

텐서플로우 시작 2

클라우드 기반 텐서플로우

AWS(Amazon Web Services)

- AWS란
 - 2006년 공식적으로 런칭한 클라우드 서비스 플랫폼
 - 전 세계 수십만 명의 고객을 보유하고 있음
 - 네트워킹 기반으로 가상 컴퓨터와 스토리지 등 다양한 서비스 제공
 - 쉽고 빠른 확장성과 비용을 절감할 수 있음
 - 클라우드를 이용하여 필요한 인프라 서비스를 사용할 수 있는 IaaS(Infrastructure as a Service)
 - 클라우드 서비스는 어떤 자원을 제공하는지에 따라 크게 3가지로 구분됨
 - IaaS(Infrastructure as a Service)
 - SaaS(Software as a Service)
 - PaaS(Platform as a Service)
- IaaS(Infrastructure as a Service) : AWS처럼 서버와 스토리지 등 컴퓨팅 인프라 장비를 제공해 주는 서비스
 - SaaS(Software as a Service) : 네이버처럼 모든 서비스를 웹 브라우저 환경에서 제공하는 서비스
 - PaaS(Platform as a Service) : 개발 시 필요한 플랫폼을 제공하는 서비스

AWS(Amazon Web Services)

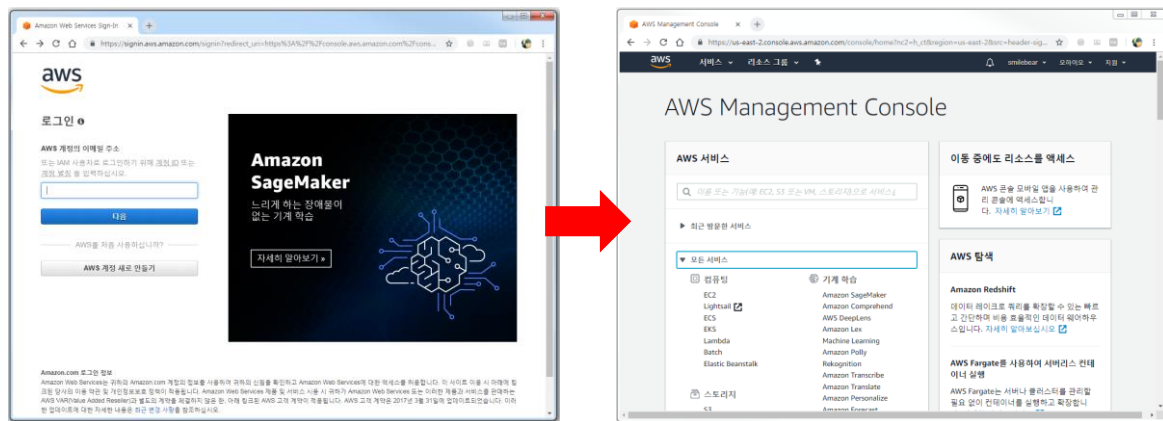
- AWS란
 - AWS는 분산되어 있는 데이터를 하나의 플랫폼에 집중시키고, 쉽고 빠르게 데이터를 수집하기 위해 구축
 - 전 세계 사람들이 관심을 가지고 사용하고 있음
 - IT 구축에 대한 모든 서비스를 제공하고 있음
- AWS 비용
 - AWS를 사용하기 위해서는 유료와 무료 제품이 있음
 - 비용은 제품 및 서비스마다 다름
 - 사용자가 사용한 만큼 비용이 계산되어 후불로 지불하는 방식
 - 12개월 동안 무료로 사용할 수 있는 AWS 프리티어로 제품 및 서비스를 체험해볼 수 있음
 - 프리티어는 AWS 신규 고객에게만 제공됨(가입일로부터 12개월 동안 사용 가능)
 - 프리티어 기간이 종료되면 항상 무료인 제품 외에는 사용한 만큼의 비용이 발생됨
 - 유/무료 제품 및 서비스, 비용 등 사전에 확인 후 사용하는 것이 좋음

AWS(Amazon Web Services)

- AWS SageMaker이란
 - AWS에서 제공하는 서비스 중 종합 관리형 기계 학습 서비스
 - 데이터 과학자와 개발자들이 기계 학습 모델을 빠르고 쉽게 구축하고 교육시킬 수 있음
 - Jupyter notebook 환경 제공
 - 콘솔에서 클릭 한 번으로 사용할 수 있기 때문에 쉽고 빠르게 서비스를 이용할 수 있음
 - 프리티어로 사용할 경우 무료로 사용가능하지만, 무료로 사용할 수 있는 범위가 있음
 - 요금 정보는 AWS 홈페이지 내 SageMaker 요금 안내(<https://aws.amazon.com/ko/sagemaker/pricing/>) 참고

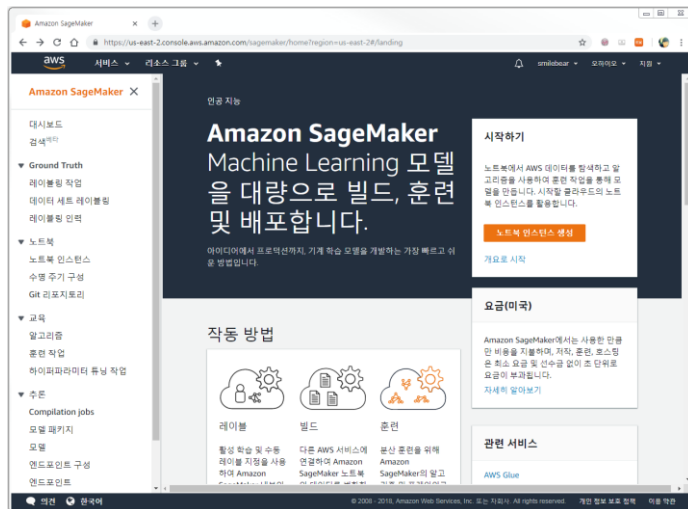
AWS(Amazon Web Services)

- AWS SageMaker 실행
 - AWS 홈페이지에 접속하여 로그인
 - 로그인 후 AWS에서 제공하는 모든 서비스 목록 확인 가능
 - 기계 학습 부분에서 Amazon SageMaker 서비스를 이용하여 Tensorflow를 사용
 - 다른 서비스에서도 Tensorflow를 사용할 수 있음
 - SageMaker 서비스가 초보자들이 가장 쉽게 사용할 수 있는 서비스



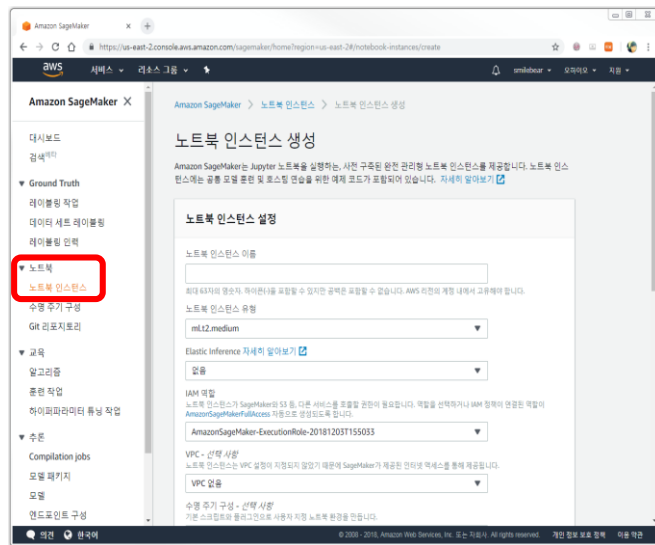
AWS(Amazon Web Services)

- AWS SageMaker 실행
 - SageMaker 서비스를 클릭하면 다양한 정보 확인 가능
 - SageMaker 소개 및 작동방법, 요금 등의 정보를 확인 할 수 있음



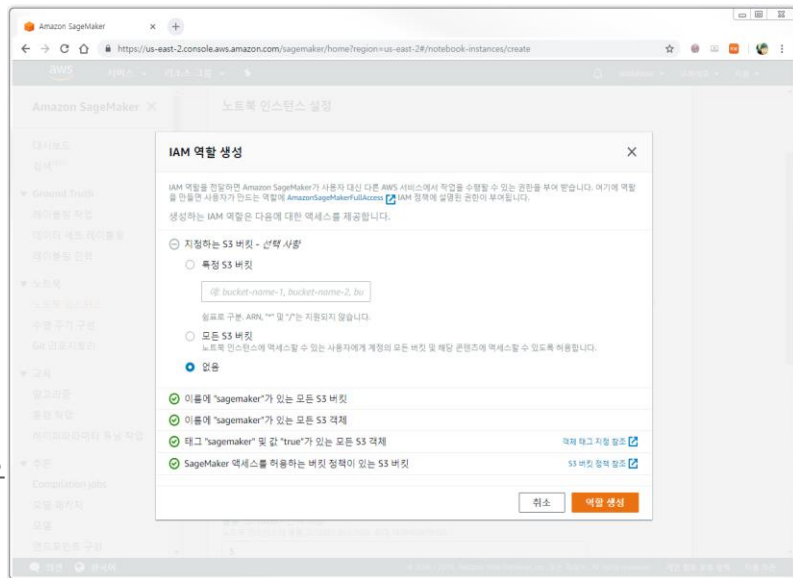
AWS(Amazon Web Services)

- AWS SageMaker 실행
 - 화면 좌측 메뉴 중 노트북 인스턴스를 통하여 인스턴스 생성
 - 노트북 인스턴스 이름
 - 조건 범위 내에서 자유롭게 설정
 - 조건 범위는 입력 아래 부분에 설명되어 있음
 - 노트북 인스턴스 유형
 - 기본으로 ml.t2.medium으로 설정되어 있음
 - AWS 무료티어를 사용하면 가입 후 첫 2개월 동안 SageMaker에서 모델 구축을 위해 t2.medium 노트북 250시간 무료 제공
 - 다른 인스턴스 유형은 홈페이지(<https://aws.amazon.com/ko/sagemaker/pricing/instance-types/>) 참고
 - Elastic Inference
 - 심화 학습 경우에 필요한 부분
 - 해당사항 없으므로 변경없이 진행



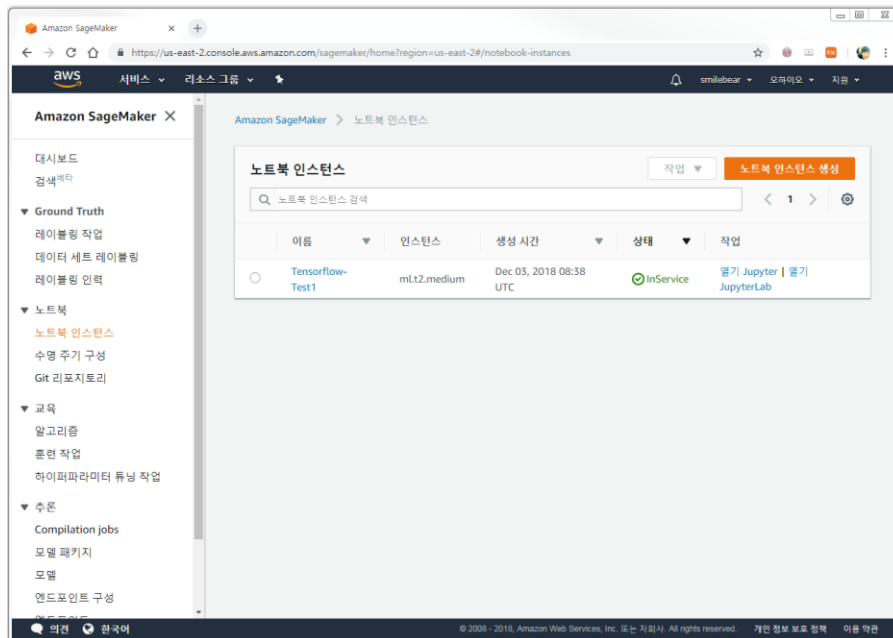
AWS(Amazon Web Services)

- AWS SageMaker 실행
 - IAM 역할
 - IAM 역할은 신뢰하는 개체에 권한을 부여하는 안전한 방법
 - IAM 역할을 전달하면 SageMaker가 사용자 대신 다른 AWS 서비스에서 작업을 수행할 수 있는 권한을 부여 받음
 - S3 버킷
 - 입력 데이터 및 출력을 저장할 때 선택
 - AWS에서 지원하는 인터넷용 스토리지 서비스
 - 특정 S3 버킷
 - 민감한 데이터가 있는 경우 선택하여 액세스 제한
 - 모든 S3 버킷
 - 더 많은 S3 버킷에 액세스하고자 할 때 선택
 - 없음
 - 액세스를 명시적으로 제어하고 싶은 경우 선택



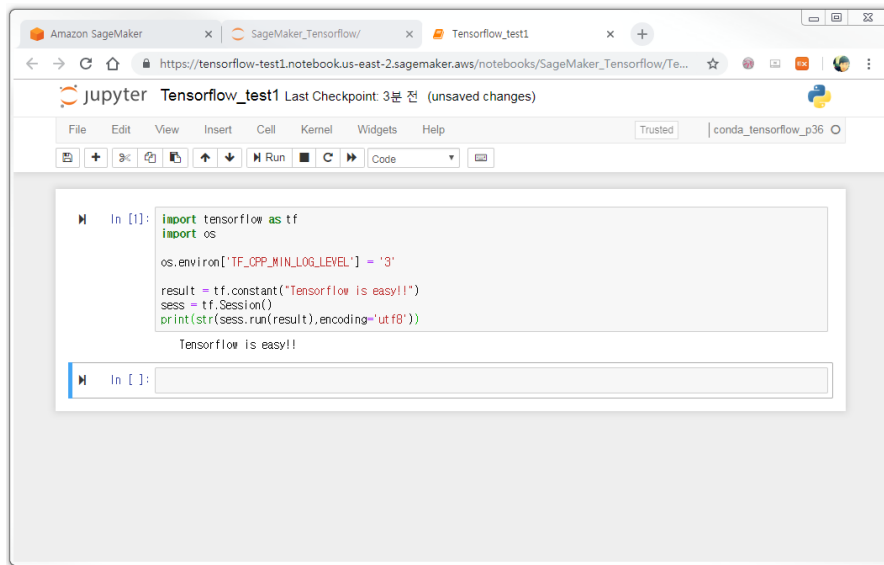
AWS(Amazon Web Services)

- AWS SageMaker 실행
 - SageMaker 노트북 인스턴스 생성
 - 인스턴스 이름 및 생성시간, 상태, 작업내용 등 확인할 수 있음
 - 상태가 InService이면 서비스 사용 가능
 - 작업 부분에서 Jupyter 열기 버튼을 클릭하여 Jupyter notebook 활성화



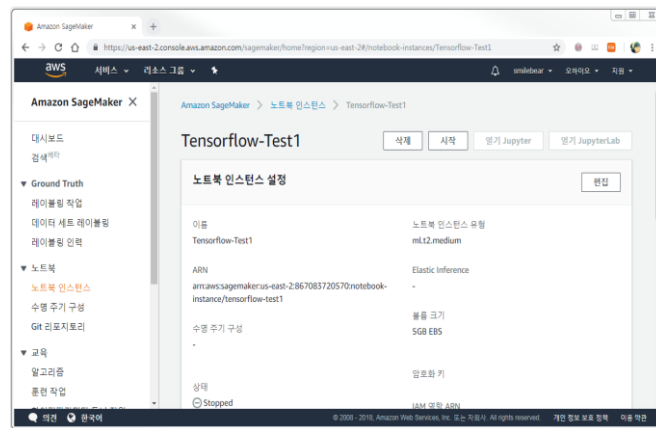
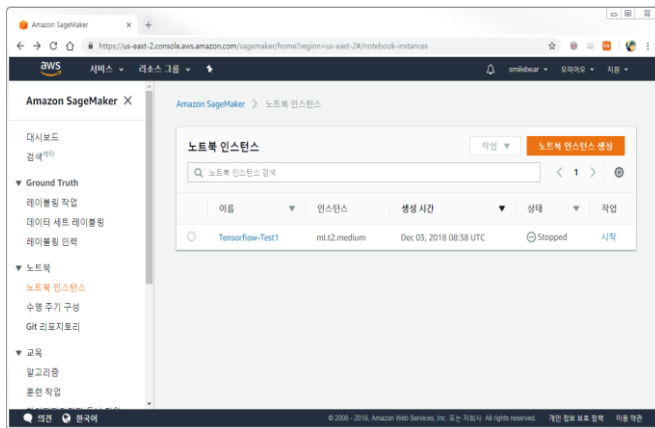
AWS(Amazon Web Services)

- AWS SageMaker 실행
 - Jupyter notebook 실행 후 폴더 및 파일 생성
 - 소스코드 입력 후 실행하여 정상적으로 출력이 되는지 확인



AWS(Amazon Web Services)

- AWS SageMaker 실행
 - 작업완료 후에는 반드시 서비스를 중지하거나 삭제해야함
 - 실행 상태 유지 시 많은 비용이 발생할 수 있음
 - 중지 : 작업한 서비스를 선택하여 작업 환경 변경
 - 삭제 : 인스턴스 이름을 클릭하여 상단에 있는 삭제 버튼 클릭



도커(Docker)

- 도커(Docker)란
 - 닷클라우드라는 기업 내부 프로젝트로 시작하여 개발된 플랫폼
 - 컨테이너 기반의 오픈 소스 가상화 플랫폼으로 2013년 3월에 오픈소스로 출시
 - 계층화된 파일 시스템을 사용하여 가상화된 컨테이너의 변경 사항을 모두 추적하고 관리
 - 컨테이너에는 라이브러리, 시스템 도구, 코드, 런타임 등 소프트웨어를 실행하는데 필요한 것들이 포함

❑ 컨테이너(Container)

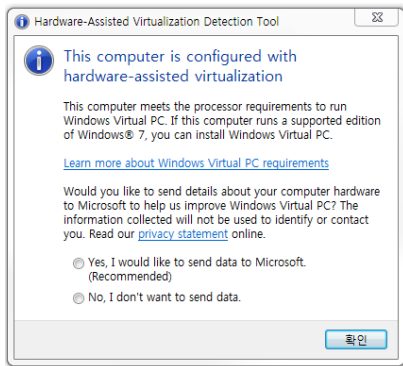
- ❑ 격리된 공간에서 프로세스가 동작하는 기술
- ❑ 애플리케이션에 필요한 바이너리 및 라이브러리를 다른 컨테이너와 커널을 공유하고 구분된 사용자 공간에서 프로세스가 실행됨

❑ 가상머신(Virtual Machine)

- ❑ 애플리케이션에 필요한 바이너리 및 라이브러리를 게스트 os에 모두 포함시키기 때문에 용량이 크고 성능 손실이 발생함

도커(Docker)

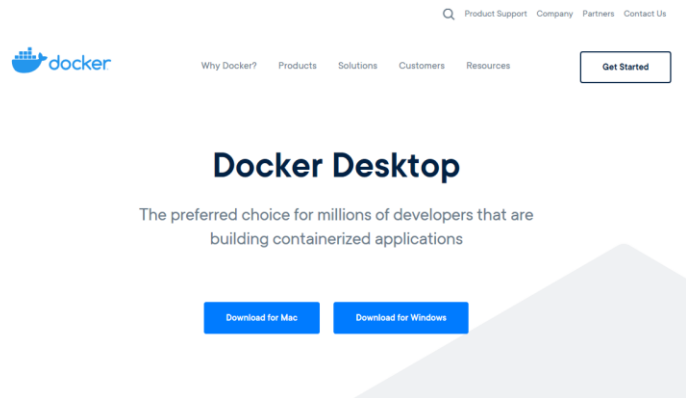
- 도커(Docker) 설치
 - 도커를 설치하기 위한 조건
 - 윈도우 7 이상(32, 64bit 모두 설치 가능)
 - 운영체제에서 가상화(Virtualization)를 지원해야 함
 - 운영체제에서 가상화 지원 여부 확인 방법
 - 윈도우 7인 경우
 - Microsoft 홈페이지(<https://www.microsoft.com/en-us/download/details.aspx?id=592>)에 접속하여 Hardware-Assisted Virtualization Detection Tool 파일 다운 후 실행



- 윈도우 8,8.1 경우 작업관리자의 성능 탭(CPU 성능)에서 가상화 사용 여부 확인

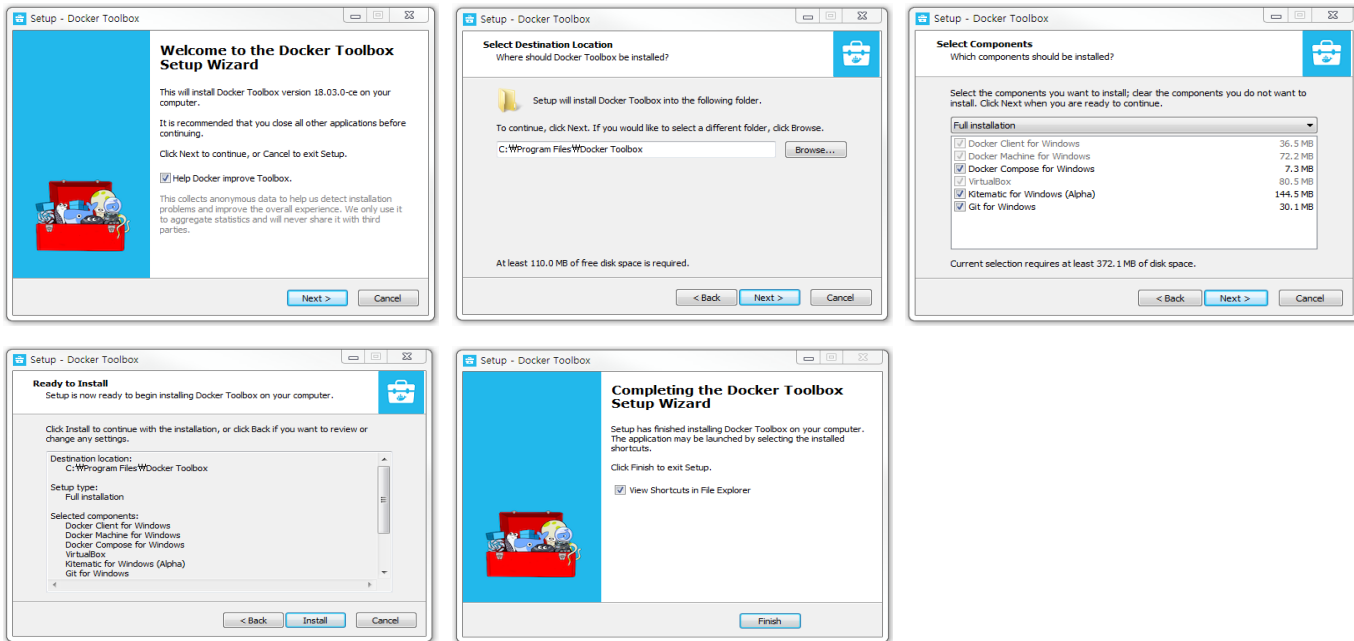
도커(Docker)

- 도커(Docker) 설치
 - 도커 공식홈페이지(<https://www.docker.com/products/docker-desktop>)에서 파일 다운로드
 - 로그인 후 파일 다운로드 가능
 - 윈도우 10 pro 혹은 Enter 버전 사용 시
 - Docker For windows 부분 참고하여 설치
 - 윈도우 7이상 10 Home 이하 버전 사용시
 - Docker ToolBox 부분 참고하여 설치



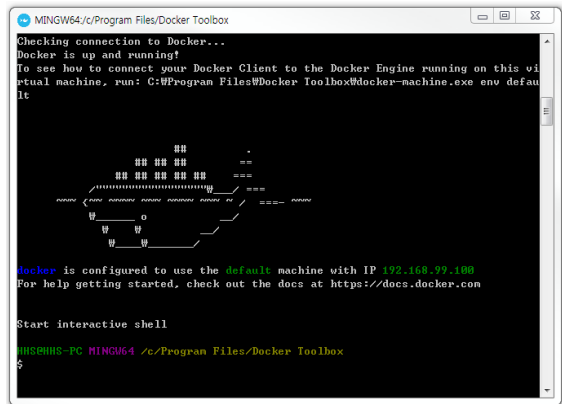
도커(Docker)

- 도커(Docker) 설치
 - 데이터 수집 동의 - 설치 경로 설정 - 설치 구성요소 설정 - 설치 준비 완료 - 설치 완료



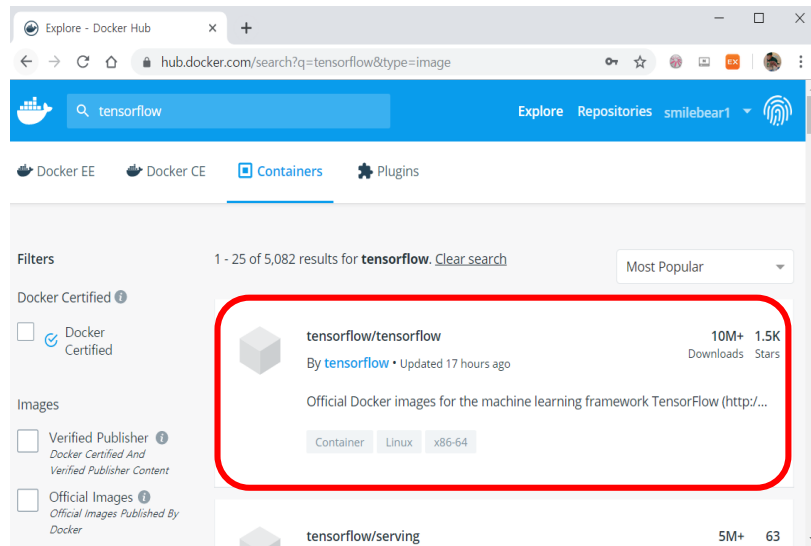
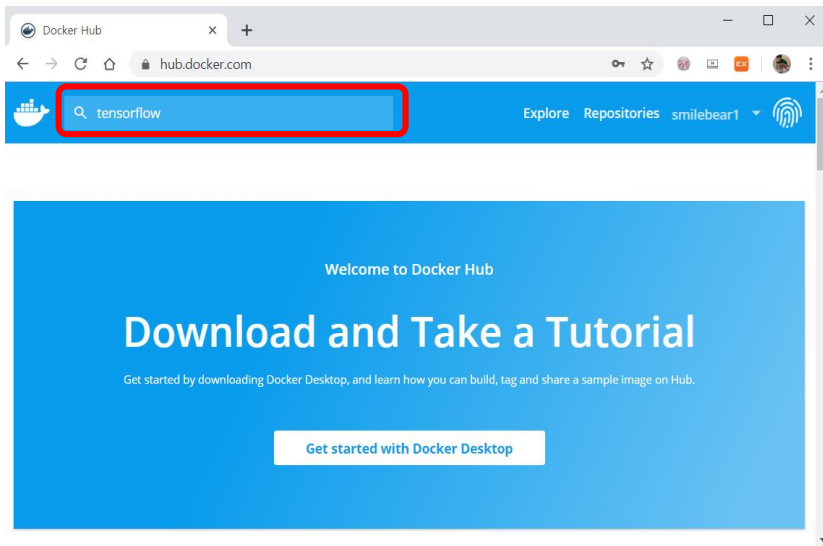
도커(Docker)

- 도커에 텐서플로우 설치
 - 도커 설치 후 생성된 Docker Quickstart Terminal 아이콘 실행
 - 도커에서 컨테이너를 실행하기 위해서는 이미지를 가져와야 함
 - 이미지는 본인이 직접 만들어서 사용할 수 있지만, 다른 사람들이 만들어 놓은 이미지를 가져다가 사용하는 것이 편함
 - 도커 이미지는 컨테이너를 실행하기 위해 필요한 파일과 설정값 등을 가지고 있음
 - 이미지를 가져와서 작업을 해도 기존 이미지가 변하는 것은 아님
 - 기존 이미지에서 본인이 필요로 하는 부분을 추가하여 새로운 이미지로 만들 수 있음
 - 도커 이미지는 Docker Hub 홈페이지(<https://hub.docker.com/>)에 등록되어 있으며, 필요한 이미지를 가져다가 사용할 수 있음



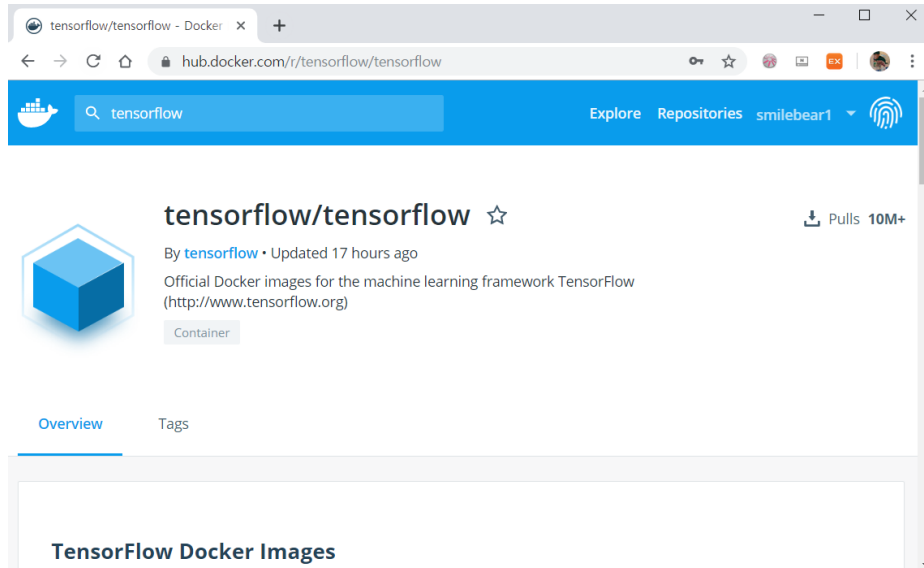
도커(Docker)

- 도커에 텐서플로우 설치
 - Docker Hub에 접속하여 tensorflow 이미지 검색
 - Tensorflow 관련 이미지 목록 출력
 - 원하는 이미지를 클릭하면 자세한 설명 확인 가능



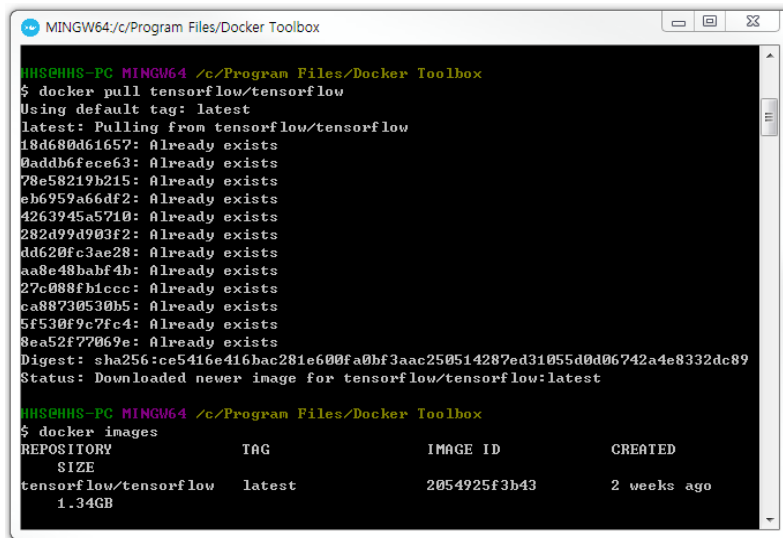
도커(Docker)

- 도커에 텐서플로우 설치
 - 사용하고자하는 이미지를 클릭하면 자세한 설명이 나와 있음
 - 이미지를 가져오는 방법
 - 가져온 이미지를 사용하여 컨테이너를 실행하는 방법



도커(Docker)

- 도커에 텐서플로우 설치
 - 도커에서 Tensorflow 이미지를 가져오는 명령어
 - 도커에서는 명령어를 사용할 때 docker ~ 형식으로 시작
 - (형식) docker <명령어> <이미지 이름> : <태그>
 - \$docker pull tensorflow/tensorflow
 - 도커에 이미지를 가져온 후 이미지 목록 확인하기
 - 이미지 목록 확인하는 명령어
 - \$docker images



The screenshot shows a Windows command prompt window titled "MINGW64/c:/Program Files/Docker Toolbox". The user is at the "NHSNHS-PC" prompt. The first command executed is `$ docker pull tensorflow/tensorflow`, which pulls the latest version of the TensorFlow image. The output shows the image ID and a list of layers that are already present. The second command is `$ docker images`, which displays a table of local Docker images.

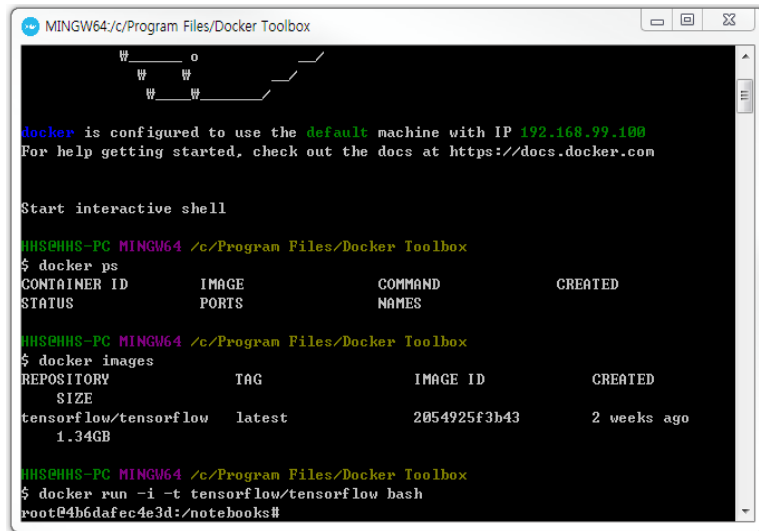
```
MINGW64/c:/Program Files/Docker Toolbox
NHSNHS-PC MINGW64 /c:/Program Files/Docker Toolbox
$ docker pull tensorflow/tensorflow
Using default tag: latest
latest: Pulling from tensorflow/tensorflow
18d680d61657: Already exists
0addb6fece63: Already exists
78e58219b215: Already exists
eb6959a66df2: Already exists
4263945a5710: Already exists
282d99d903f2: Already exists
dd620fc3ac28: Already exists
aa8e48baf4b: Already exists
27c088fbiccc: Already exists
ca88730530b5: Already exists
5f530f9c7fc4: Already exists
8ea52f77069e: Already exists
Digest: sha256:ce5416e416bac281e600fa0bf3aac250514287ed31055d0d06742a4e8332dc89
Status: Downloaded newer image for tensorflow/tensorflow:latest

NHSNHS-PC MINGW64 /c:/Program Files/Docker Toolbox
$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED
tensorflow/tensorflow	latest	2054925f3b43	2 weeks ago
		1.34GB	

도커(Docker)

- 도커에 텐서플로우 설치
 - 가져온 Tensorflow 이미지로 컨테이너를 실행하는 명령어
 - (형식) `docker <명령어> <옵션> <이미지 이름>`
 - `$docker run -i -t tensorflow/tensorflow bash`
 - 옵션 설명
 - `i` : interactive이며, 표준 입출력을 키보드와 화면을 통해 가능하도록 하는 옵션
 - `t` : tty이며, 텍스트 기반의 터미널 환경을 제공해 주는 옵션
 - `bash` : 컨테이너를 사용할 때 bash 쉘을 사용하겠다는 의미



```
MINGW64/c/Program Files/Docker Toolbox

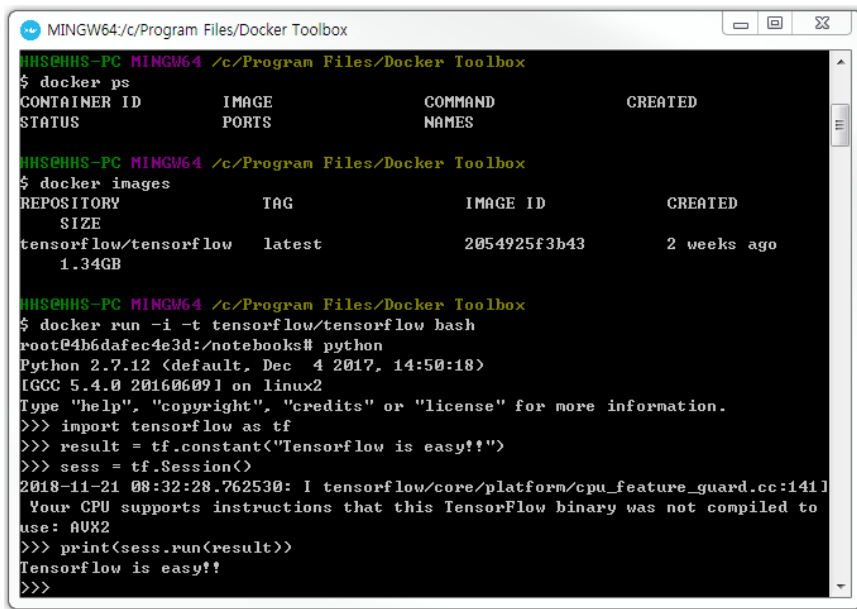
docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com

Start interactive shell

MHSEHNS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
MHSEHNS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
tensorflow/tensorflow  latest            2054925f3b43       2 weeks ago
1.34GB
MHSEHNS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker run -i -t tensorflow/tensorflow bash
root@4b6dafec4e3d:/notebooks#
```

도커(Docker)

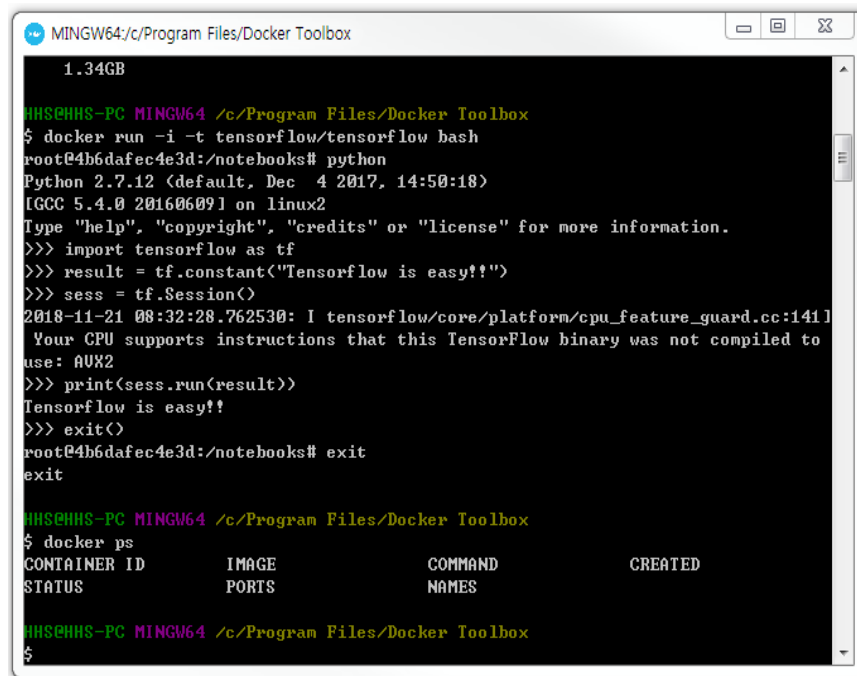
- 도커에 텐서플로우 실행
 - 현재 도커에 컨테이너가 실행되고 있는지 확인 명령어
 - 명령어 : `docker ps`
 - tensorflow 컨테이너 실행 후 테스트
 - 컨테이너를 실행하면 bash 셸을 사용할 수 있는 환경이 제공됨
 - python 실행 후 소스코드 입력
 - 정상적으로 작동되는지 확인



```
MINGW64/c/Program Files/Docker Toolbox
MSSEHHS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS
MSSEHHS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
tensorflow/tensorflow  latest             2054925f3b43       2 weeks ago
1.34GB
MSSEHHS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker run -i -t tensorflow/tensorflow bash
root@4b6dafec4e3d:/notebooks# python
Python 2.7.12 (default, Dec  4 2017, 14:50:18)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> result = tf.constant("Tensorflow is easy!!")
>>> sess = tf.Session()
2018-11-21 08:32:28.762530: I tensorflow/core/platform/cpu_feature_guard.cc:141
Your CPU supports instructions that this TensorFlow binary was not compiled to
use: AVX2
>>> print(sess.run(result))
Tensorflow is easy!!
>>>
```

도커(Docker)

- 도커에 텐서플로우 실행
 - 컨테이너 사용 후에는 종료를 해주어야 함
 - 백그라운드에서 계속 실행되고 있기 때문에 불필요한 자원 낭비
 - python 사용 후 exit()를 이용하여 종료
 - bash 셸 환경에서 exit 명령어로 컨테이너 종료
 - 컨테이너가 정상적으로 종료되었는지 docker ps 명령어를 이용하여 확인



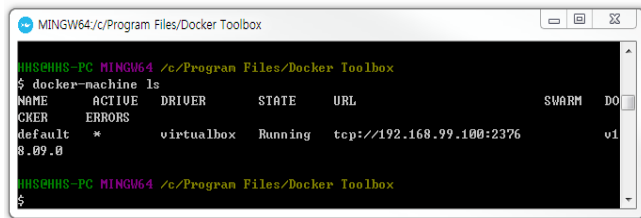
```
MINGW64/c/Program Files/Docker Toolbox
1.34GB

HNS@HNS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker run -i -t tensorflow/tensorflow bash
root@4b6dafec4e3d:/notebooks# python
Python 2.7.12 <default, Dec  4 2017, 14:50:18>
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> result = tf.constant("Tensorflow is easy!!")
>>> sess = tf.Session()
2018-11-21 08:32:28.762530: I tensorflow/core/platform/cpu_feature_guard.cc:141
Your CPU supports instructions that this TensorFlow binary was not compiled to
use: AVX2
>>> print(sess.run(result))
Tensorflow is easy!!
>>> exit()
root@4b6dafec4e3d:/notebooks# exit
exit

HNS@HNS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS             PORTS              NAMES
HNS@HNS-PC MINGW64 /c/Program Files/Docker Toolbox
$
```

도커(Docker)

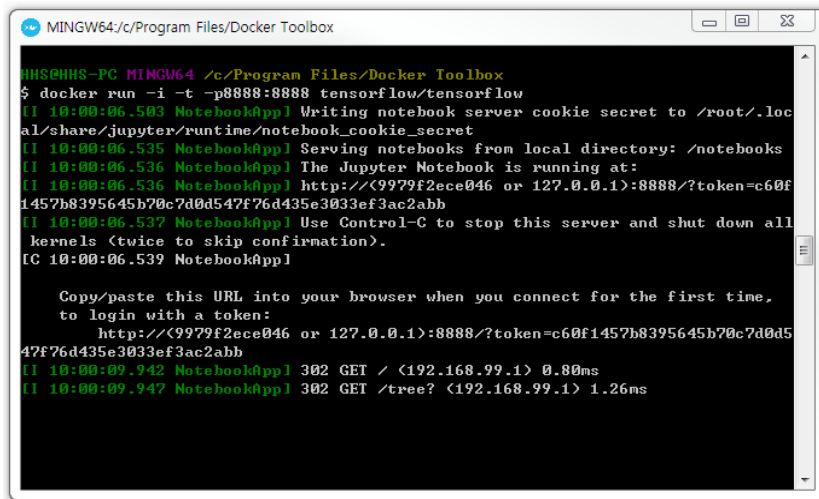
- 도커에 Jupyter notebook 사용
 - tensorflow 컨테이너 실행 시 옵션 추가
 - 명령어 : `$docker run -i -t -p 8888:8888 tensorflow/tensorflow`
 - 옵션 설명
 - `p` : 컨테이너에 서비스를 구동한 후 외부에서 해당 서비스에 접근하기 위한 옵션
 - 앞쪽 `8888` : 도커를 설치한 호스트의 포트번호
 - 뒤쪽 `8888` : 컨테이너의 포트번호
 - `dockermachine ip` 번호 확인
 - Jupyter notebook 사용을 위해 `dockermachine ip`를 확인해야함
 - `dockermachine ip:8888`로 접속하여 Jupyter notebook 사용
 - 명령어 : `docker-machine ls`



```
MINGW64/c/Program Files/Docker Toolbox
$ docker-machine ls
NAME      ACTIVE   DRIVER        STATE     URL                         SWARM   DOCKER
 ERRORS
default   *        virtualbox     Running   tcp://192.168.99.100:2376   v1
3.09.0
```


도커(Docker)

- 도커에 Jupyter notebook 사용
 - Jupyter notebook을 사용하기 위해 명령어를 입력하면 URL 정보가 제공됨
 - 제공된 URL로 접속하면 Jupyter notebook 사용 가능
 - 다른 방법으로는 dockermachine ip:port로 접속하여 제공받은 token 정보 입력

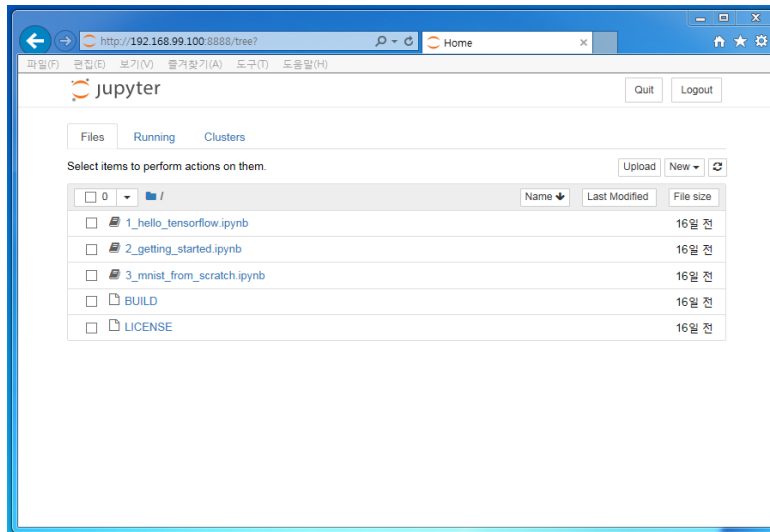
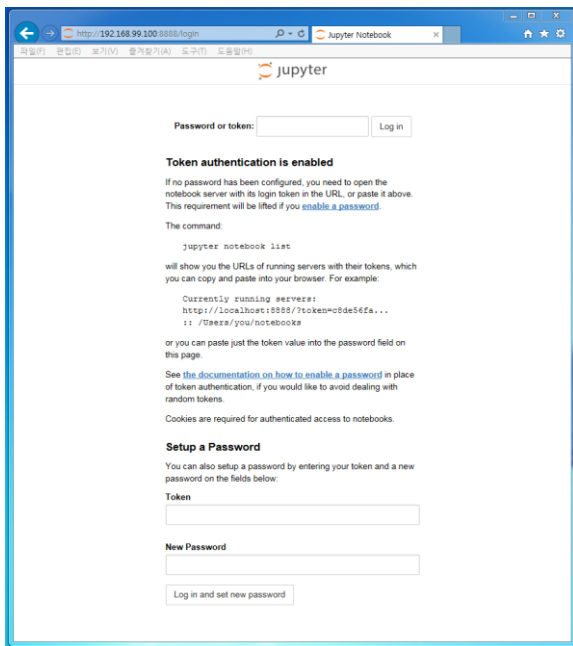


```
MINGW64/c/Program Files/Docker Toolbox
HHSENHS-PC MINGW64 /c/Program Files/Docker Toolbox
$ docker run -i -t -p8888:8888 tensorflow/tensorflow
[1 10:00:06.503 NotebookApp] Writing notebook server cookie secret to /root/.local/share/jupyter/runtime/notebook_cookie_secret
[1 10:00:06.535 NotebookApp] Serving notebooks from local directory: /notebooks
[1 10:00:06.536 NotebookApp] The Jupyter Notebook is running at:
[1 10:00:06.536 NotebookApp] http://<9979f2ece046 or 127.0.0.1>:8888/?token=c60f1457b8395645b70c7d0d547f76d435e3033ef3ac2abb
[1 10:00:06.537 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[1 10:00:06.539 NotebookApp]

Copy/paste this URL into your browser when you connect for the first time,
to login with a token:
    http://<9979f2ece046 or 127.0.0.1>:8888/?token=c60f1457b8395645b70c7d0d547f76d435e3033ef3ac2abb
[1 10:00:09.942 NotebookApp] 302 GET / <192.168.99.1> 0.80ms
[1 10:00:09.947 NotebookApp] 302 GET /tree? <192.168.99.1> 1.26ms
```

도커(Docker)

- 도커에 Jupyter notebook 사용
 - Jupyter notebook에 접속하여 사용
 - 사용 후에는 반드시 컨테이너 종료해야함



Google Colaboratory

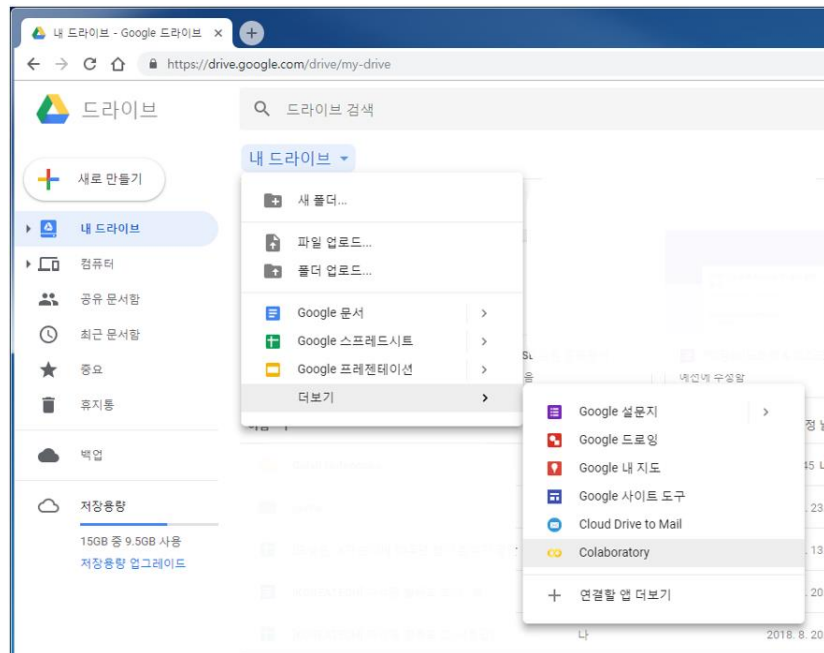
- Google Colaboratory이란
 - Google에서 제공하는 클라우드 서비스
 - Google 내부에서 직원들이 연구와 교육 목적으로 개발한 서비스
 - Jupyter notebook 기반으로 구축되었음
 - 작업환경을 하나의 공동 작업 문서로 통합하여 제공해주는 머신러닝 및 연구를 위한 데이터 분석 도구
 - 웹 브라우저에 접속하여 Jupyter notebook으로 사용

Google Colaboratory

- Jupyter notebook과의 차이점
 - 가장 큰 차이점은 사용자 측면에서의 편리함
 - Jupyter notebook
 - Jupyter notebook은 별도의 프로그램을 설치해야함
 - Colaboratory
 - 프로그램 설치없이 구글 계정과 웹 브라우저만 있으면 무료로 사용할 수 있음
 - 클라우드 환경에서 작업하기 때문에 다른 사람들과 협업할 수 있음
 - 본인 계정 전용의 가상 머신(docker)에서 실행

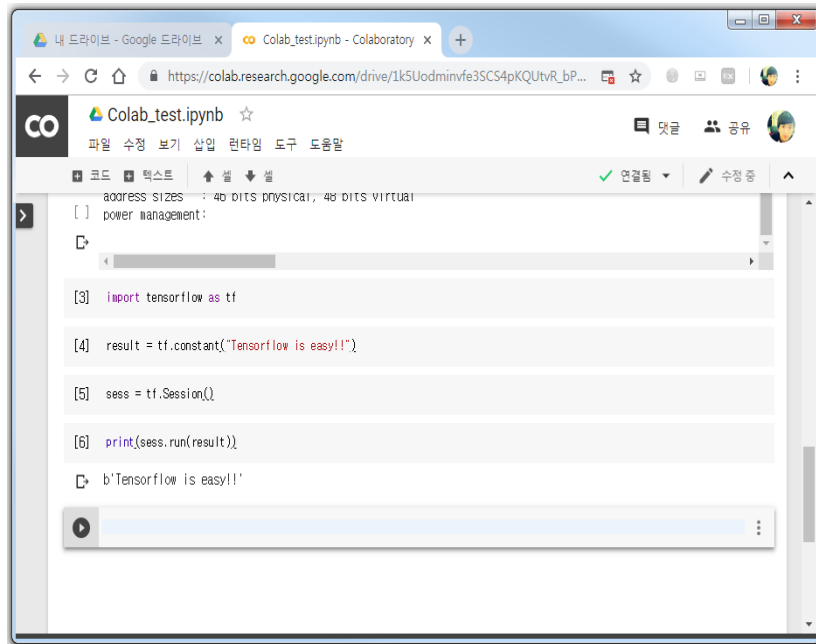
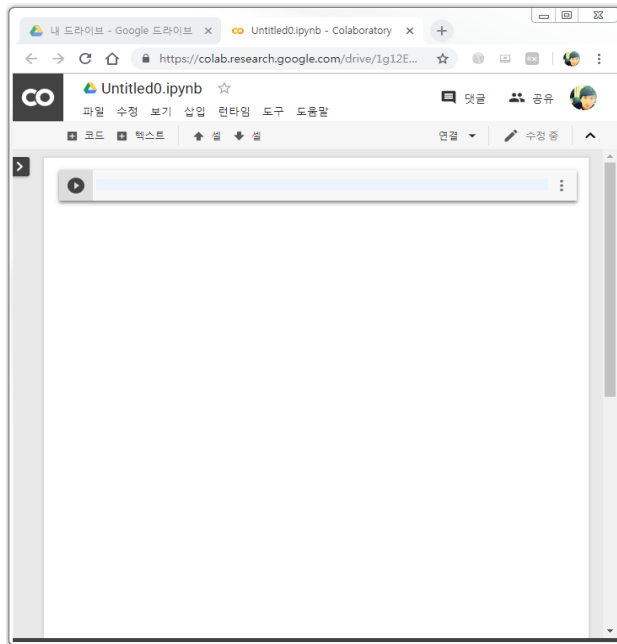
Google Colaboratory

- Google Colaboratory 설정
 - Google 계정에 로그인하여 구글 드라이브 접속
 - [내 드라이브] - [더보기] - [Colaboratory] 실행



Google Colaboratory

- Google Colaboratory 설정
 - Jupyter notebook과 비슷하여 사용하기 편리함



Microsoft Azure

- Azure란
 - Microsoft 데이터 센터에서 응용 프로그램을 빌드 및 배포하고 관리할 수 있는 개방형 클라우드 플랫폼
 - 다른 클라우드와 다르게 일관된 하이브리드 클라우드
 - 하이브리드 클라우드란
 - 퍼블릭 클라우드와 프라이빗 클라우드 환경이 조합됨
 - 퍼블릭 클라우드 : 인터넷을 통해 다수의 고객들에게 서비스를 제공하는 클라우드
 - 프라이빗 클라우드 : 자체적으로 만들어 사용할 수 있는 서비스를 제공하는 클라우드
 - 온프레미스 및 클라우드 환경에서 동작이 되도록 설계된 플랫폼, 도구, 서비스를 통해 위험을 줄임
 - 보안 및 개인 정보 요구사항을 명확하게 설정
 - 고객들의 데이터를 안전하게 유지

Microsoft Azure

- Azure 비용
 - AWS 처럼 사용한 만큼 비용을 지불해야함
 - 체험 계정을 통해 12개월 무료로 사용 가능함
 - 12개월 간 무료로 제공되는 서비스도 있으며, 항상 무료인 서비스도 제공
 - 체험계정 생성 후 30일동안 모든 서비스를 탐색할 수 있도록 224,930원 크레딧 제공

Microsoft Azure

지금 Azure 체험 계정 만들기 | X

https://azure.microsoft.com/ko-kr/free/

영입팀에 문의: 080-822-1183 | 검색 | 내 계정 | 포털 | 로그인

개요 | 솔루션 | 제품 | 설명서 | 가격 | 교육 | Marketplace | 파트너 | 지원 | 블로그 | 기타

지금 Azure 체험 계정 만들기

12개월 체험 서비스 시작

체험 계정 만들기 >

또는 지금 바로 구입 >

어떤 것들이 포함되어 있습니까?

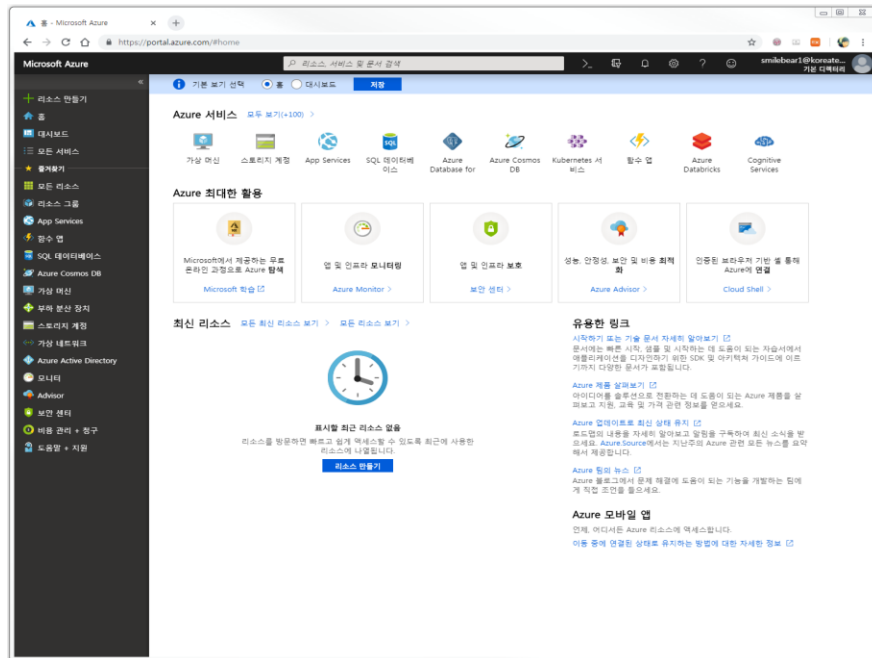
Azure 체험 계정에는 다음과 같은 내용이 포함되어 있으며, 업그레이드를 선택하기 전까지는 요금이 청구되지 않습니다.

12개월 언제 무료 서비스	+	₩224,930 (대략) 크레딧 30일 동안 모든 Azure 서비스 탐색	+	항상 무료 제공 25개 이상의 서비스
--------------------------	---	--	---	--------------------------------

Microsoft에 대해서 보내기

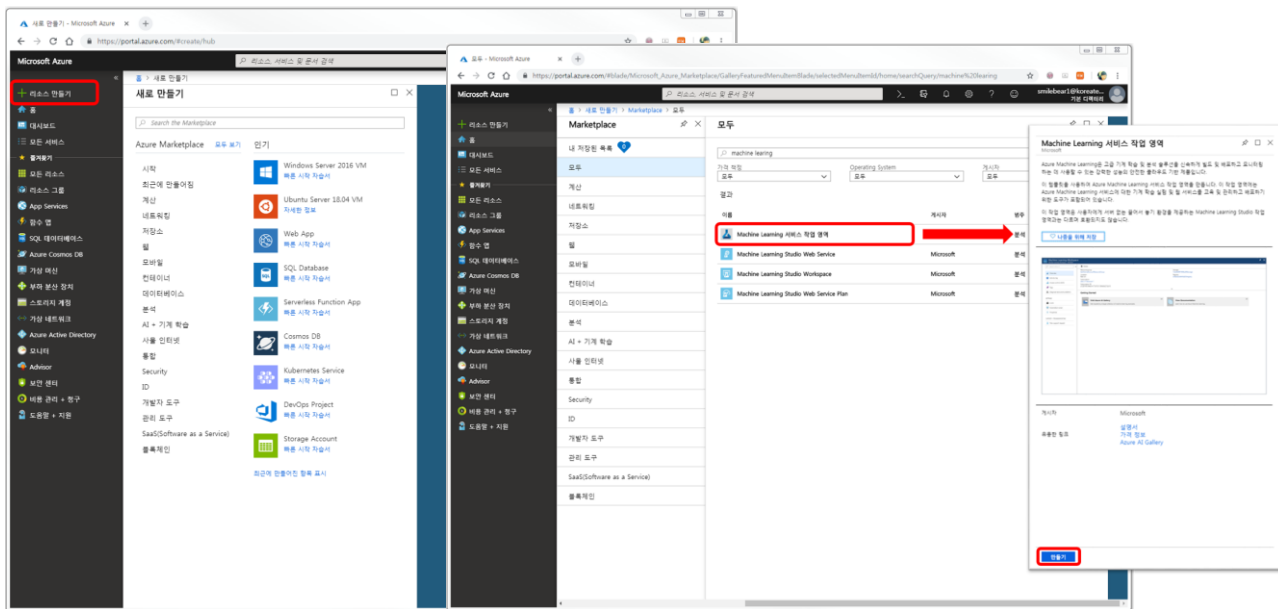
Microsoft Azure

- Azure 설정
 - 체험계정 생성 후 로그인
 - Azure portal은 Chrome 및 Edge, Firefox 브라우저에서만 지원
 - 좌측메뉴는 자주 사용하는 서비스 목록
 - 메인 화면에는 서비스 및 최신 리소스 목록 등 다양한 정보 제공



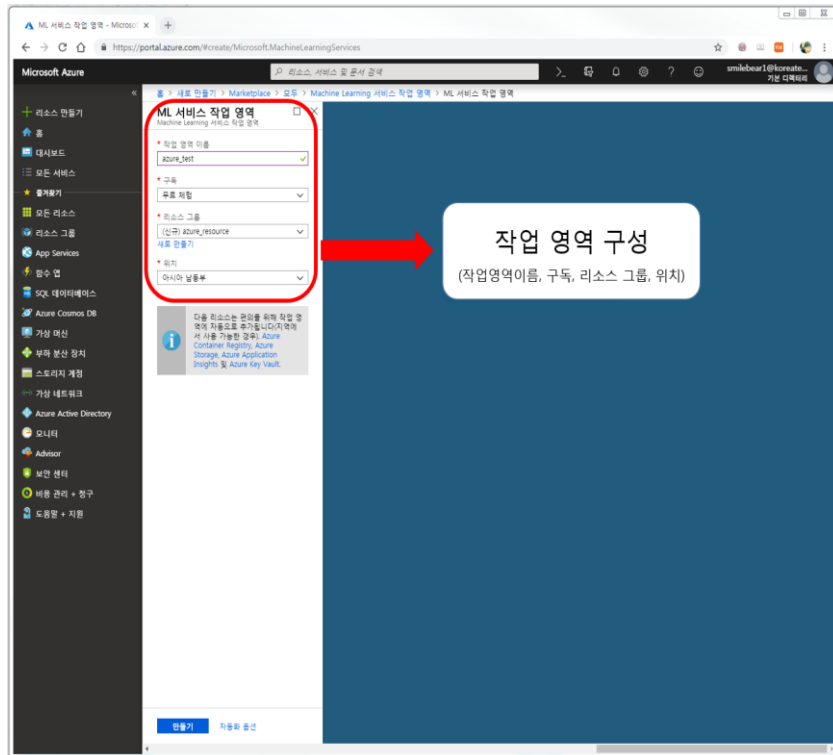
Microsoft Azure

- Azure 설정
 - 좌측 메뉴에 있는 [리소스 만들기] 선택
 - 'machine learning' 검색
 - Machine Learning 서비스 작업 영역 선택 - 설명 확인 후 만들기 버튼 클릭



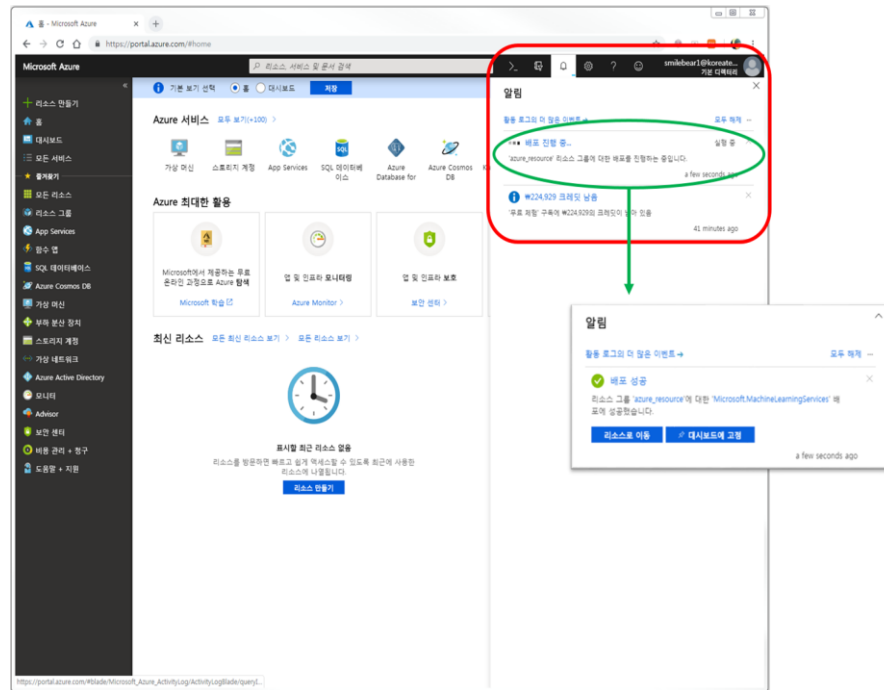
Microsoft Azure

- Azure 설정
 - ML 서비스 작업 영역 구성
 - 이름 및 구독, 리소스 그룹, 위치 설정
 - 이름 : 고유 이름 설정
 - 구독 : Azure 구독 선택
 - 무료 체험 선택
 - 리소스 그룹 : Azure 솔루션에 관련된 리소스를 보유하는 컨테이너
 - 위치 : 사용자 및 데이터 리소스와 가장 가까운 위치 선택



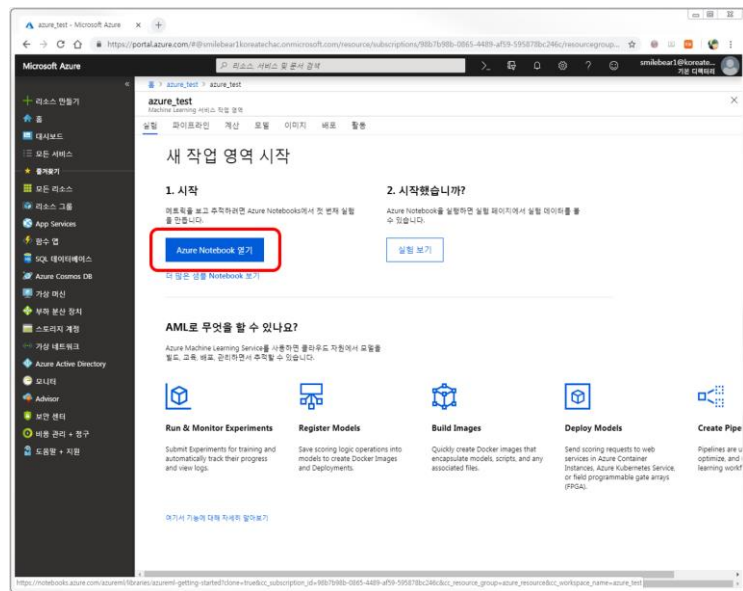
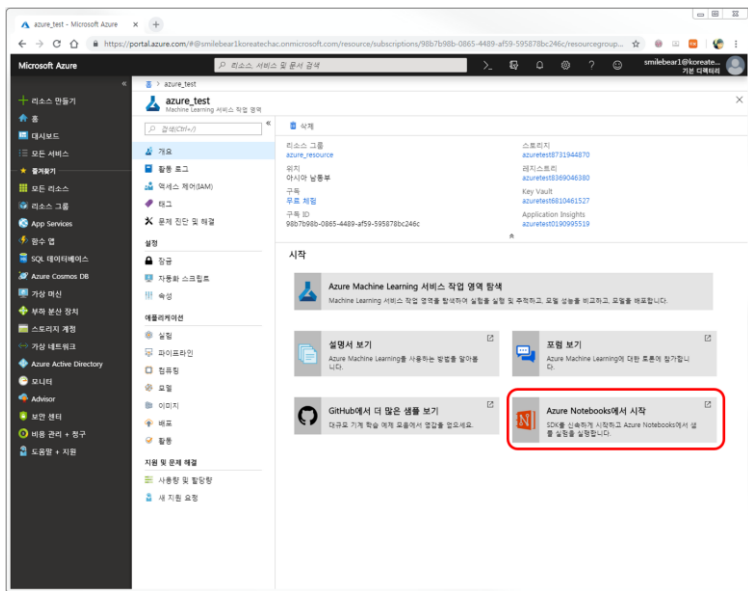
Microsoft Azure

- Azure 설정
 - 작업 영역 구성 후 리소스 배포과정을 통해 최종적인 리소스 생성
 - 배포 완료 시 우측 상단에 알림으로 확인 가능
 - 크레딧 정보도 확인할 수 있음



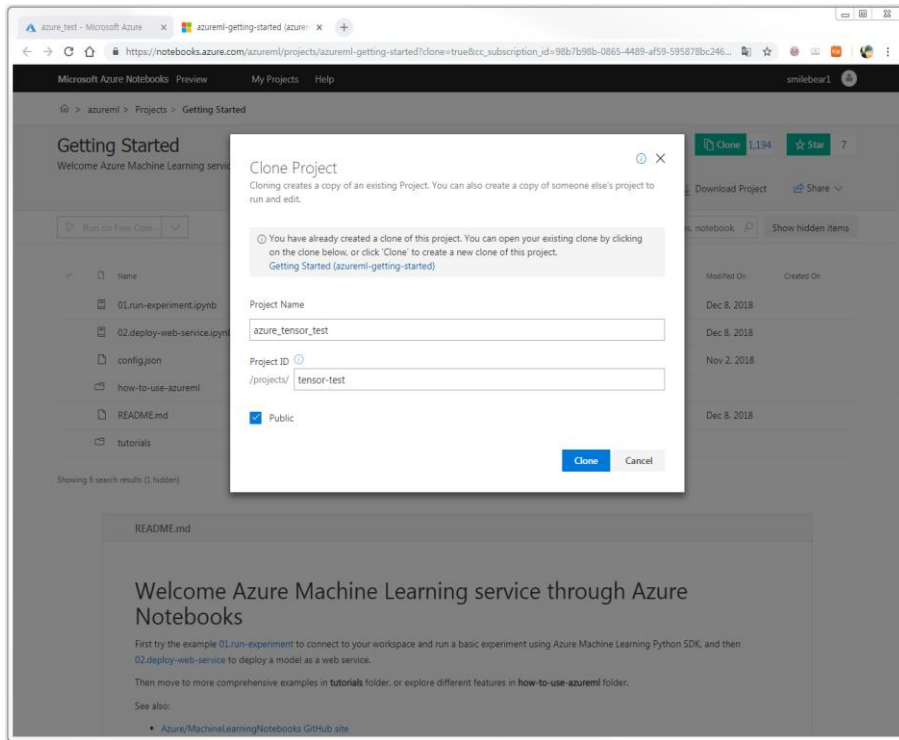
Microsoft Azure

- Azure 설정
 - 생성된 리소스 클릭 후 Azure Notebook 선택
 - Azure Notebook 열기 선택



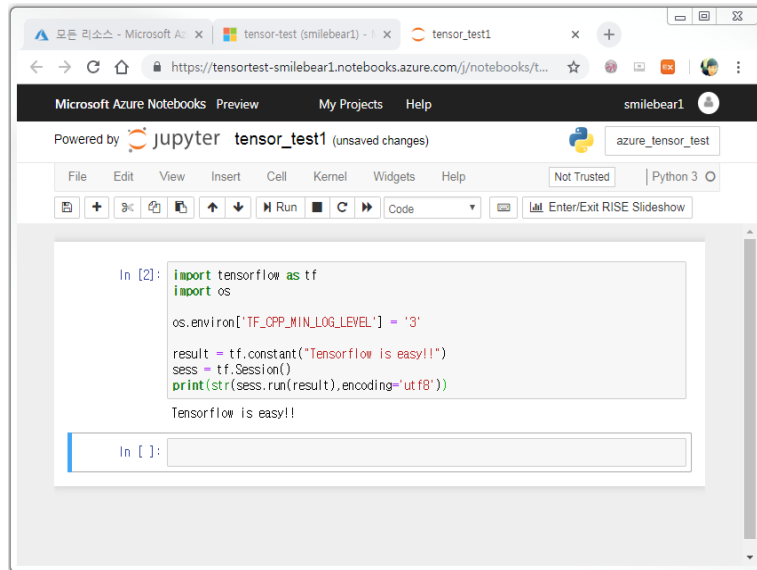
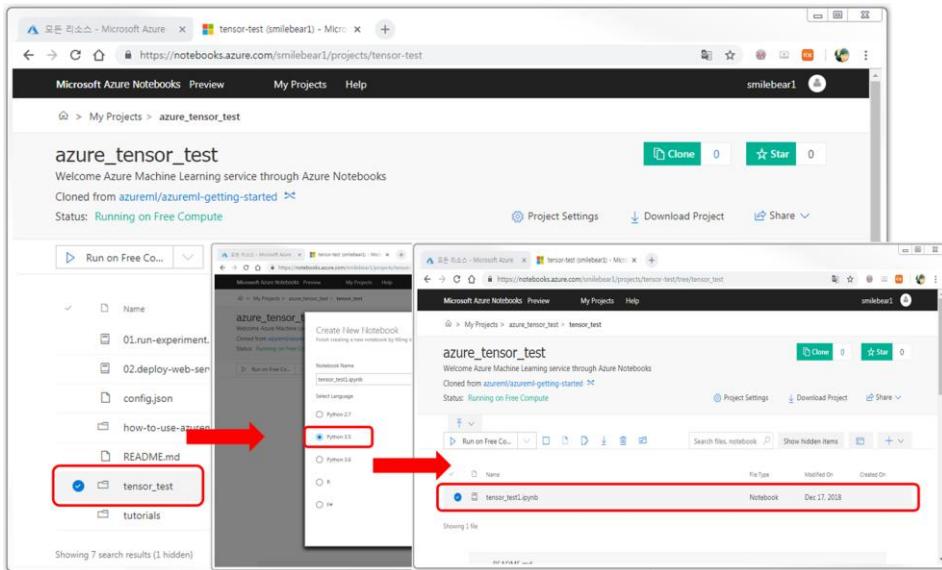
Microsoft Azure

- Azure 설정
 - 프로젝트 명 및 ID 입력
 - 프로젝트 ID는 다른 사람들과 공유하고 싶을 경우 사용하는 고유한 이름



Microsoft Azure

- Azure 설정
 - 프로젝트 생성 후 작업 폴더 생성
 - 테스트 파일을 만들어 Tensorflow 소스코드 입력



Microsoft Azure

- Azure 설정
 - 작업종료 후에는 반드시 서비스 중단
 - 서비스가 계속 실행 중이면 사용한 만큼 비용 발생
 - My Projects에서 프로젝트 목록 확인
 - 작업했던 프로젝트 상태 확인 후 Shutdown 버튼을 이용하여 중지
 - Azure portal 왼쪽 메뉴 - 모든 리소스 - 생성된 모든 리소스 목록 중 사용하지 않는 리소스 삭제

