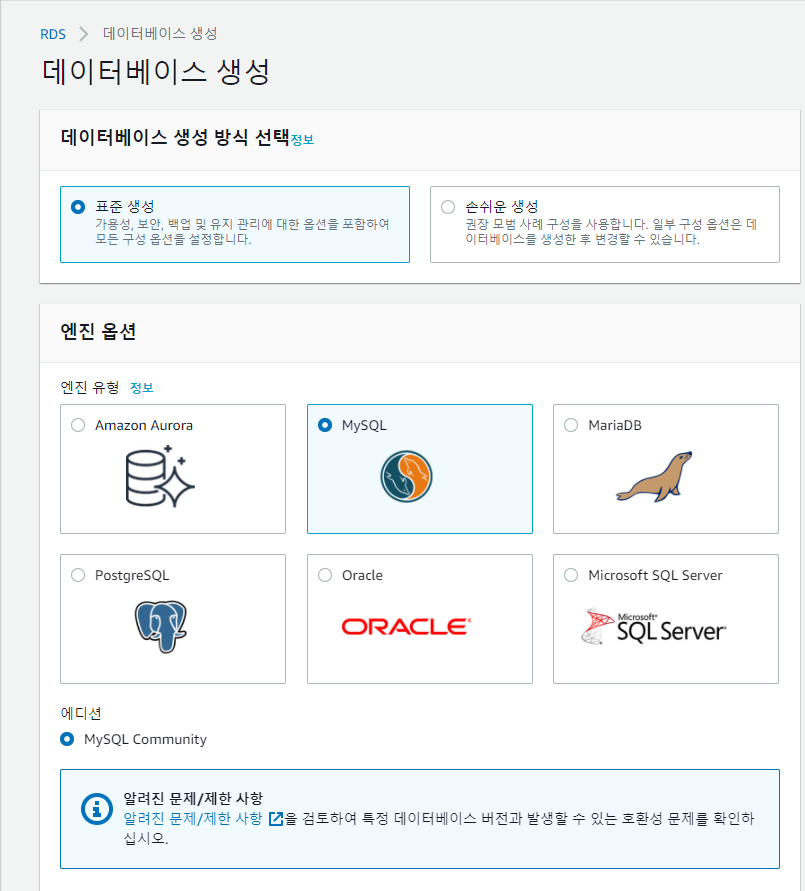
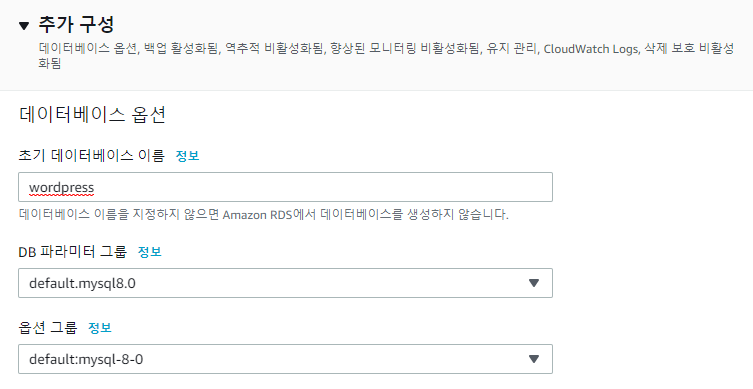
1. RDS로 MySQL 데이터베이스를 생성

1) MySQL 데이터베이스 생성

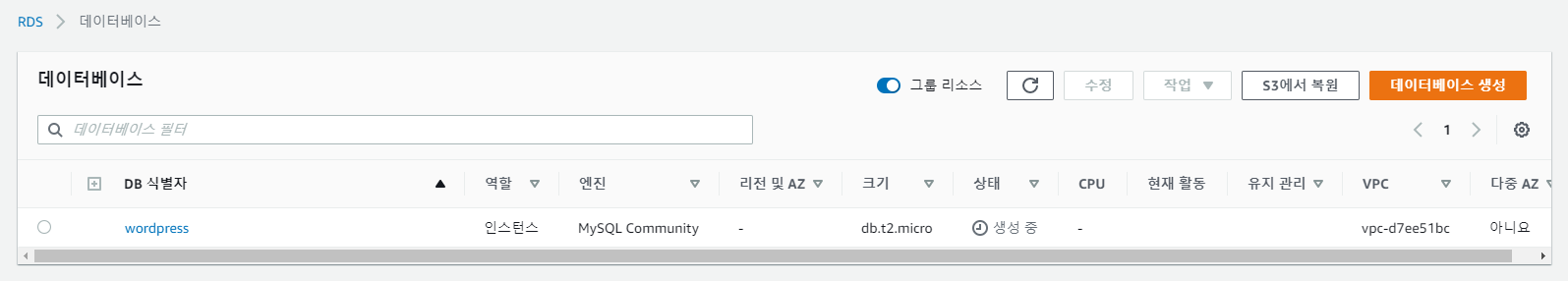




템플릿을 프리 티어로 지정해주고, 설정에서 식별자와 사용자 이름, 암호를 지정한다.



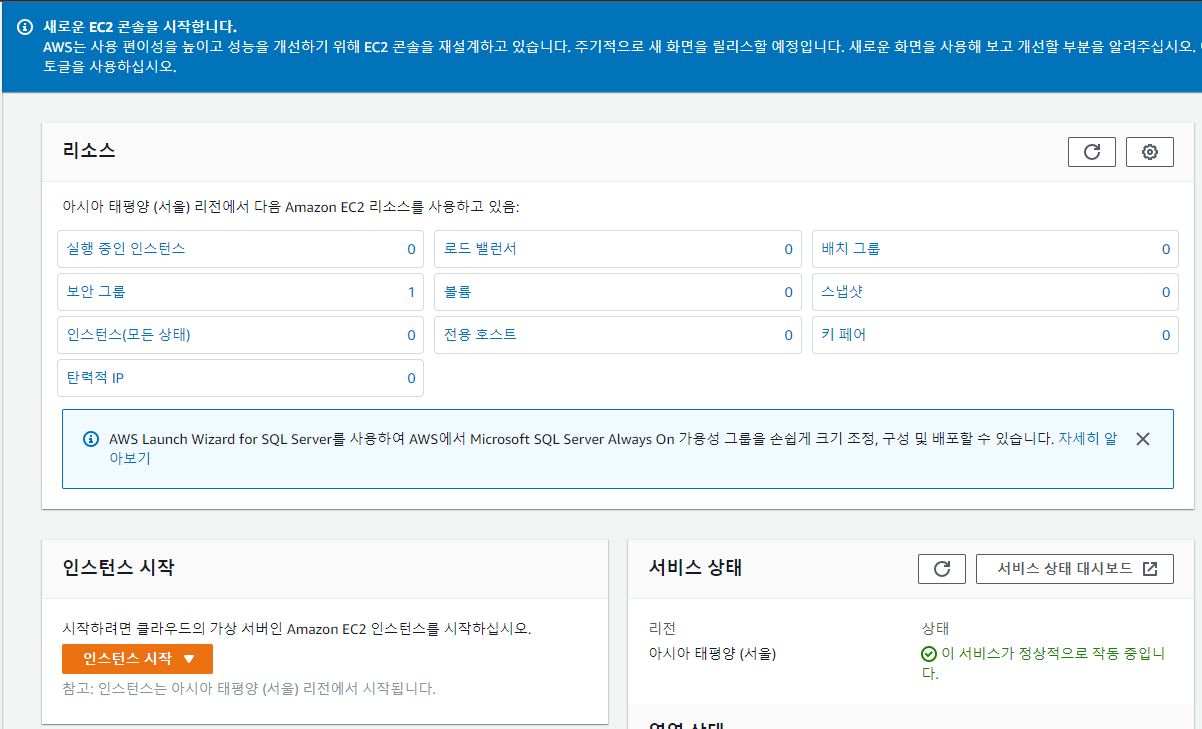
**초기 데이터베이스** 이름을 **wordpress**로 설정합니다. 이렇게 하면 초기화와 동시에 RDS가 MySQL 인스턴스에서 데이터베이스를 생성합니다. 이 데이터베이스 이름은 데이터베이스에 연결할 때 사용합니다.



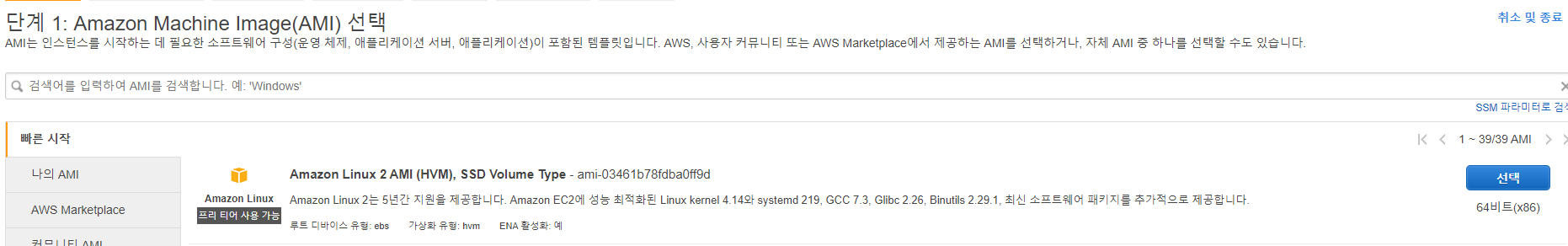
데이터베이스를 생성해준다.

2. EC2 인스턴스 생성

1) Amazon Machine Image 선택

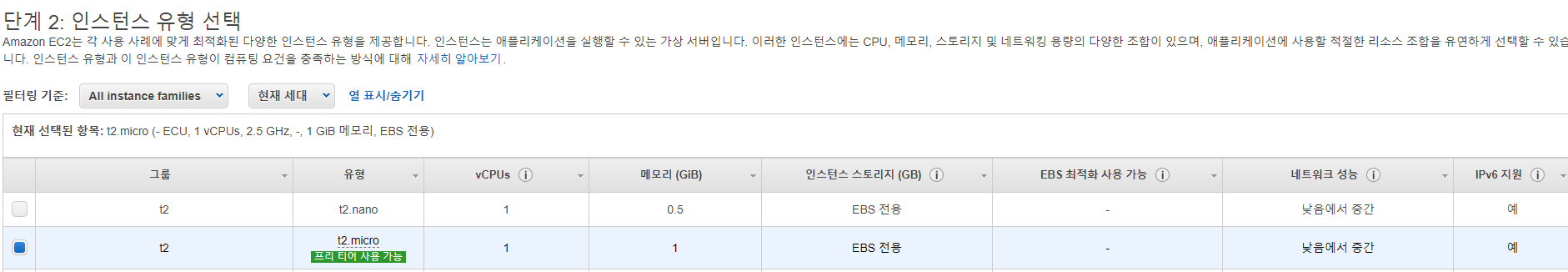


AWS 콘솔에서 Amazon EC2로 이동한다. 인스턴스 시작을 클릭하여 인스턴스 생성 마법사를 연다.



AMI(프리티어)를 선택한다.

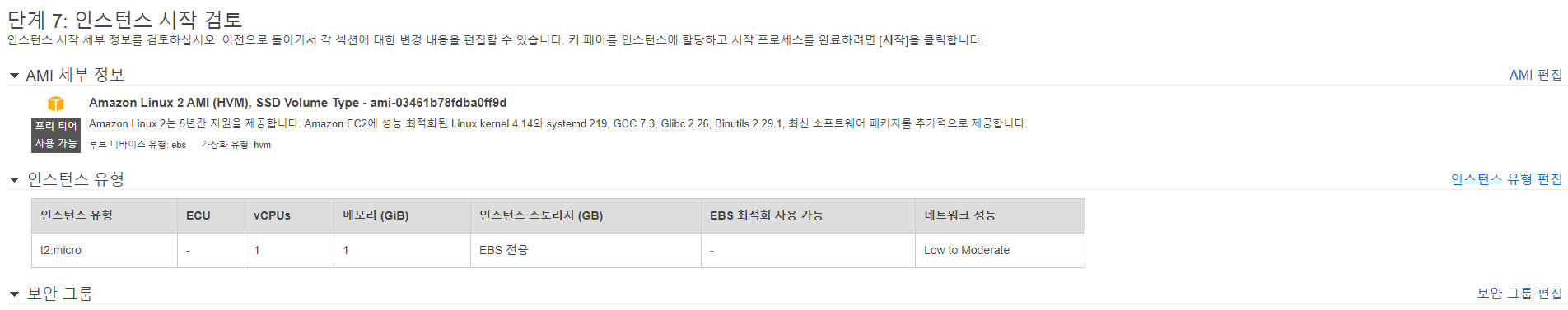
2) 인스턴스 유형 선택



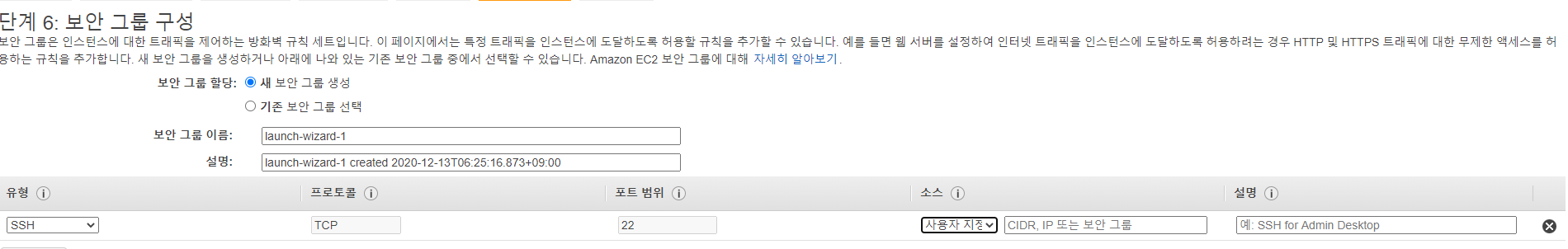
추가 요금이 청구되지 않도록 t2.micro 인스턴스를 선택한다.(매월 750시간 실행 가능)

검토 및 시작 버튼을 클릭하여 몇 가지 구성 단계를 건너뛴다.

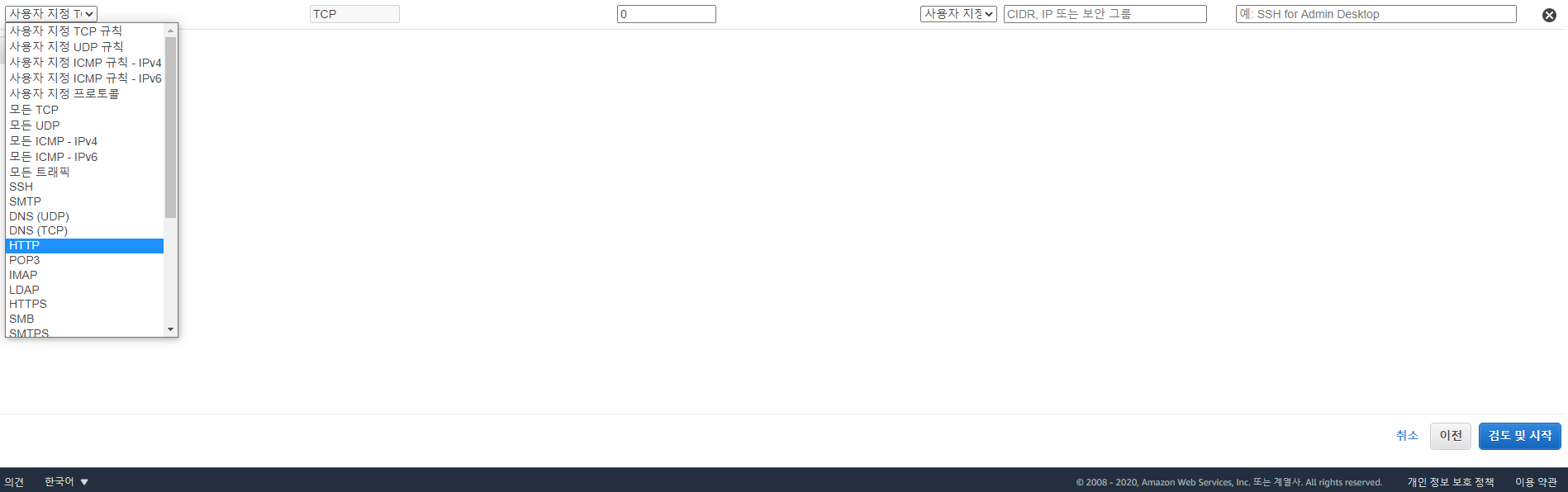
3) 보안 그룹 구성



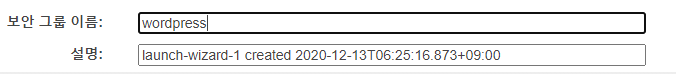
보안 그룹 편집 링크를 클릭한다.



소스를 클릭하여 나의 IP로 변경해준다.

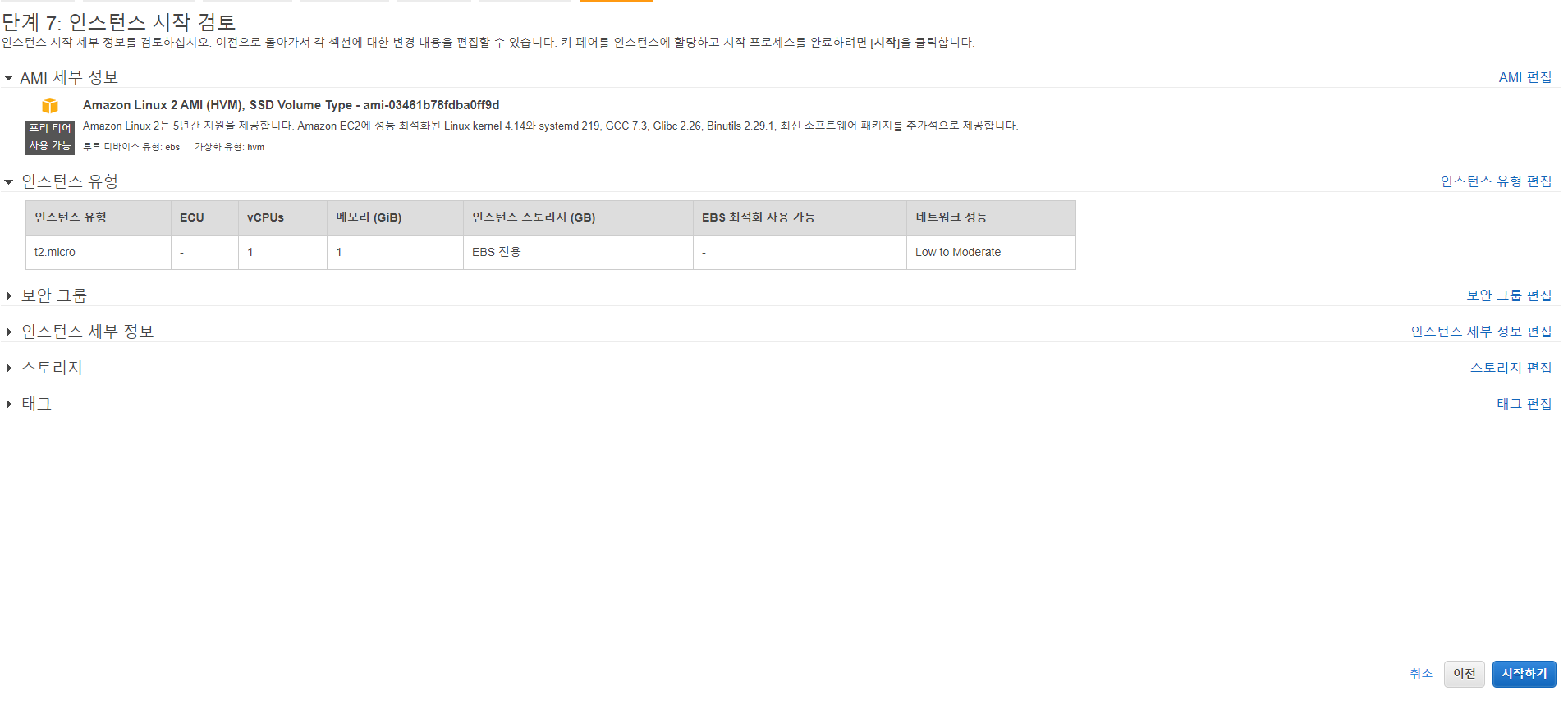


새 규칙을 추가하여 유형을 HTTP로 변경해준다.



보안 그룹 이름을 알기 쉽게 변경해주고, 검토 및 시작 버튼을 클릭한다.

4) 시작 및 SSH 키 받기



시작하기 버튼으로 EC2 인스턴스를 생성한다.



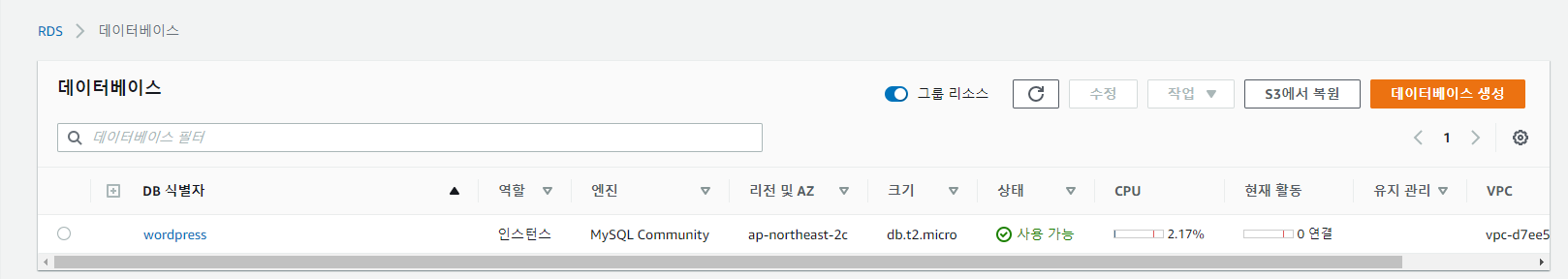
키 페어를 새로 생성해주고, 키 페어 다운로드를 클릭하여 .pem 파일을 다운로드한다.

키 페어가 다운로드 되면 인스턴스 시작 버튼을 클릭하여 EC2 인스턴스를 시작한다.

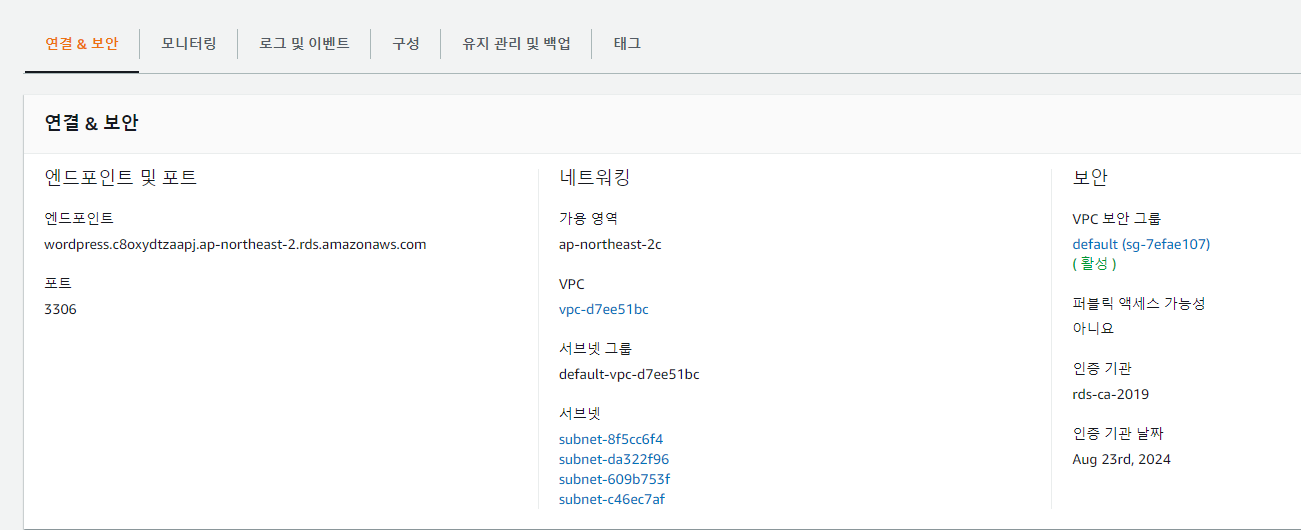
3. RDS 데이터베이스 구성

네트워크 보안과 암호 인증 및 승인을 구성한다.

1) EC2 인스턴스가 RDS 데이터베이스에 액세스하도록 허용



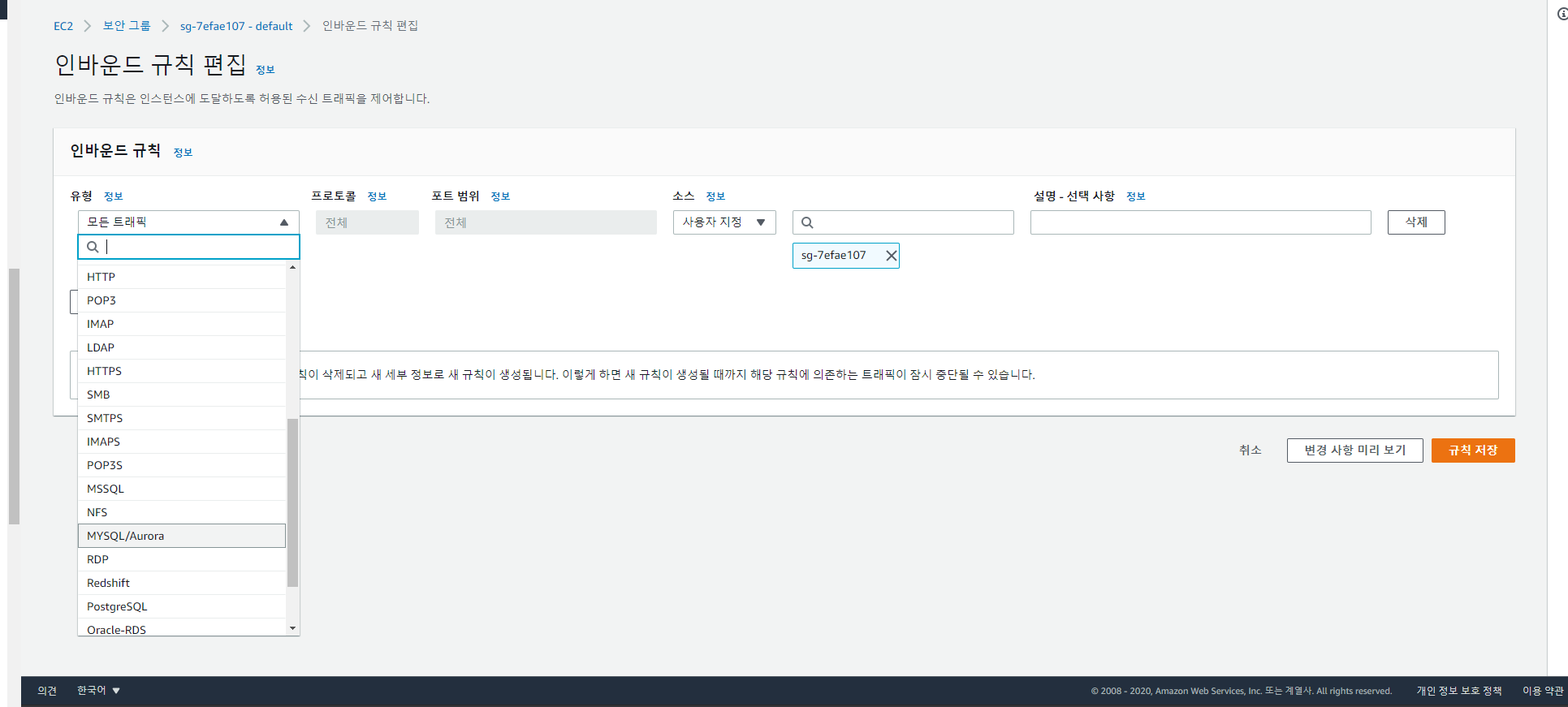
앞서 생성한 MySQL 데이터베이스(wordpress)를 클릭한다.



연결 & 보안 탭에서 VPC 보안 그룹에 나열된 보안 그룹을 클릭한다.

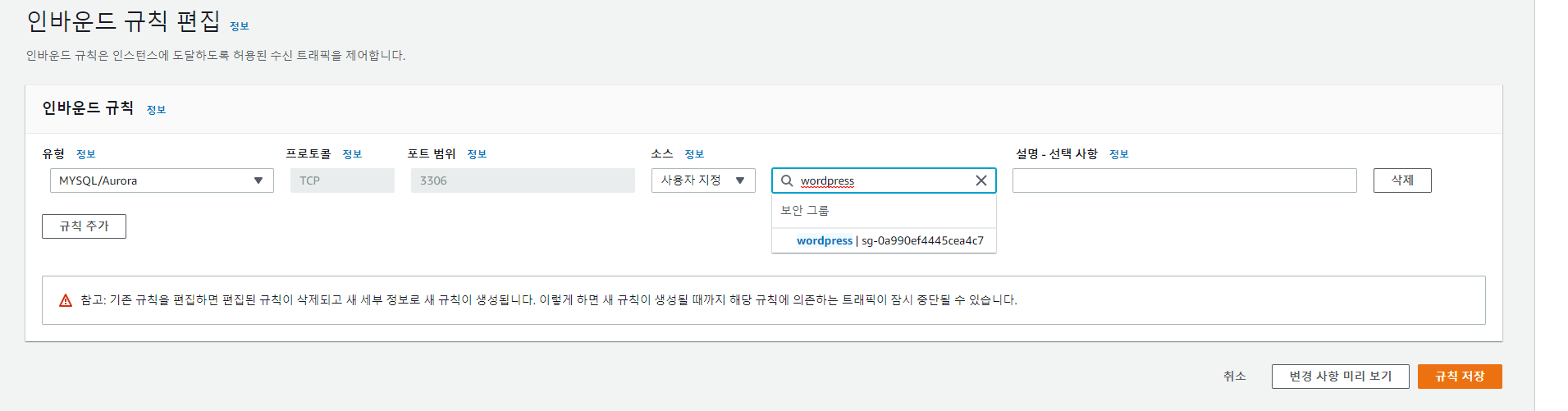


인바운드 탭을 클릭한 후, 편집 버튼을 클릭하여 보안 그룹에 대한 규칙을 변경한다.



기본 보안 그룹은 기본 보안 그룹의 다른 인스턴스에서 발생하는 모든 인바운드 트래픽을 허용하는 규칙이 있다. 그러나 Wordpress EC2 인스턴스는 해당 보안 그룹에 없기 때문에 RDS 데이터베이스에 액세스할 권한이 없다.

유형 속성을 MYSQL/Aurora로 변경하면 프로토콜과 포트 범위가 적절한 값으로 업데이트 된다.



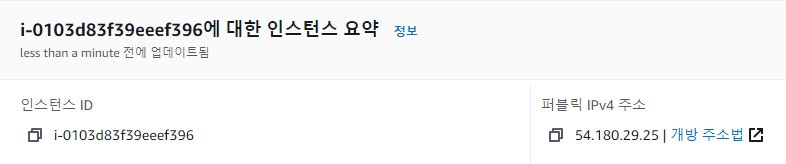
Wordpress를 입력해주고, EC2 인스턴스에서 사용한 wordpress 보안 그룹을 클릭한다.

클릭하면 보안 그룹 ID가 채워진다. 이 규칙을 설정하면 MySQL이 해당 보안 그룹이 구성된 모든 EC2 인스턴스에 액세스할 수 있게 된다. 저장 버튼을 클릭한다.

2) EC2 인스턴스에 SSH 적용

EC2 인스턴스가 RDS 데이터베이스에 액세스할 수 있게 되었으므로 EC2 인스턴스에 SSH를 적용하고 몇 가지 구성 명령을 실행한다.

EC2 인스턴스 페이지로 이동한 후, EC2 인스턴스를 클릭하고, 인스턴스 설명에 IPv4 퍼블릭 IP를 잘 기억해 둔다. SSH를 인스턴스에 적용할 때 필요하다.



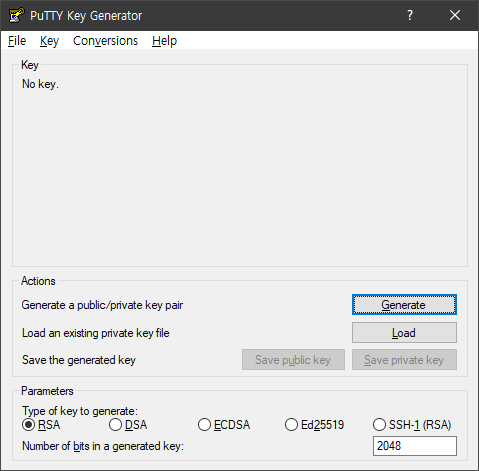
------------------------------------------------------------------------------------------------------------

Windows용 SSH 클라이언트인 PuTTY를 사용하여 EC2 인스턴스에 연결한다. .pem 파일과 퍼블릭 IP 주소가 필요하다.



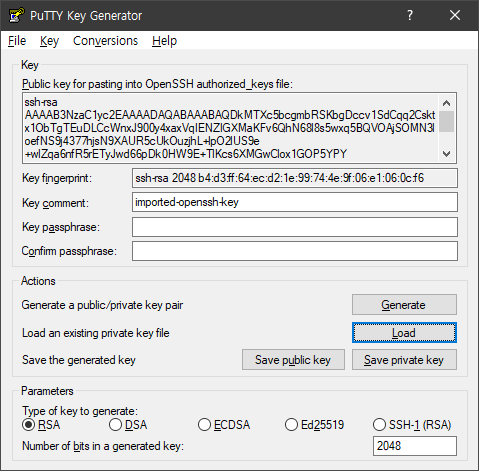
PuTTY를 설치해준다.

1) PuTTYgen을 사용하여 프라이빗 키 변환



PuTTY Gen을 실행한 후, Type of key to generate를 RSA로 설정해준다.

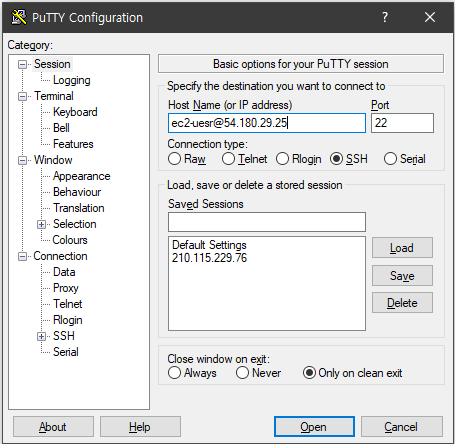
Load를 해준 후, .pem 파일을 Load 해준다.



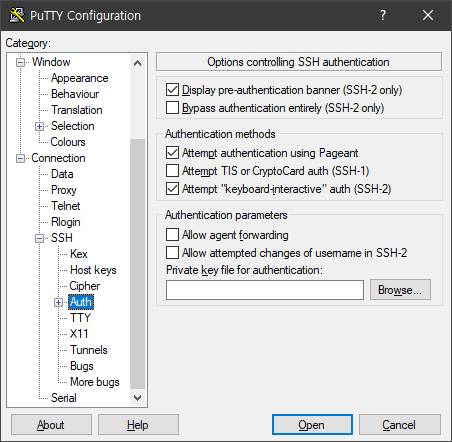
Save private key를 클릭하여 PuTTY에서 사용할 수 있는 형식으로 키를 저장한다.

이제 개인 키가 PuTTY에 사용하기에 올바른 형식으로 되어 있으므로 PuTTY의 SSH 클라이언트를 사용하여 인스턴스에 연결할 수 있다.

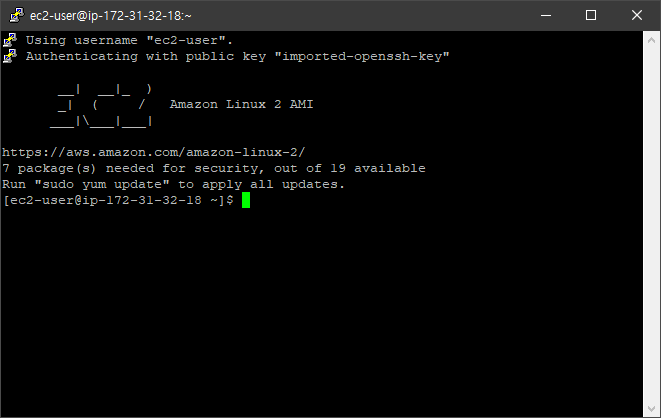
2) Linux 인스턴스에 연결



PuTTY를 실행한 후, 다음과 같이 Host Name을 지정해준다. 사용자이름@IPv4주소

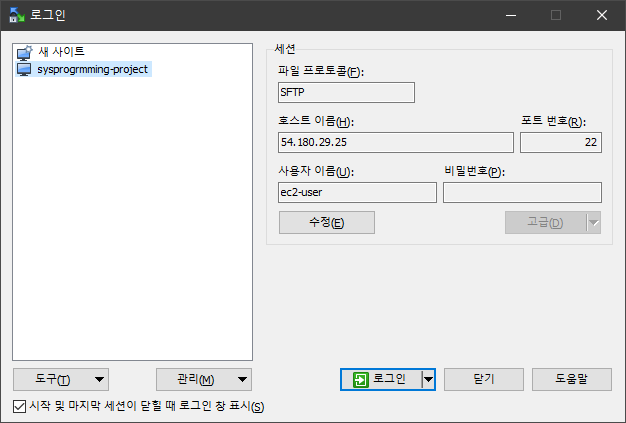


Browse를 클릭하여 앞서 저장한 프라이빗 키 파일(.ppk)를 열고 Open을 클릭한다.

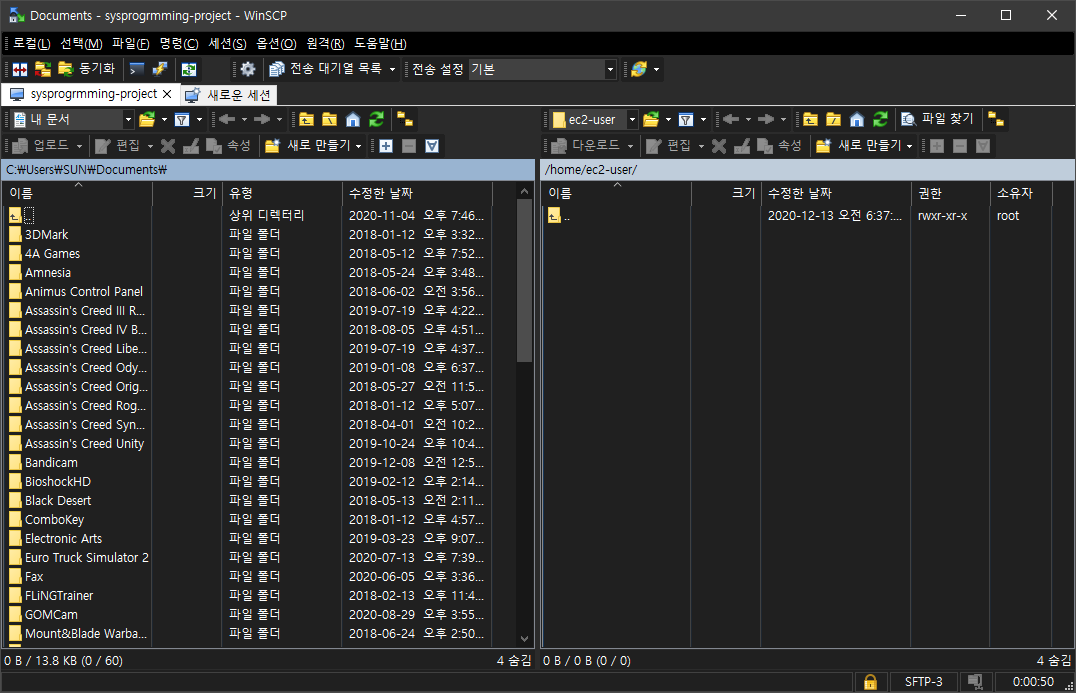


다음과 같이 성공적으로, 접속했다.

3) WinSCP를 사용하여 Linux 인스턴스로 파일 전송



호스트 이름에 퍼블릭 IP를 작성해주고, 사용자 이름은 ec2-user로 작성해준다. 로그인 클릭.

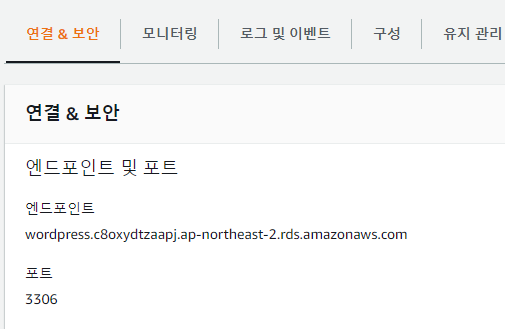


다음과 같이 왼쪽에는 로컬 시스템, 오른쪽은 Linux 인스턴스가 위치함을 알 수 있다.

3) 데이터베이스 사용자 생성

sudo yum install -y mysql

mysql 클라이언트 설치



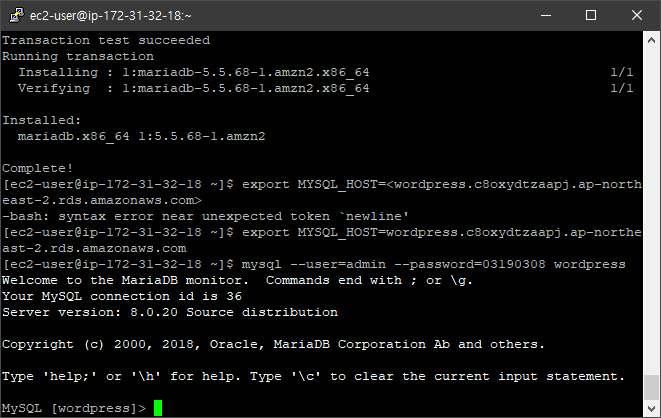
AWS 콘솔에서 RDS 데이터베이스 세부 정보의 연결 및 보안 섹션에서 엔드포인트의 호스트 이름을 찾는다.

export MYSQL\_HOST=<your-endpoint>

터미널에서 다음의 명령어를 입력하여 MySQL 호스트에 대한 환경 변수를 설정한다. <your-endpoint>는 RDS 인스턴스의 호스트 이름으로 대체한다.

mysql --user=<user> --password=<password> wordpress

다음에는 터미널에서 다음의 명령어를 실행하여 MySQL 데이터베이스에 연결한다. “<user>” 및 “<password>”는 RDS 데이터베이스를 생성할 때 구성한 마스터 사용자 이름과 암호로 대체한다.



다음과 같이 성고적으로 MySQL 데이터베이스에 연결된 것을 알 수 있다.

마지막으로 WordPress 애플리케이션의 데이터베이스 사용자를 생성하고 “wordpress” 데이터베이스에 액세스할 권한을 부여한다.

CREATE USER 'wordpress' IDENTIFIED BY 'wordpress-pass';

GRANT ALL PRIVILEGES ON wordpress.\* TO wordpress;

FLUSH PRIVILEGES;

Exit

터미널에서 다음과 같은 명령을 수행한다.

이번에는 RDS 데이터베이스의 네트워크 및 암호 보안을 구성하는 방법을 했다.

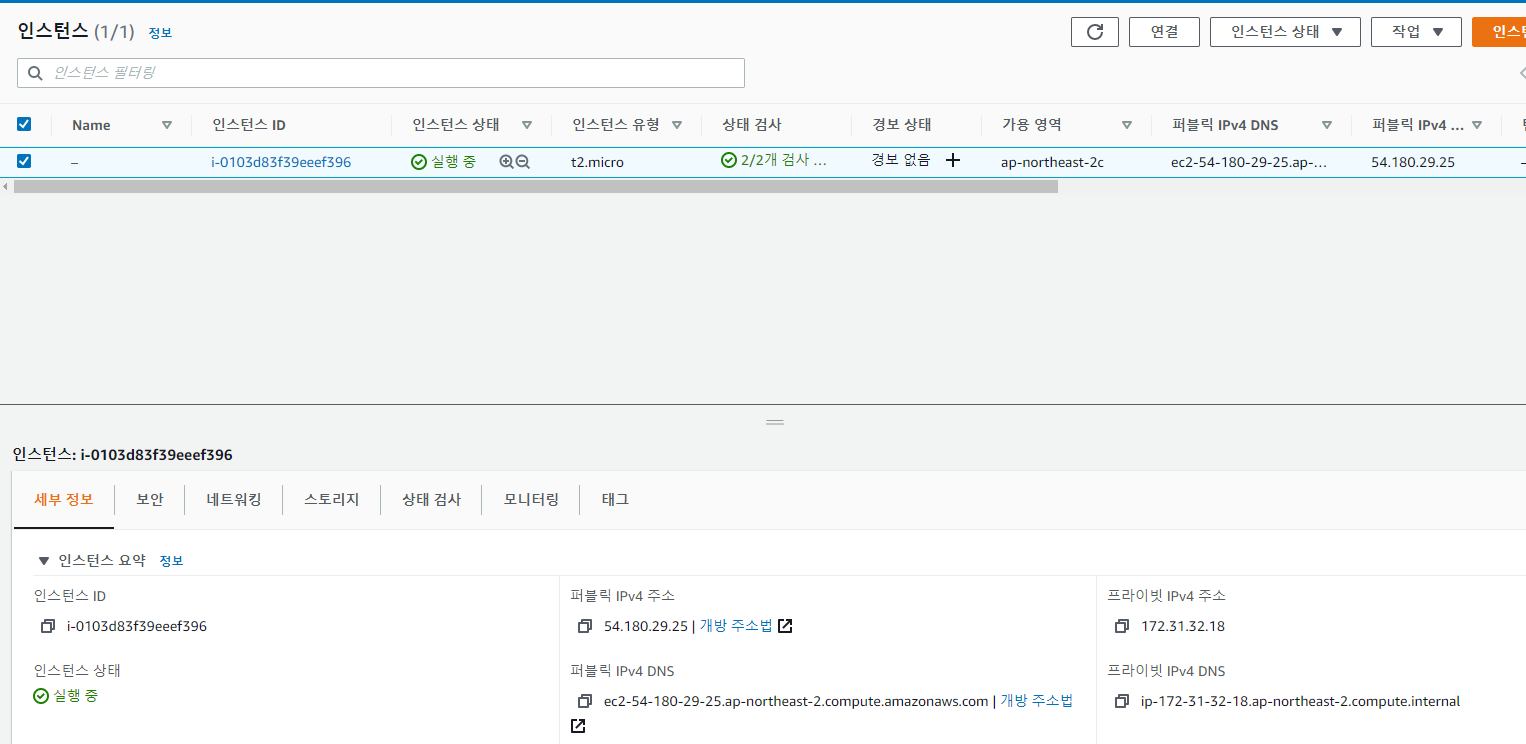
4. EC2에서 WordPress 구성

1) Apache 웹 서버 설치

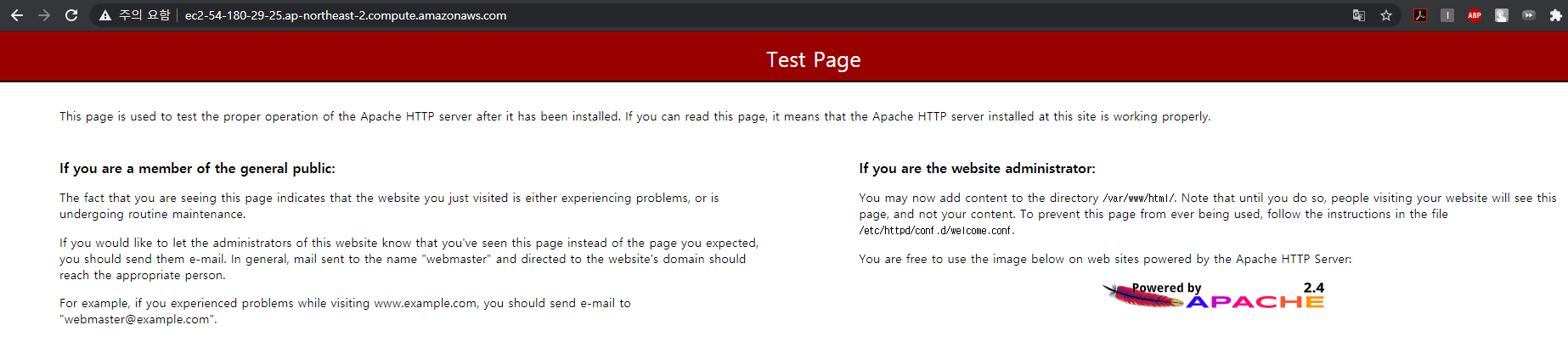
WordPress를 실행하려면 EC2 인스턴스에서 웹 서버를 실행해야 한다.

sudo yum install -y httpd

다음의 명령어를 실행하여 EC2 인스턴스에 Apache를 설치해준다.



인스턴스 페이지의 세부 정보 탭의 퍼블릭 IPv4 DNS를 찾고, 웹 브라우저에 그 값을 입력한다.



다음과 같이 Apache 테스트 페이지가 표시된다.

이제 wordpress를 다운로드하고 구성해야한다.

2) WordPress 다운로드 및 구성

WordPress 소프트웨어를 설치하고 구성을 설정한다.

wget https://wordpress.org/latest.tar.gz

tar -xzf latest.tar.gz

먼저, 터미널에서 다음의 명령어로 소프트웨어를 다운로드하고 압축을 해제한다.



Ls를 입력하면 tar 파일과 압축이 해제된 내용이 들어 있는 wordpress라는 디렉토리가 보인다.

cd wordpress

cp wp-config-sample.php wp-config.php

다음 명령을 사용하여 디렉토리를 wordpress로 변경하고 기본 구성 파일의 사본을 생성한다.

nano wp-config.php

다음 명령을 실행하여 nano 편집기로 wp-config.php 파일을 연다.

다음 행을 변경하여 데이터베이스 구성을 편집한다.

// \*\* MySQL settings - You can get this info from your web host \*\* //

/\*\* The name of the database for WordPress \*/

define( 'DB\_NAME', 'database\_name\_here' );

/\*\* MySQL database username \*/

define( 'DB\_USER', 'username\_here' );

/\*\* MySQL database password \*/

define( 'DB\_PASSWORD', 'password\_here' );

/\*\* MySQL hostname \*/

define( 'DB\_HOST', 'localhost' );

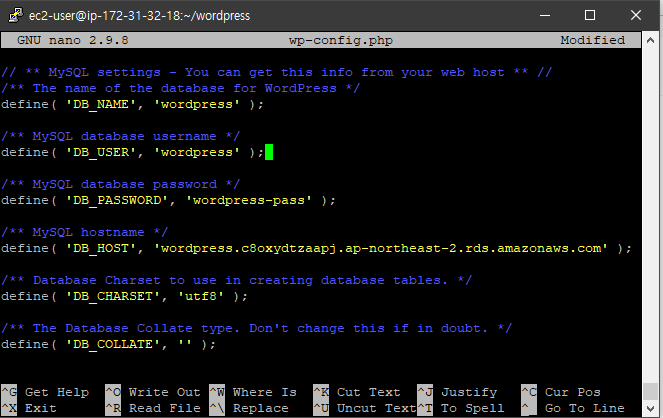
값은 다음과 같이 설정해야 한다.

DB\_NAME: “wordpress”

DB\_USER: 이전 모듈에서 데이터베이스에 생성한 사용자 이름

DB\_PASSWORD: 이전 모듈에서 생성한 사용자의 암호

DB\_HOST: 이전 모듈에서 찾은 데이터베이스 호스트 이름



Ctrl-O를 입력하고 Ctrl-X를 입력하여 nano 편집기에서 빠져나온다.

3) WordPress 배포

이 단계에서는 Apache 웹 서버가 Wordpress에 대한 요청을 처리하도록 설정한다.

첫째, Wordpress에 필요한 애플리케이션 종속성을 설치한다.

sudo amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2

둘째, 다음 명령을 실행하여 올바른 디렉토리로 변경한다.

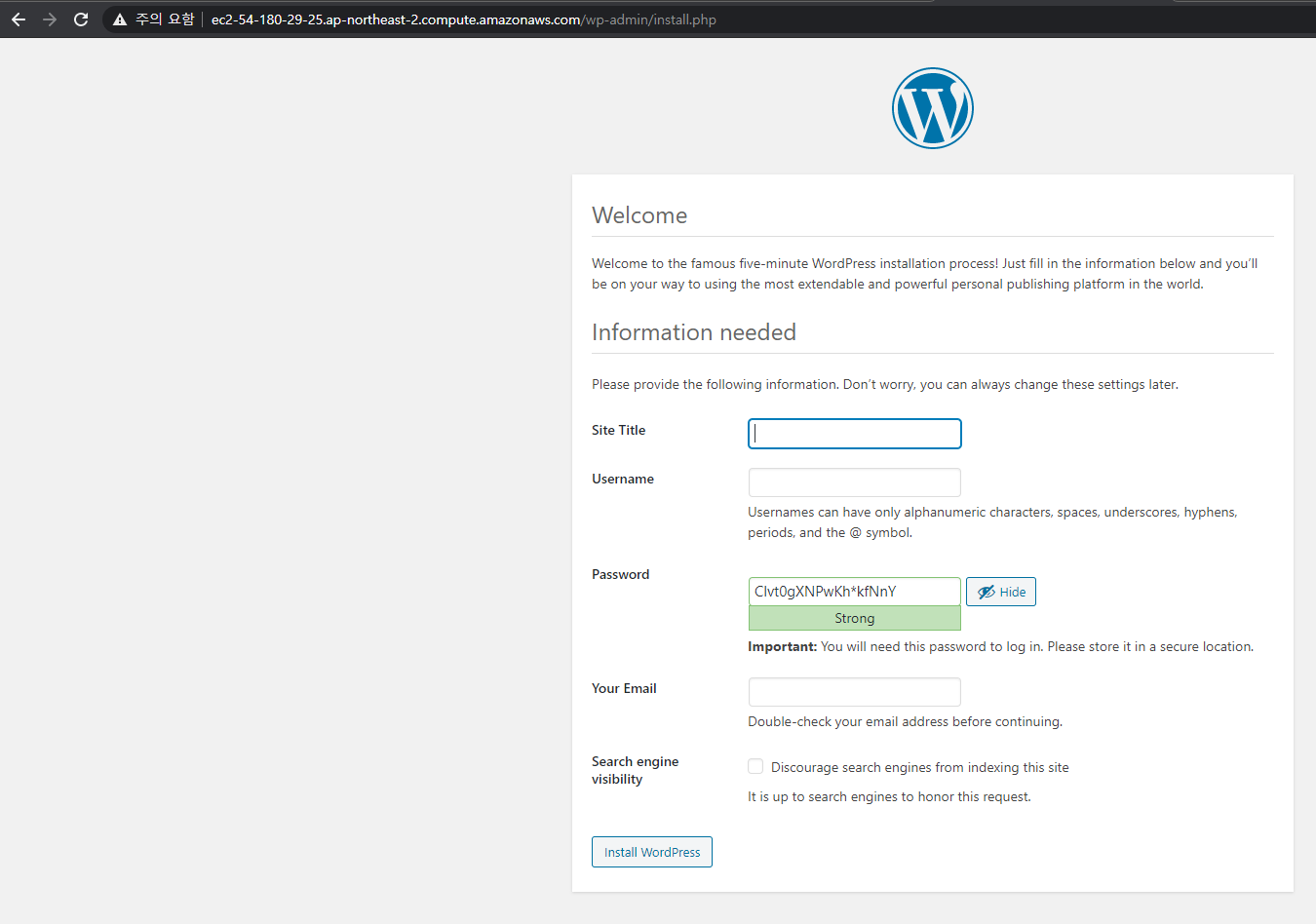
cd /home/ec2-user

셋째, Wordpress 애플리케이션 파일을 Apache에서 사용하는 /var/www/html 디렉토리에 복사한다.

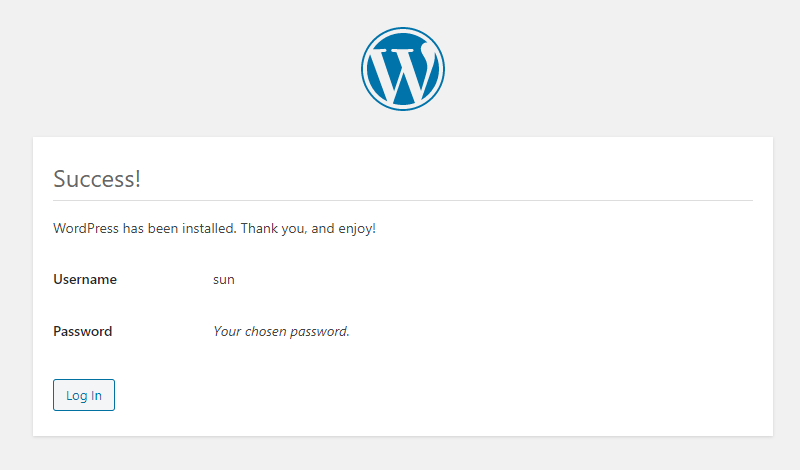
sudo cp -r wordpress/\* /var/www/html/

마지막으로, 변경 사항이 적용되도록 Apache 웹 서버를 다시 시작한다.

sudo service httpd restart



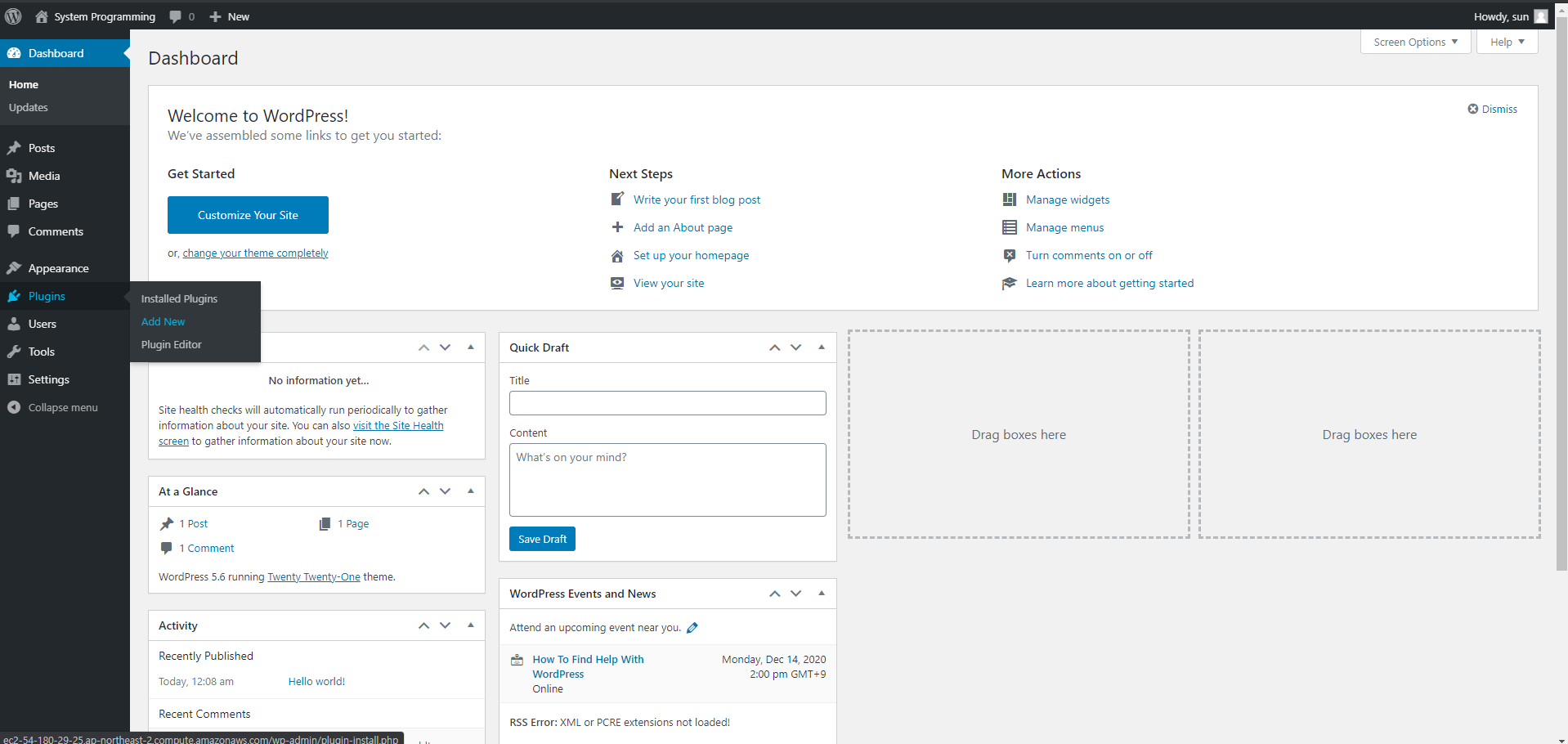
이전에 띄운 아파치 웹사이트를 새로 고침하면 다음과 같이 Wordpress 시작 페이지가 표시된다.

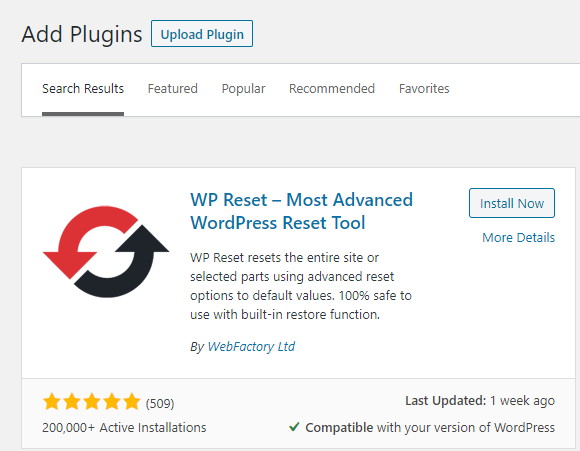


이를 통해 Amazon RDS 기반의 완전관리형 MySQL 데이터베이스를 사용하며 퍼블릭 액세스가 가능한 WordPress 사이트가 시작되었다.

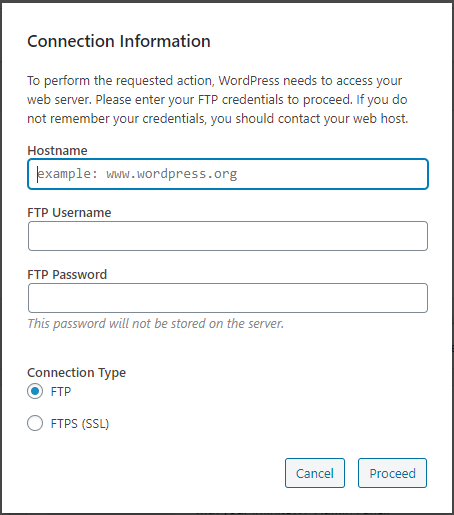
5. 본격적으로 블로그 꾸미기

먼저, Plugins의 Add new를 클릭한다.





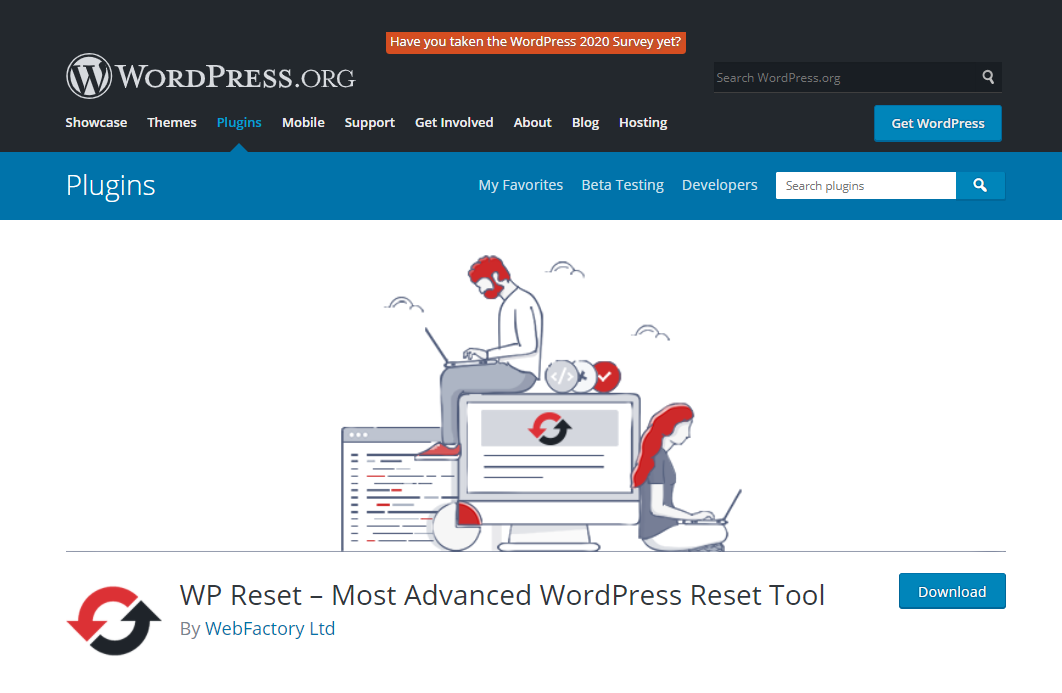
Reset을 검색하여 WP Reset 플러그인을 install 한다.



Install 하려고 하면 다음과 같은 팝업이 뜨는데 구글링을 해서 wordpress-config.php 파일을 아무리 수정해봐도 해결이 되지 않았다.

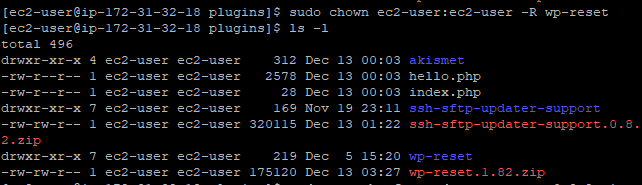
그래서 수동으로 플러그인을 설치하기로 했다.

wp-reset를 검색한 후, more details로 들어가서 홈페이지에서 직접 다운받아서 winscp를 사용하여 로컬에서 ec2 인스턴스의 /var/www/html/wp-content/plugins 에 붙여 넣었다.

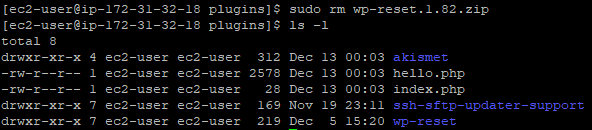


그 후, 다음의 명령어를 통해 압축을 해제하였다.

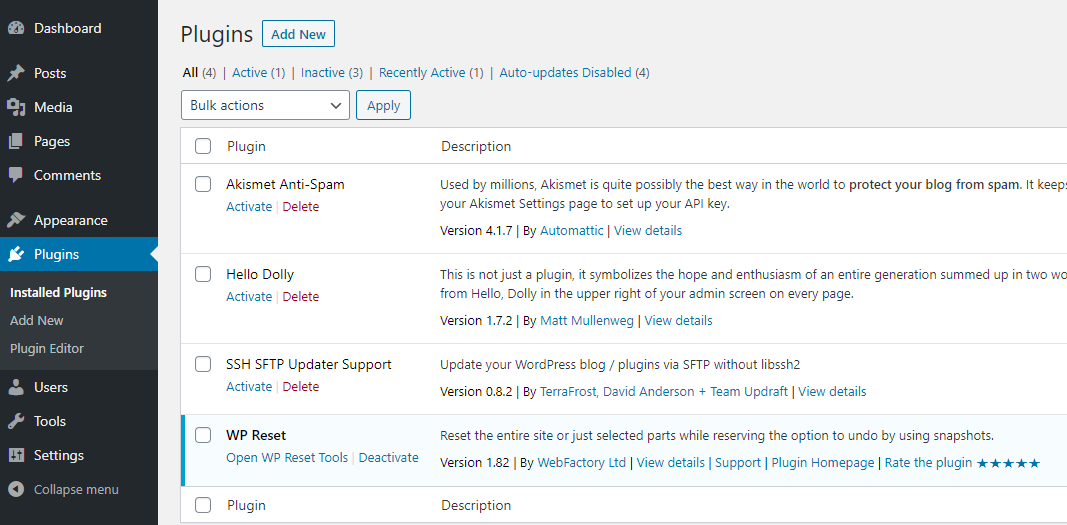




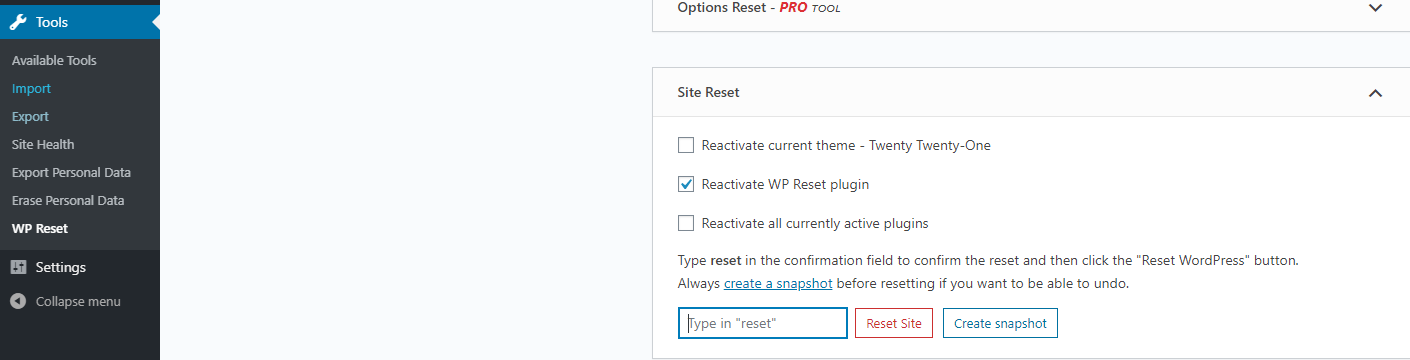
압축이 풀린 wp-reset 플러그인을 하위 디렉토리 및 파일을 포함하여 소유자를 ec2-user로 변경해 준다.



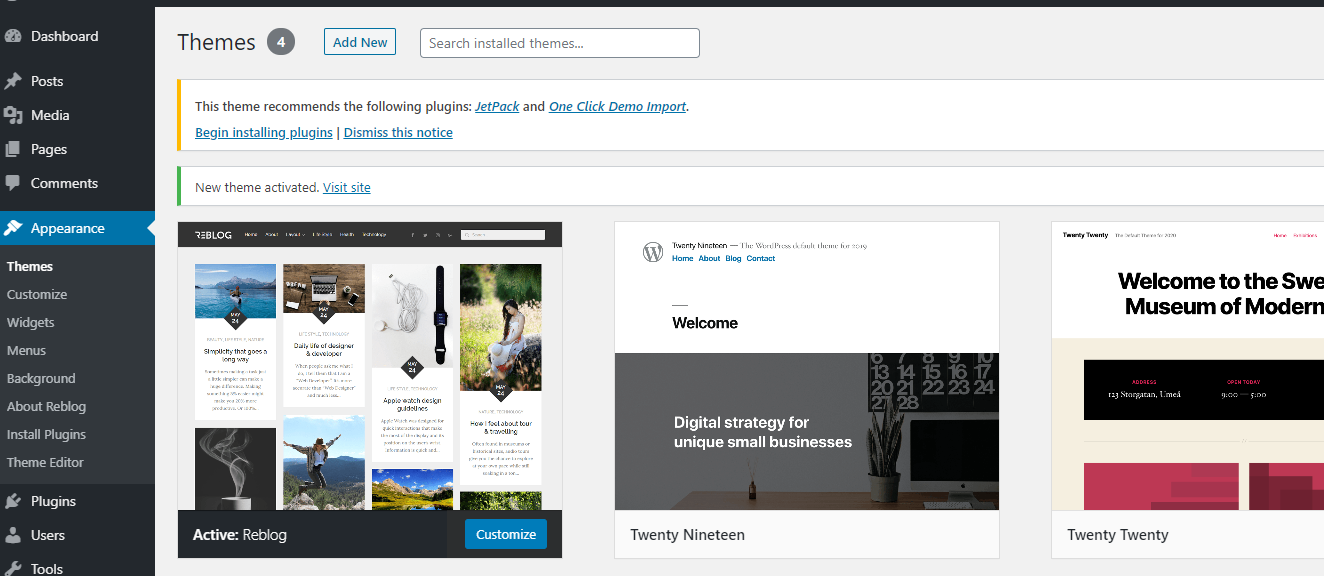
복사한 압축 파일은 삭제한다.



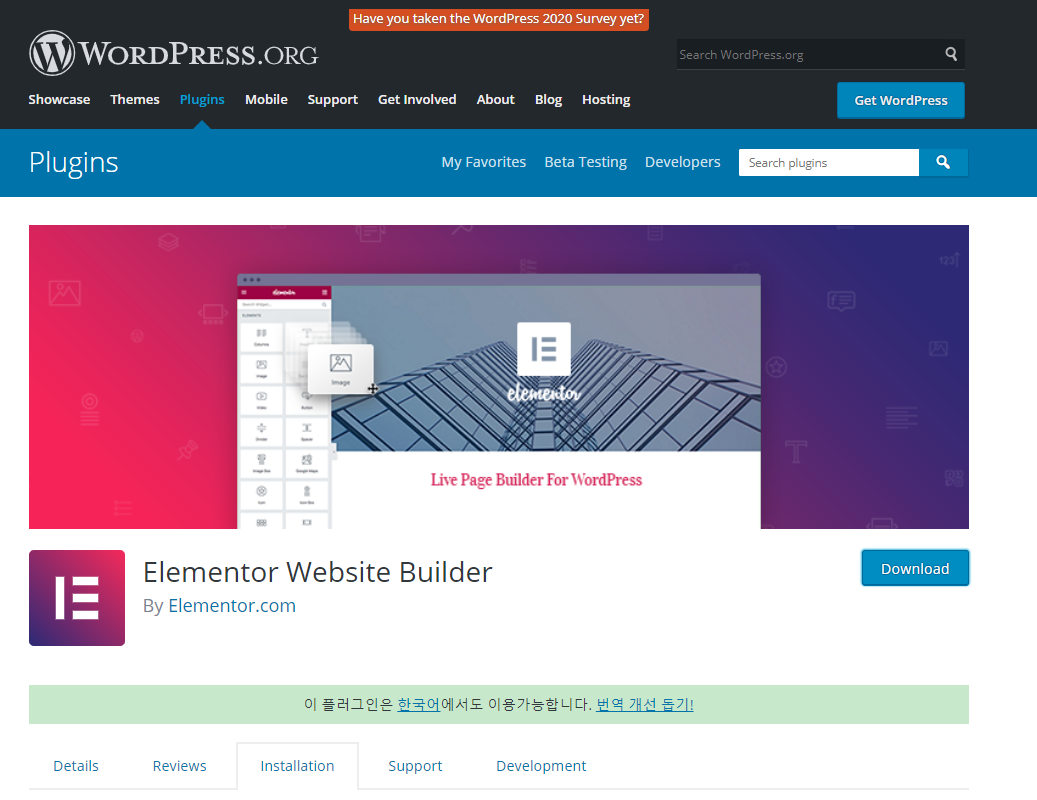
대시보드로 돌아와서 installed Plugins에 WP Reset이 있음을 확인하고 Active를 한다.

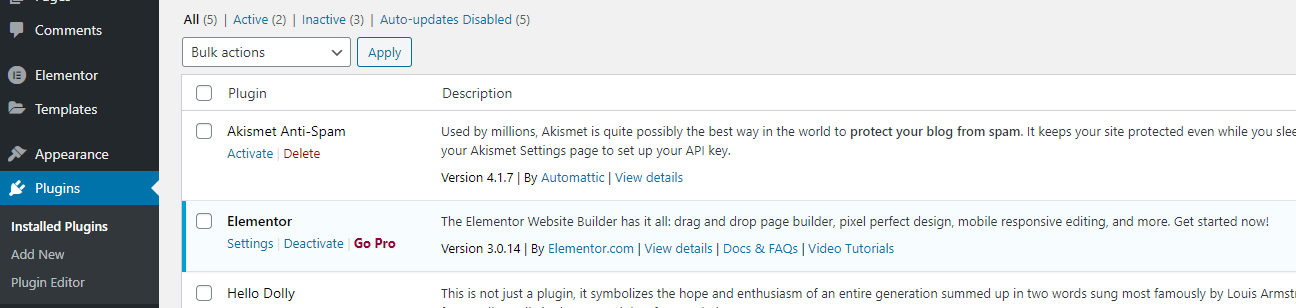


Tools의 WP Reset을 클릭하고 쭉 내려와서 Site Reset 탭에 reset을 입력하고 Reset Site를 클릭한다.

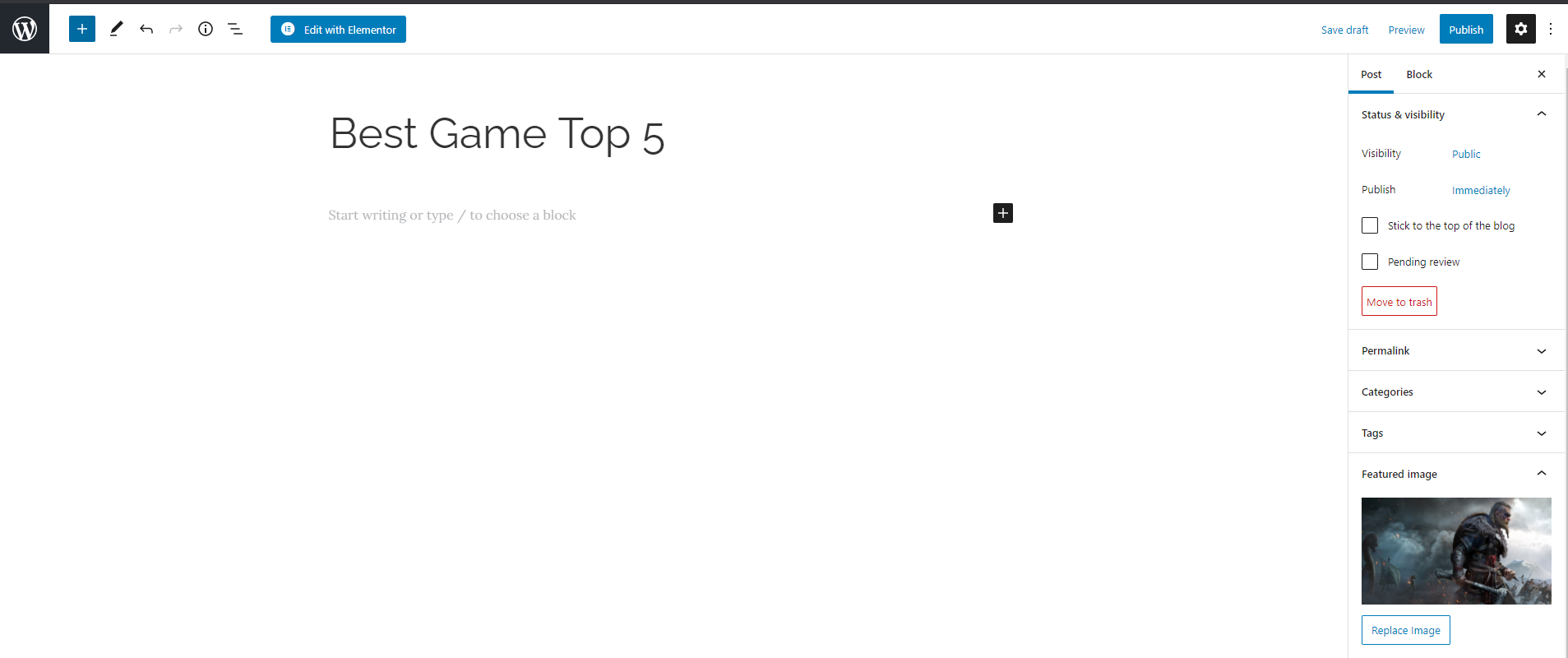


플러그인 설치와 마찬가지로 Reblog 테마를 다운받아 Active 해준다.

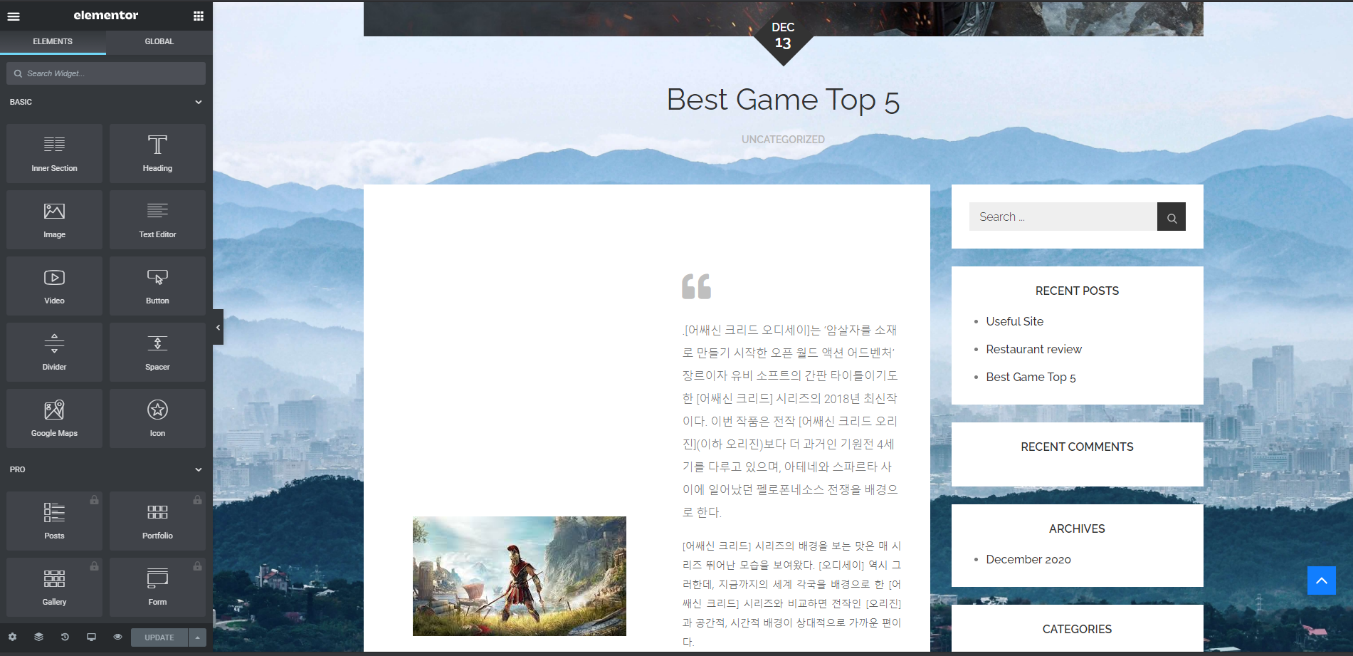




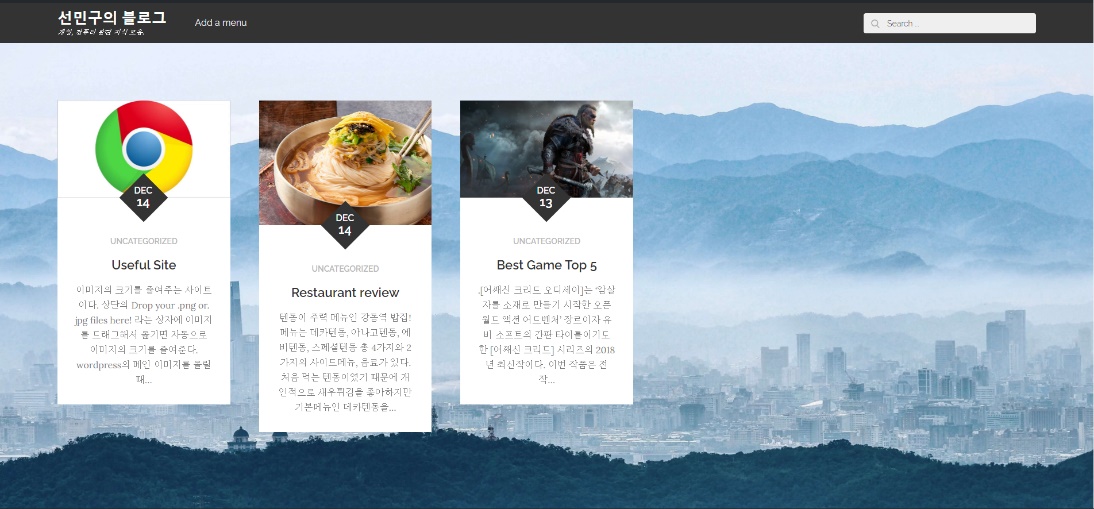
마찬가지로 Elementor 플러그인도 설치해준다.



글을 새로 추가하여 포스트의 제목을 쓰고, Featured Image로 배경 이미지를 넣어주고, 상단의 Edit with Elementor를 클릭한다.



원하는 템플릿을 설정해주고, 이미지와 글 등의 내용을 작성한다.



Publish로 저장을 하고, 사이트에 재접속하면 다음과 같은 메인 화면이 나온다. 이로써 AWS로 서버를 개설하여 설치형 wordpress로 블로그를 만들어보았다.