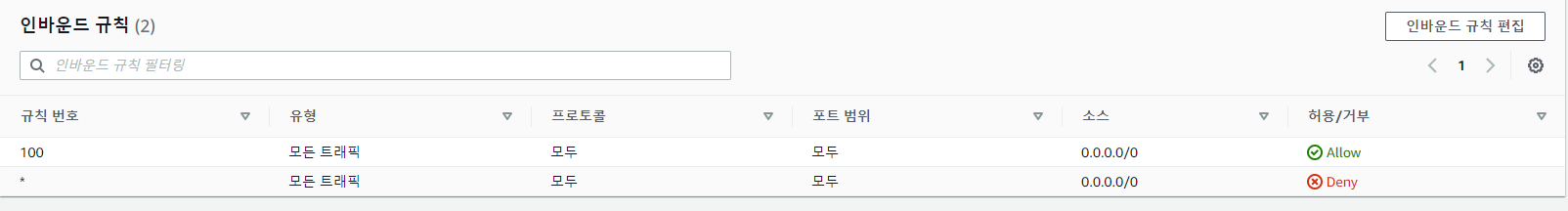


마찬가지로 DB1/DB2를 선택 후 저장한다

## 인바운드 / 아웃바운드 규칙 변경

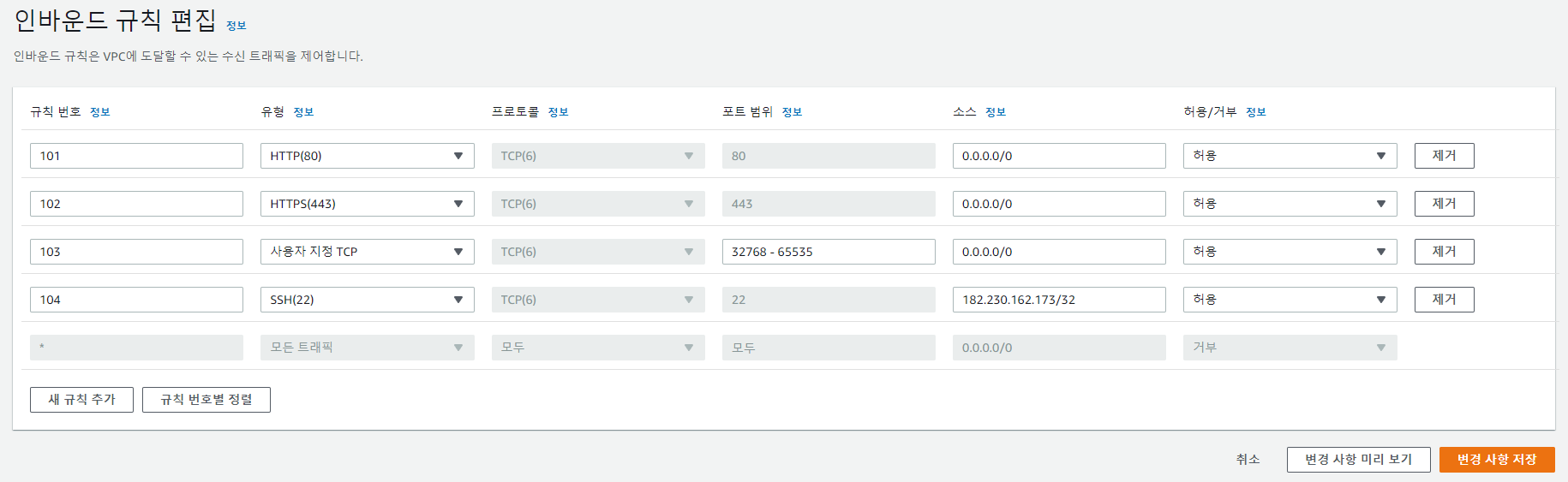


기본값



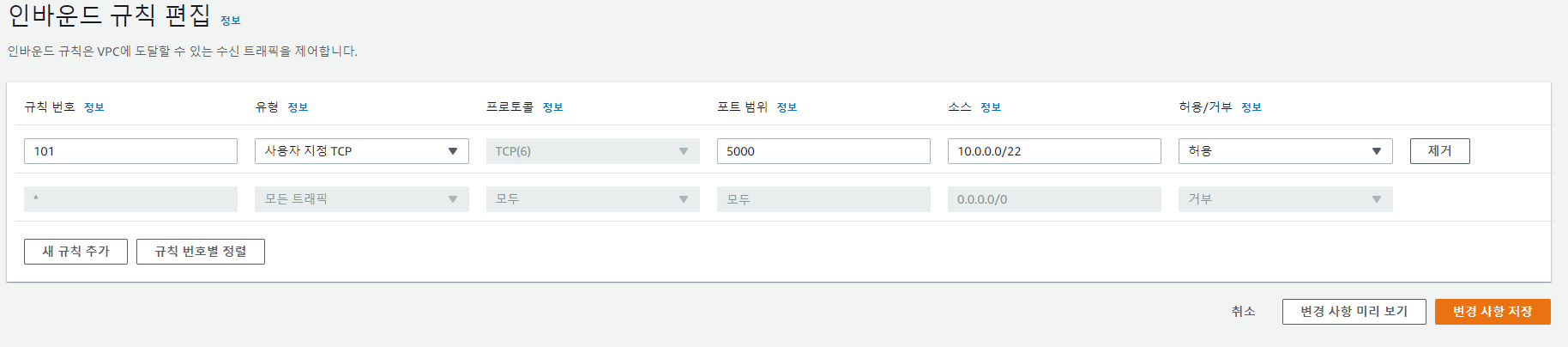
다음과 같이 기본값이 정해져 있을 것이다.

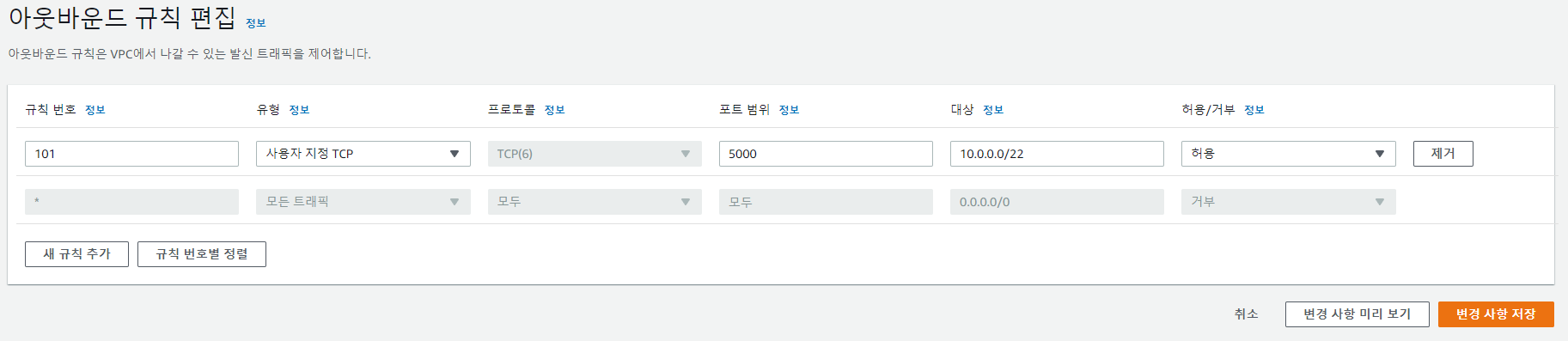
## Web NACL 인 바운드 규칙



Web에 경우 http / https / 동적포트 / ssh 내 IP주소 각각 입력 후 IP 는 0.0.0.0/0 인터넷으로 설정 한다. 내 IP 주소만 허용은 /32를 붙여 입력한다.

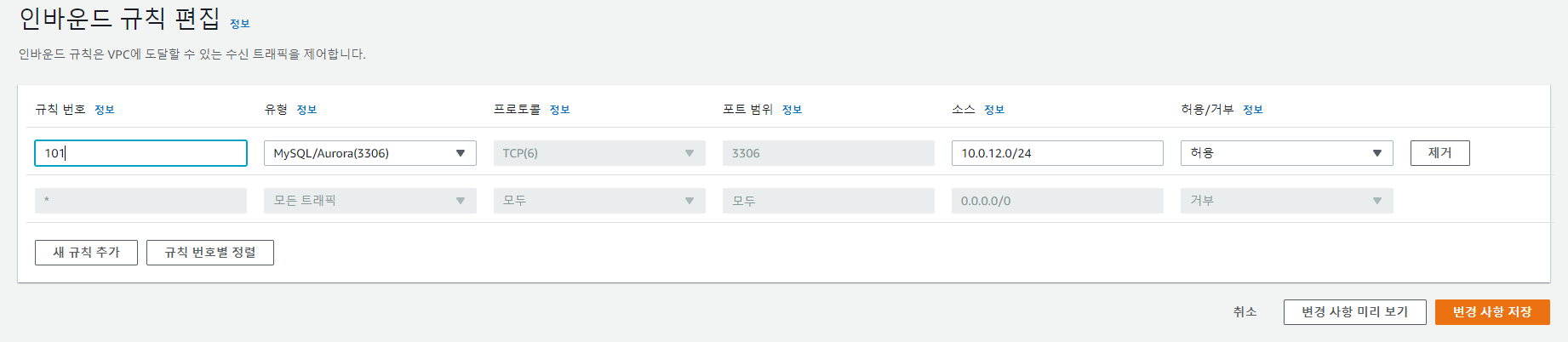
## API 인바운드 / 아웃바운드 규칙

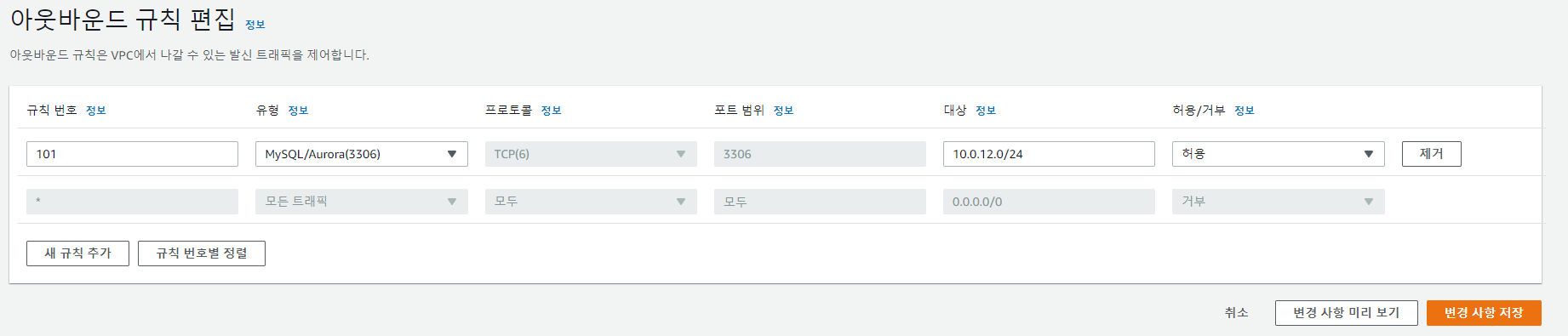




API 의 경우 Flask 를 사용하므로 포트번호 5000번을 입력해주고, Web으로 오는 IP만 허용하기 위해 10.0.0.0/22 를 입력해준다.

## DB 인바운드 / 아웃바운드 규칙

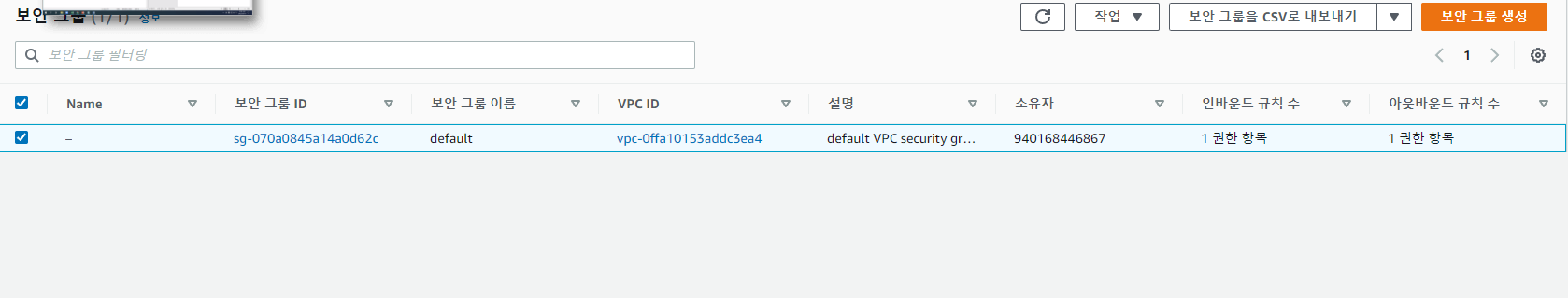




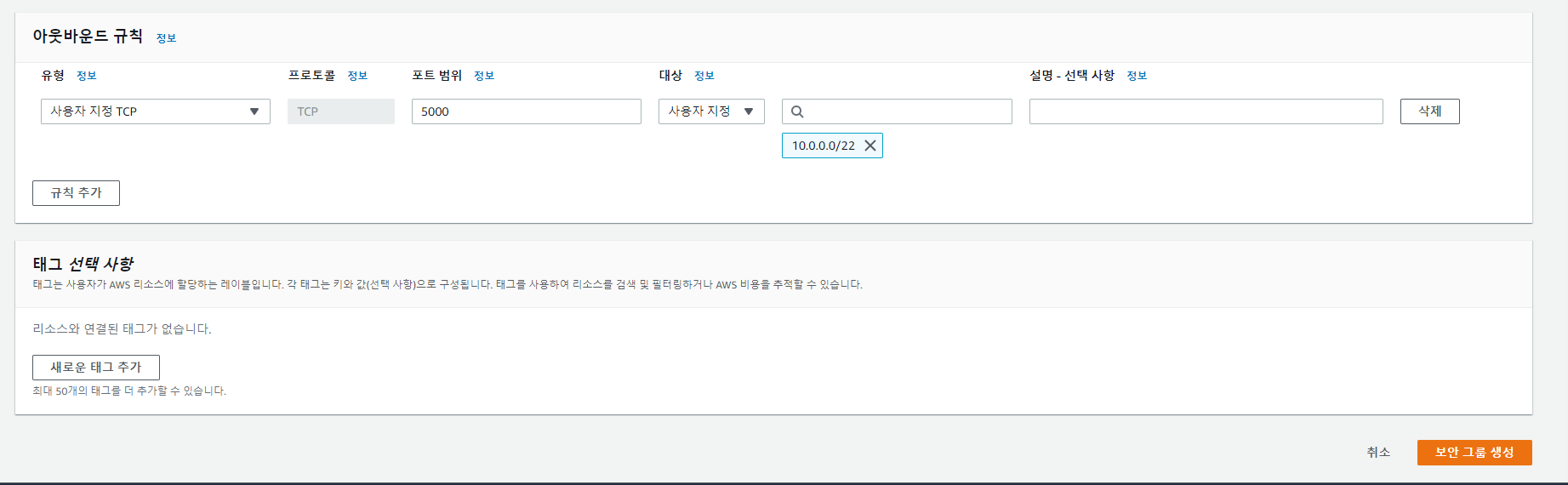
DB의 경우 Maria/MySQL 을 사용하므로 알맞게 선택 후 API로 오는 경로만 허용하기 위해

10.0.12.0/24 를 입력해준다.

# 보안그룹 생성

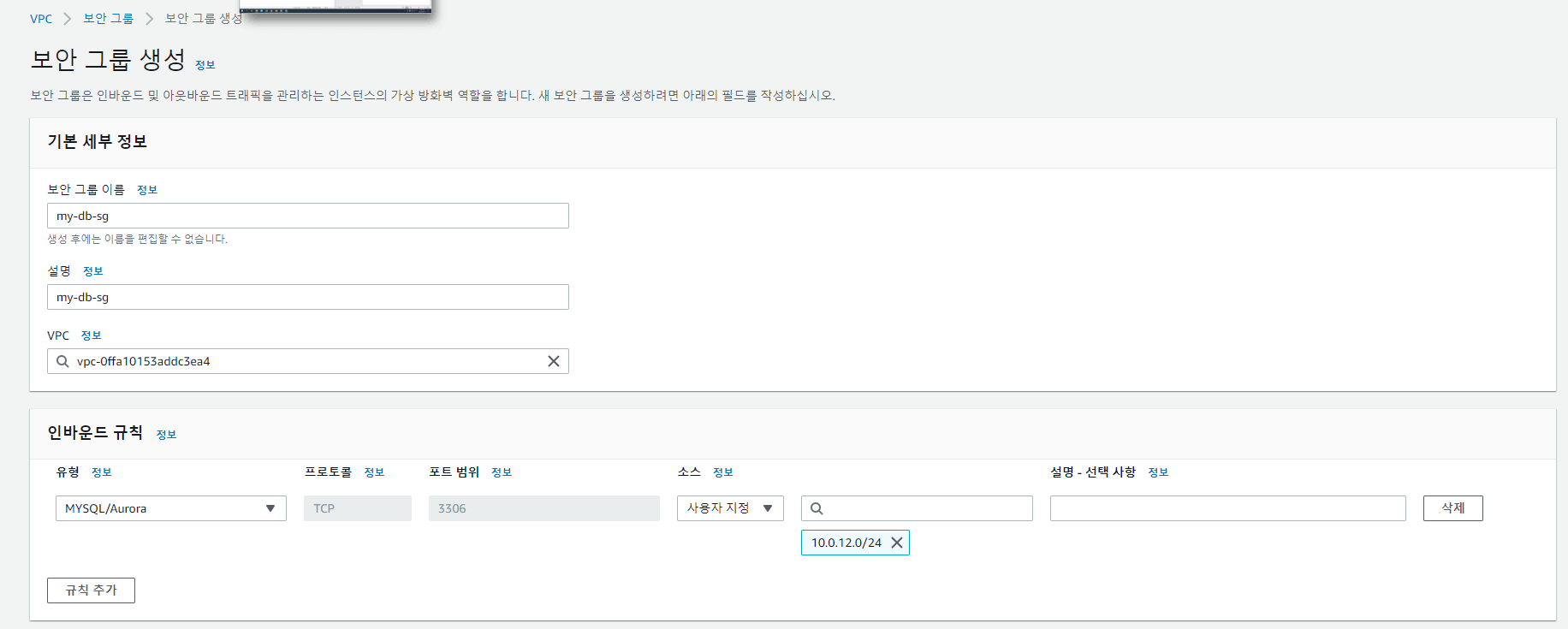


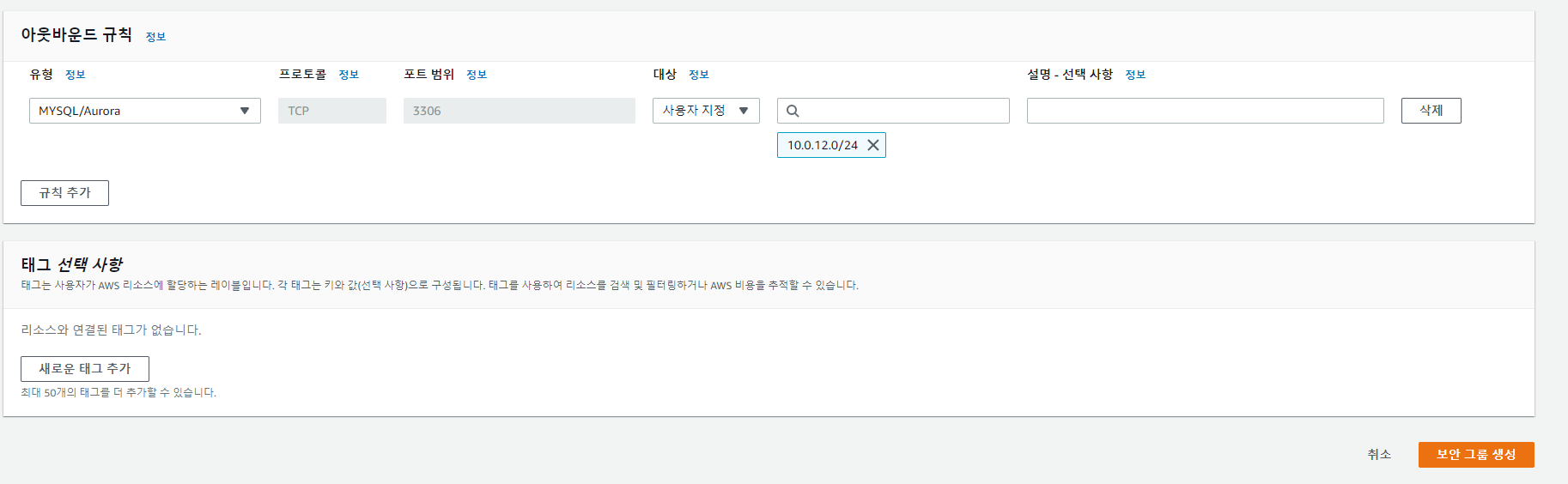




보안그룹 생성 시 인바운드 규칙 내 IP주소 및 동적포트를 설정해 주고, // 나갈 때는 모두 허용으로 열어준다.? (기억이..)

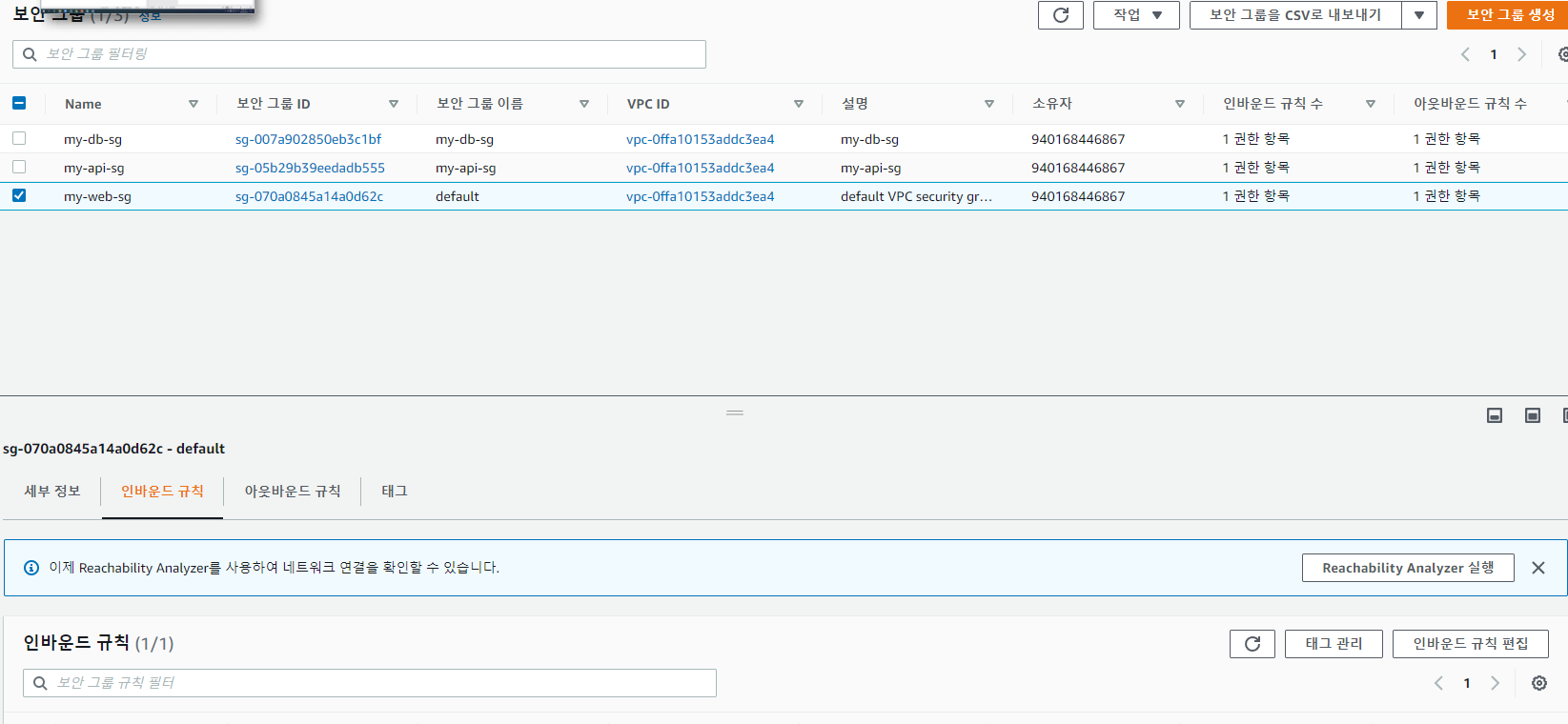
## DB 보안그룹 생성





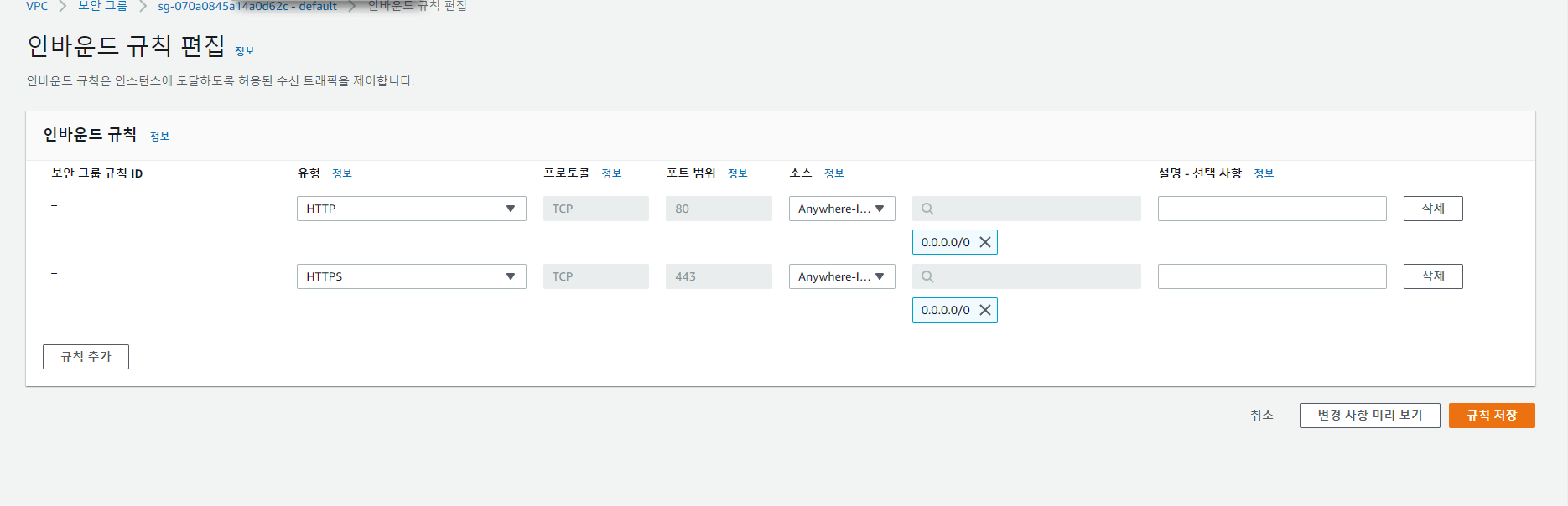
마찬가지로 Mair/MySQL 용도에 맞게 선택 후 API로 오는 경로만 허용하도록 설정 후 저장해준다.

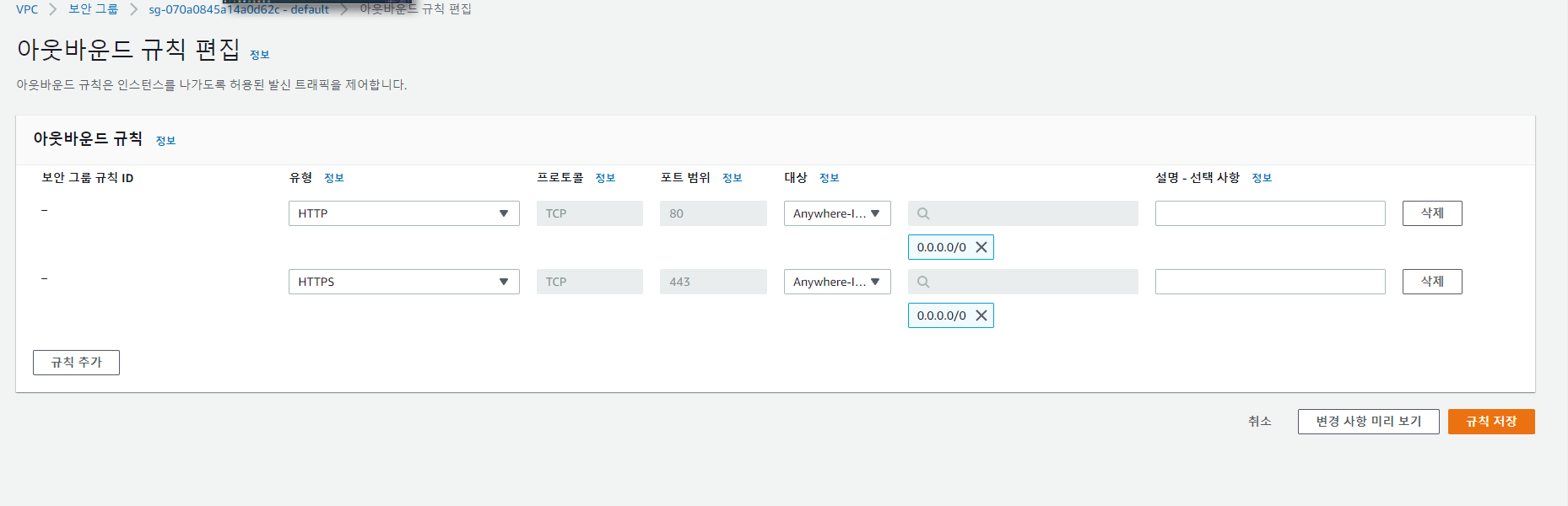
## Web 보안그룹



미리 만들어진 default값을 사용한다.

인바운드 / 아웃바운드 규칙

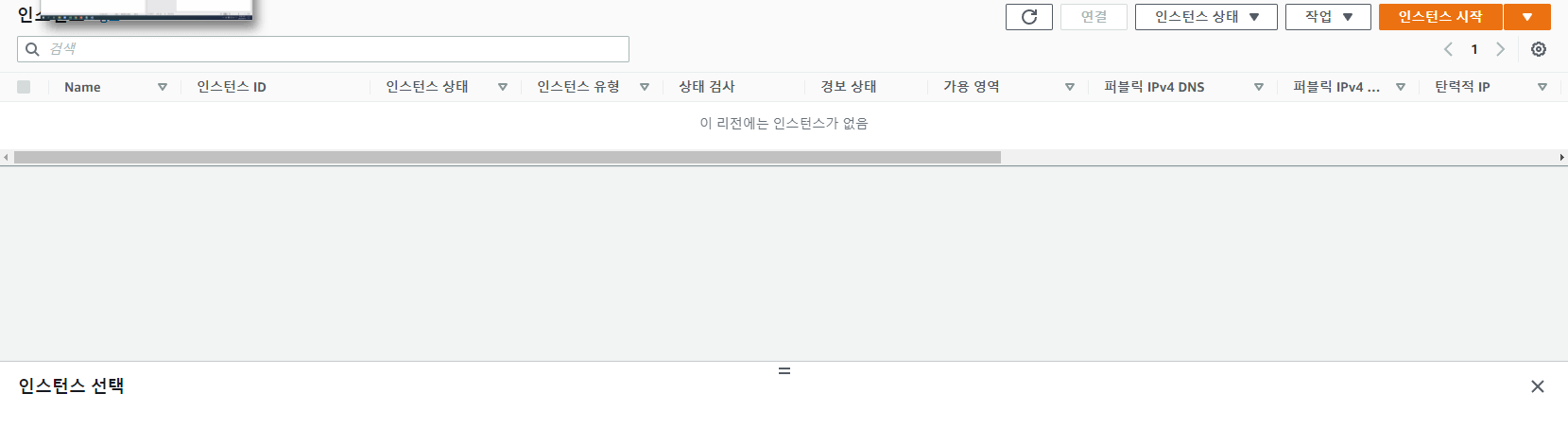




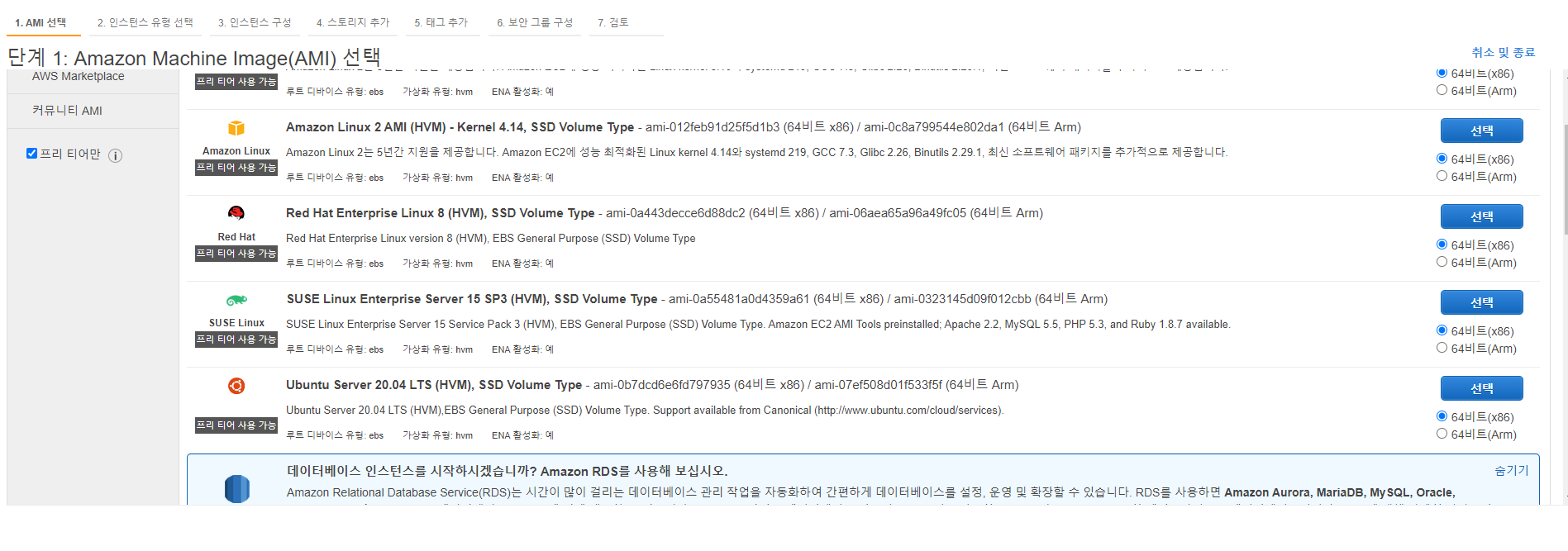
지금 http 및 https를 이용하지는 않고 내 아이피주소 /32 – 및 동적포트를 사용하고

아웃바운드의 경우 모두 열어준다.

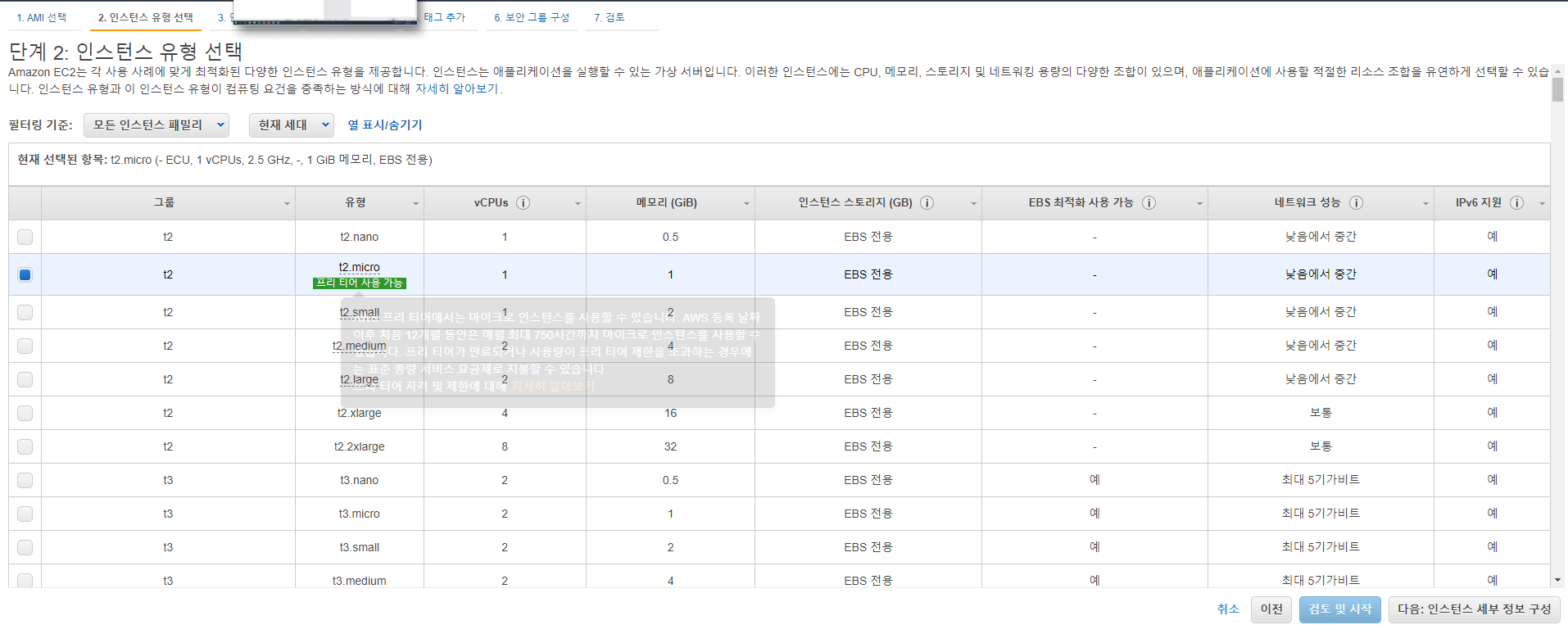
# EC2 인스턴스 생성



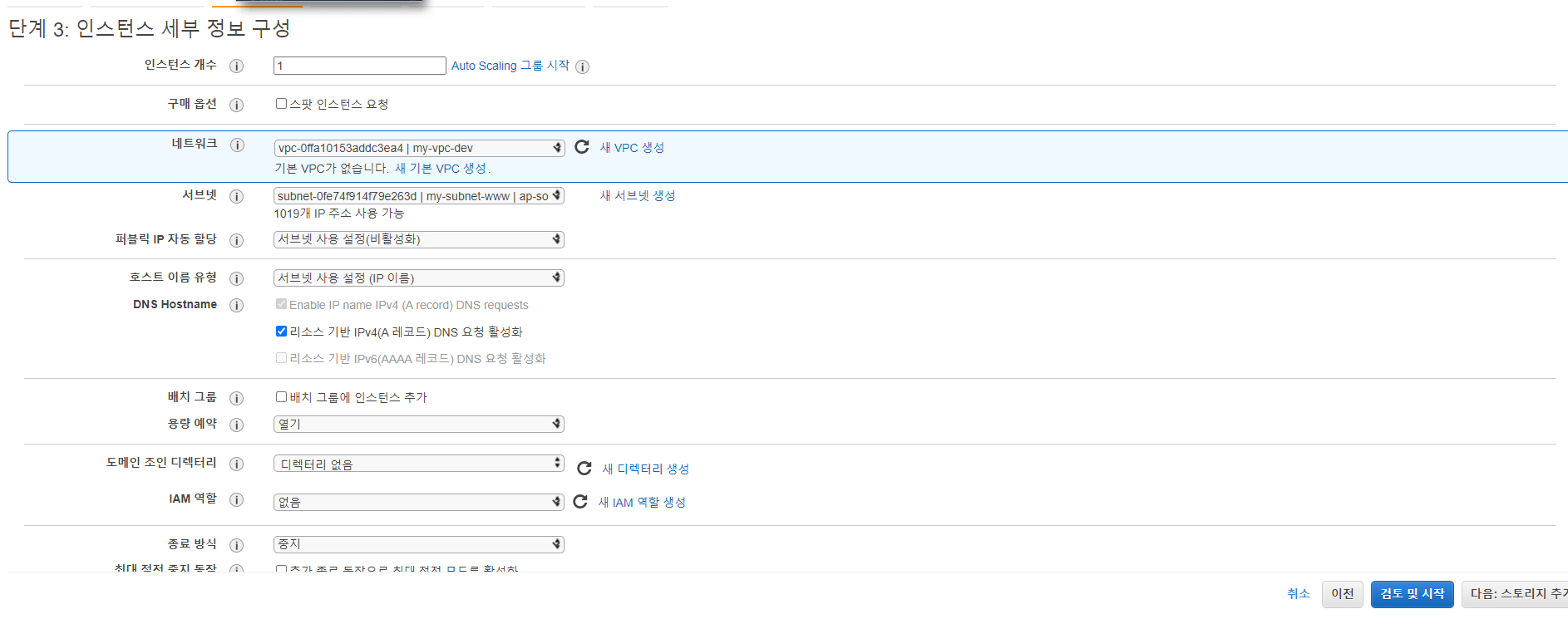
## 프리티어 선택



## 프리티어 인스턴스 선택

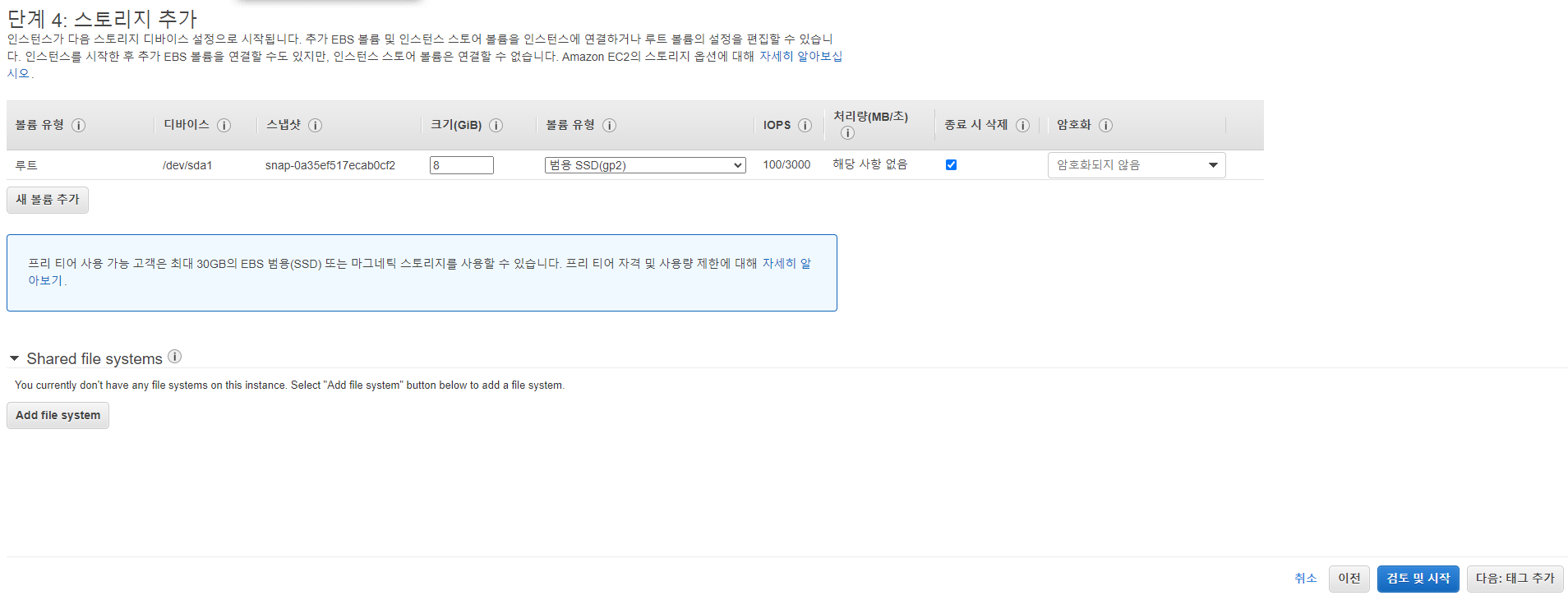


## 인스턴스 세부 정보 구성

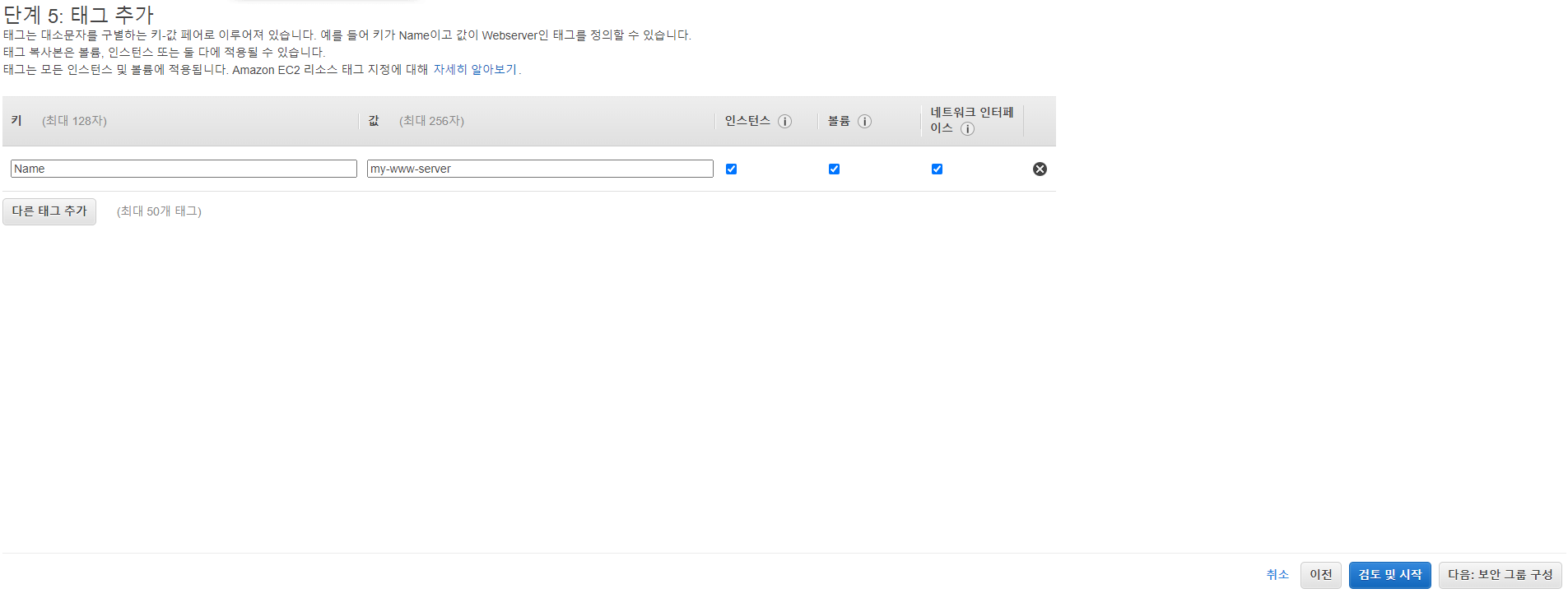


Web으로만 변경하고 나머지는 기본값을 사용한다.

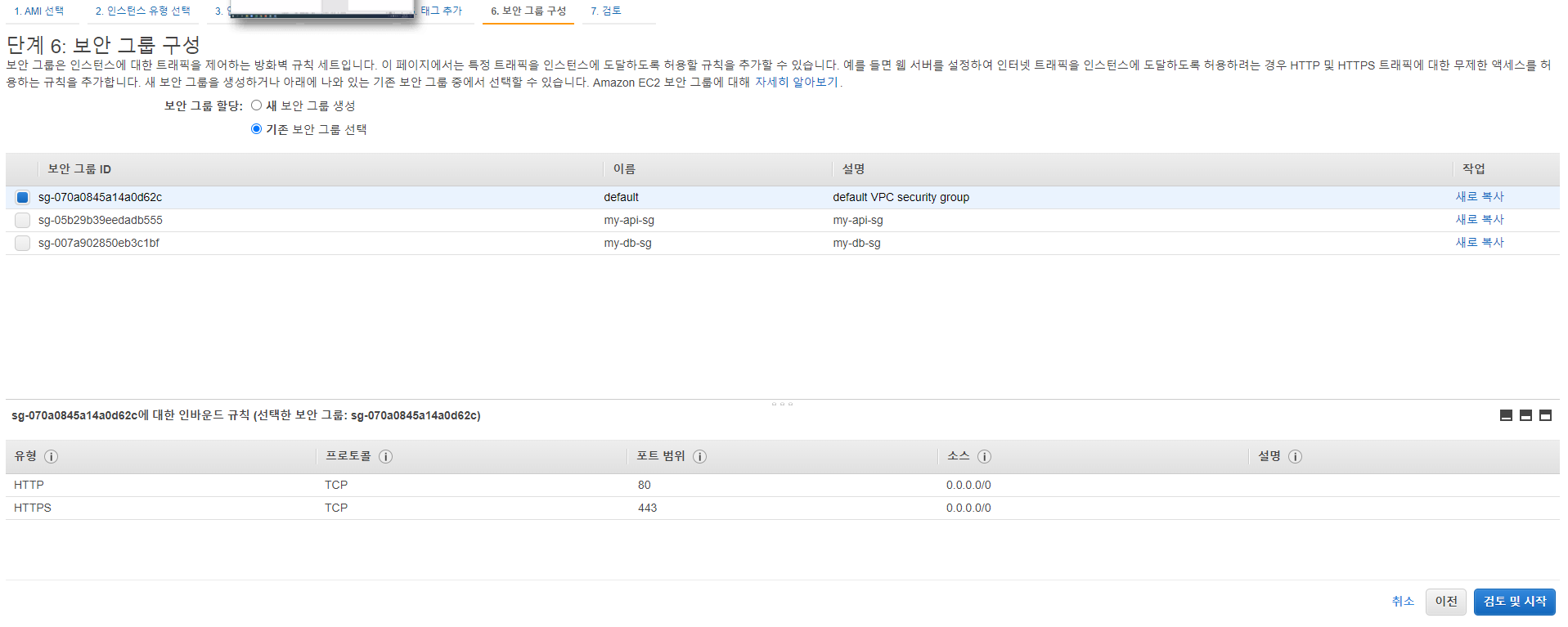
## 스토리지 추가



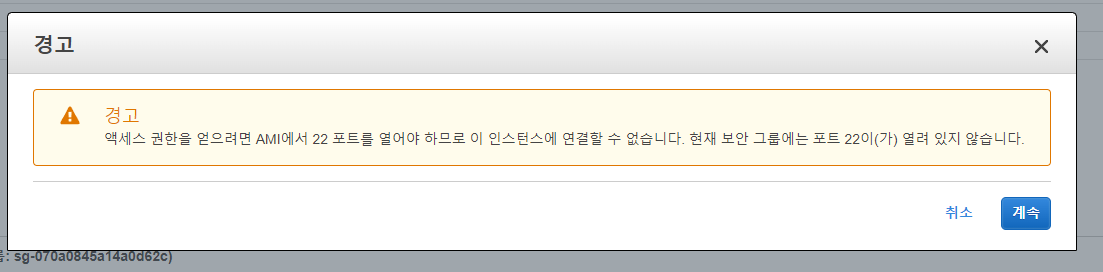
## 테그 추가



## 보안그룹 선택



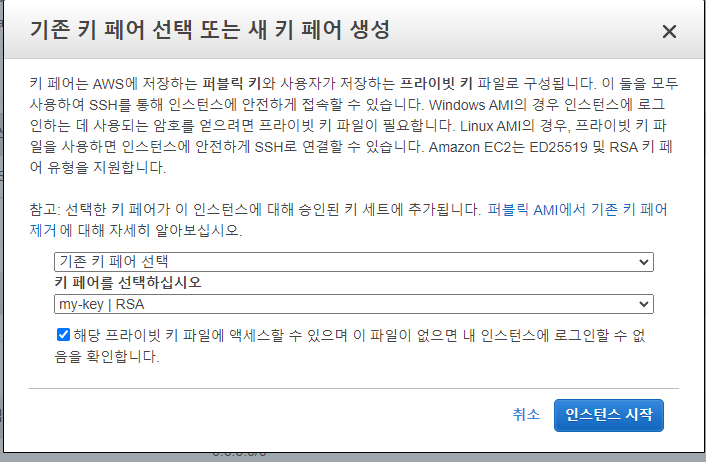
미리 생성한 web보안 그룹을 선택한다.



## 생성완료

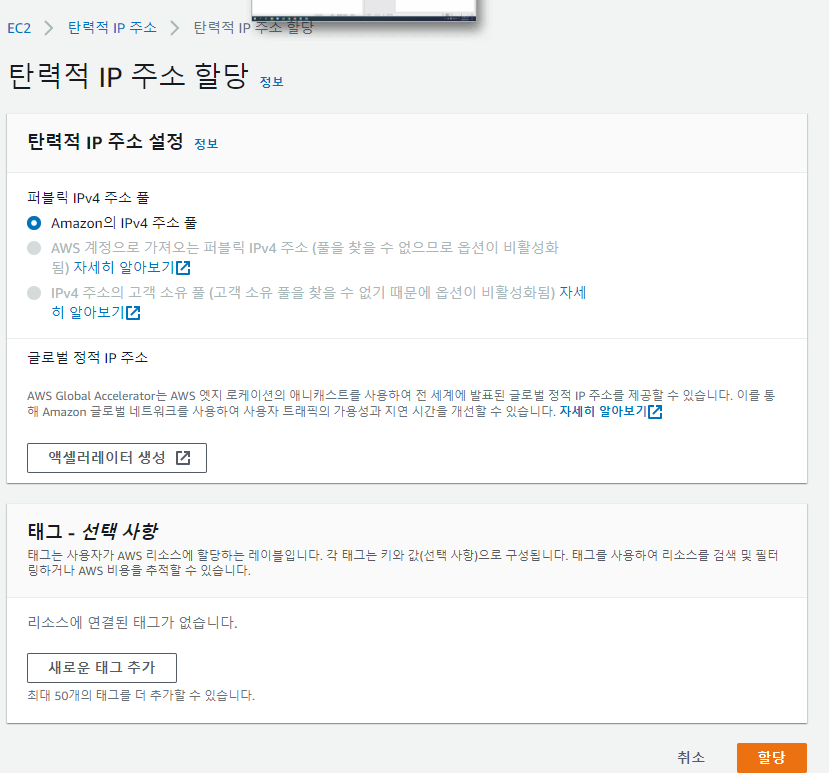


## 키페어 생성



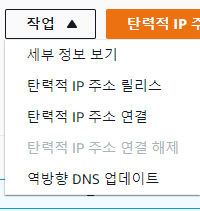
다운로드 경로에 그대로 둬야 한다. 위치를 바꿀 경우 나중에 실행이 안되는 오류가 발생한다.

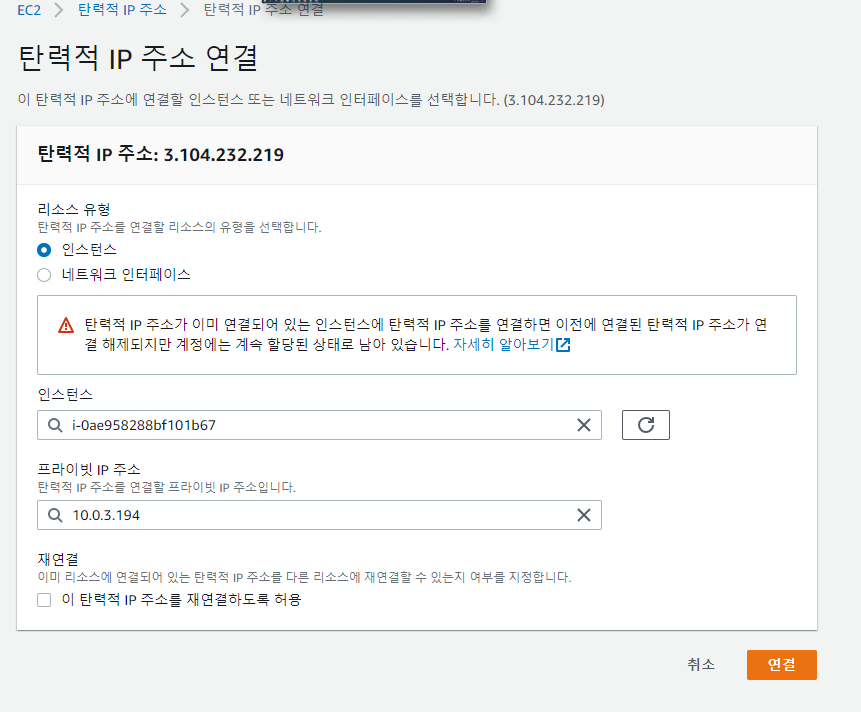
# 탄력적 IP주소 할당



* 기본값 그대로 할당

## 탄력적 IP주소 연결





* 인스턴스 및 프라이빗 IP주소를 선택하여 연결 버튼을 누른다.

인스턴스에서 퍼블릭 주소 확인



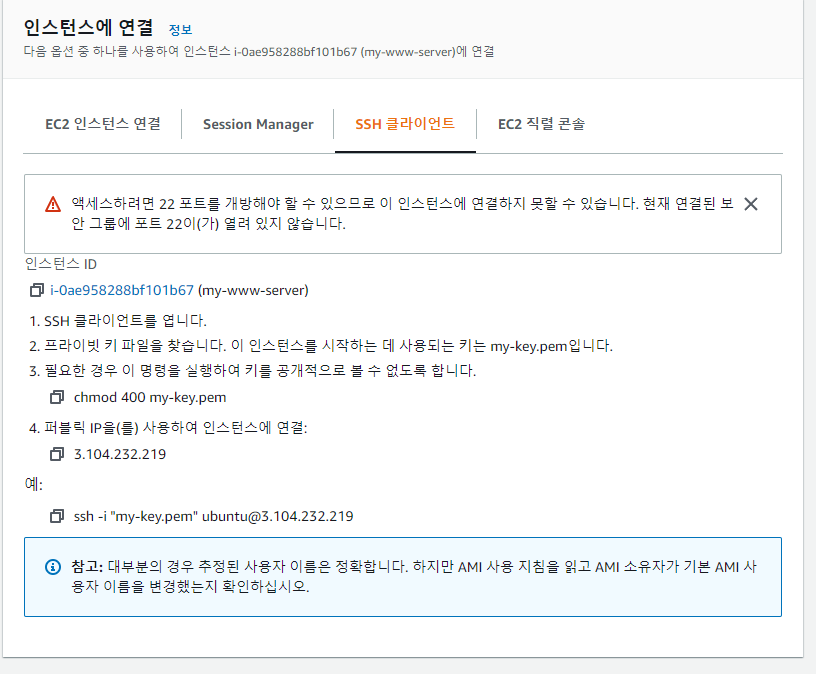
* 다음과 같이 생성이 완료되었으며 5분정도 기다려야 할당이 완료된다.

## IP주소와 EC2인스턴스 연결



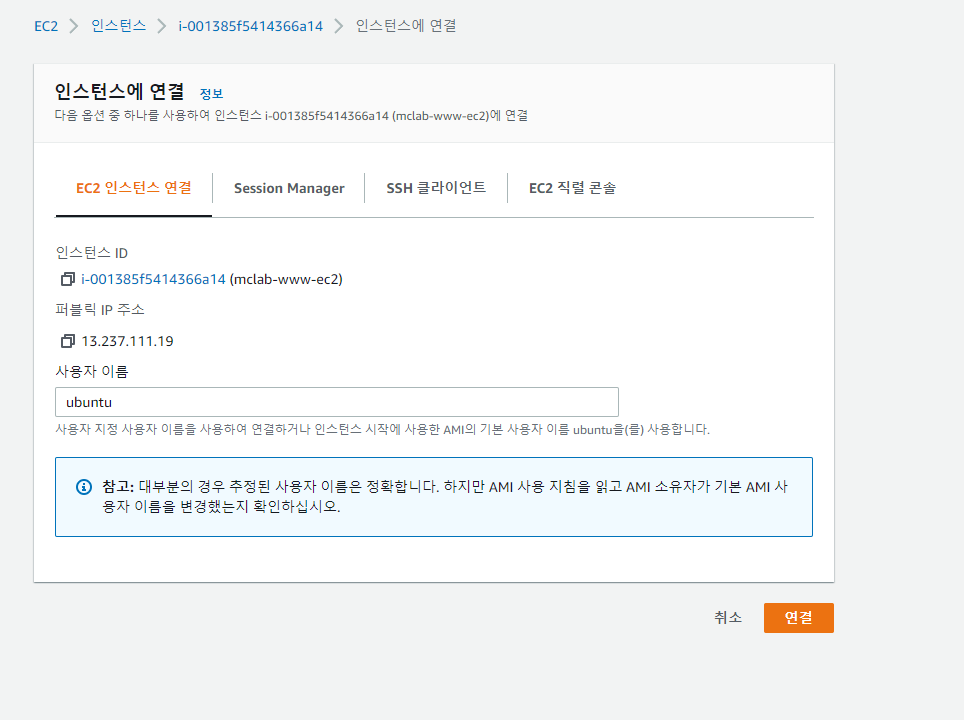
* 연결 버튼을 클릭한다.

## 퍼블릭 IP 확인



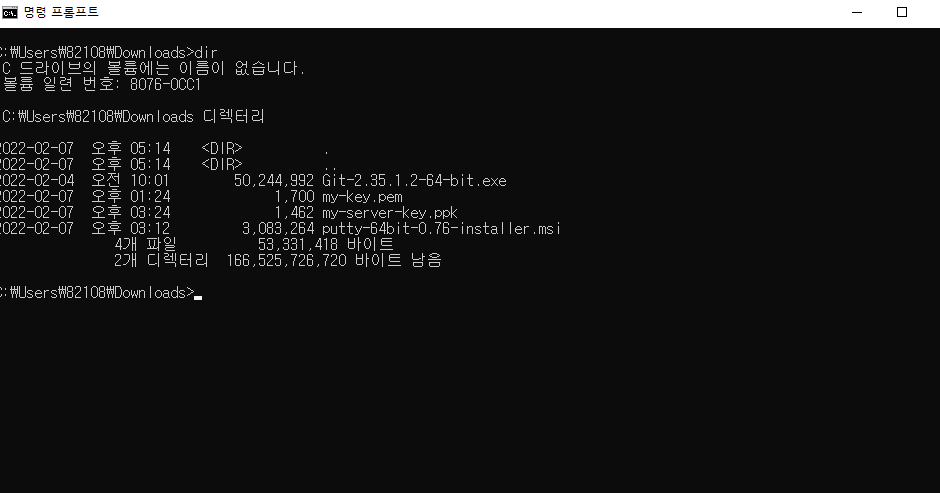
* 예: 이후에 있는 값을 복사해준다.

## EC2 인스턴스 연결

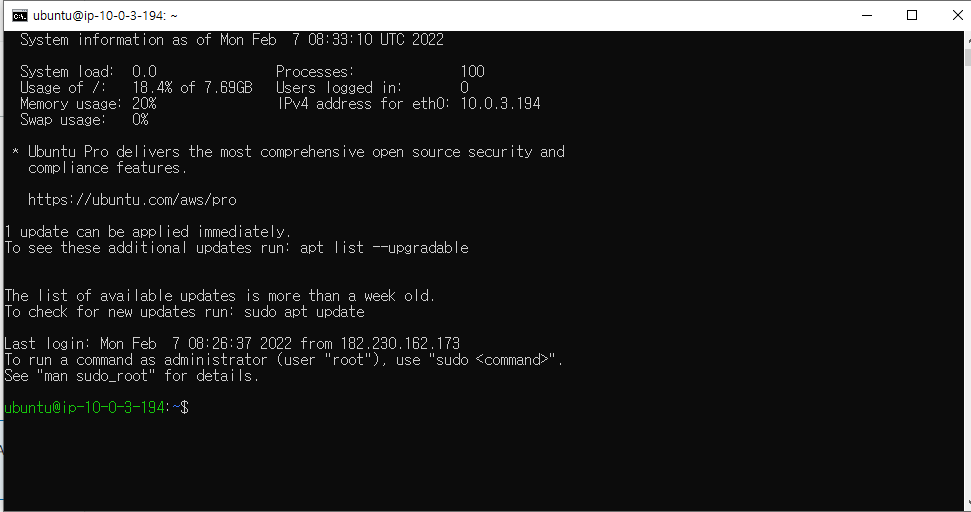


* 여기 퍼블릭 IP 주소 및 사용자 이름은 나중에 PUTTY에서 사용하므로
* 필요할 때마다 가서 확인 해 준다.

# cmd로접속하기



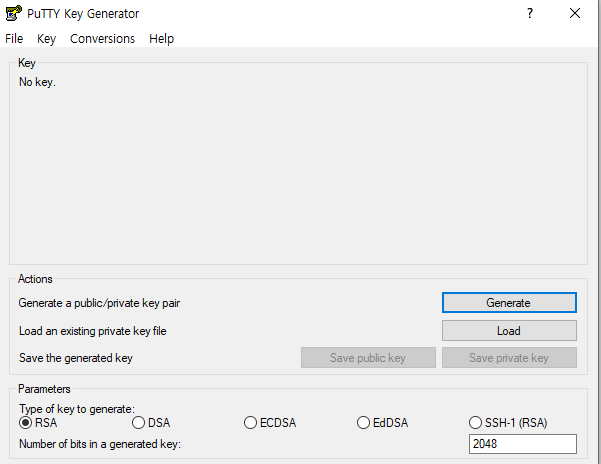
* 아까 저장해둔 값을 key 값이 있는 디렉토리에서 붙여넣기를 해준다.



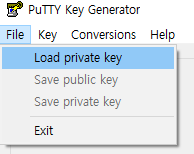
* 다음과 같이 Ubuntu에 접속할 수 있다.
* sudo apt update 명령어를 입력하여 업데이트
* sudo apt install nginx 입력하여 nginx 설치

# Putty로 연결

## 실행

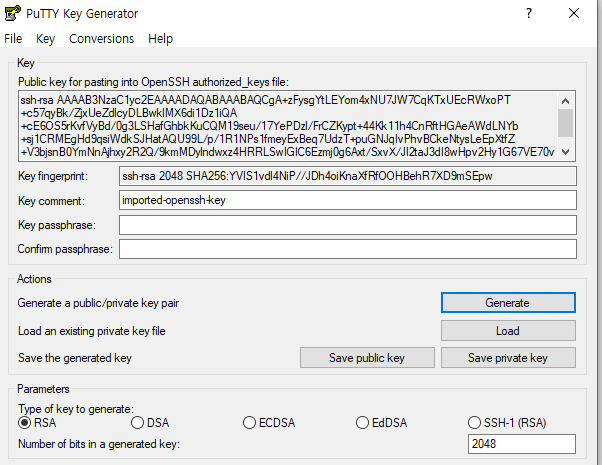


## 등록



* file : Load private key 를 선택한다.

## Save private key



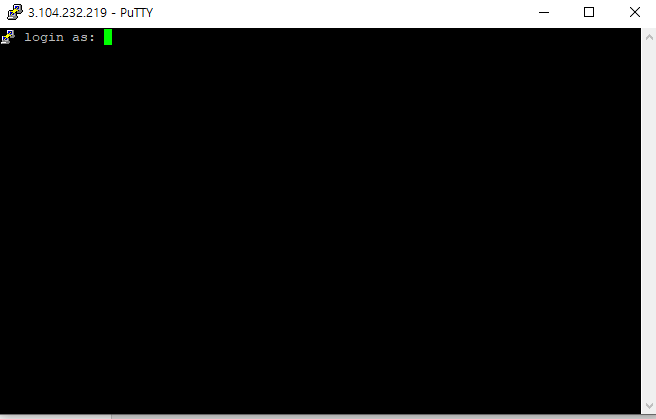
* save private key버튼을 클릭하여 확장자를 바꿔준다.

## putty실행



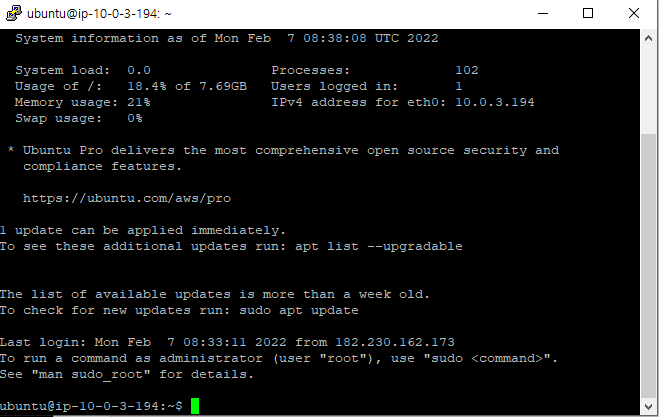
* putty를 실행해 주고 방금 전 확인한 IP주소를 입력해주고 SSH//Auth로 들어가 변환된 key값을 입력해 준다.

## Login as값 입력



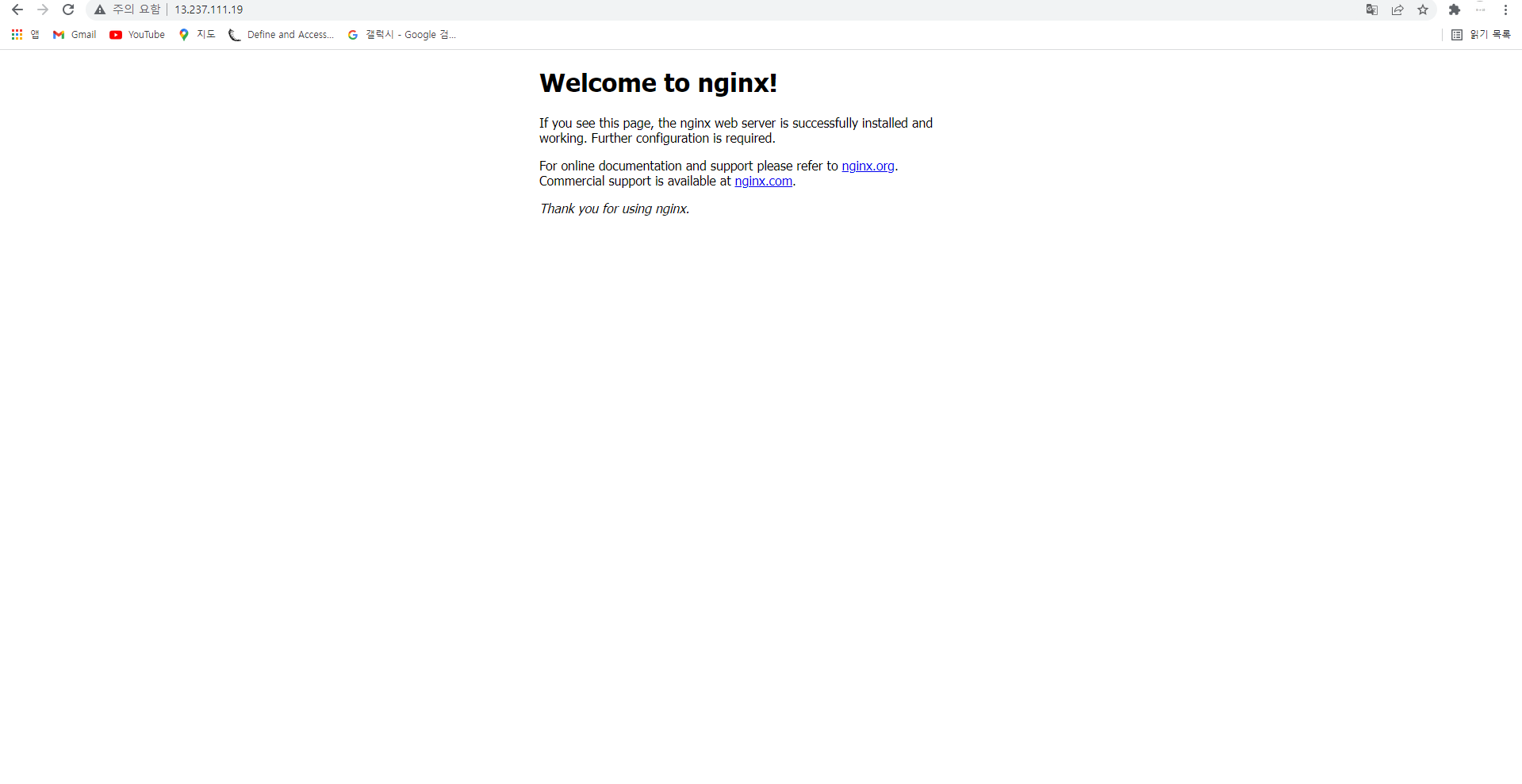
* 여기에 사용자 이름인 ubuntu를 입력해 준다.

## 완성



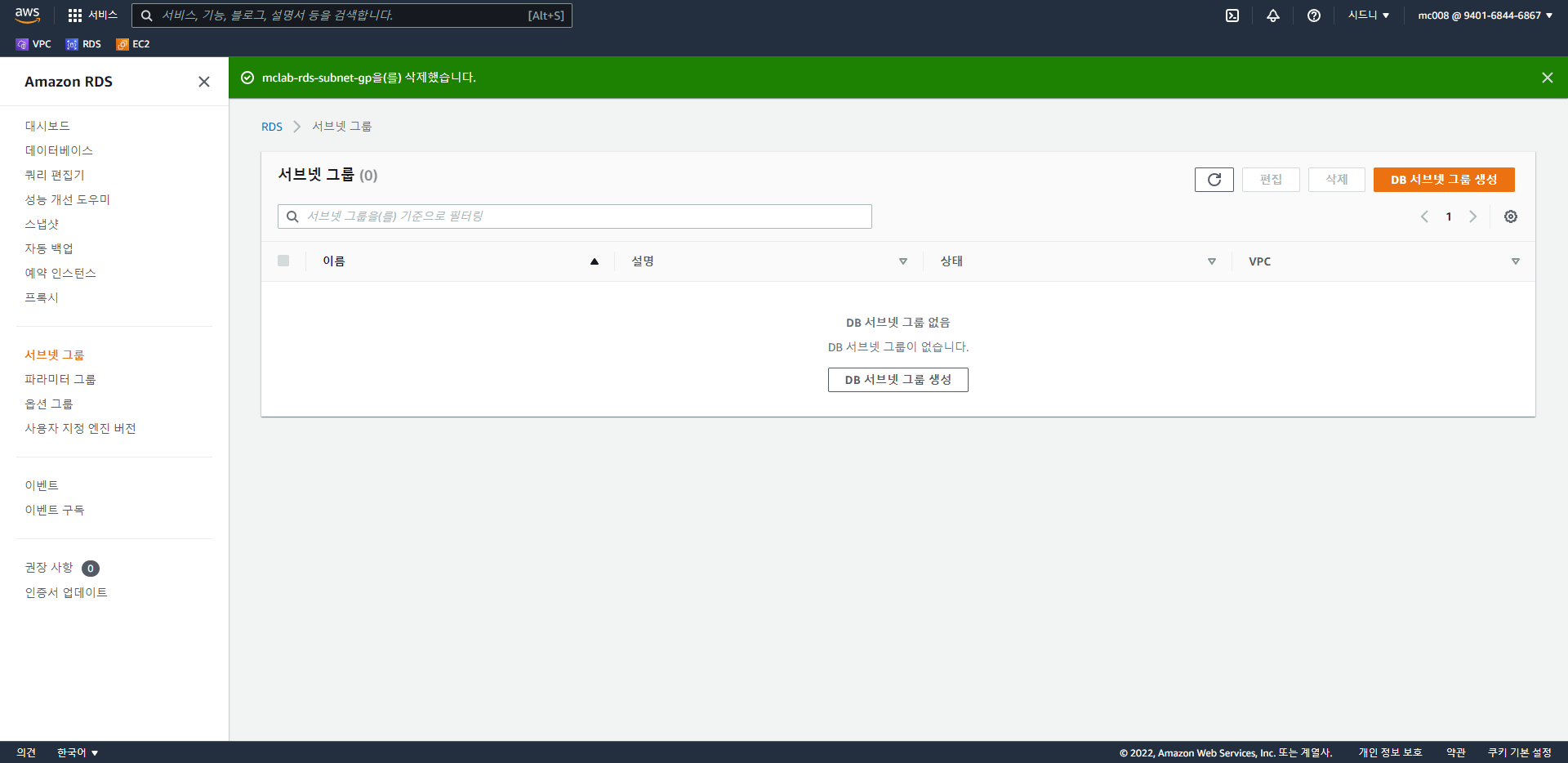
* 다음과 같이 성공적으로 실행 되었음을 알 수 있다.

## nginx설치완료

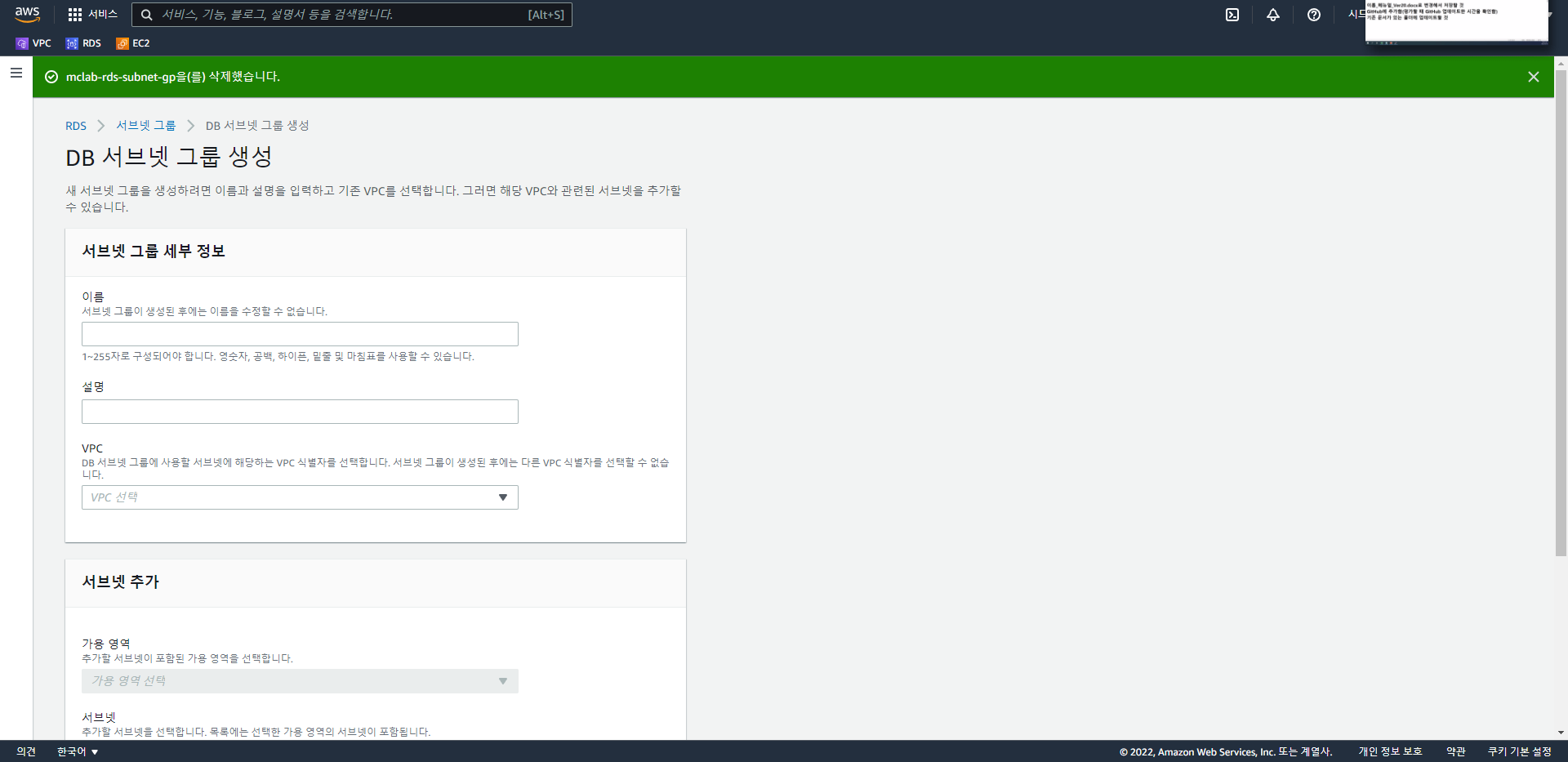


* 이제 ip주소를 입력해 주면 다음과같이 성공적으로 화면이 나타난다.

# RDS(서브 넷 그룹 생성)

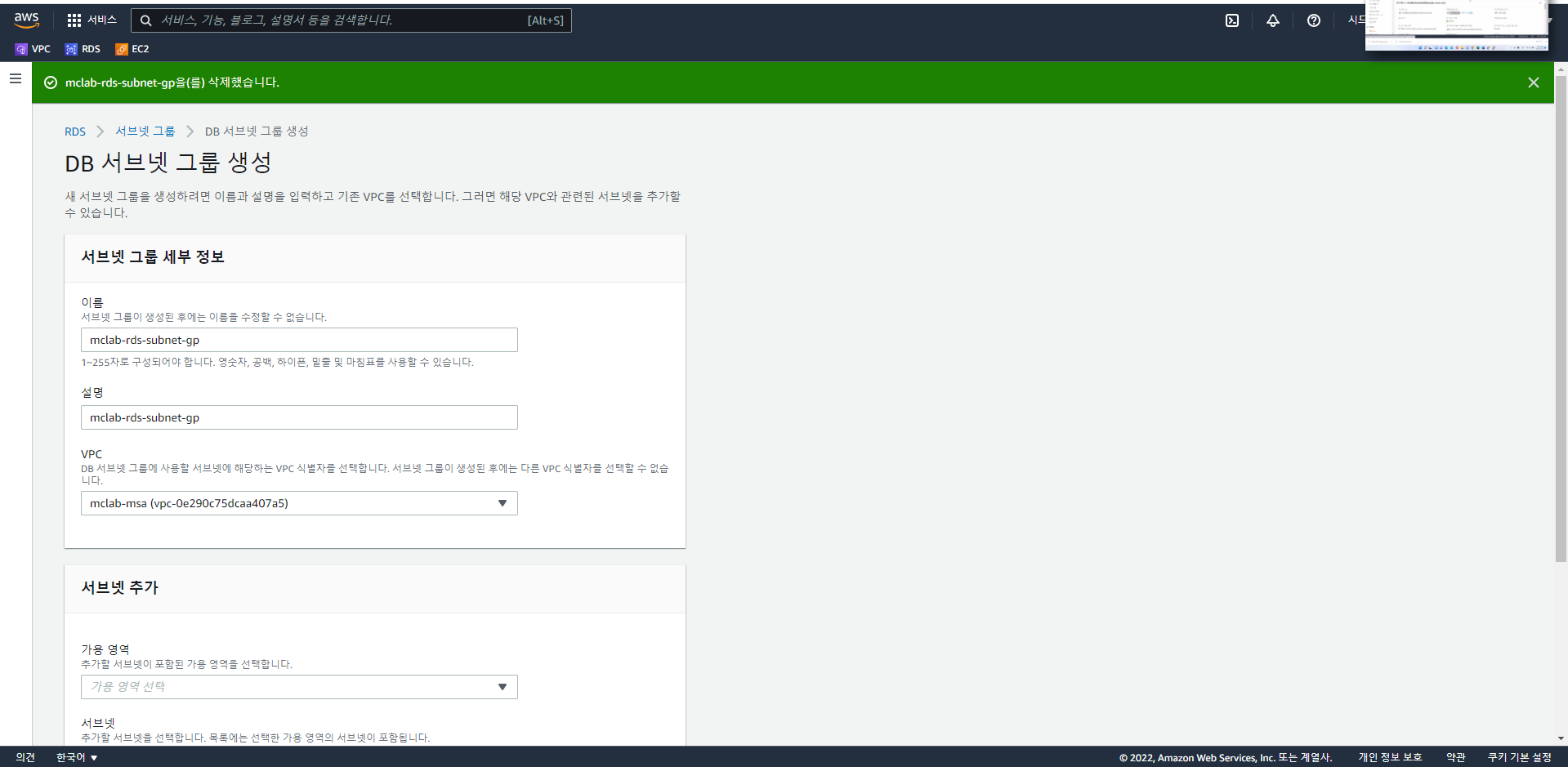


* RDS를 생성하기 위해 서브 넷 그룹을 생성해야 한다.
* 지금 화면이 바로 서븐 넷 그룹의 기본 화면이다.
* 다음 서브 넷 그룹 생성 버튼을 클릭해 준다.

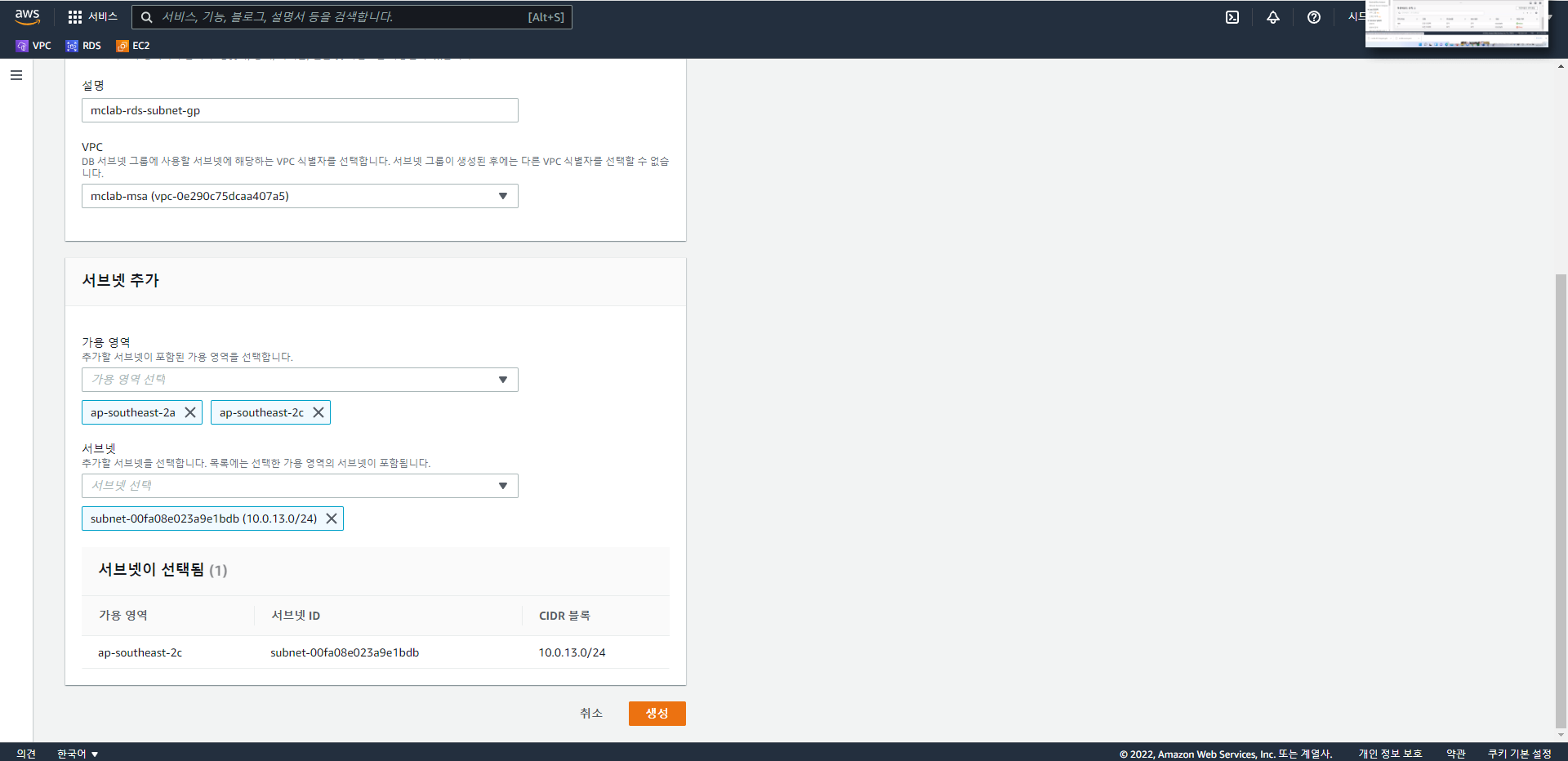


* 기본 화면

## 정보 입력

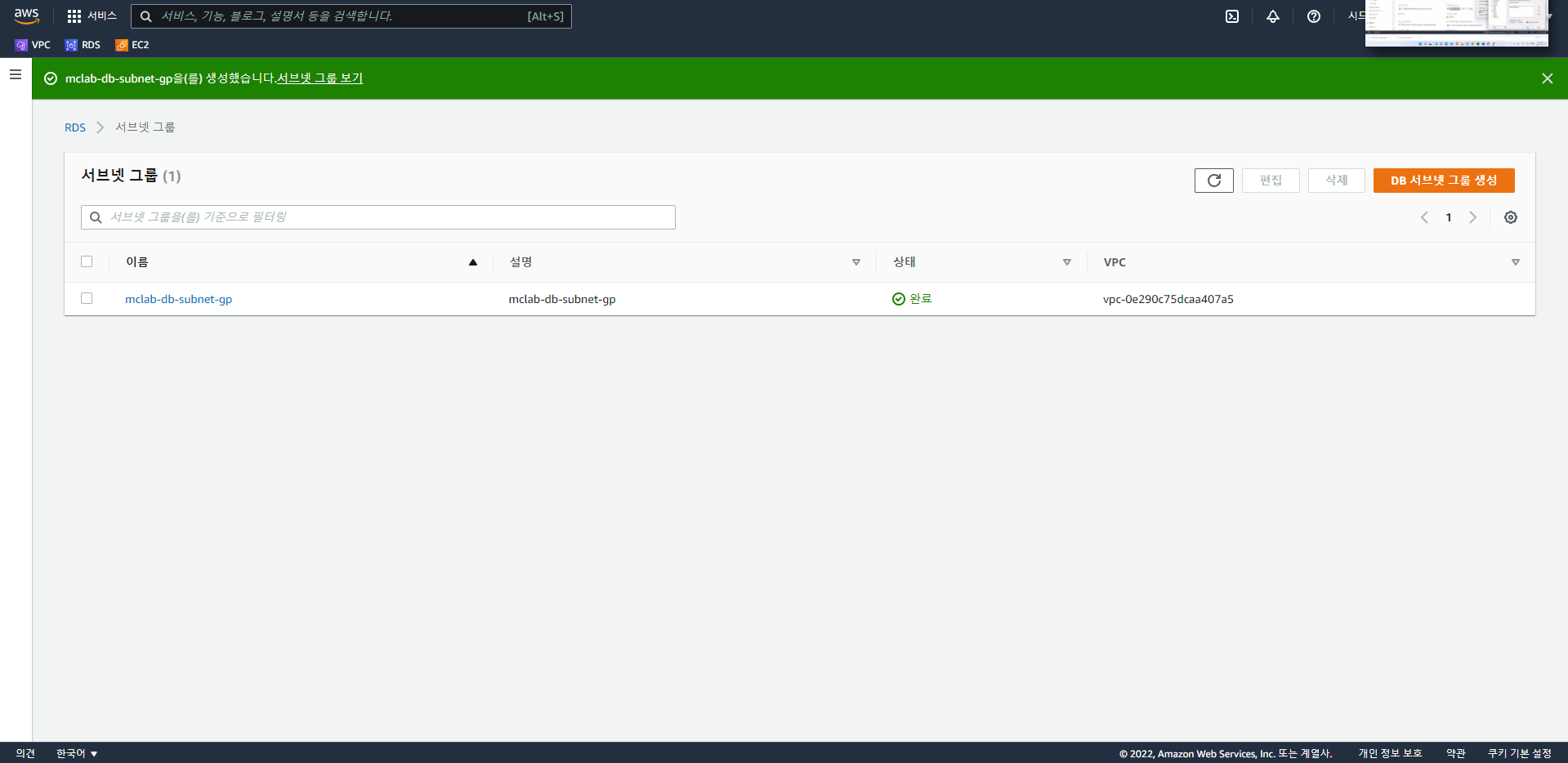


* 이름 및 설명 VPC를 입력 및 설정해 준다.



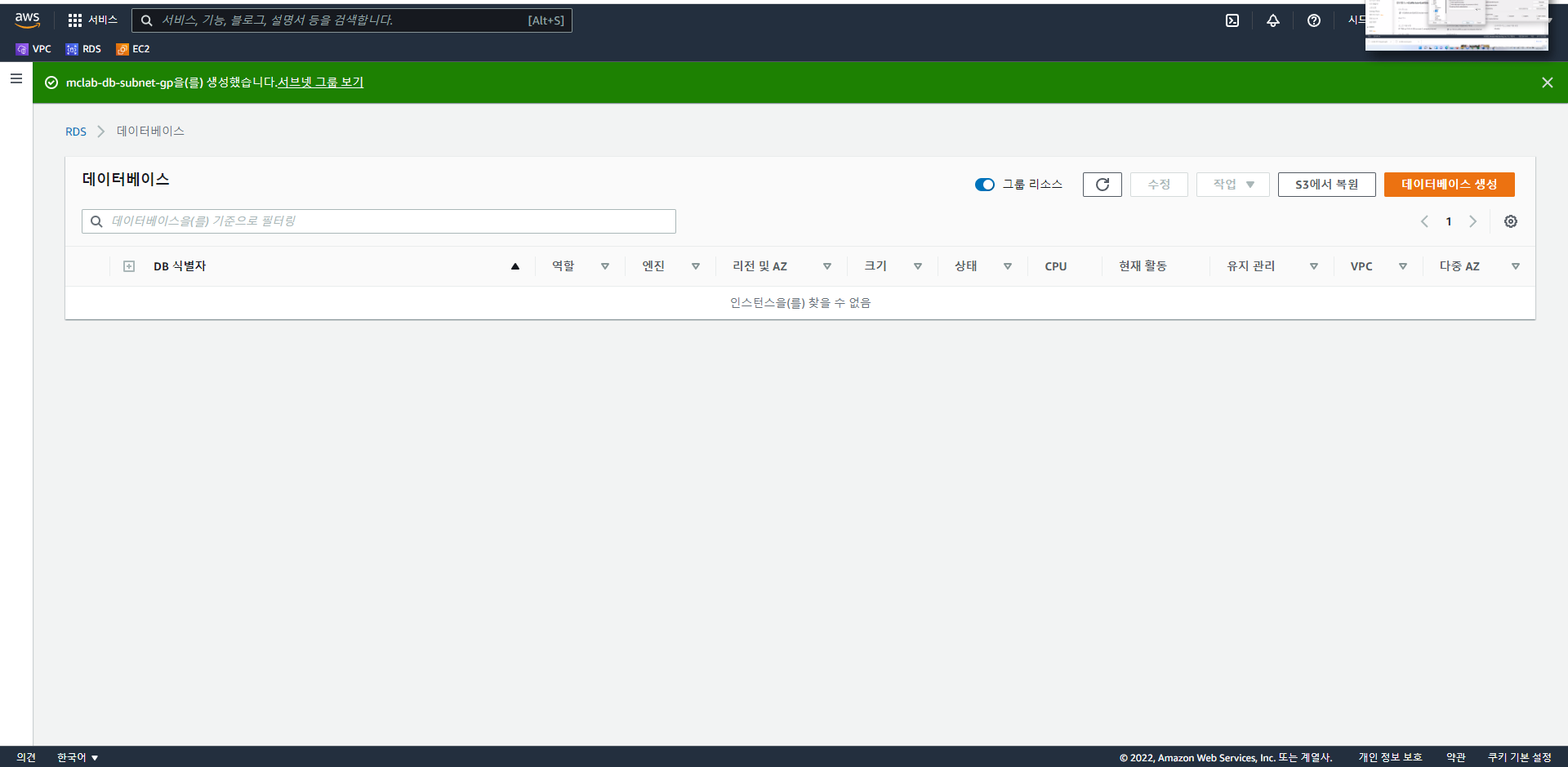
* DB가 설치된 subnet가용 영역을 입력하고, 서브넷을 선택하여 입력하고 생성
* 이때 db subnet이 2개이므로 가용역역 또한 2개 선택해준다.

## 생성 완료



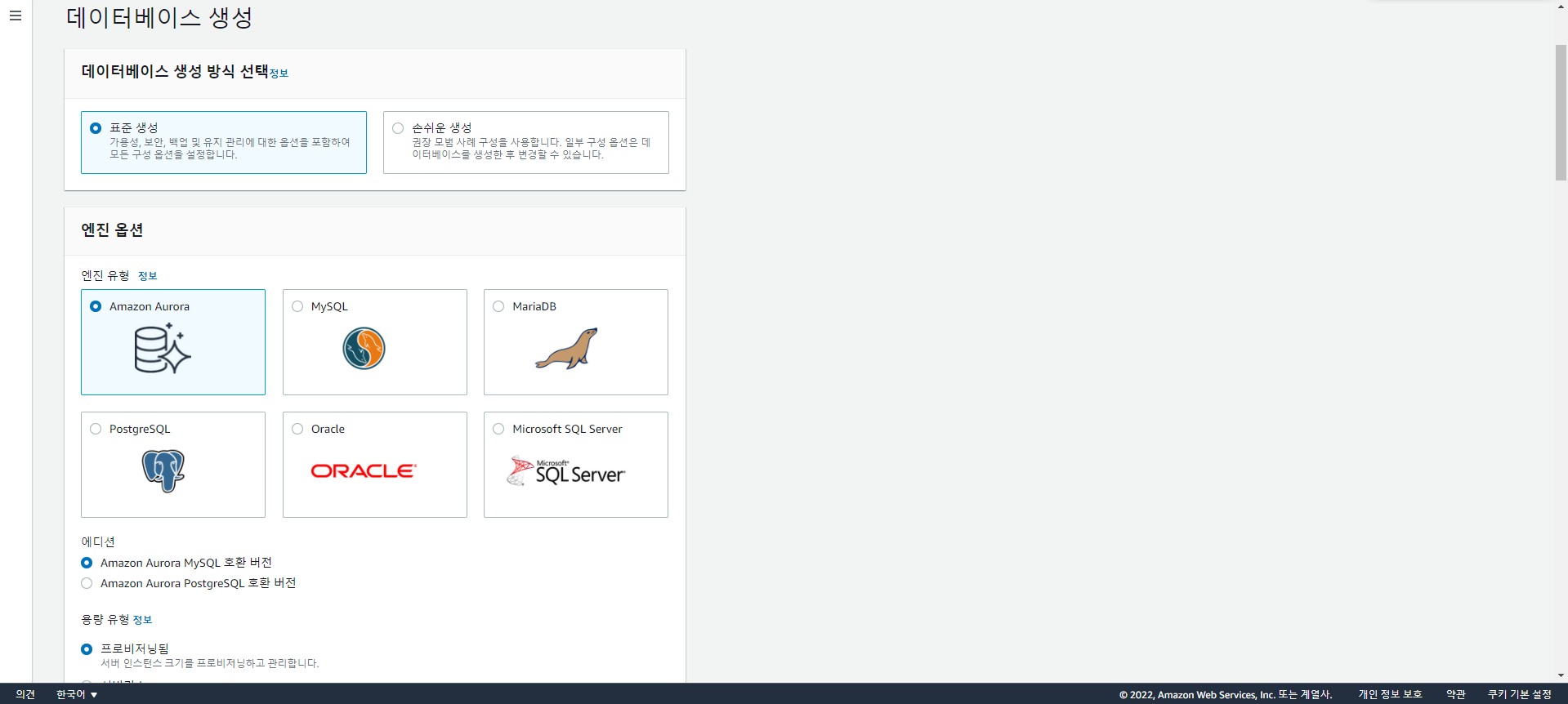
* 정상적으로 만들어 졌을 때 화면

# RDS(DB생성)



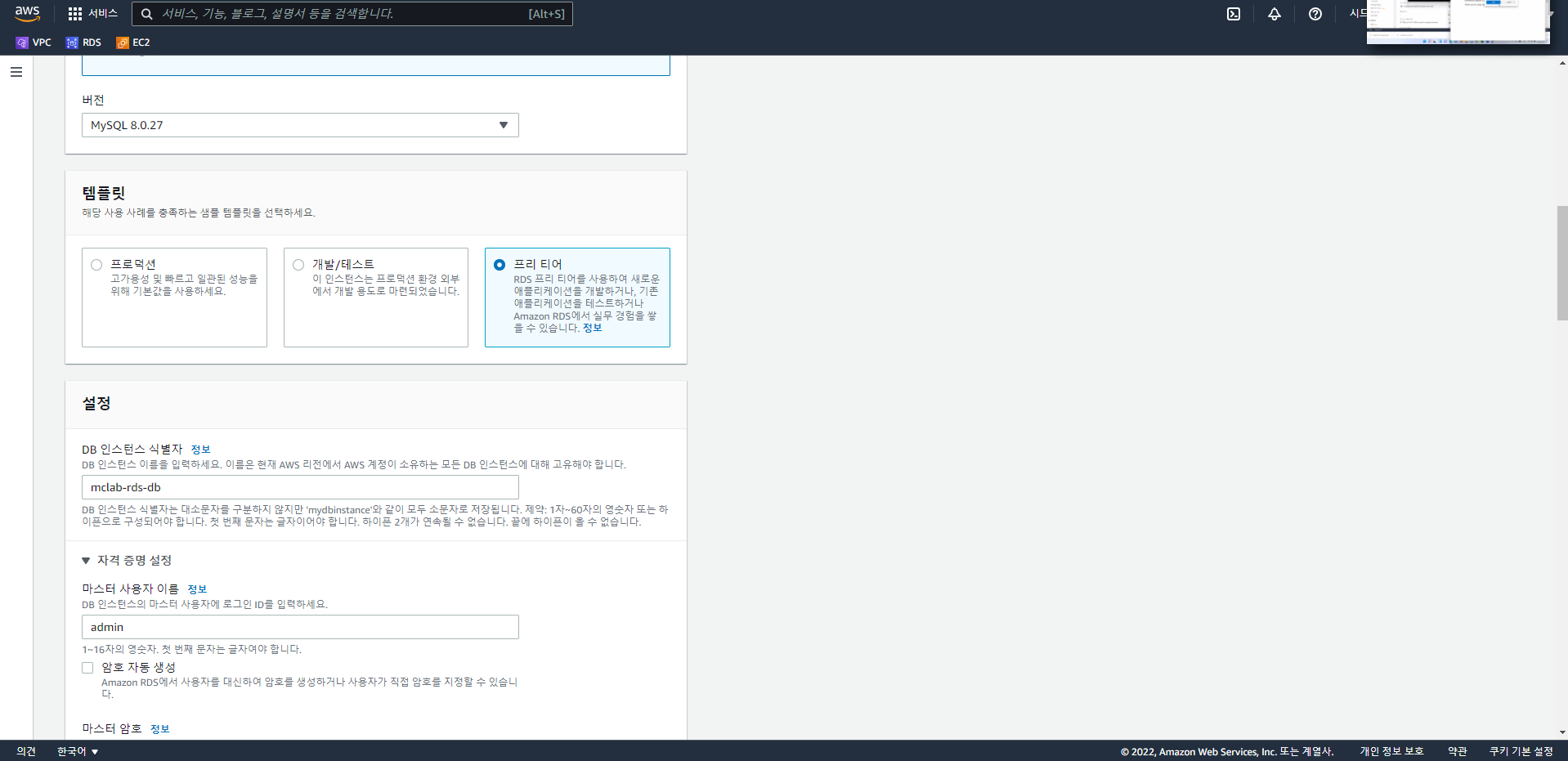
* 기본화면에서 데이터베이스 생성 버튼을 클릭한다.

## 엔진옵션



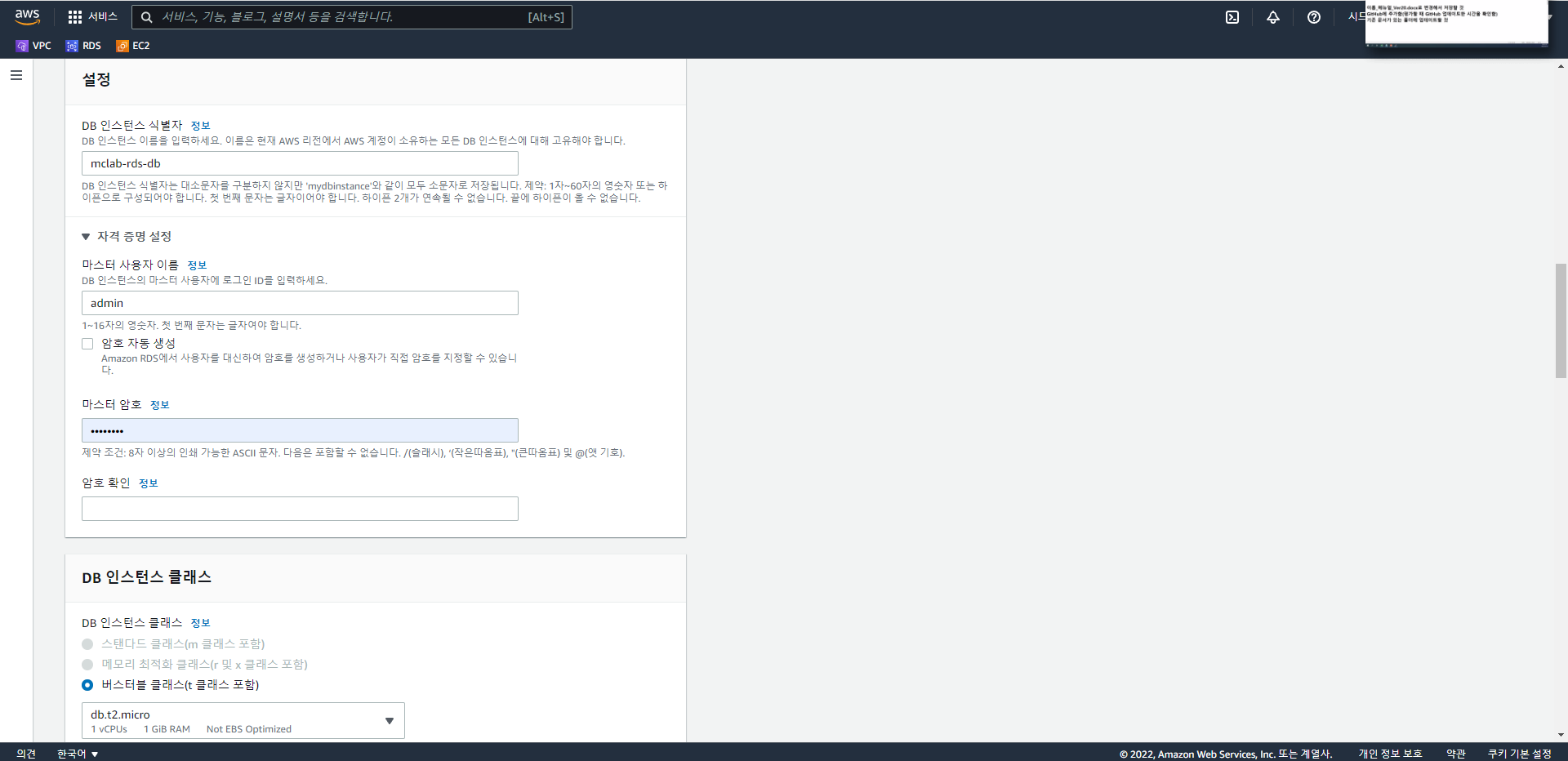
* 사용 하고자 하는 엔진을 선택한다.

## 버전/템플릿 설정

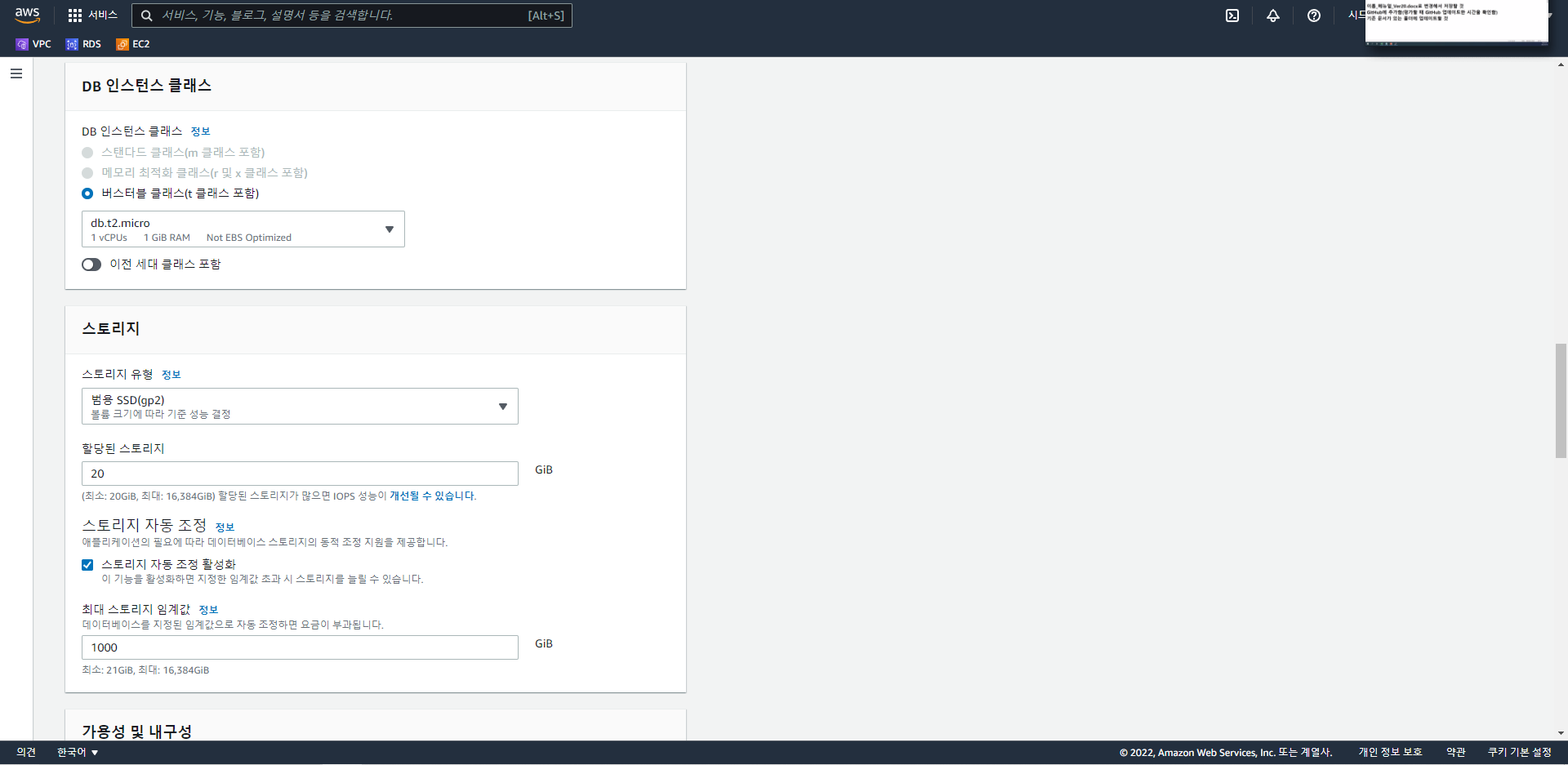


* 버전과 템플릿을 선택한다.

## 설정

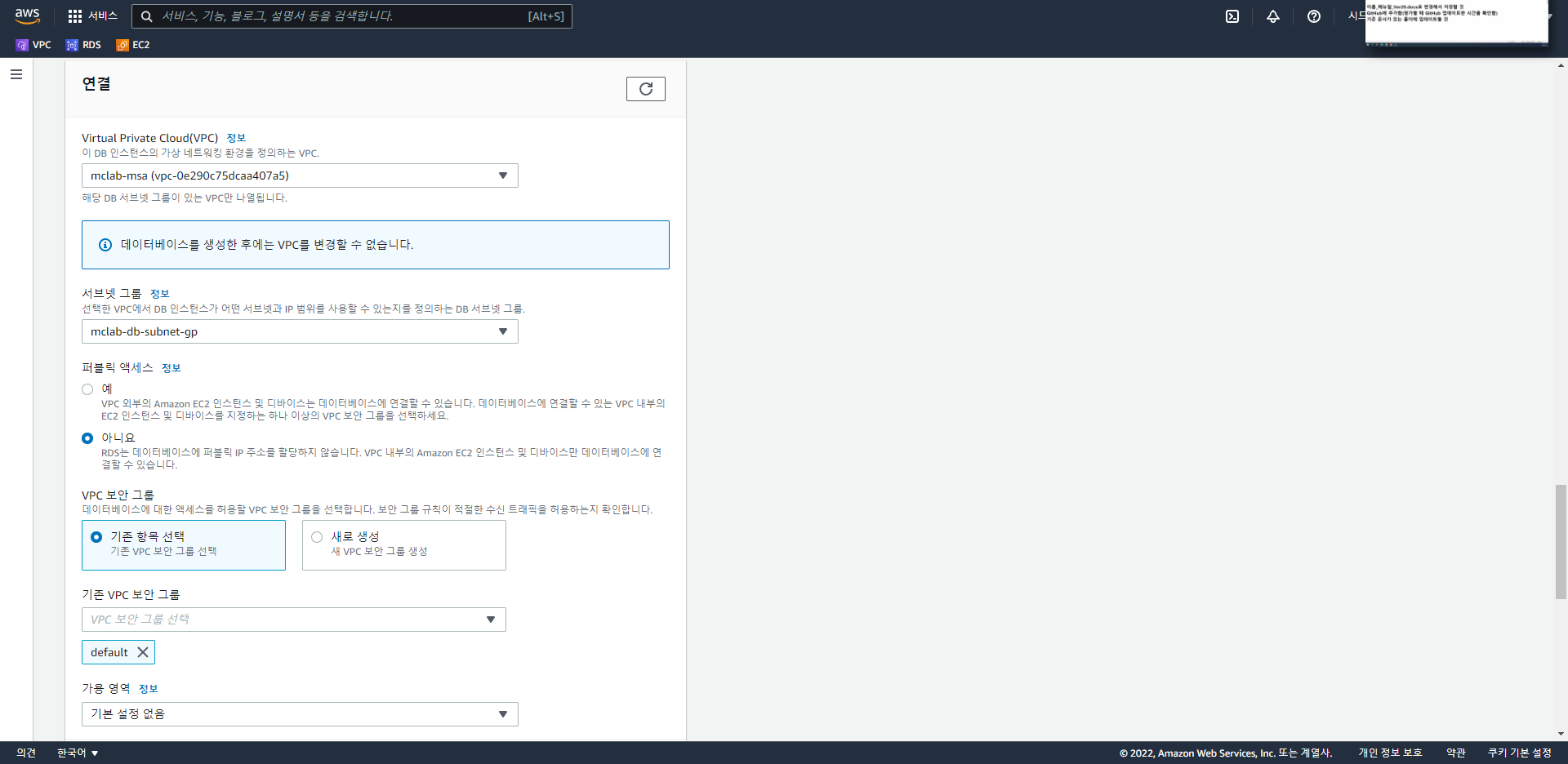


* DB 인스턴스 식별자를 입력해 준다.
* 자격 증명 설정의 비밀번호 값을 설정해 준다.

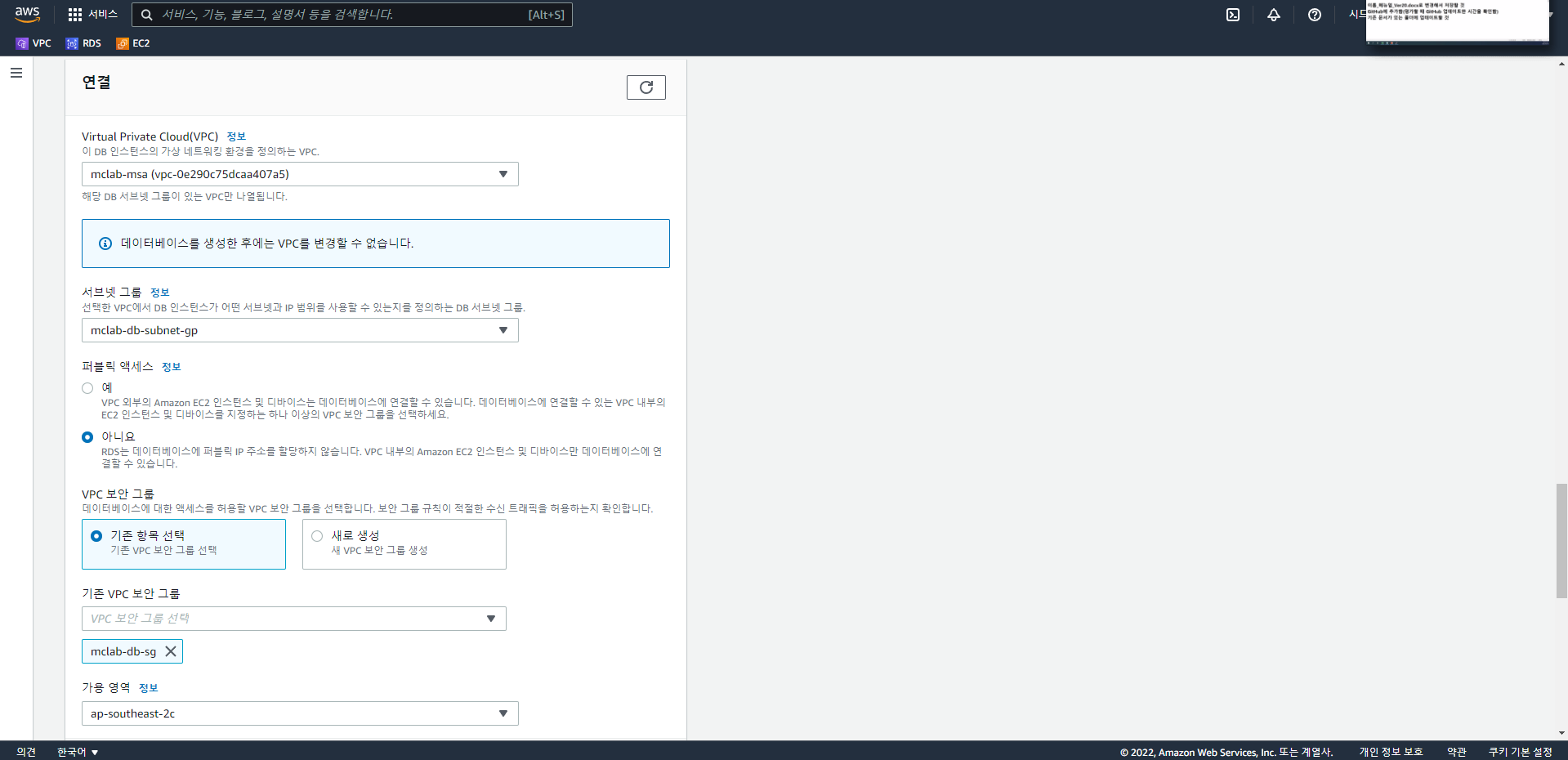


* 다음 값들은 기본값 그대로 둔다.

## 연결

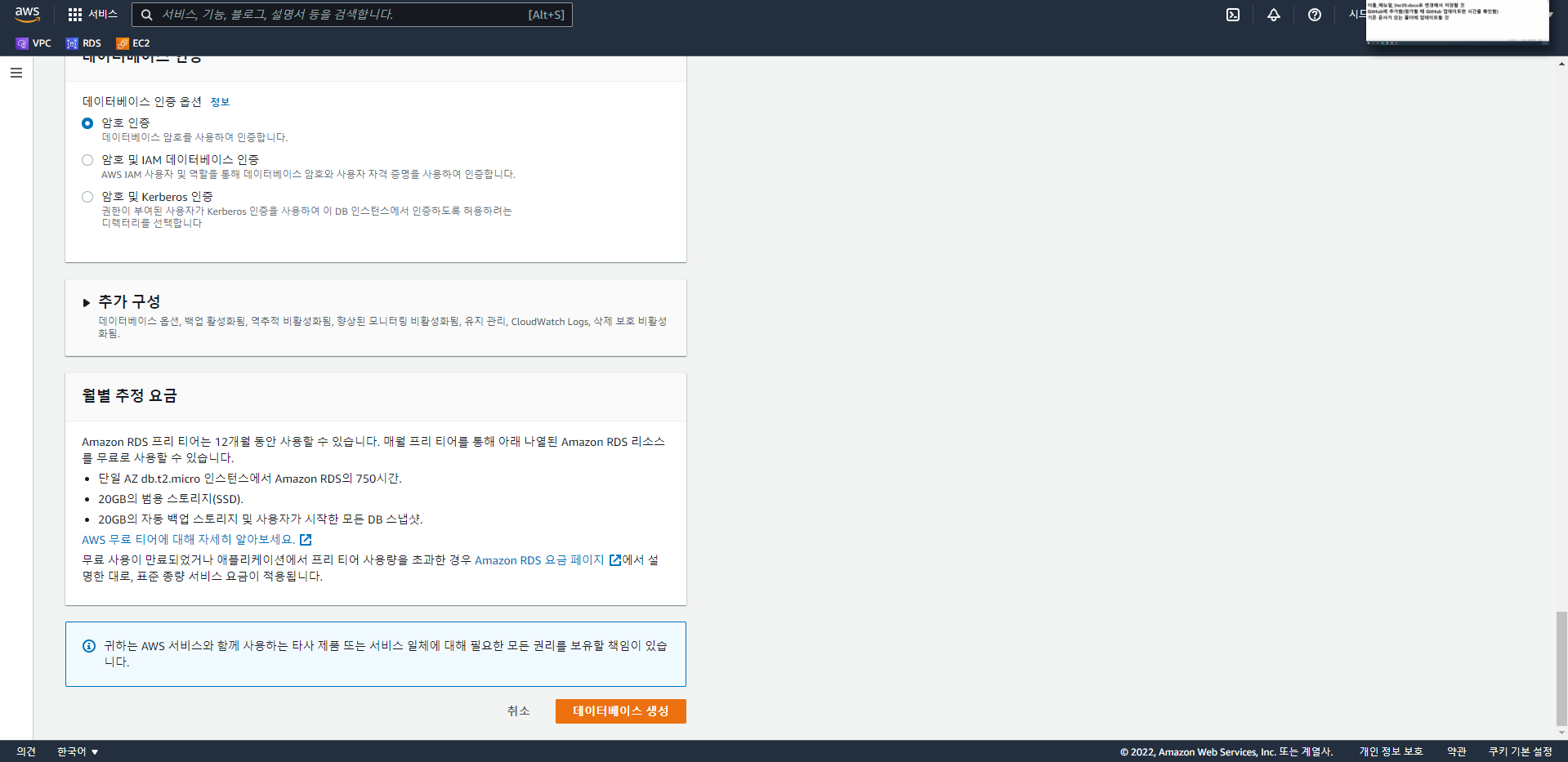


* 나머지 값들은 건들지 말고 기본 VCP보안그룹에 default 값을 x키를 눌러 없앤다.
* 다음에 내가 만든 DB보안을 클릭해준다.
* 다음 서브 넷 그룹을 참조하여 가용역역을 설정해 준다.



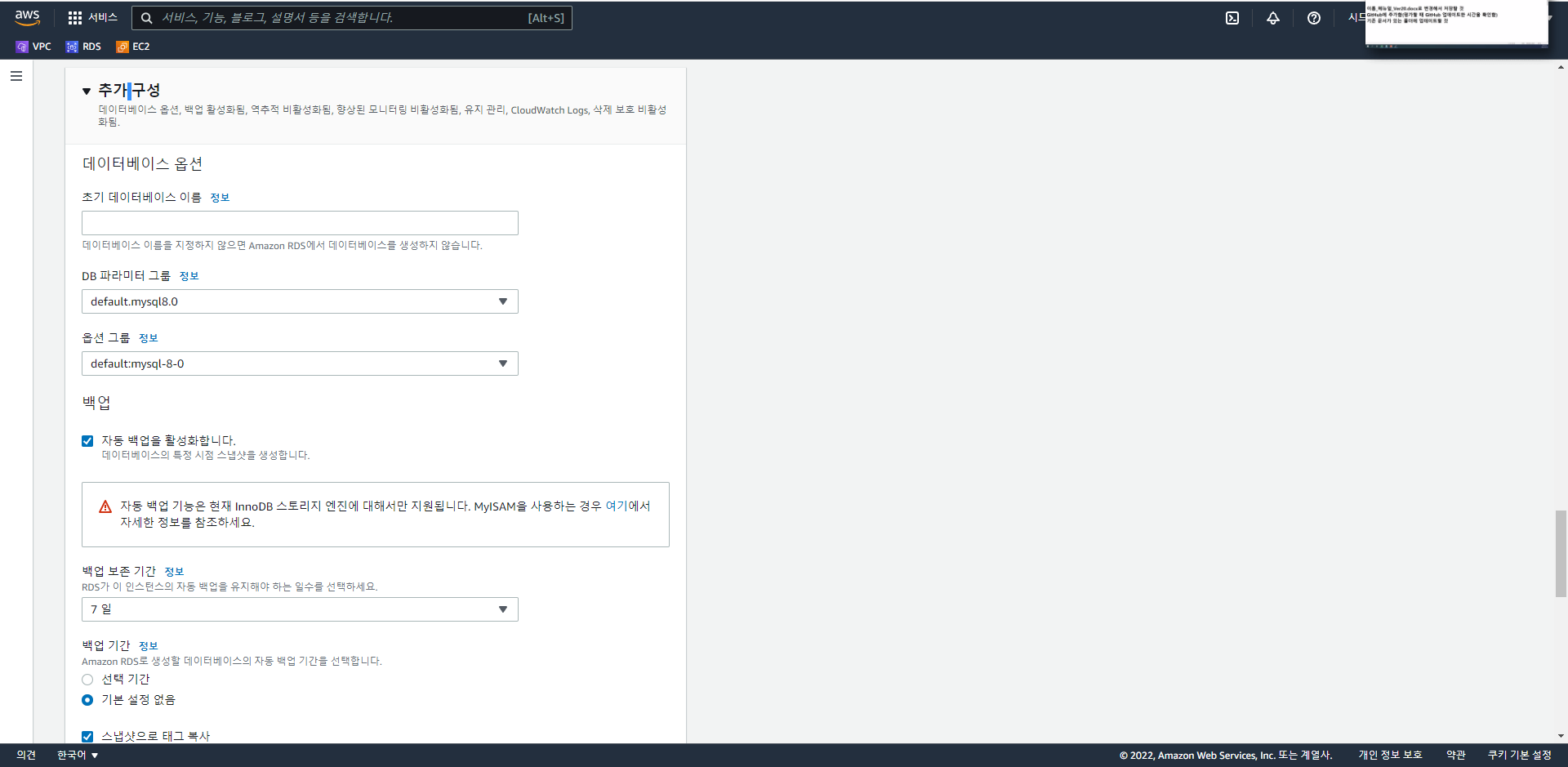
* 입력이 완료 되었을 때 화면

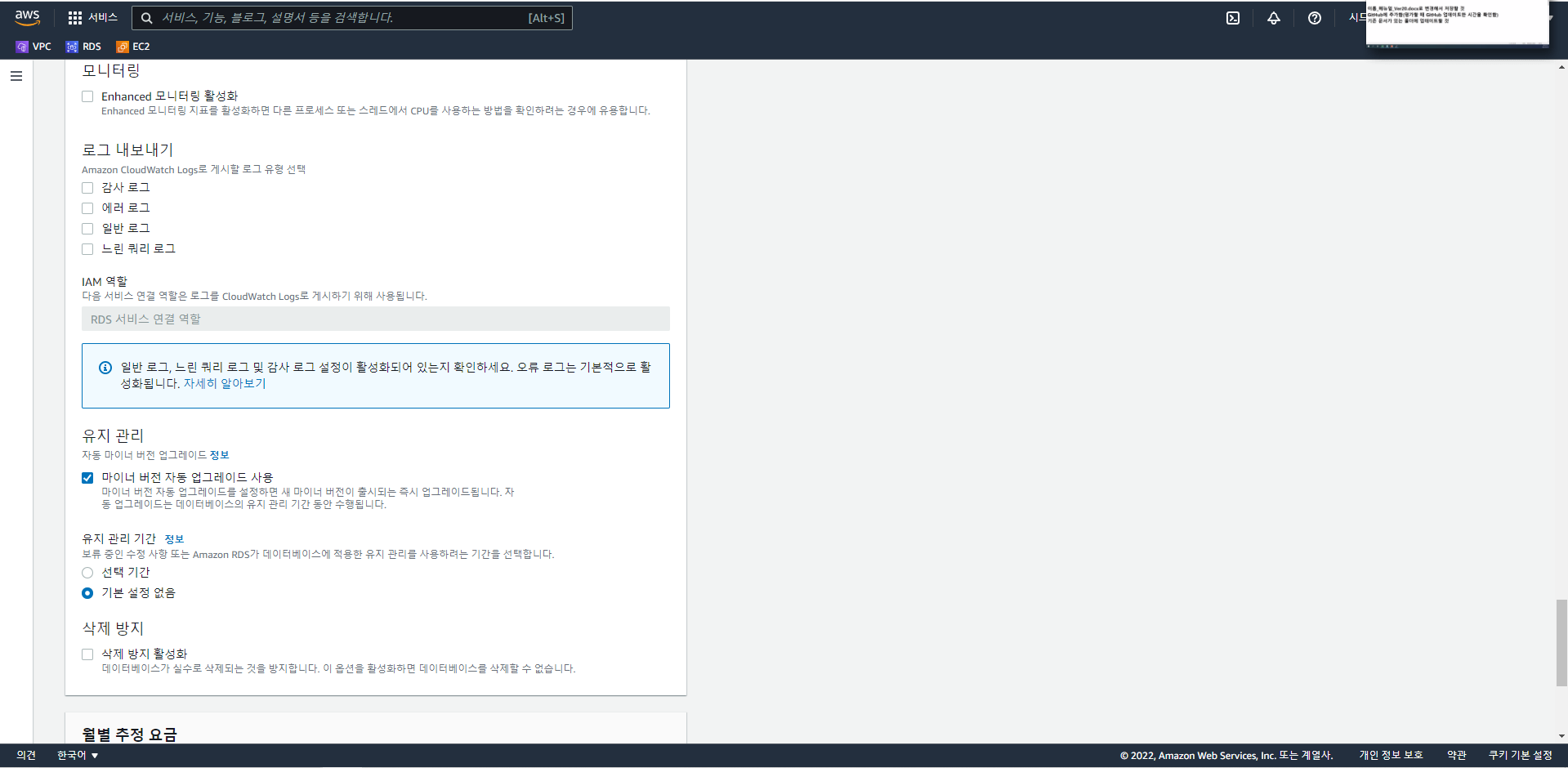
## 추가 구성

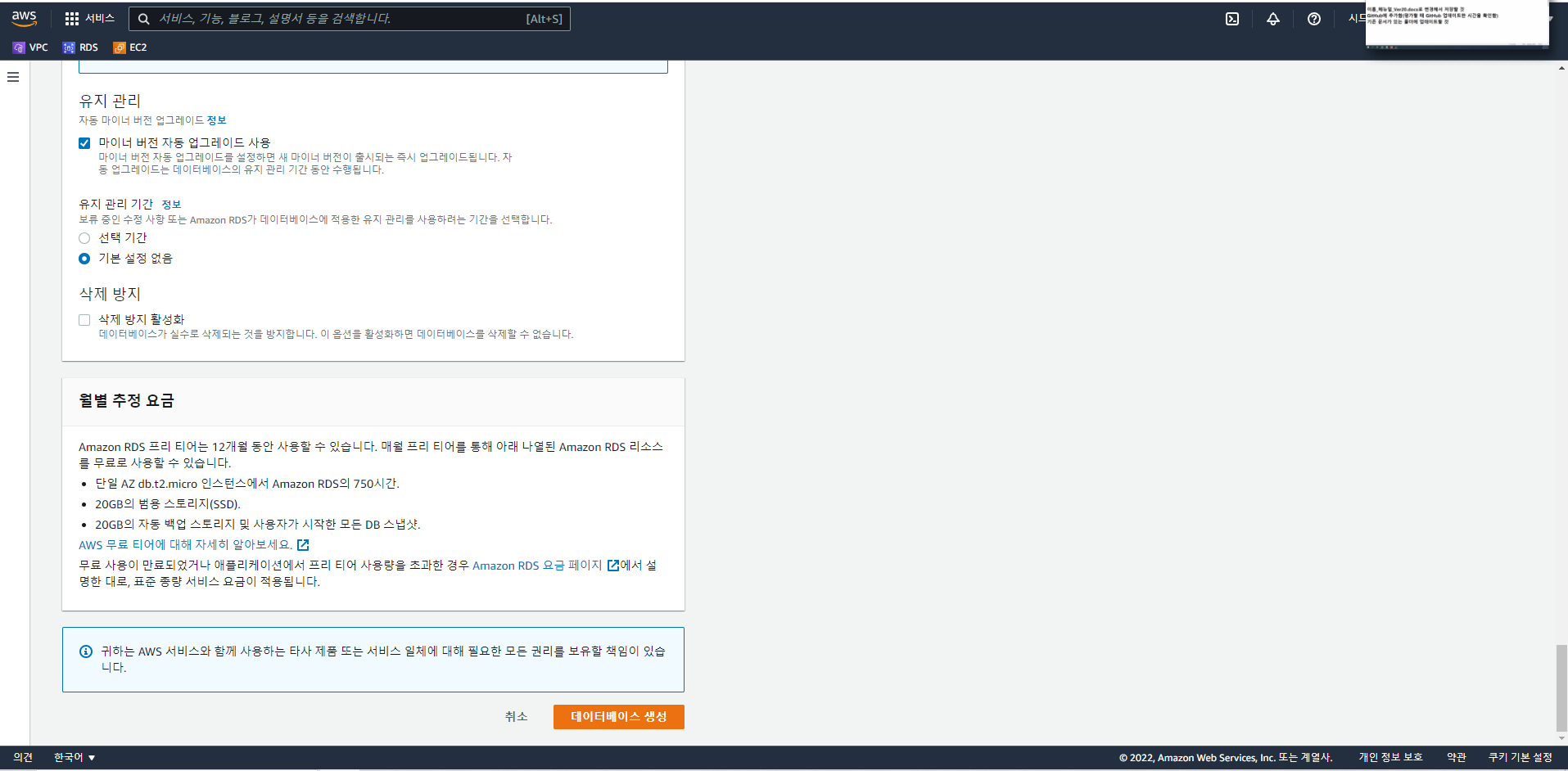


* 추가 구성 기본화면에서 추가구성을 클릭한다.

추가구성 기본 선택 화면

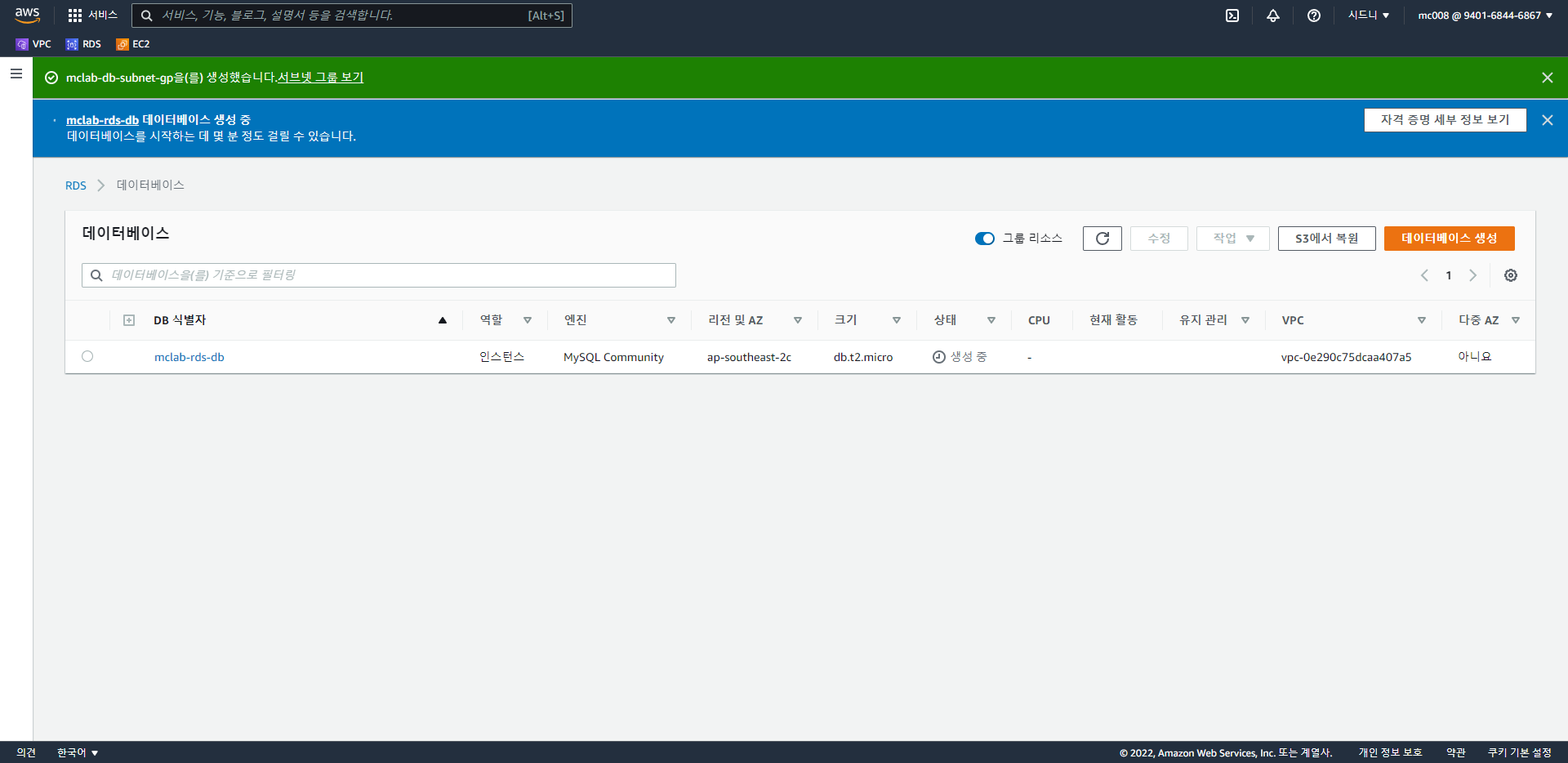






* 다음과 같은 기본 화면에서 먼저 1. 초기 데이터 베이스 이름을 입력
* 2. 백업 비활성화
* 3. 유지관리 비활성화
* 4. 데이터 베이스 생성 버튼 클릭

## 생성 완료



* 정상적으로 생성되었을 때 화면