



Amazon SageMaker Hands-on Guidebook

8 Dec. 2020

Table of Contents

준비 조건	2
Module 1: Notebook Instance 생성하기	3
Module 2. Git clone 으로 Source Code 복제하기	7
Module 3: Sagemaker Example Code 복제 하기	9

준비 조건

- AWS 계정: AWS IAM, S3, SageMaker 자원을 생성할 수 있는 권한이 필요합니다.
- AWS Region: SageMaker 는 지원되는 region 은 <https://aws.amazon.com/about-aws/global-infrastructure/regional-product-services/> 에서 확인하실 수 있습니다.
이번 HoL에서는 **US-East-1 (North Virgin)** 에서 실행합니다.
- Browser: 최신 버전의 **Chrome, Firefox** 를 사용하세요.

※ 주의 사항: Notebook 안의 Cell 에서 코드 실행후 결과 값이 나오는 데는 수 초가 걸립니다. 훈련 Job 을 실행하는 경우 수 분이 걸릴 수도 있습니다. 실습 완료 후에는 아래 가이드에 따라 생성된 자원을 꼭 종료/삭제해 주세요.

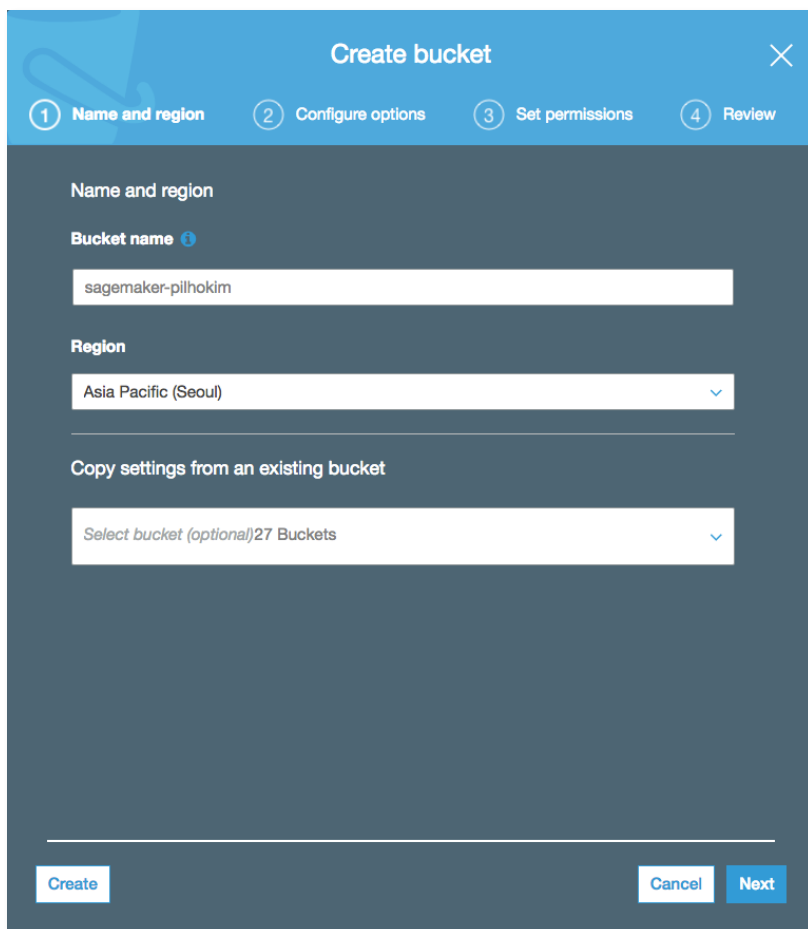


Module 1: Notebook Instance 생성하기

1. S3 Bucket 생성하기

SageMaker 는 S3 를 데이터와 모델 저장소로 사용합니다. 여기서는 해당 목적으로 S3 Bucket 을 생성합니다.

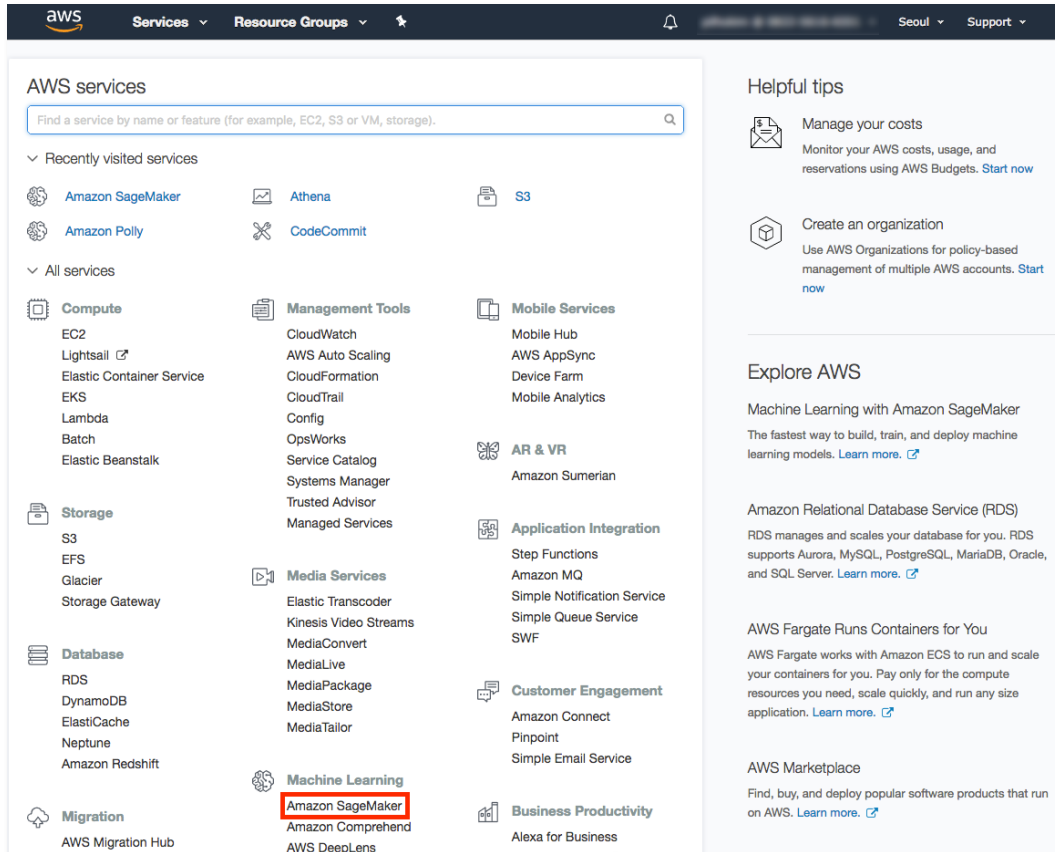
- 1) AWS 관리 콘솔 (<https://console.aws.amazon.com/>)에 Sign in 합니다.
- 2) AWS Services 리스트에서 S3 로 이동합니다.
- 3) "+ Create Bucket" 버튼을 선택합니다.
- 4) 아래 내용 설정 후 화면 왼쪽 아래 Create 클릭합니다.
 - Bucket name: sagemaker-{userid} [반드시 고유한 값 설정]
 - Region : US-East-1 (North Virginia)



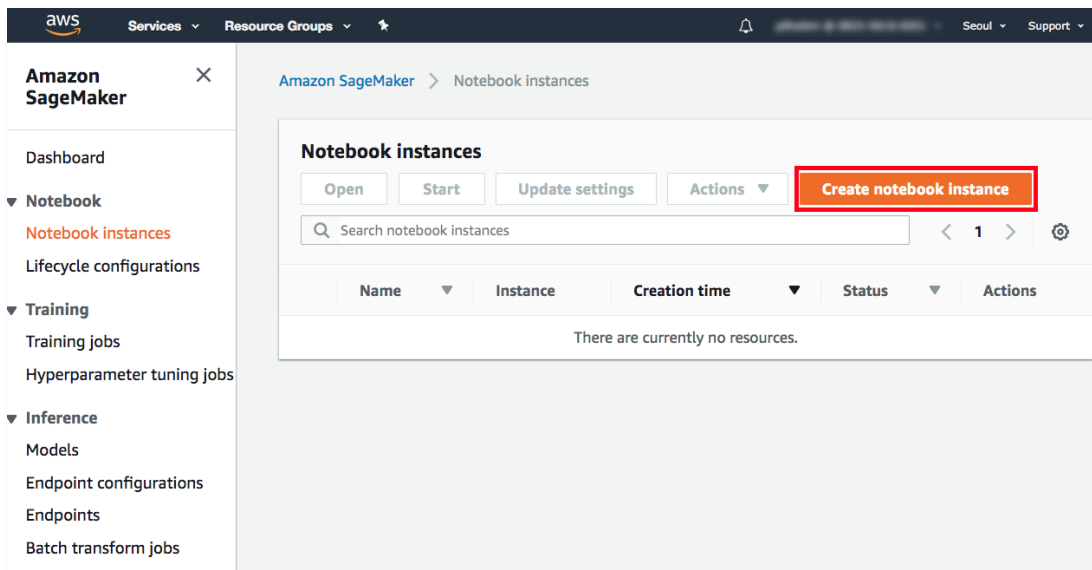
The screenshot shows the 'Create bucket' wizard in the AWS Management Console. The title bar says 'Create bucket' with a close button. Below the title bar are four steps: 1 Name and region, 2 Configure options, 3 Set permissions, and 4 Review. The first step, 'Name and region', is active. It contains a 'Bucket name' field with the value 'sagemaker-pilhokim' and a 'Region' dropdown menu set to 'Asia Pacific (Seoul)'. There is also a section for 'Copy settings from an existing bucket' with a dropdown menu showing 'Select bucket (optional) 27 Buckets'. At the bottom of the form are three buttons: 'Create', 'Cancel', and 'Next'.

2. Notebook instance 생성

1) AWS 관리 콘솔에서 오른쪽 상단에서 North Virginia Region 선택 후 AWS Services 리스트에서 Amazon SageMaker 서비스를 선택합니다.



2) 새로운 Notebook instance 를 생성하기 위해 왼쪽 패널 메뉴 중 Notebook Instances 선택 후 오른쪽 상단의 Create notebook instance 버튼을 클릭 합니다.



3) Notebook instance 이름으로 [First Name]-[Last Name]-workshop 으로 넣은 뒤 **ml.m5.xlarge** 인스턴스 타입을 선택 합니다.

The screenshot displays the 'Create notebook instance' form. It includes a title 'Create notebook instance' and a brief description of SageMaker notebook instances. The form contains several sections: 'Notebook instance settings' with fields for 'Notebook instance name' and 'Notebook instance type' (where 'ml.m5.xlarge' is selected and highlighted with a yellow box); 'IAM role' with a dropdown menu; 'VPC - optional' with a 'No VPC' selection; 'Lifecycle configuration - optional' with a 'No configuration' selection; and 'Encryption key - optional' with a 'No Custom Encryption' selection. At the bottom, there is a section for 'Tags - optional'.

4) IAM role 은 Create a new role 을 선택하고, 생성된 팝업 창에서는 S3 buckets you specify – optional 밑에 Specific S3 Bucket 을 선택 합니다. 그리고 텍스트 필드에 위에서 만든 S3 bucket 이름(예: sagemaker-xxxxx)을 선택 합니다. 이후 Create role 을 클릭합니다.

Create an IAM role

Passing an IAM role gives Amazon SageMaker permission to perform actions in other AWS services on your behalf. Creating a role here will grant permissions described by the [AmazonSageMakerFullAccess](#) IAM policy to the role you create.

The IAM role you create will provide access to:

- ☒ **S3 buckets you specify - optional**
 - ☒ **Specific S3 buckets**

 Comma delimited. ARNs, "*" and "/" are not supported.
 - ☐ Any S3 bucket
 Allow users that have access to your notebook instance access to any bucket and its contents in your account.
 - ☐ None
- ☒ Any S3 bucket with "sagemaker" in the name
- ☒ Any S3 object with "sagemaker" in the name
- ☒ Any S3 object with the tag "sagemaker" and value "true" [See Object tagging](#)
- ☒ S3 bucket with a Bucket Policy allowing access to SageMaker [See S3 bucket policies](#)

Cancel **Create role**

5) 다시 Create Notebook instance 페이지로 돌아온 뒤 Create notebook instance 를 클릭합니다.

3. Notebook Instance 접근하기

1) 서버 상태가 InService 로 바뀔 때까지 기다립니다. 보통 5 분정도의 시간이 소요 됩니다.

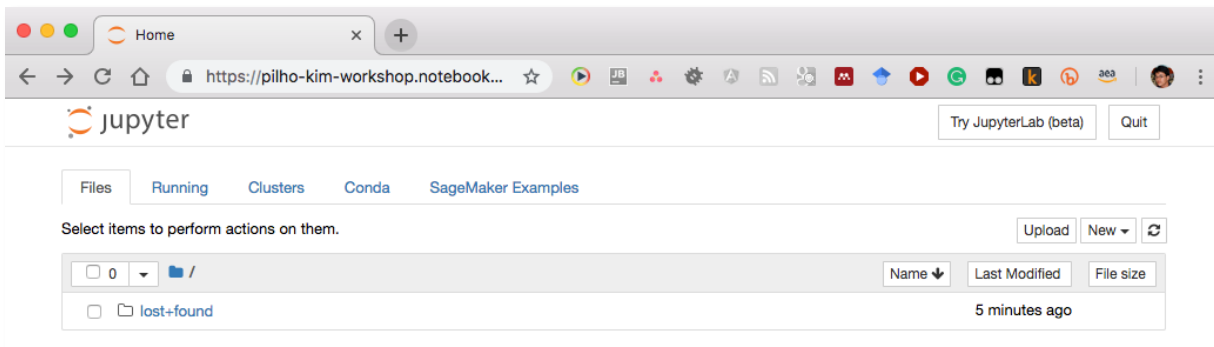
Amazon SageMaker > Notebook instances

Notebook instances [Open](#) [Start](#) [Update settings](#) [Actions](#) [Create notebook instance](#)

Search notebook instances

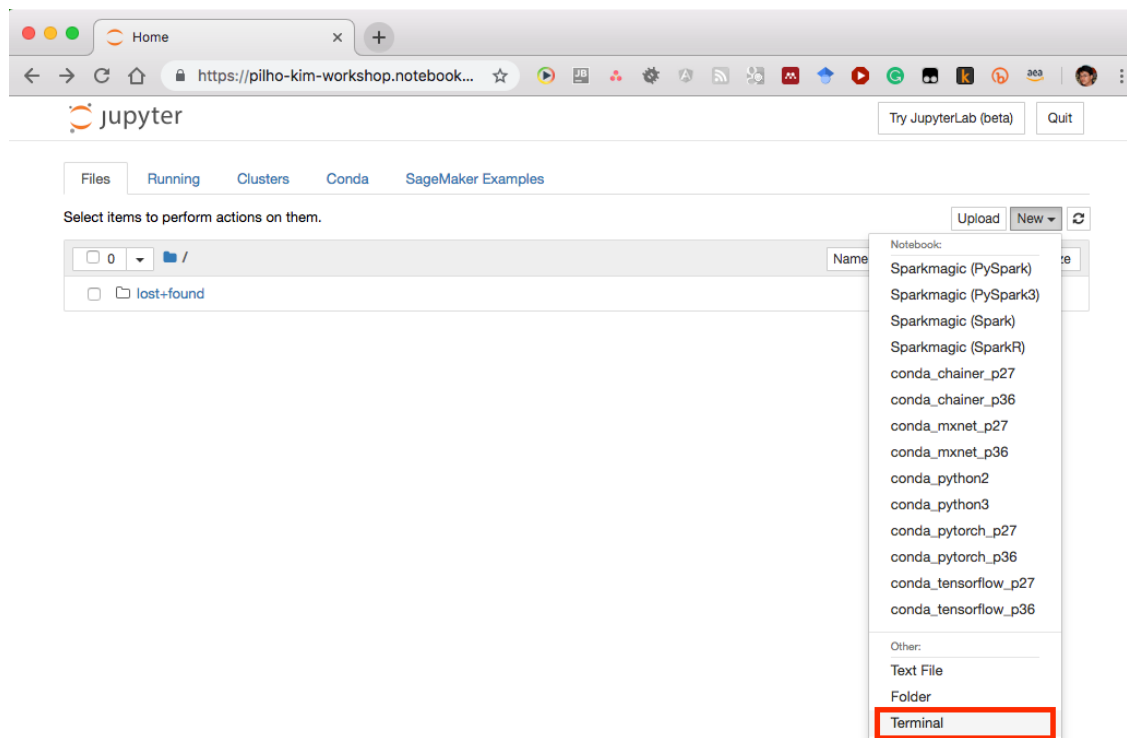
	Name	Instance	Creation time	Status	Actions
<input type="radio"/>	pilho-kim-workshop	mL.m4.xlarge	Sep 17, 2018 08:24 UTC	✔ InService	Open Stop

2) Open 을 클릭하면 방금 생성한 notebook instance 의 Jupyter 홈페이지로 이동하게 됩니다.



Moduel 2. Git clone 으로 Source Code 복제하기

SageMaker 의 Jupyter 노트북도 Linux 기반의 서버입니다. Jupyter 노트북에서 서버의 Terminal 을 바로 실행하는 기능을 제공하고 있습니다. **Error! Reference source not found.**와 같이 Terminal 을 선택합니다.



SageMaker 노트북 서버에 접속하기 위한 Terminal 실행 화면.

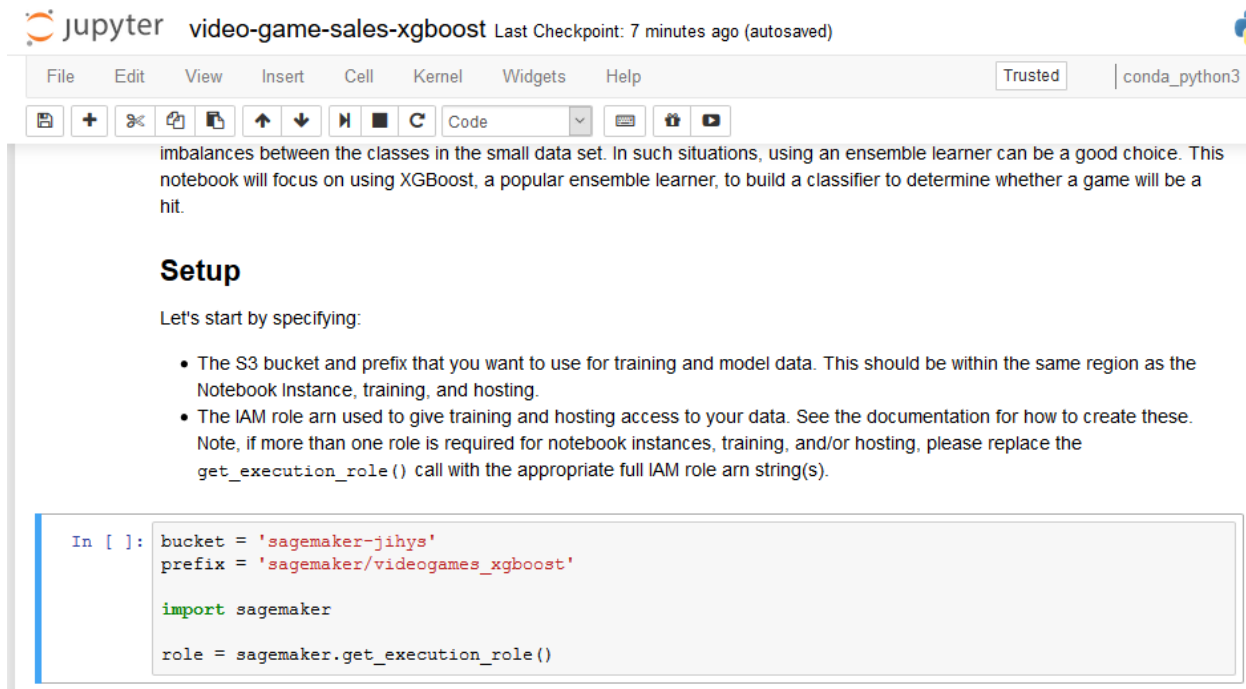
터미널이 실행되면 아래의 명령어들을 입력해서 실행합니다.

```
cd SageMaker/
git clone https://github.com/jihys/startup-sagemaker-workshop
```

실습을 위해서 현재 설치되어 있는 SageMaker 의 Jupyter 노트북을 강사자의 안내에 따라 선택하도록 합니다.

만약 Cell 에서 **bucket= '<your_s3_bucket_name_here>'**인 부분이 있다면 모듈 1 에서 만든 S3 bucket 이름(예: sagemaker-xxxxx)을 적습니다. S3://...와 같은 경로 이름은 적지 않습니다.

<예시>



imbalances between the classes in the small data set. In such situations, using an ensemble learner can be a good choice. This notebook will focus on using XGBoost, a popular ensemble learner, to build a classifier to determine whether a game will be a hit.

Setup

Let's start by specifying:

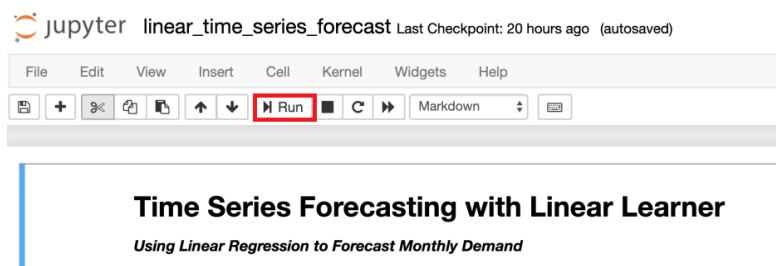
- The S3 bucket and prefix that you want to use for training and model data. This should be within the same region as the Notebook Instance, training, and hosting.
- The IAM role arn used to give training and hosting access to your data. See the documentation for how to create these. Note, if more than one role is required for notebook instances, training, and/or hosting, please replace the `get_execution_role()` call with the appropriate full IAM role arn string(s).

```
In [ ]: bucket = 'sagemaker-jihys'
        prefix = 'sagemaker/videogames_xgboost'

import sagemaker

role = sagemaker.get_execution_role()
```

Jupyter notebook 은 코드와 주석을 같이 저장합니다. Jupyter notebook 에는 두 가지의 Cell(Code Cell 과 markdown Cell)이 있습니다. Code 를 실행하려면 실행 버튼을 클릭합니다. (Control+Enter 도 동일한 기능이며, 실제 사용하는 실행 후 셀을 이동하는 Shift+Enter 가 더 편리합니다.)



linear_time_series_forecast Last Checkpoint: 20 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run

Time Series Forecasting with Linear Learner

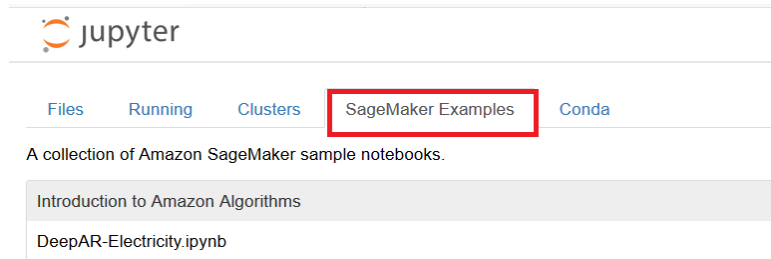
Using Linear Regression to Forecast Monthly Demand

Code 가 실행되면 Code Cell 왼쪽의 "In []" 라는 부분이 "In [*]"로 변경이 되고 완료시에는 실행 순서를 나타내는 숫자로 변경 됩니다.

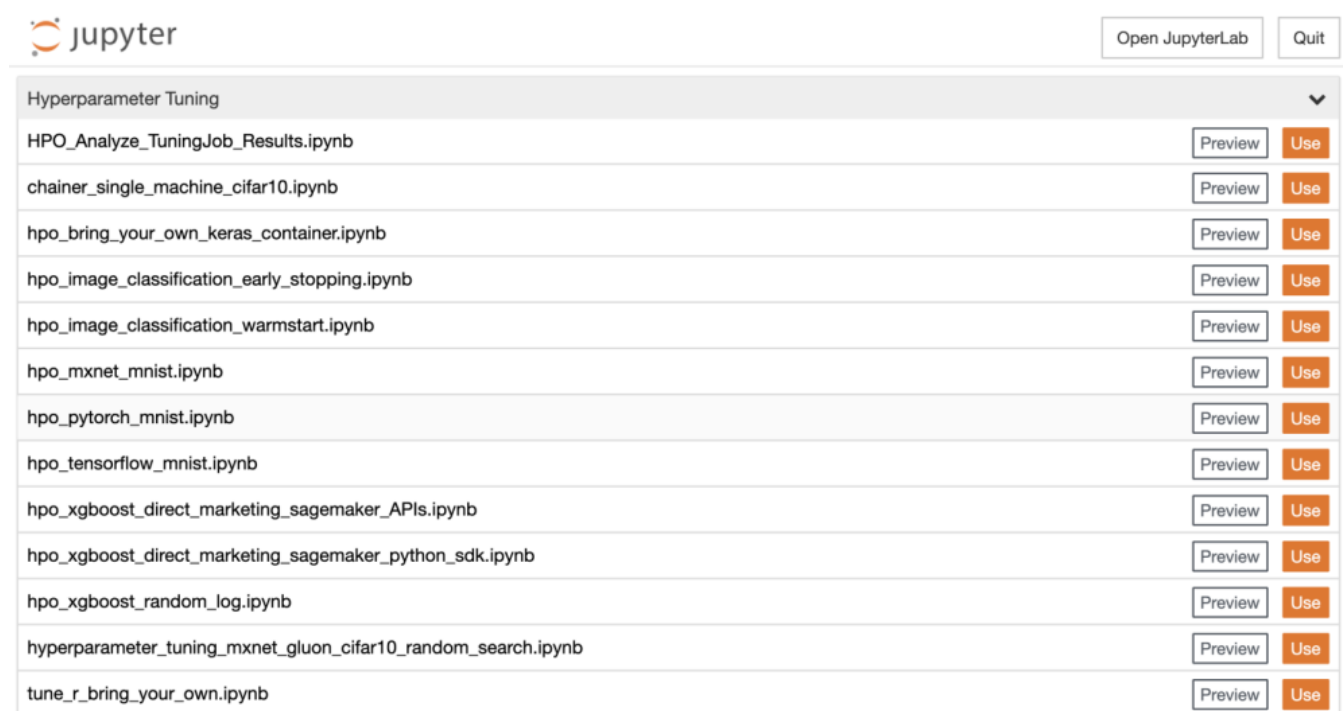
※ 코드는 Code Cell 에 나타난 순서대로 실행하고 반복 작업을 피하기 위해서 한 번만 실행합니다.
같은 훈련 job cell 을 반복 실행하게 되면 두 개의 훈련 job 을 실행하게 되어 서비스 제한을 넘을 수도 있습니다.

Module 3: Sagemaker Example Code 복제 하기

SageMaker 의 Jupyter 노트북 페이지 상단의 탭메뉴에서 "SageMaker Examples"를 클릭합니다.

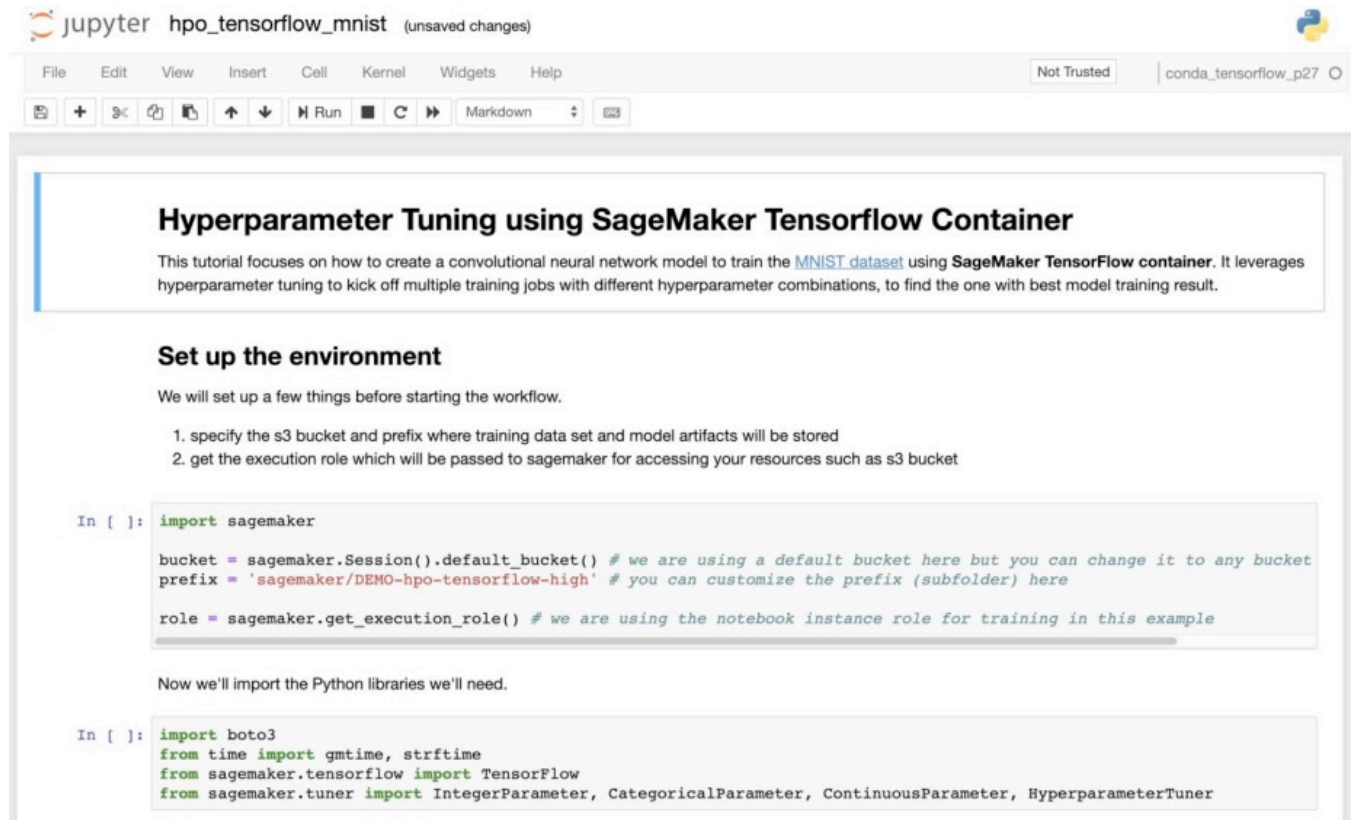


샘플 노트북 목록에서 원하는 노트북을 선택합니다.



원하는 노트북의 우측의 **Use** 버튼을 클릭합니다. 다음과 같은 팝업창에서 Create copy 버튼을 클릭하여 관련 파일들을 사용자의 홈디렉토리로 복사를 진행합니다.

새로운 브라우저 탭에서 노트북이 오픈되면 준비가 완료됩니다.



The image shows a Jupyter Notebook interface with the title "hpo_tensorflow_mnist" and a status of "(unsaved changes)". The top bar includes a menu (File, Edit, View, Insert, Cell, Kernel, Widgets, Help), a "Not Trusted" warning, and the environment "conda_tensorflow_p27". The notebook content is a markdown document titled "Hyperparameter Tuning using SageMaker Tensorflow Container". It explains that the tutorial focuses on creating a CNN model to train the MNIST dataset using SageMaker TensorFlow container, leveraging hyperparameter tuning. The first section, "Set up the environment", lists two steps: specifying the S3 bucket and prefix, and getting the execution role. Below this, there are two code blocks. The first code block imports sagemaker and sets up bucket, prefix, and role variables with comments. The second code block imports boto3, time, tensorflow, and sagemaker tuner classes. The interface includes standard Jupyter controls like run, undo, redo, and a markdown preview button.

Hyperparameter Tuning using SageMaker Tensorflow Container

This tutorial focuses on how to create a convolutional neural network model to train the [MNIST dataset](#) using **SageMaker TensorFlow container**. It leverages hyperparameter tuning to kick off multiple training jobs with different hyperparameter combinations, to find the one with best model training result.

Set up the environment

We will set up a few things before starting the workflow.

1. specify the s3 bucket and prefix where training data set and model artifacts will be stored
2. get the execution role which will be passed to sagemaker for accessing your resources such as s3 bucket

```
In [ ]: import sagemaker

bucket = sagemaker.Session().default_bucket() # we are using a default bucket here but you can change it to any bucket
prefix = 'sagemaker/DEMO-hpo-tensorflow-high' # you can customize the prefix (subfolder) here

role = sagemaker.get_execution_role() # we are using the notebook instance role for training in this example
```

Now we'll import the Python libraries we'll need.

```
In [ ]: import boto3
from time import gmtime, strftime
from sagemaker.tensorflow import TensorFlow
from sagemaker.tuner import IntegerParameter, CategoricalParameter, ContinuousParameter, HyperparameterTuner
```