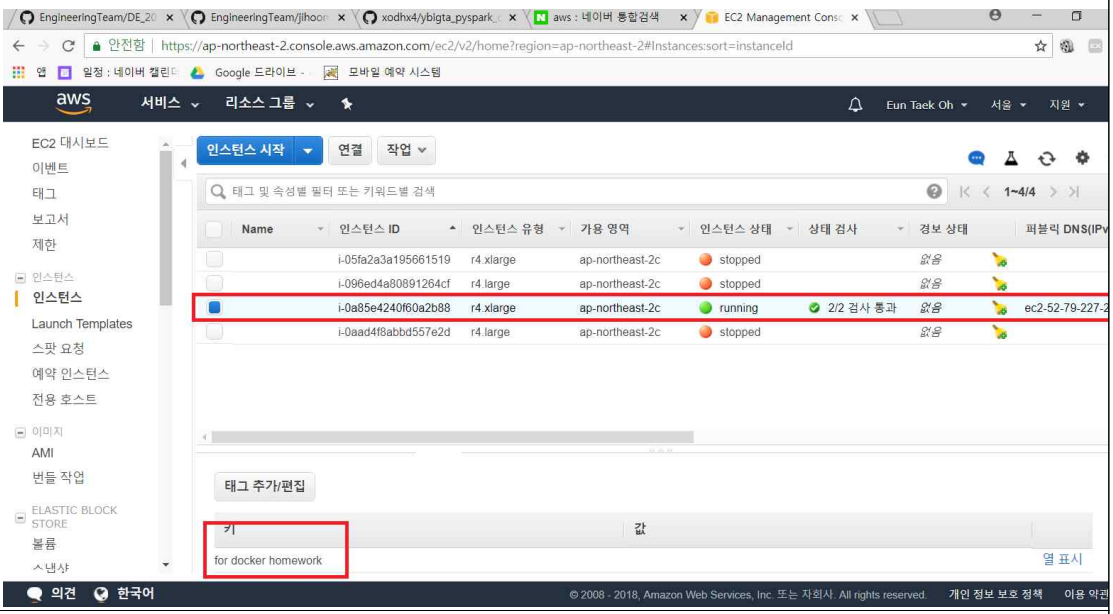
인스턴스 생성



필요한 패키지 설치

```
sudo apt-get update
sudo apt-get install \
    apt-transport-https \
    ca-certificates \
    curl \
    software-properties-common
```

도커 repo 추가하기

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
| sudo apt-key add -

sudo apt-key fingerprint 0EBFCD88

sudo add-apt-repository \
   "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
   $(lsb_release -cs) \
   stable"
```

도커 CE(Communitiy edition) 설치

```
sudo apt-get update
sudo apt-get install docker-ce
```

sudo 없이 도커쓰기

```
sudo groupadd docker
```

```
sudo usermod -aG docker $USER
```

```
root@d685cbfdeb70: /
faribausc/imagename                                         0

arcangel12456/imagename                                     0

xandyx/imagename                                            0

tycy/imagename                              dev-nodejs-ubuntu   0

rteja/imagename                                             0

ubuntu@ip-172-31-21-180:~$ docker pull ubuntu:16.04
16.04: Pulling from library/ubuntu
3620e2d282dc: Pull complete
ef22f5e4b3b2: Pull complete
99f229f854da: Pull complete
4fe433abe16a: Pull complete
c9b72a16d85e: Pull complete
Digest: sha256:14066a391d902c386d6164d44ade3460ba044abcdf8df88b0ff79a6f635be8d3
Status: Downloaded newer image for ubuntu:16.04
ubuntu@ip-172-31-21-180:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              16.04               e13f3d529b1a        3 days ago          115MB
hello-world         latest              2cb0d9787c4d        9 days ago          1.85kB
ubuntu@ip-172-31-21-180:~$ docker run -it --name homework 0p 10001:10001 ubuntu:16.04 /bin/bash
Unable to find image '0p:latest' locally
docker: Error response from daemon: pull access denied for 0p, repository does not exist or may r
ire 'docker login'.
See 'docker run --help'.
ubuntu@ip-172-31-21-180:~$ docker run -it --name homework -p 10001:10001 ubuntu:16.04 /bin/bash
root@d685cbfdeb70:/# docker ps
bash: docker: command not found
root@d685cbfdeb70:/# ls
bin   dev  home  lib64  mnt  proc  run   srv  tmp  var
boot  etc  lib   media  opt  root  sbin  sys  usr
root@d685cbfdeb70:/#
```

docker images 쳐서 image 있는지 확인

```
ubuntu@ip-172-31-30-56: ~
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/

ubuntu@ip-172-31-30-56:~$ docker pull ubuntu:16.04
16.04: Pulling from library/ubuntu
3620e2d282dc: Pull complete
ef22f5e4b3b2: Pull complete
99f229f854da: Pull complete
4fe433abe16a: Pull complete
c9b72a16d85e: Pull complete
Digest: sha256:14066a391d902c386d6164d44ade3460ba044abcdf8df88b0ff79a6f635be8d3
Status: Downloaded newer image for ubuntu:16.04
ubuntu@ip-172-31-30-56:~$ docker run -it --name homework -p 10001:10001 ubuntu:16.04 /bin/bash
root@e157521982a6:/# exit
exit
ubuntu@ip-172-31-30-56:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
PORTS               NAMES
ubuntu@ip-172-31-30-56:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              16.04               e13f3d529b1a        3 days ago          115MB
hello-world         latest              2cb0d9787c4d        9 days ago          1.85kB
ubuntu@ip-172-31-30-56:~$
```

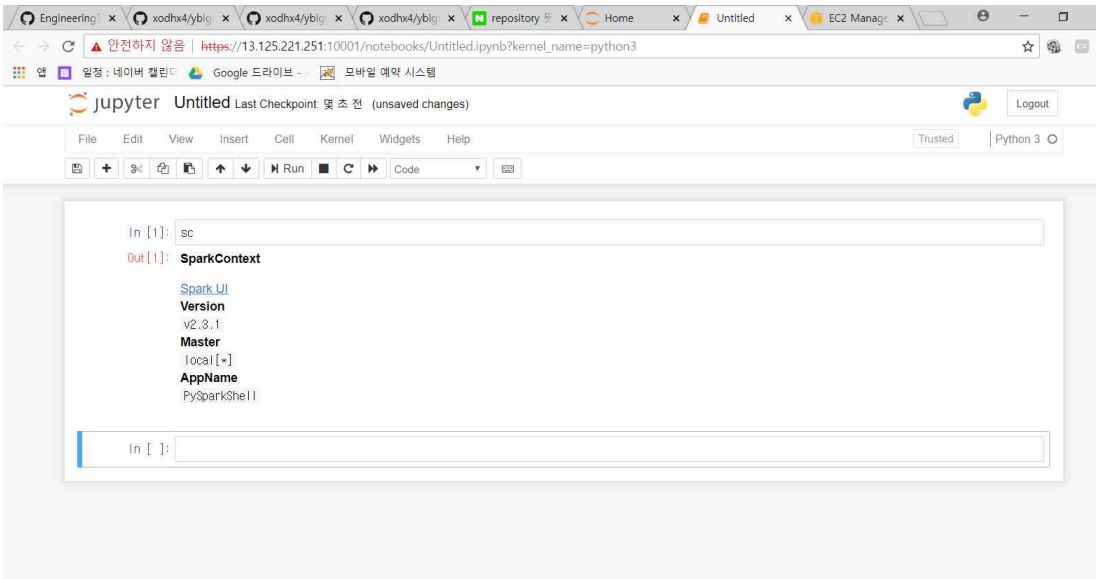
homework3 : spark 컨테이너 만들기

git clone https://github.com/xodhx4/ybigta_pyspark_docker.git
을 입력해서 repo 클론하기

docker build -tab pyspark:1.0 .
일케 쳐서 docker image를 만들자

docker run -it -name homework2 -p 10001:10001 -v ~/workspace:/root/workspace pyspark:1.0

쳐서 컨테이너 만들기. 동시에 실행됨
https://{ec-2 ip}:10001 치면 jupyter notebook 들어가지고 비밀번호는 admin



ctrl+c 눌러서 pyspark중지
kill $(ps -ef | grep hive | awk '{print $2}')
hive 종료
$HADOOP_HOME/sbin/stop-yarn.sh
$HADOOP_HOME/sbin/stop-dfs.sh

```
kip confirmation).
13.[I 11:52:30.769 NotebookApp] 302 GET / (165.132.5.144) 0.60ms
[I 11:52:30.779 NotebookApp] 302 GET /tree? (165.132.5.144) 0.54ms
[I 11:52:34.133 NotebookApp] 302 POST /login?next=%2Ftree%3F (165.132.5.144) 1.07ms
[I 11:52:37.306 NotebookApp] Creating new notebook in
[I 11:52:37.318 NotebookApp] Writing notebook-signing key to /root/.local/share/jupyter/notebook_sec
ret
[I 11:52:39.064 NotebookApp] Kernel started: d282378b-d946-4657-8ee3-084c2e3bf712
2018-07-20 11:52:41 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platfo
rm... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
[I 11:52:43.435 NotebookApp] Adapting to protocol v5.1 for kernel d282378b-d946-4657-8ee3-084c2e3bf7
12
^C
root@24c8002b7dba:~#
```

```
[I 11:52:39.064 NotebookApp] Kernel started: d282378b-d946-4657-8ee3-084c2e3bf712
2018-07-20 11:52:41 WARN  NativeCodeLoader:62 - Unable to load native-hadoop library for your platf
rm... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
[I 11:52:43.435 NotebookApp] Adapting to protocol v5.1 for kernel d282378b-d946-4657-8ee3-084c2e3bf
12
^C
root@24c8002b7dba:~# ^C
root@24c8002b7dba:~# kill $(ps -ef | grep hive | awk '{print $2}')[I 11:54:39.860 NotebookApp] Savi
g file at /Untitled.ipynb

bash: kill: (1266) - No such process
root@24c8002b7dba:~# $HADOOP_HOME/sbin/stop-yarn.sh
stopping yarn daemons
stopping resourcemanager
localhost: stopping nodemanager
localhost: nodemanager did not stop gracefully after 5 seconds: killing with kill -9
no proxyserver to stop
root@24c8002b7dba:~# $HADOOP_HOME/sbin/stop-dfs.sh
Stopping namenodes on [localhost]
localhost: stopping namenode
localhost: stopping datanode
Stopping secondary namenodes [localhost]
localhost: stopping secondarynamenode
root@24c8002b7dba:~# jps
1186 SparkSubmit
1751 Jps
root@24c8002b7dba:~#
```

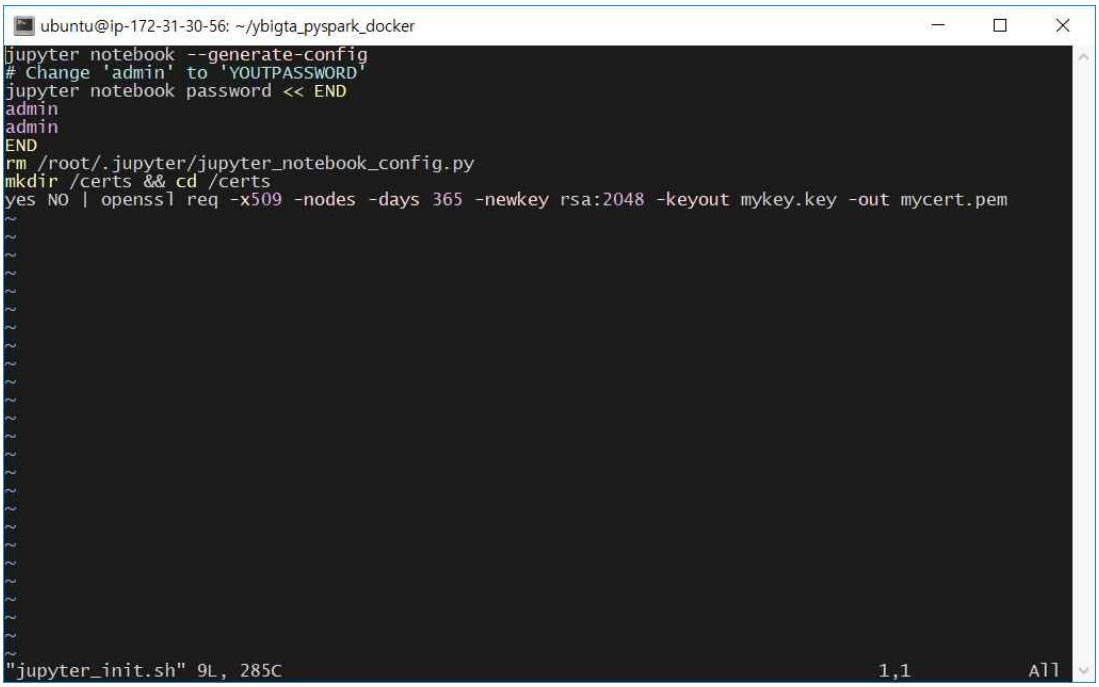
나오는 법은 exit 하면 나와지고,
컨테이너 재실행

```
root@24c8002b7dba: ~
ubuntu@ip-172-31-30-56:~$ docker start homework3
homework3
ubuntu@ip-172-31-30-56:~$ docker stop homework3
docker start homework2
^C
ubuntu@ip-172-31-30-56:~$ docker start homework2
Error response from daemon: No such container: homework2
Error: failed to start containers: homework2
ubuntu@ip-172-31-30-56:~$ docker start homework3
homework3
ubuntu@ip-172-31-30-56:~$ docker stop homework3
^C
ubuntu@ip-172-31-30-56:~$ docker stop homework3
homework3
ubuntu@ip-172-31-30-56:~$ docker start homework3
homework3
ubuntu@ip-172-31-30-56:~$ docker attach homework3
2018-07-20 12:00:29: Starting Hive Metastore Server
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/root/apache-hive-2.3.3-bin/lib/log4j-slf4j-impl-2.6.2.jar!/org/sl
f4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/root/hadoop-2.9.0/share/hadoop/common/lib/slf4j-log4j12-1.7.25.ja
r!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
root@24c8002b7dba:~# [I 12:00:33.797 NotebookApp] JupyterLab beta preview extension loaded from /opt
/conda/lib/python3.6/site-packages/jupyterlab
[I 12:00:33.798 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 12:00:33.806 NotebookApp] Serving notebooks from local directory: /root
[I 12:00:33.807 NotebookApp] 0 active kernels
[I 12:00:33.807 NotebookApp] The Jupyter Notebook is running at:
[I 12:00:33.807 NotebookApp] https://24c8002b7dba:10001/
[I 12:00:33.807 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to s
kip confirmation).
```

docker start homework3 하면 컨ㅌ네이너가 켜지고
docker attach homework3하면 들어가짐
exit하던지, 에 나와서는 docker stop homework3하면 homework가 꺼짐

vim jupyter_init.sh 해서 들어가서

원하는 비번입력후 저장



후에 다시 build하면 댐~
다시 처음부터 시작하라는 뜻