

### 3. 클라우드 가상서버 EC2

#### 3강. EC2 인스턴스 실습

##### 학습목표

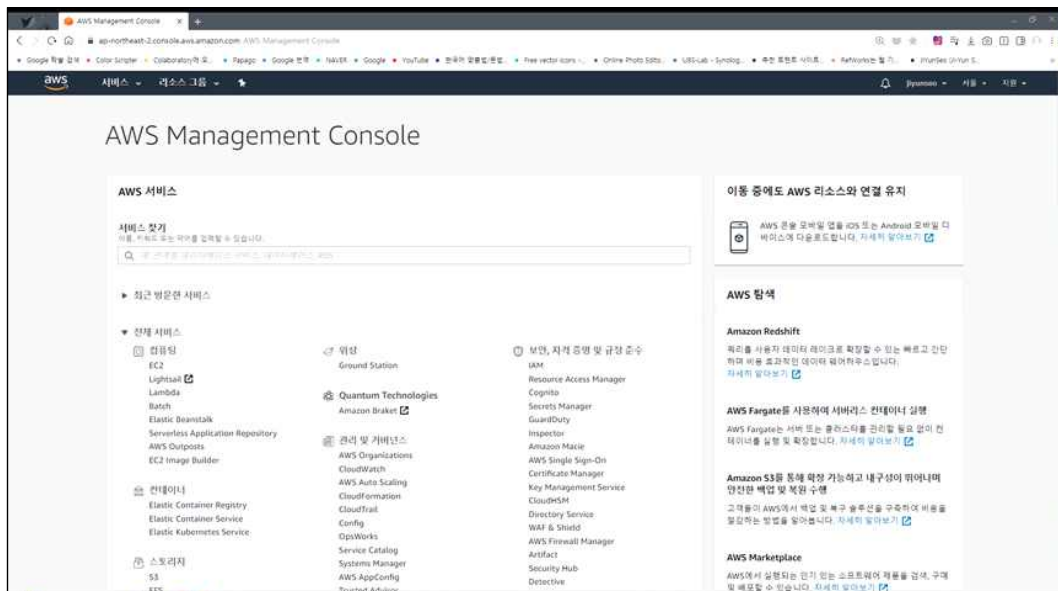
- EC2 인스턴스를 생성하고 원격 접속할 수 있다.
- EC2 인스턴스에 주피터 노트북을 설치 할 수 있다.

##### 학습내용

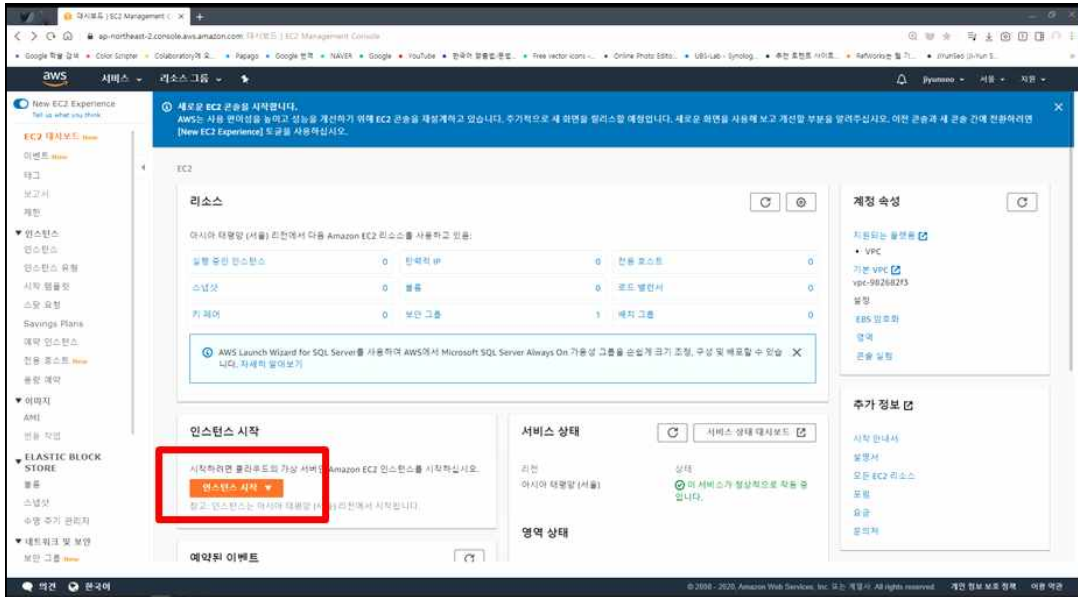
- EC2 인스턴스 생성하기
- EC2 인스턴스에 원격 접속하기
- 주피터 노트북 설치 및 설정

#### 실습 1. EC2 인스턴스 생성하기

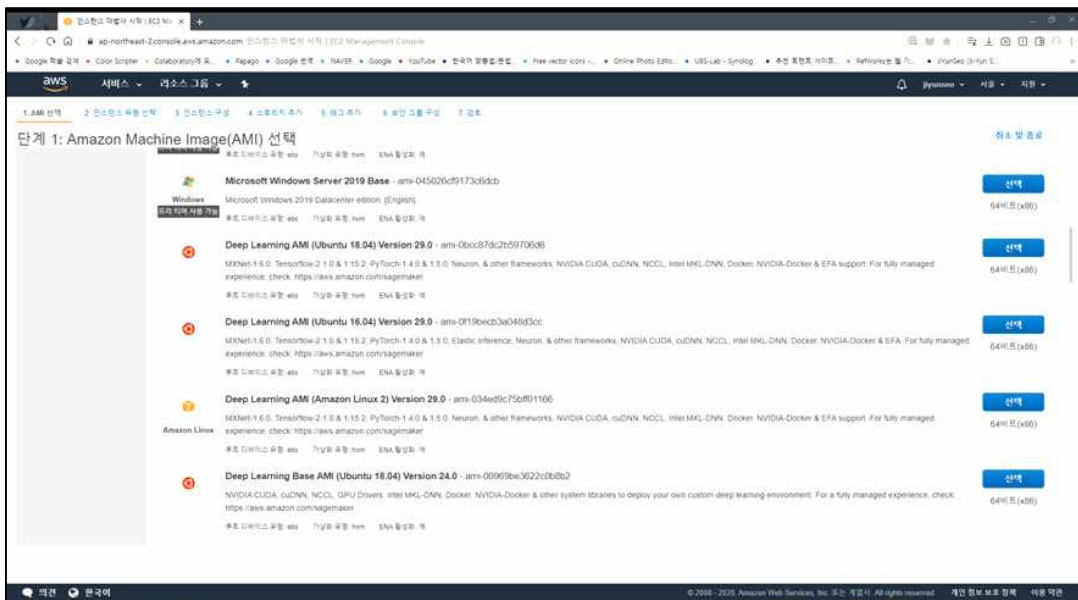
- AWS 홈페이지 접속 후 EC2 솔루션 접속



- AWS 홈페이지 접속 후 EC2로 이동 후 인스턴스 시작 클릭

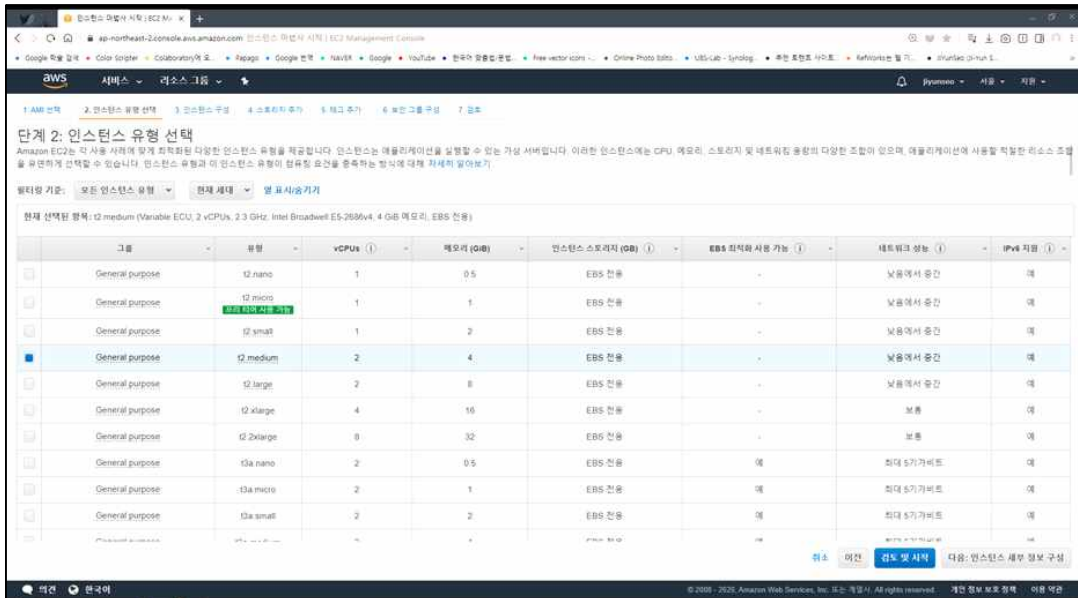


- AMI 선택 - Deep Learning AMI(Ubuntu)

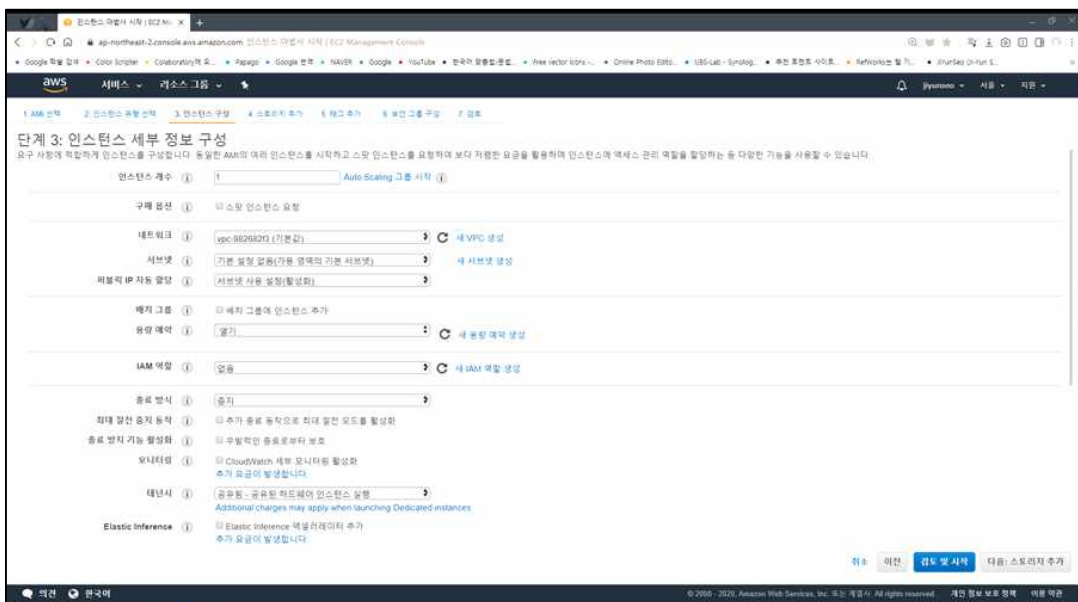


- 인스턴스 유형 선택

- ✓ CPU/메모리가 클 수록 속도적에서 차이가 있음
- ✓ 사양이 좋을수록 비싼 요금이 청구



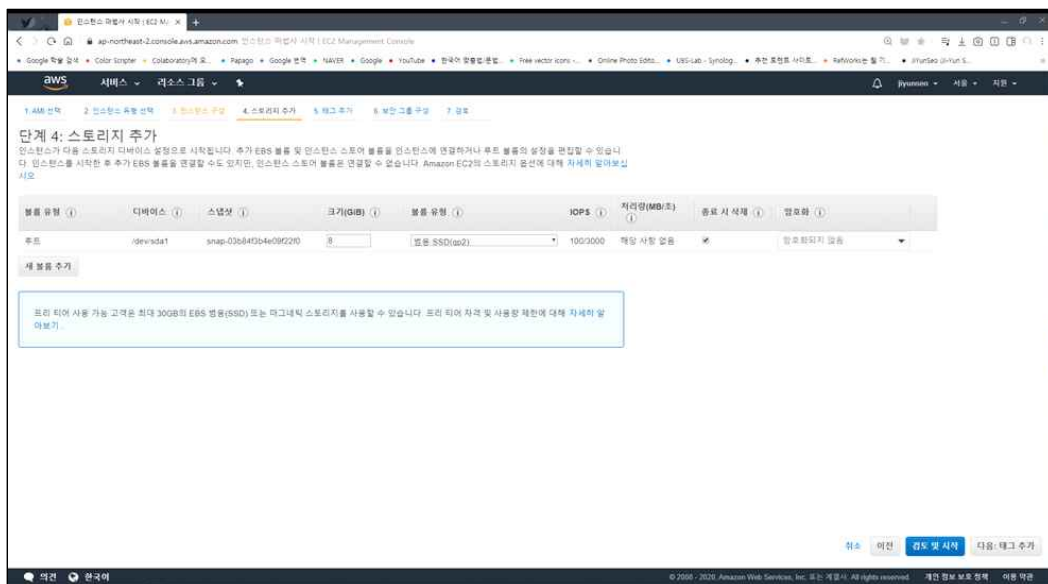
- 인스턴스 세부 정보 구성



- ✓ 인스턴스 수 - 1(기본값)
- ✓ 구매 옵션 - 스팟 인스턴스 요청 선택 취소(기본값). 스팟 인스턴스에는 프리 티어 옵션이 적용되지 않음
- ✓ 네트워크 - 기존 VPC를 선택하거나 새 VPC를 생성하여 인스턴스를 시작

할 위치를 결정

- ✓ 서브넷 - 키 페어를 생성한 것과 동일한 리전에서 서브넷을 선택
  - ✓ 퍼블릭 IP 자동 할당 - Enable(활성화)을 선택
  - ✓ IAM 역할 - 없음(기본값)
  - ✓ 종료 방식 - 중지(기본값)
  - ✓ 종료 방지 기능 활성화 - 실수로 종료되지 않도록 보호를 선택
  - ✓ 모니터링 - CloudWatch 세부 모니터링 활성화 선택 취소(기본값)
  - ✓ 테넌시 - 공유 테넌시(다중 테넌트 하드웨어)(기본값). 전용 서버에는 프리 티어 옵션이 적용되지 않음
- 스토리지 추가
    - ✓ 인스턴스의 저장 공간을 정할 수 있음

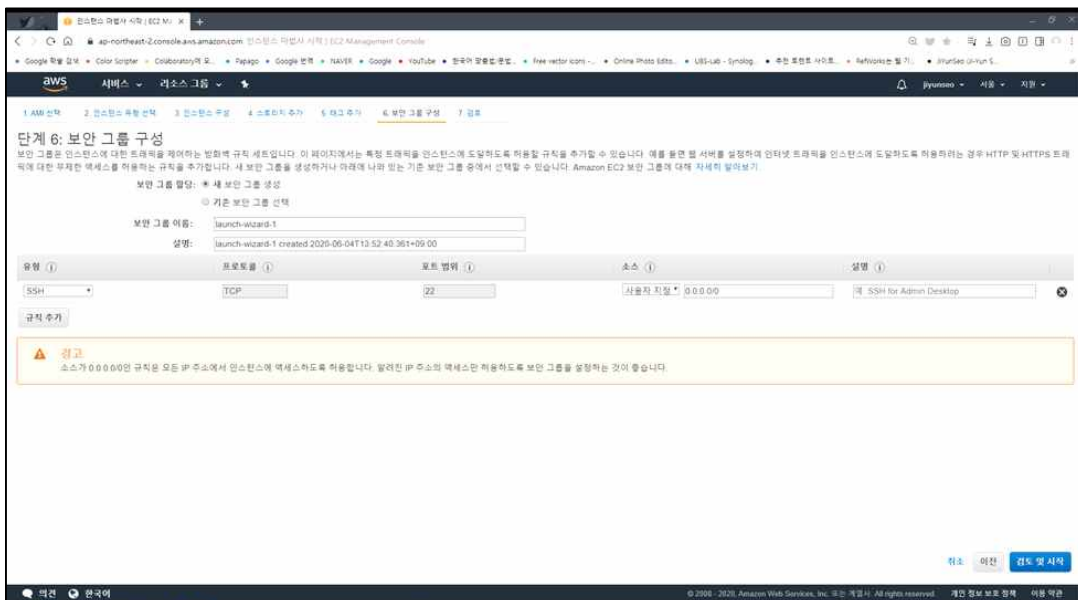


- 태그 추가

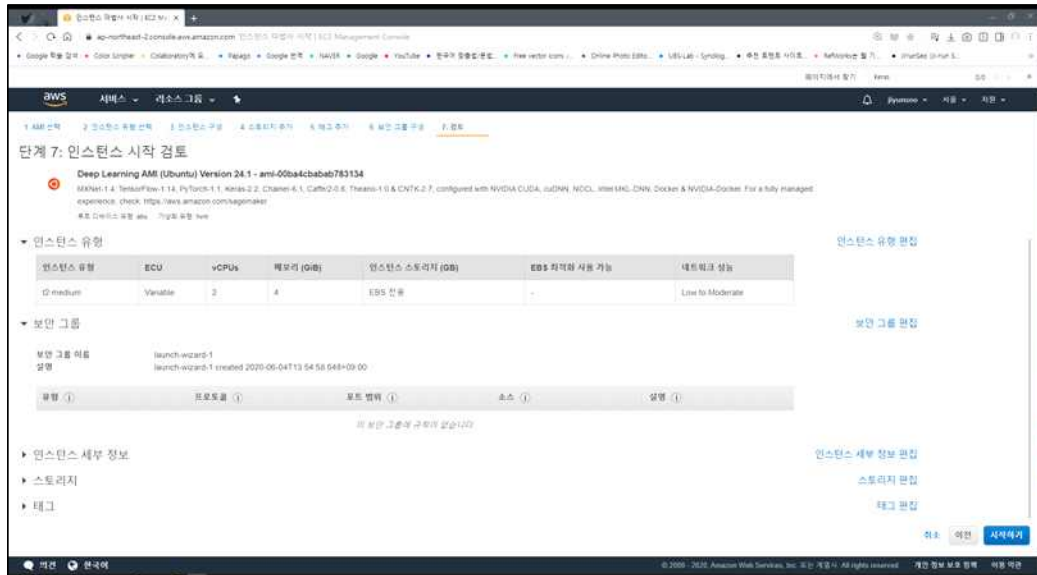


- 보안 그룹 구성

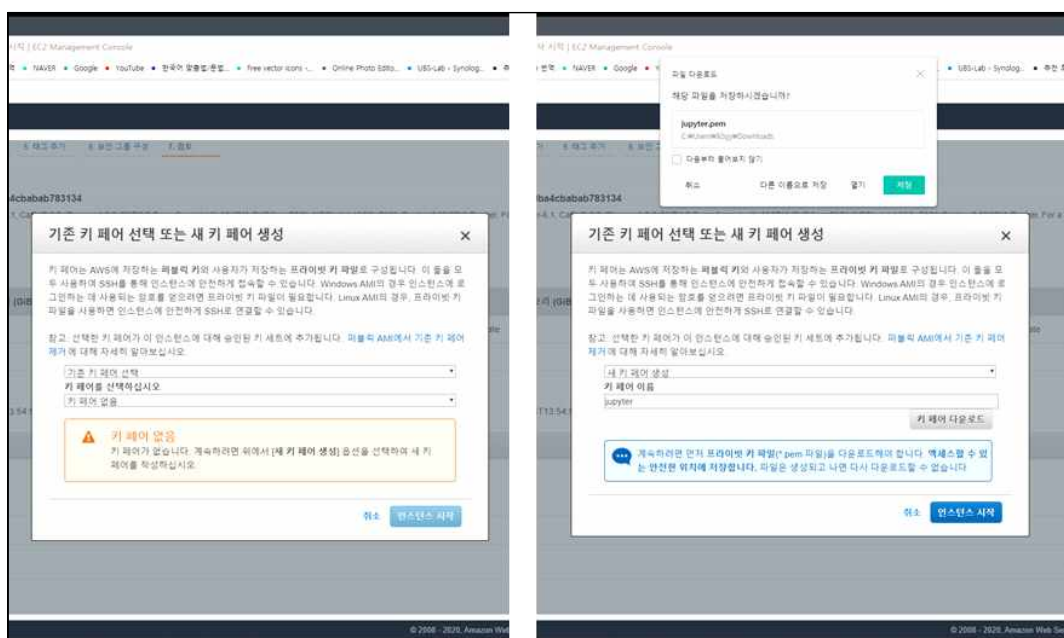
- ✓ 보안 그룹 구성에서는 그룹 이름과 설명을 나중에 알아보기 쉽게 바꿔주고 보안 규칙을 설정할 수 있음



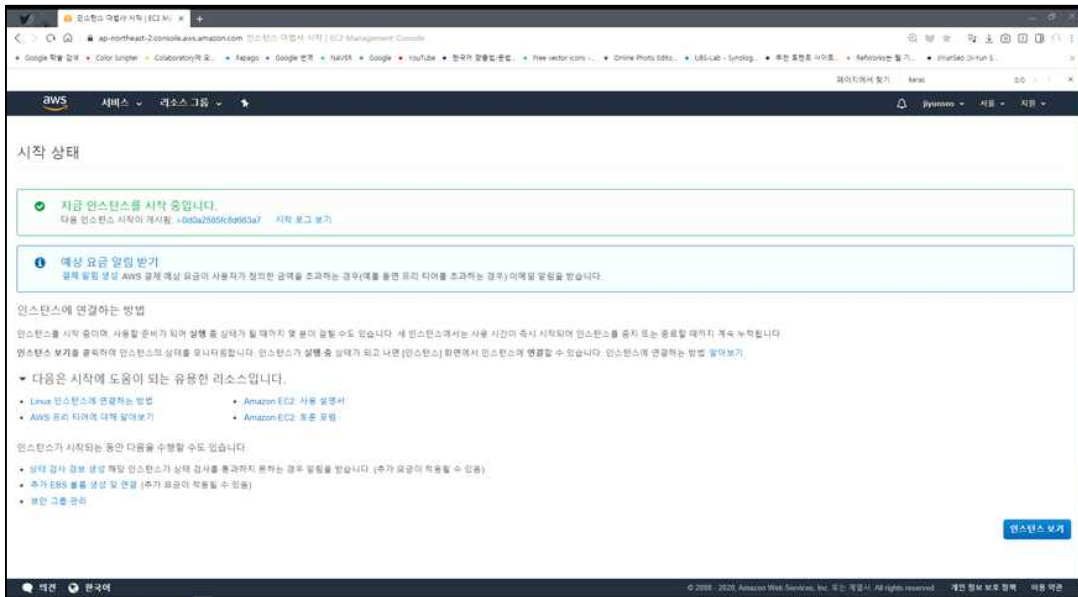
- 인스턴스 시작 검토
  - ✓ 지금까지 설정해온 인스턴스 최종 확인
  - ✓ 확인 후 시작하기 클릭



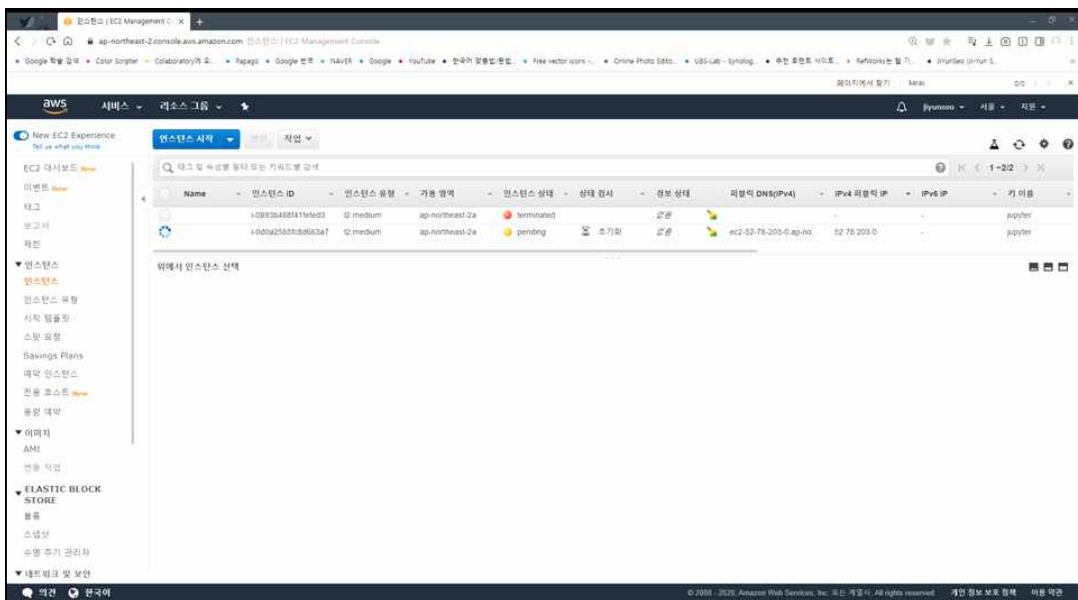
- 키 페어 생성
  - ✓ 키 페어는 나중에 원격으로 이 인스턴스에 접속하기 위해서 반드시 필요
  - ✓ 새 키 페어 생성을 누른 후 키 페어 이름을 작성하고 키 페어 다운로드 다운로드 해주며, 키 페어는 한번 발급 받은 후, 다시 받을 방법이 없으므로 반드시 잘 보관해야함



- 예상 요금 알람 받기
  - ✓ 인스턴스가 생성되는 동안, 예상 요금을 이메일로 알림 받을 수 있도록 세팅이 가능



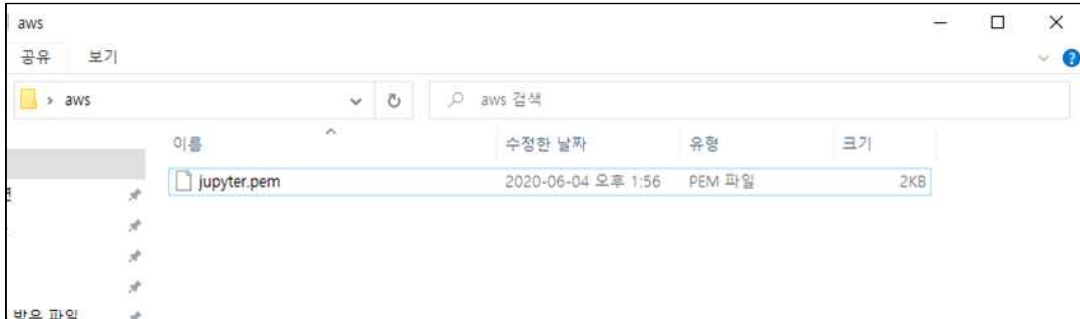
- EC2 인스턴스 생성완료



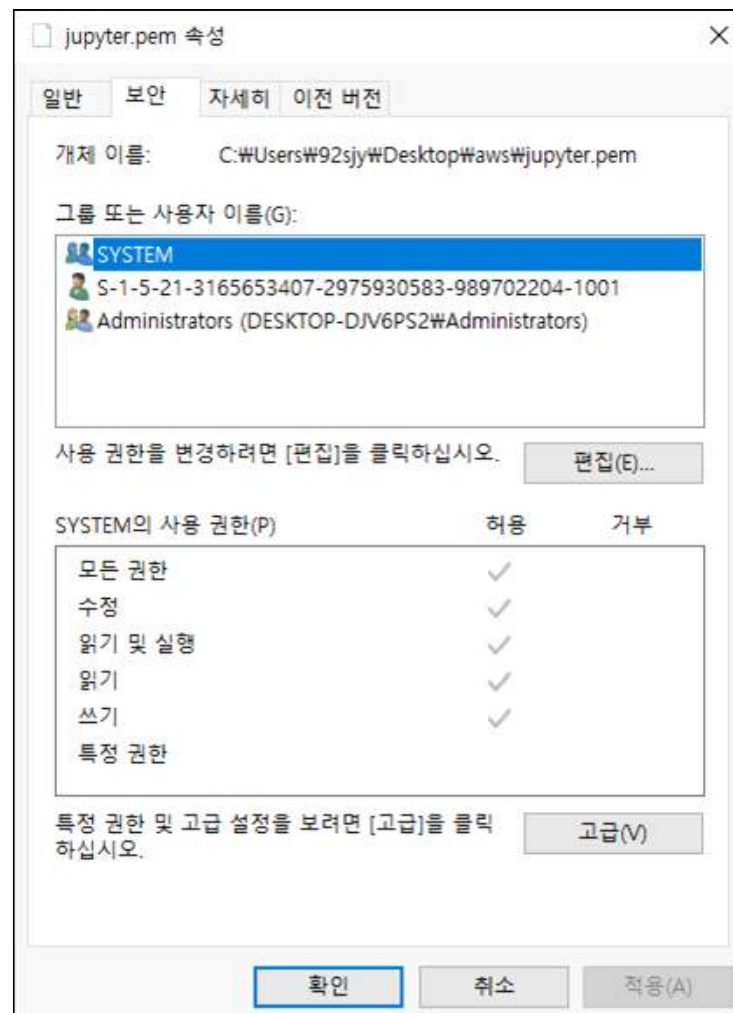


## 실습 2. EC2 인스턴스 실행하기

- 인스턴스 실행을 위하여 키페어 권한 설정

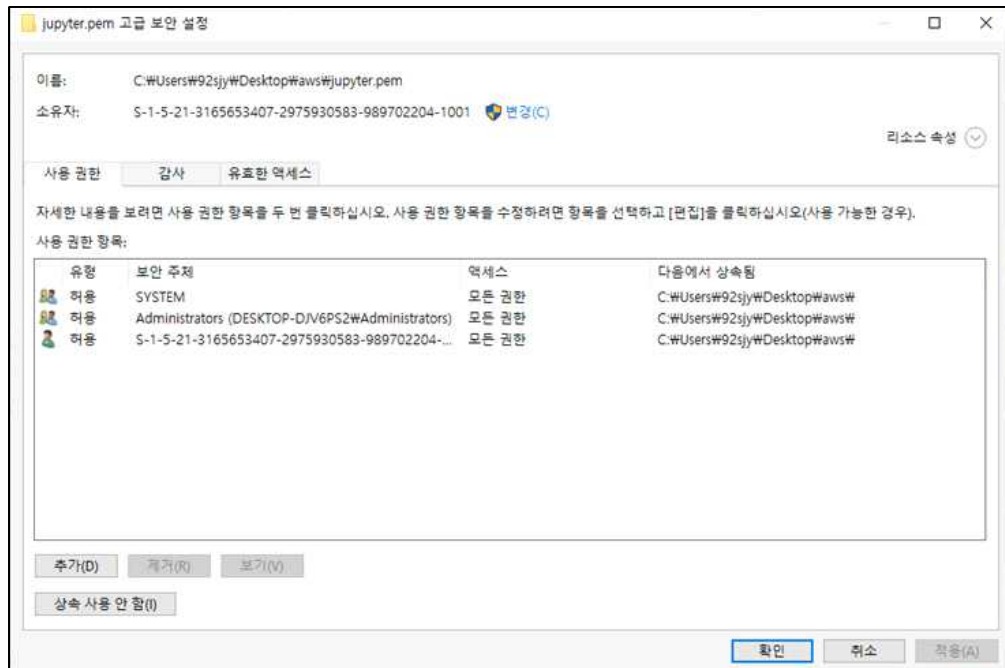


- 키 페어 속성 → 보안 → 고급 클릭

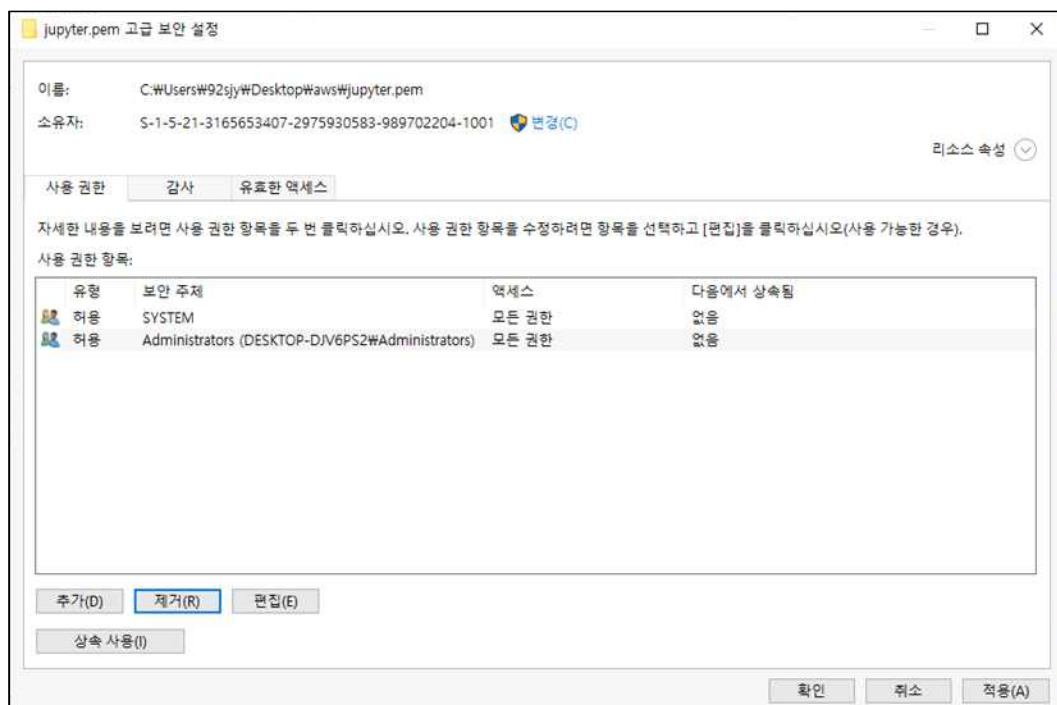




- 상속안함 클릭 → 상속된 사용 권한을 이 개체에 대한 명시적 사용 권한으로 변환 클릭

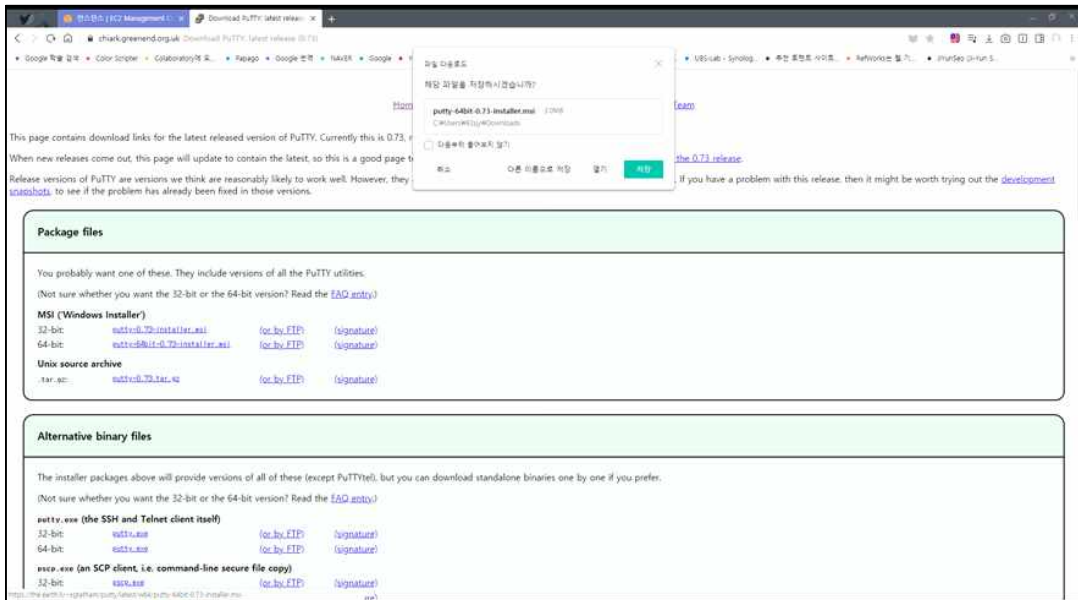


- 상속안함 클릭 → 상속된 사용 권한을 이 개체에 대한 명시적 사용 권한으로 변환 클릭
- 클릭 후 system, administrators 제외한 모든 유형 삭제



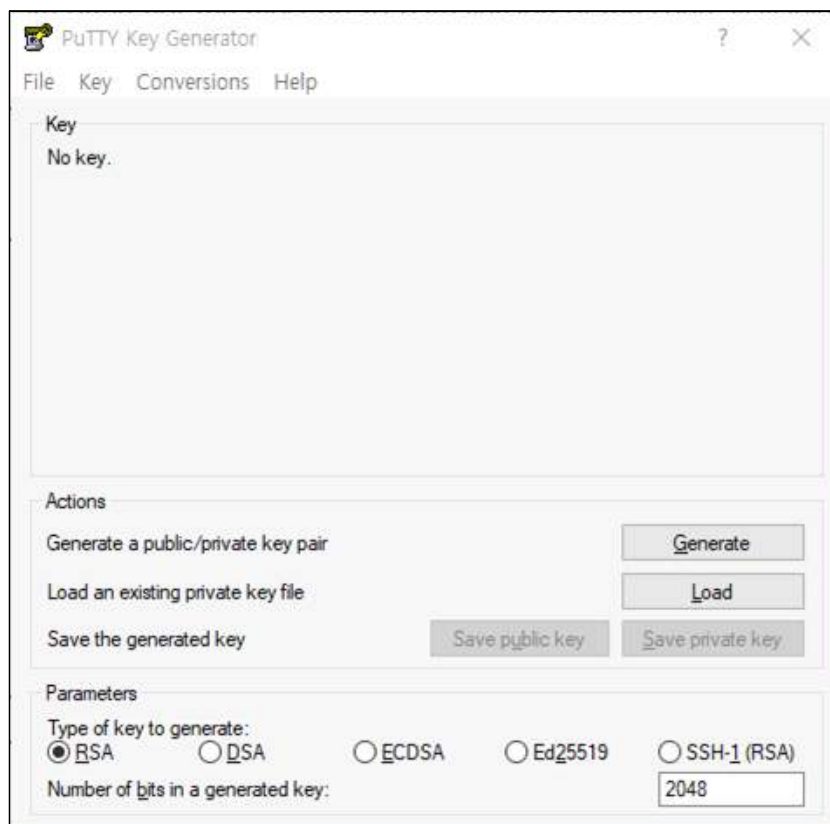
- 인스턴스 실행을 위한 putty 설치

✓ <https://www.chiark.greenend.org.uk/~sgtatham/putty/>

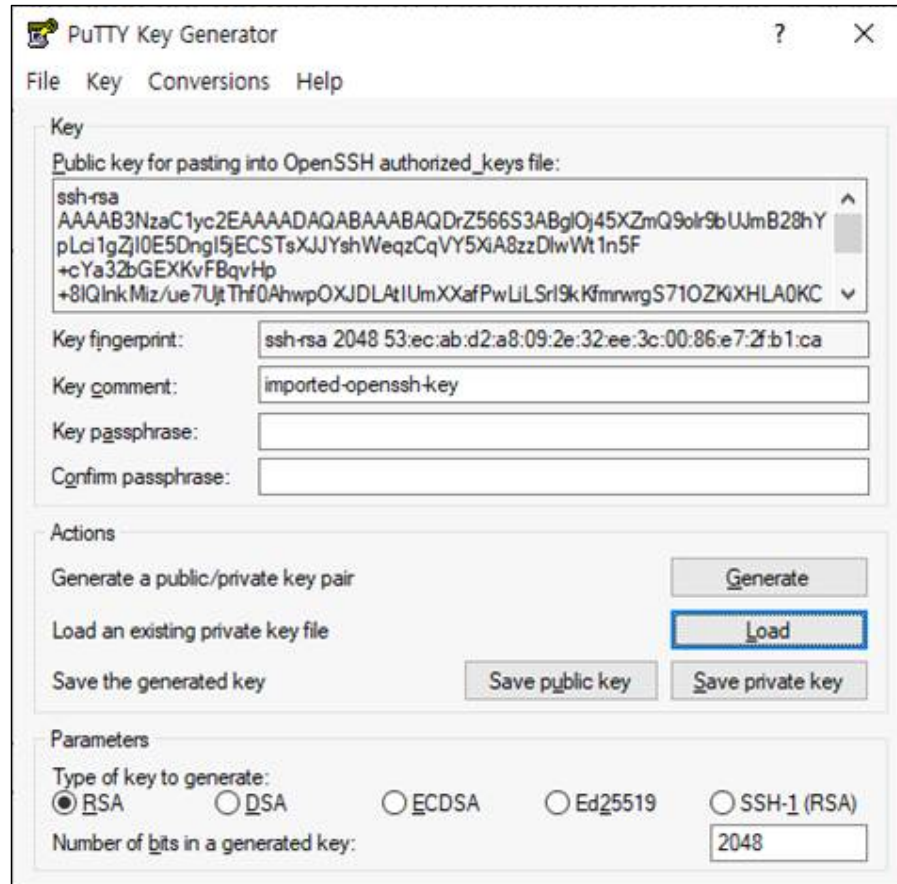


- putty 설치가 완료되면 putty, puttygen 두 가지 파일이 설치됨

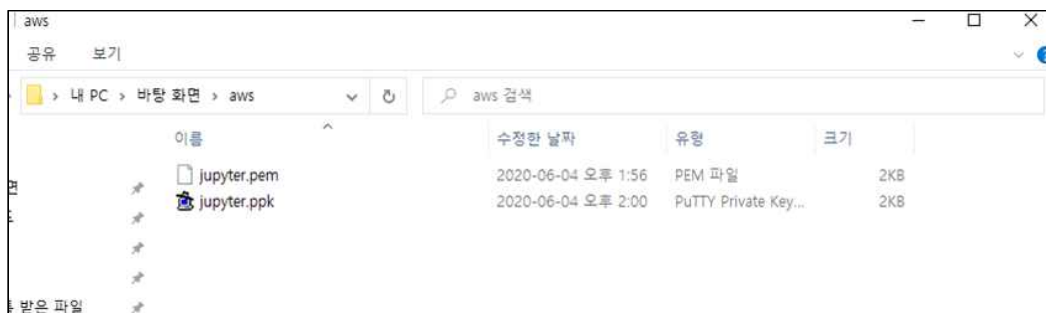
✓ putty gen을 실행하여 키페어.pem을 키페어.ppk로 변환



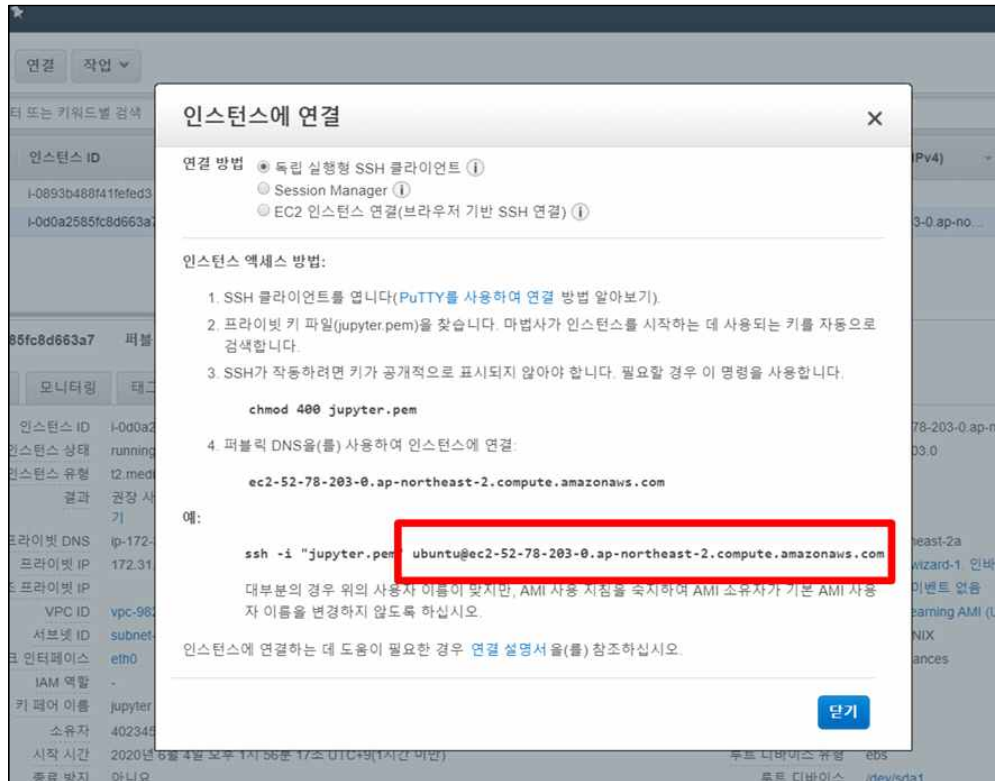
- Load 클릭 후 파일 유형을 모두 all files로 변환 후 pem 파일 불러오기
  - ✓ 파라미터를 RSA, 2048로 체크 후 Save private key 클릭



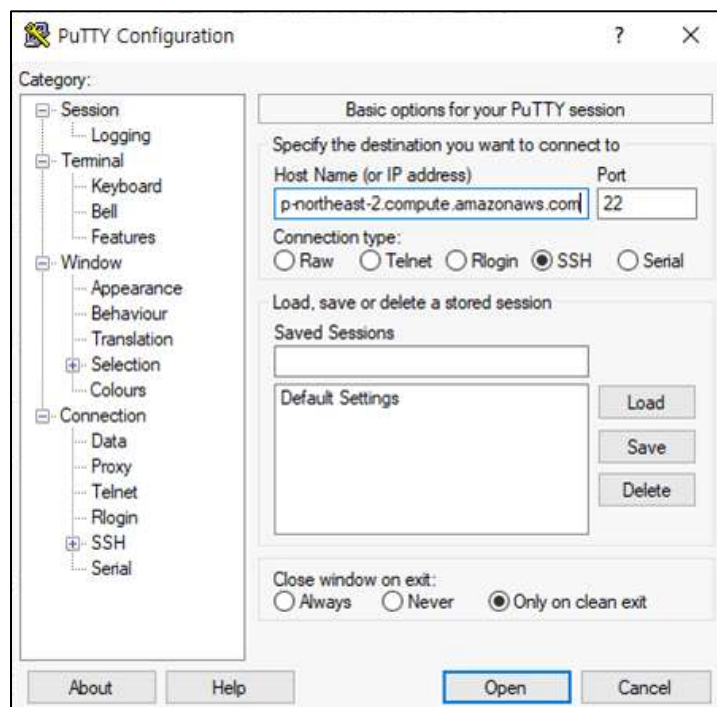
- 팝업창이 뜨면 [예] 선택 후 안전한 폴더에 저장



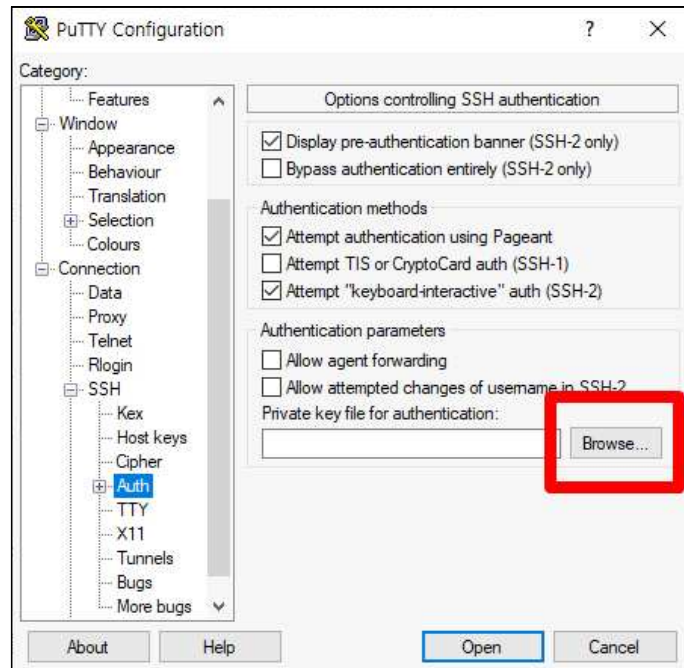
- AWS 인스턴스 페이지에서 연결 클릭 후 붉은 영역 복사  
✓ putty의 host name으로 들어갈 예정



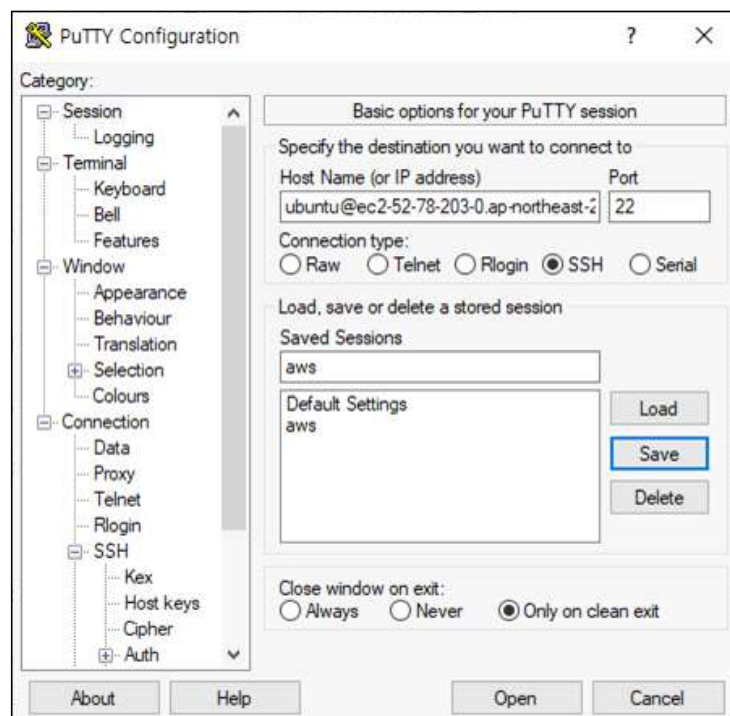
- putty 실행 후 host name에 붙여넣기



- Category에서 Connection → SSH → Auth로 들어간 후 Brower 클릭 하여 변환된 키페어 입력



- Session으로 돌아와 세션 이름을 설정하고 save 클릭 후 open 클릭
- ✓ 이후 저장된 세션을 이용하여 이전의 작업과정을 수행하지 않더라도 인스턴스 실행 가능



- 인스턴스가 실행되고 팝업창이 뜰 경우 [예] 클릭



- 인스턴스 실행 완료

```
ubuntu@ip-172-31-5-206:~$
for Tensorflow(GPU) with Python3 (CUDA 10.0 and Intel MKL-DNN)
    source activate tensorflow_g2
    source activate tensorflow_g2
for Tensorflow(GPU) with Python2
    source activate tensorflow_g2
for Theano(Keras) with Python3 (CUDA 9.0)
    source activate theano_g3
for Theano(Keras) with Python2 (CUDA 9.0)
    source activate theano_g2
for PyTorch with Python3 (CUDA 10.0 and Intel MKL)
    source activate pytorch_g3
for PyTorch with Python2 (CUDA 10.0 and Intel MKL)
    source activate pytorch_g2
for CHYI(Keras) with Python3 (CUDA 9.0 and Intel MKL-DNN)
    source activate chyi_g3
for CHYI(Keras) with Python2 (CUDA 9.0 and Intel MKL-DNN)
    source activate chyi_g2
for Caffe2 with Python3 (CUDA 9.0)
    source activate caffe2_g3
for Caffe2 with Python2 (CUDA 9.0)
    source activate caffe2_g2
for Caffe with Python3 (CUDA 9.0)
    source activate caffe_g3
for Caffe with Python2 (CUDA 9.0)
    source activate caffe_g2
for Chainer with Python3 (CUDA 10.0 and Intel MKL)
    source activate chainer_g3
for Chainer with Python2 (CUDA 10.0 and Intel MKL)
    source activate chainer_g2
for base Python3 (CUDA 9.0)
    source activate python3
for base Python2 (CUDA 9.0)
    source activate python2

Official Conda User Guide: https://docs.conda.io/projects/conda/en/latest/user-g
ides.html
AWS Deep Learning AMI Homepage: https://aws.amazon.com/machine-learning/ami/
Developer Guide and Release Notes: https://docs.aws.amazon.com/dlami/latest/deve
loper/what-is-dlami.html
Support: https://forums.aws.amazon.com/forum.jspa?forumID=243
For a fully managed experience, check out Amazon SageMaker at https://aws.amazon
.com/sagemaker

-----
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
https://www.ubuntu.com/business/services/cloud

164 packages can be updated.
169 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ip-172-31-5-206:~$
```







- 주피터 노트북 생성
  - ① `jupyter notebook --generate-config`
  - ② `sudo vi /home/ubuntu/.jupyter/jupyter_notebook_config.py`

```
ubuntu@ip-172-31-5-206:~$  
ubuntu@ip-172-31-5-206:~$ jupyter notebook --generate-config  
C  
ubuntu@ip-172-31-5-206:~$ clear  
ubuntu@ip-172-31-5-206:~$ jupyter notebook --generate-config  
Override /home/ubuntu/.jupyter/jupyter_notebook_config.py with default config? [Y/N]y  
Writing default config to: /home/ubuntu/.jupyter/jupyter_notebook_config.py  
ubuntu@ip-172-31-5-206:~$ sudo vi /home/ubuntu/.jupyter/jupyter_notebook_config.py  
# Config generated from the Jupyter Notebook  
  
#-----  
# Application configuration options  
#-----  
  
# Most IPython applications.  
  
# The main kernel used by launching applications in interactive  
# Applications may start here - it's best to leave this as the default  
  
# The language classes module  
# Applications may start here - it's best to leave this as the default  
  
# Set the log level for various components  
# Applications may start here - it's best to leave this as the default  
  
# JupyterLab application configuration  
# Applications may start here - it's best to leave this as the default  
  
# Some other tool specific applications  
  
# Launching an application  
# Applications may start here - it's best to leave this as the default  
  
# Some other tool specific applications  
  
# Launching an application  
# Applications may start here - it's best to leave this as the default
```

- 가장 아래로 내려와서 1-3 입력하고 4번으로 빠져나가기
  - c = get\_config()
  - c.NotebookApp.password = u'해시값(sha1:~)' ← 저장해둔 해시값
  - c.NotebookApp.ip = 'ip입력'
  - esc 누르고 :wq

```

ubuntu@ip-172-31-5-206: ~
# The right file is used to store Docker signatures. By default, this
# is in your $HOME and directory. The can set it to /usr/local/bin.
# Right setting is the file name.
c.NotebookApp.signature_file = ""

# The secret key with which Docker signatures are signed.
c.NotebookApp.secret_key = ""

# The file where the secret key is stored.
c.NotebookApp.secret_key_file = ""

# A challenge requiring the storage backend for Docker signatures. The default
# is set to 0.0.0.0.
c.NotebookApp.storage_backend = 'sqlite3,sqlite3'

# Verbose logging configuration.
# Verbose logging configuration.

# If there is no Python kernel installed and the IPython kernel is
# available, ensure it is added to the open list.
c.NotebookApp.kernels_start_kernel = True

# The kernel spec class. This is configurable to allow overriding of the
# NotebookApp.kernels for persistent kernels.
c.NotebookApp.kernels_start_kernel = 'IPythonKernel, NotebookApp'

# Whether to allow kernel specs.
# By default, all installed kernels are allowed.
c.NotebookApp.kernels_start_kernel = True

c = get_config()
c.NotebookApp.password = 'sha1:21a3e5d2217:cf2e459e23a9406bdc437f0117b2b8e4d943'
c.NotebookApp.ip = '172.31.5.206'

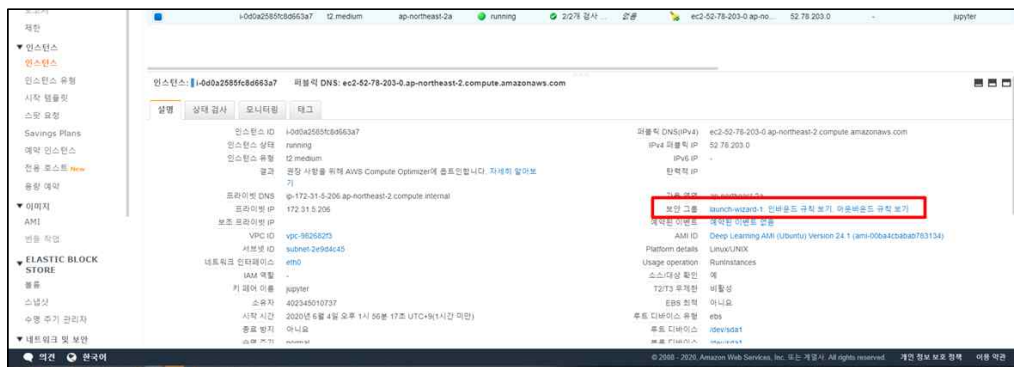
-- INSERT --

```

- 주피터 노트북 활성화
  - ✓ `sudo jupyter-notebook --allow-root`

```
ubuntu@ip-172-31-5-206:~$ sudo vi /home/ubuntu/.jupyter/jupyter_notebook_config.py
ubuntu@ip-172-31-5-206:~$ sudo jupyter-notebook --allow-root
[I 05:28:12.116 NotebookApp] Writing notebook server cookie secret to /home/ubuntu/.local/share/jupyter/runtime/notebook_cookie_secret
[I 05:28:16.015 NotebookApp] Serving notebooks from local directory: /home/ubuntu
[I 05:28:16.016 NotebookApp] The Jupyter Notebook is running at:
[I 05:28:16.016 NotebookApp] http://172.31.5.206:8888/
[I 05:28:16.016 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation)
[W 05:28:16.020 NotebookApp] No web browser found: could not locate runnable browser.
```

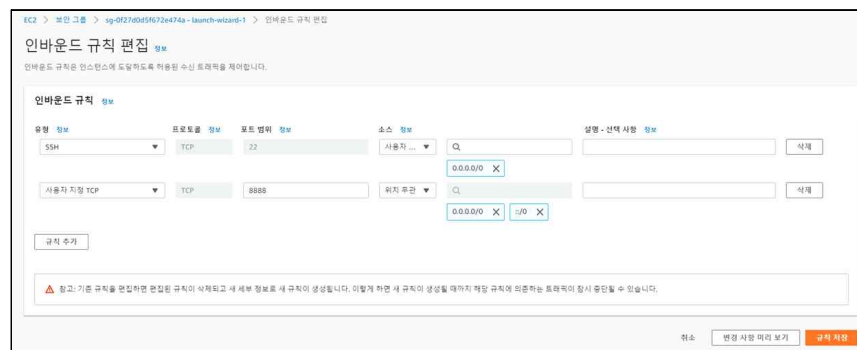
- 보안 그룹 클릭(본인이 설정한 보안그룹 이름)



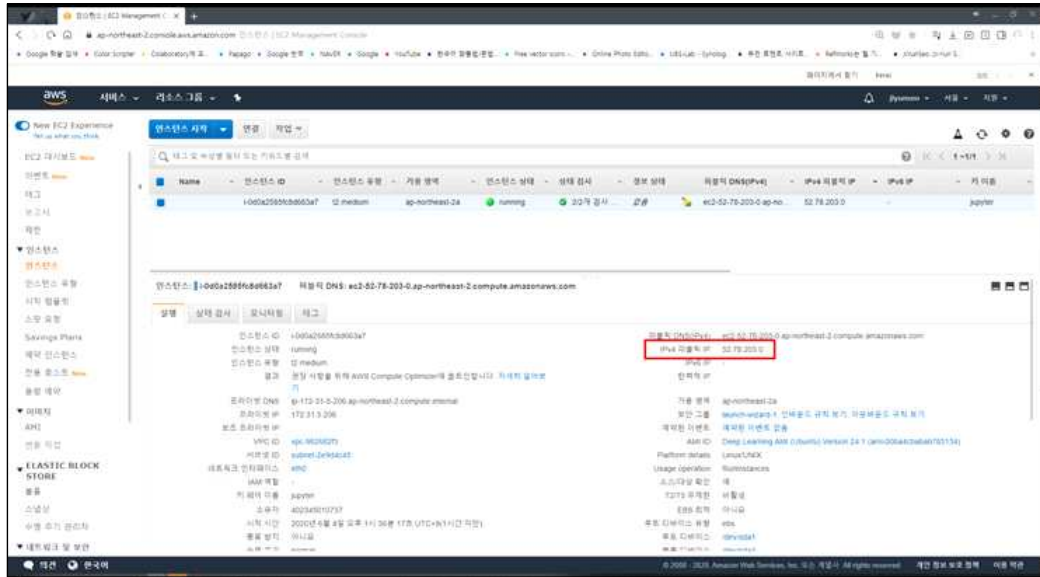
- 작업 → 인바운드 규칙 편집



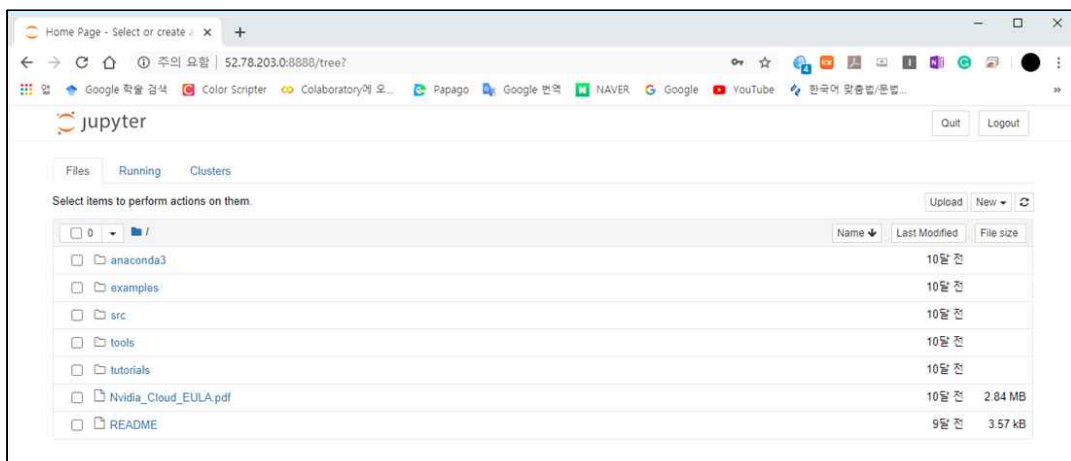
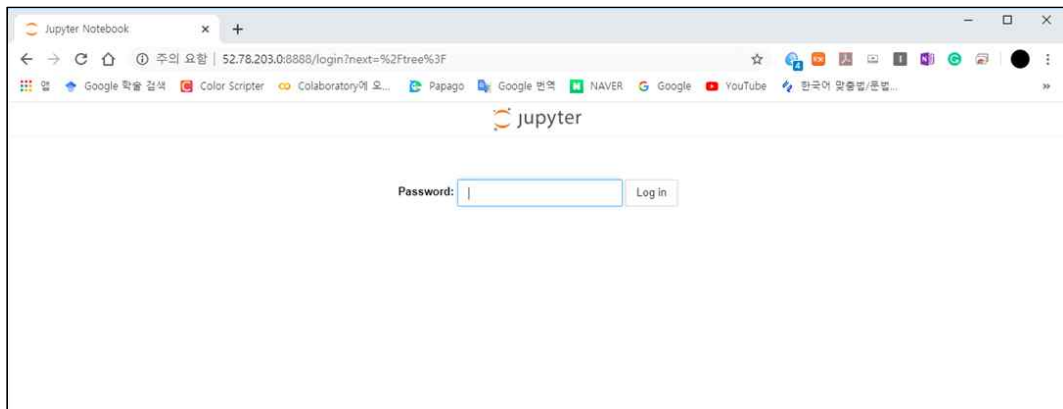
- 규칙 추가 클릭 후 규칙정보 입력 후 저장 클릭
  - ✓ 유형 : 사용자 지정 TCP, 포트범위 : 8888, 소스 : 위치무관



- IPv4 퍼블릭 ip 확인

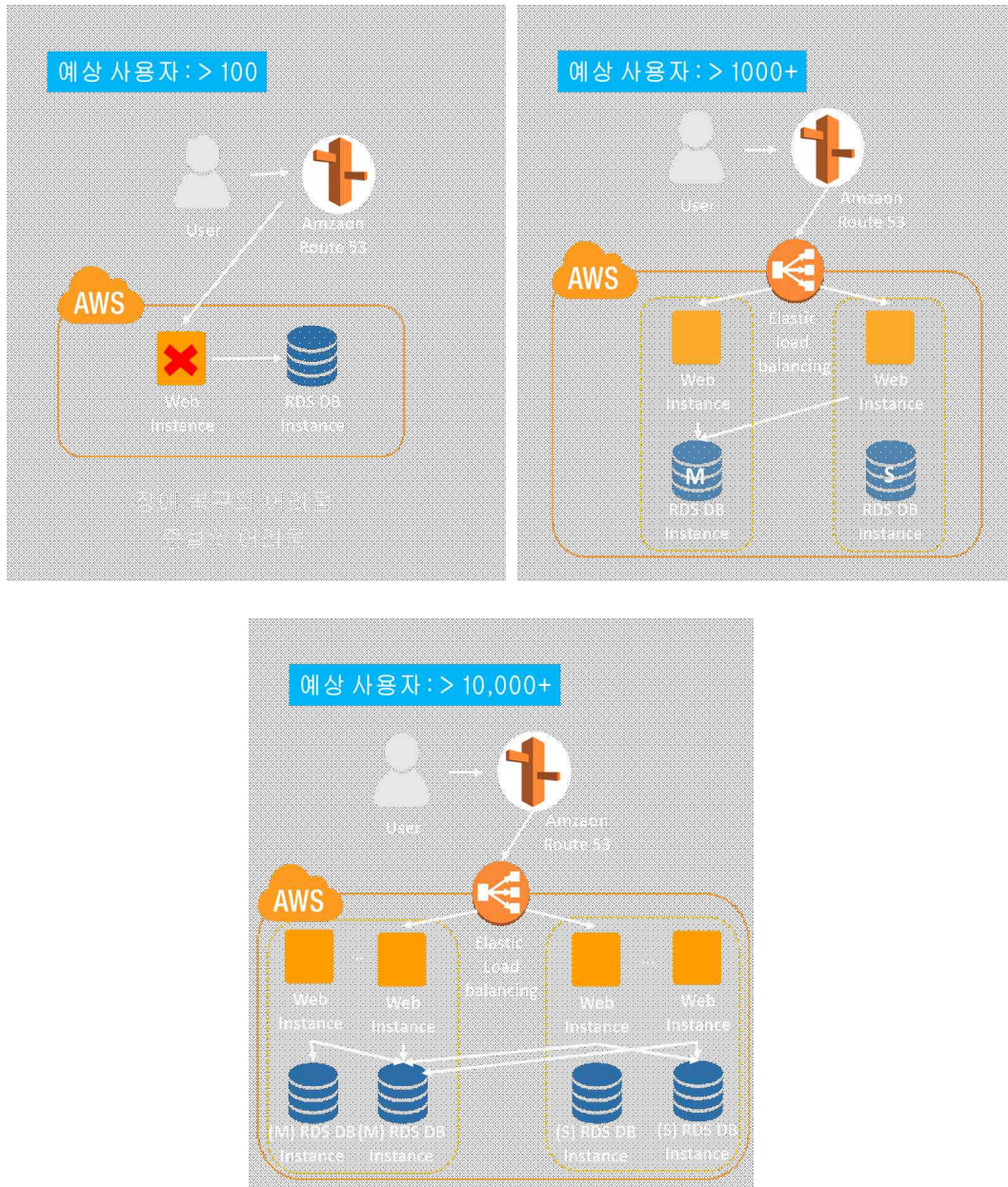


- 주소창에 IPv4 퍼블릭 ip:8888 입력을 하면 주피터 실행 완료



## ※ 나도 전문가다

- EC2 오토 스케일링



- 사례 : 언론사 홈페이지
  - ✓ 기존 IDC 인프라의 한계점
    - 특정 기사가 올라왔을 때 트래픽 처리가 어려움
    - 기존 IDC는 트래픽 최고치 기준으로 요금 책정되어 불합리
    - 안정적(고가용성)이고 효율적(비용)인 인프라 운영이 중요

✓ AWS로 전환 후

- 서버 부하 문제 해결 및 확장성 확보(Auto Scaling 활용)
- 서버 인프라 비용 절감
- 개발 및 이전 시간 단축
- 웹사이트 성능 향상



### 평가하기

#### 1. 빈 칸에 알맞은 단어는?

- EC2 인스턴스에 원격으로 접속하기 위하여  를 사용한다.

- 정답 : Putty

해설 : 인스턴스 원격 접속을 위해서는 Putty를 사용합니다. 하지만 다른 응용프로그램도 사용할 수 있습니다.

#### 2. 빈 칸에 알맞은 단어는?

- 실행 결과를 웹 브라우저를 통해 실시간으로 확인하기 위하여  를 사용한다.

- 정답 : 주피터 노트북

해설 : 주피터 노트북은 데이터 시각화, 코드 공유, 코드와의 실시간 대화, 코드 샘플 기록 등의 장점을 가지고 있습니다.

### 학습정리

#### 1. EC2 인스턴스 생성하기

- AMI 선택 - Deep Learning AMI(Ubuntu)
- 인스턴스 유형 선택
- 인스턴스 세부 정보 구성
- 스토리지 추가
- 태그 추가
- 보안 그룹 구성
- 인스턴스 시작 검토
- 키 페어 생성
- 예상 요금 알람 받기
- EC2 인스턴스 생성완료