

Amazon SageMaker

End-to-End ML pipeline

1. Data acquisition
2. Data preparation
3. Data Analysis
4. Modeling
5. Deploy/Productionization
6. Monitoring



→ Problem-specific
Sub-stages
& variations

End-to-End ML pipeline

1. Data acquisition
2. Data preparation
3. Data Analysis
4. Modeling
5. Deploy/Productionization
6. Monitoring

[On-premise]



DB/DW/Log Servers

compute / cluster servers

Pros & Cons of on-premise setup:

- (-) up front high hardware costs
- (-) less flexible to changes in workload
- ~~(-)~~ Administrative & ops overhead → install, update, h/w setup
- (-) security overhead
- (+) full-control of h/w, s/w & security

Cloud

[AWS, Azure, GCP]

→ renting hardware, software, Ops
as you need.

Popular Open-source tools/libraries

- Python , Matplotlib, Numpy / Scipy, Seaborn, Notebooks, Pandas
- SKlearn, Xgboost, Light GBM, CatBoost, . . .
- TF, Keras, PyTorch, Caffe
- Spark (MLlib), Dask ,
- Kubeflow, MLFlow,

	On-premise + open source tools	Cloud + opensource tools	Cloud ML platforms
Hardware	x	✓	✓
Software	x	x	✓
Security	x	(partial)	✓
Migration	easy	easy	hard

Cloud-based ML platforms

Amazon Sagemaker

Azure ML

GCP Vertex AI + Other services



Hardware

Open-source software

Ops

[well-packaged]

Pros & Cons of cloud ML platforms

- (-) lock-in ; harder to move to others
- (+) no H/w & S/w & ops issues ; focus on core -ML
- (+) best practices & tools available by default
- (+) can speed-up , ML : idea to product , by a few times-

e.g.: why we love Google Colab?

→ (Notebook Server)

No installation headaches

Accessible anywhere

Pro for heavier workloads.

Makes life simpler & easier.

When to use Amazon SageMaker?

- data & team on AWS
- No need to migrate in foreseeable future.
- No internal compliance issues/conflicts -

NOT
when to use Amazon SageMaker?

- full control of everything
- large team (Admin, ops, Engineers ...)
- Internal compliance & security

SageMaker

Key Features →

Prepare

Data Wrangler

Feature Store

Ground Truth

Clarify

Build

Studio

Autopilot

JumpStart

Train & Tune

Debugger

Distributed Training

Deploy & Manage

Pipelines

Model Monitor

Kubernetes Integration

Edge Manager

Neo

<https://aws.amazon.com/sagemaker/>

Simple end-end system

- UCI credit card Default
- Xgboost model
- Focus on The Sagemaker specific details
- we will dive deeper later

Very
easy
to
follow

Refer : <https://aws.amazon.com/getting-started/hands-on/build-train-deploy-monitor-machine-learning-model-sagemaker-studio/>

AWS

- create an account
- login
- needs a payment method.

Sagemaker - home page

The screenshot shows the Amazon SageMaker homepage. On the left, a sidebar menu is open under the heading "Amazon SageMaker". A red box highlights the "Amazon SageMaker Studio" option, which is also the target of a red arrow pointing from the top-left of the image. Other menu items include "Dashboard", "Search", "Images", and several sections starting with "▶".

The main content area features the title "Amazon SageMaker" and the subtitle "Build, train, and deploy machine learning models at scale". Below this, a sub-subtitle reads "The quickest and easiest way to get ML models from idea to production." To the right, there's a "Get started" section with a "SageMaker Studio" button, a "Pricing (US)" section, and a "Related services" section listing "AWS Glue", "Amazon EC2", and "Amazon Elastic Block Store (EBS)".

In the center, there's a "How it works" section with three cloud icons labeled "Label", "Build", and "Train". Each icon has a brief description below it.

Label	Build	Train
Set up and manage labeling jobs for highly accurate training datasets within Amazon SageMaker, using active learning and human labeling.	Connect to other AWS services and transform data in Amazon SageMaker notebooks.	Use Amazon SageMaker's algorithms and frameworks, or bring your own, for distributed training.

SETUP

Quickstart vs Standard Setup

SageMaker Studio

Get started

[Learn more about getting started with SageMaker Studio](#)

Quick start
Let Amazon SageMaker handle configuring account and setting the permissions that you or a team in your organization need to use SageMaker Studio. Choosing this option uses standard encryption, which you can't change. If you need more control over configuration, choose Standard setup.

User name

The user name can have up to 63 characters. Valid characters: A-Z, a-z, 0-9, and - (hyphen)

Execution role
SageMaker Studio requires permissions to access other AWS services, such as Amazon SageMaker and Amazon S3. The execution role must have the [AmazonSageMakerFullAccess policy](#) attached. If you don't have a role with this policy attached, we can create one for you.

⚠ Please provide a valid Arn.

SageMaker Projects and JumpStart New
Enable access and provisioning of AWS Service Catalog Portfolio of products in Amazon SageMaker Studio for Amazon SageMaker Projects and JumpStart. [Learn more](#)

Enable Amazon SageMaker project templates and JumpStart for this account and Studio users
If enabled, the administrator can view the Amazon SageMaker provided project templates and JumpStart solutions published in AWS Service Catalog and users who are configured to use the domain execution are allowed to create projects using those templates and solutions with JumpStart. A launch constraint role and a project use role are automatically generated in IAM for your account.

Standard setup
Control all aspects of account configuration, including permissions and encryption. Choose this option if you are an administrator setting up SageMaker Studio for your organization.

[Cancel](#) [Submit](#)

Execution role:

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*

Any S3 bucket
Allow users that have access to your notebook instance access to any bucket and its contents in your account.

Specific S3 buckets
Example: bucket-name-1, bucket
Comma delimited. ARNs, "*" and "/" are not supported.

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](#)

S3 : simple storage service

Get started

[Learn more about getting started with SageMaker Studio](#)

Quick start

Let Amazon SageMaker handle configuring account and setting the permissions that you or a team in your organization need to use SageMaker Studio. Choosing this option uses standard encryption, which you can't change. If you need more control over configuration, choose Standard setup.

User name

tutorial-appliedroots

The user name can have up to 63 characters. Valid characters: A-Z, a-z, 0-9, and - (hyphen)

Execution role

SageMaker Studio requires permissions to access other AWS services, such as Amazon SageMaker and Amazon S3. The execution role must have the [AmazonSageMakerFullAccess policy](#) attached. If you don't have a role with this policy attached, we can create one for you.

AmazonSageMaker-ExecutionRole-20210815T163173



Success! You created an IAM role.

[AmazonSageMaker-ExecutionRole-20210815T163173](#)



SageMaker Projects and JumpStart [New](#)

Enable access and provisioning of AWS Service Catalog Portfolio of products in Amazon SageMaker Studio for Amazon SageMaker Projects and JumpStart. [Learn more](#)

Enable Amazon SageMaker project templates and JumpStart for this account and Studio users

If enabled, the administrator can view the Amazon SageMaker provided project templates and JumpStart solutions published in AWS Service Catalog and users who are configured to use the domain execution are allowed to create projects using those templates and solutions with JumpStart. A launch constraint role and a project use role are automatically generated in IAM for your account.

Standard setup

Control all aspects of account configuration, including permissions and encryption. Choose this option if you are an administrator setting up SageMaker Studio for your organization.

Cancel

Submit

Data to S3

after few mins

SageMaker Studio is ready
Choose your user name, then choose Open Studio to get started.

Amazon SageMaker > SageMaker Studio > Control Panel

SageMaker Studio Control Panel

Choose your user name, then choose Open Studio to get started

User name	Last modified	Created	
tutorial-appliedroots	Aug 15, 2021 11:11 UTC	Aug 15, 2021 11:11 UTC	<input checked="" type="checkbox"/> Open Studio

Studio Summary

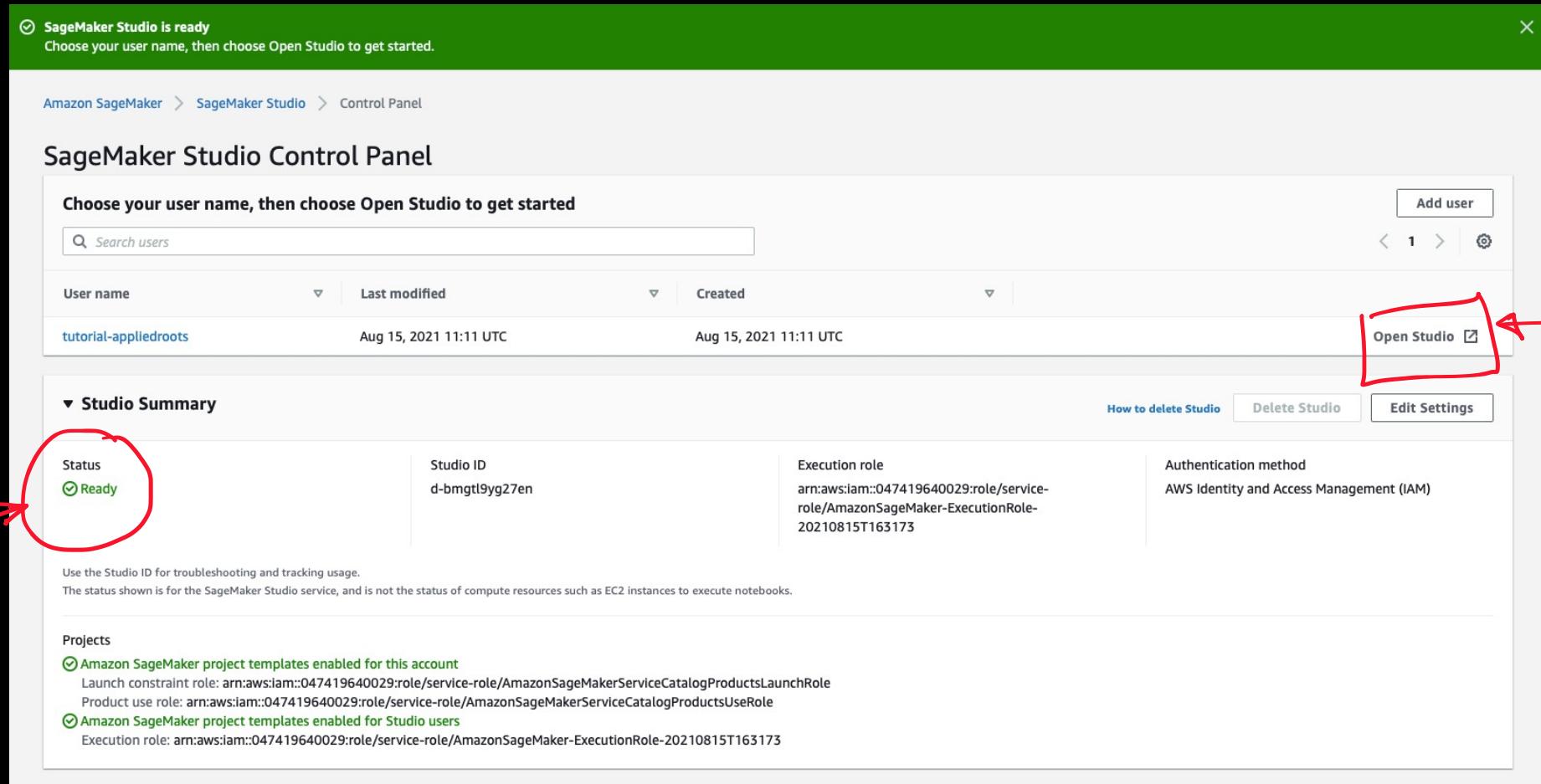
Status Ready	Studio ID d-bmgtl9yg27en	Execution role arn:aws:iam::047419640029:role/service-role/AmazonSageMaker-ExecutionRole-20210815T163173	Authentication method AWS Identity and Access Management (IAM)
------------------------	-----------------------------	---	---

Use the Studio ID for troubleshooting and tracking usage.
The status shown is for the SageMaker Studio service, and is not the status of compute resources such as EC2 instances to execute notebooks.

Projects

- Amazon SageMaker project templates enabled for this account
 - Launch constraint role: arn:aws:iam::047419640029:role/service-role/AmazonSageMakerServiceCatalogProductsLaunchRole
 - Product use role: arn:aws:iam::047419640029:role/service-role/AmazonSageMakerServiceCatalogProductsUserRole
- Amazon SageMaker project templates enabled for Studio users
 - Execution role: arn:aws:iam::047419640029:role/service-role/AmazonSageMaker-ExecutionRole-20210815T163173

[How to delete Studio](#) [Delete Studio](#) [Edit Settings](#)





Amazon SageMaker Studio

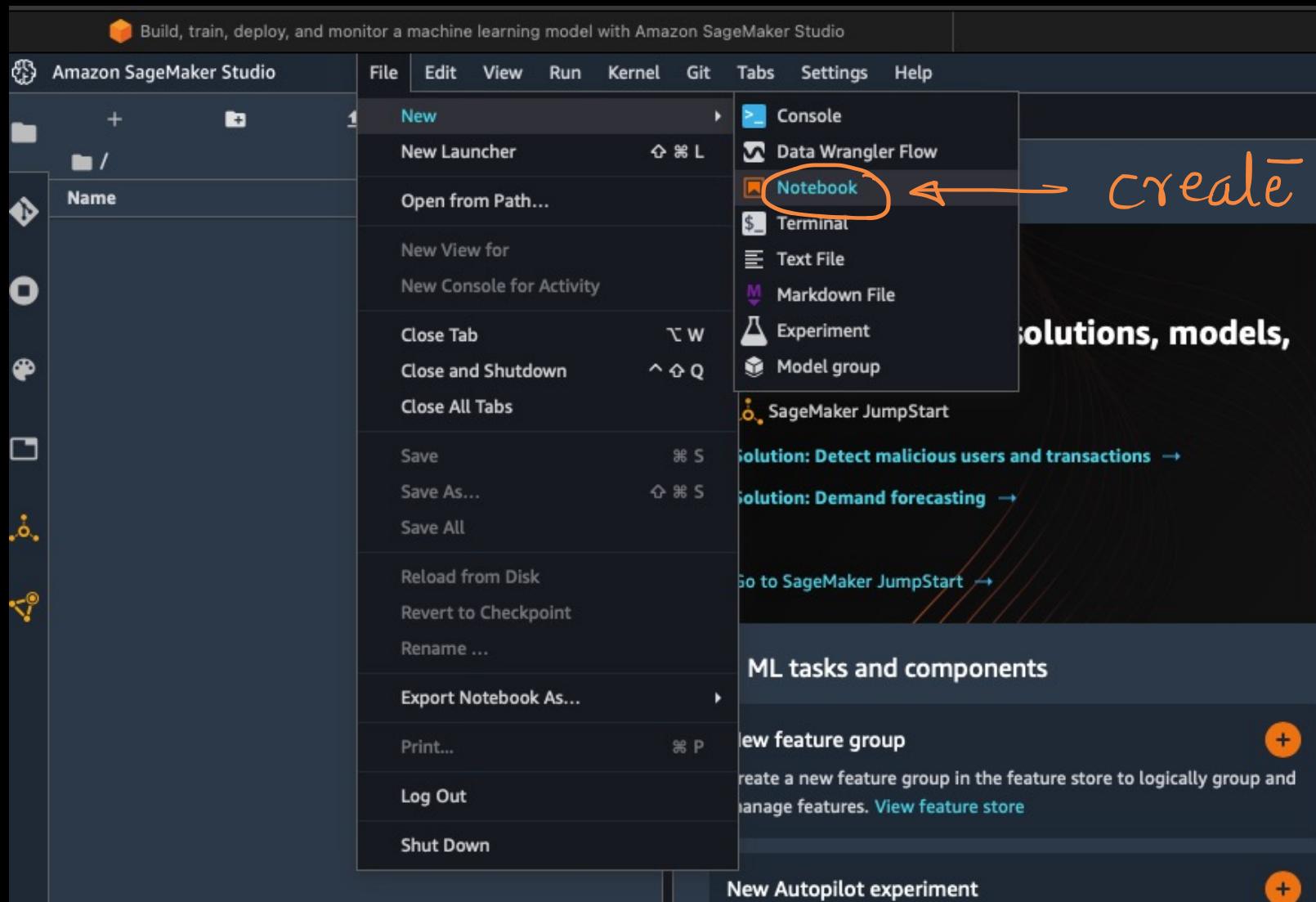
Creating the JupyterServer application default...

Jupyter-like

The screenshot shows the Amazon SageMaker Studio Launcher interface. The top navigation bar includes File, Edit, View, Run, Kernel, Git, Tabs, Settings, and Help. On the left, there's a file browser with a sidebar containing icons for files, folders, databases, and other data sources.

The main content area is titled "Launcher" and displays several sections:

- Get started**:
 - Explore one-click solutions, models, and tutorials**: Includes "SageMaker JumpStart" with links to "Detect malicious users and transactions" and "Demand forecasting". A "Go to SageMaker JumpStart" button is also present.
 - Build models automatically**: Features "SageMaker Autopilot" with a "Video: Get started with Autopilot" and a "Blog: Getting started with Autopilot". A "New autopilot experiment" button is available.
 - Run open-source models with one click**: Features "SageMaker JumpStart" with links to "Popular image classification based on ResNet" and "State-of-the-art BERT text processing". An "Explore models" button is present.
- ML tasks and components**:
 - New feature group**: Create a new feature group in the feature store to logically group and manage features. A "View feature store" link is provided.
 - New data flow**: Prepare and visualize your data with SageMaker Data Wrangler. A "View data flows" link is provided.
 - New project**: Organize ML components and automate MLOps with built-in or custom project templates. A "View projects" link is provided.
- Notebooks and compute resources**:
 - Select a SageMaker image: Data Science dropdown.
 - Options: Notebook (Python 3), Console (Python 3), Image Terminal (Image Terminal).
- Utilities and files**:
 - Show Contextual Help, System terminal, Text File, Markdown File.



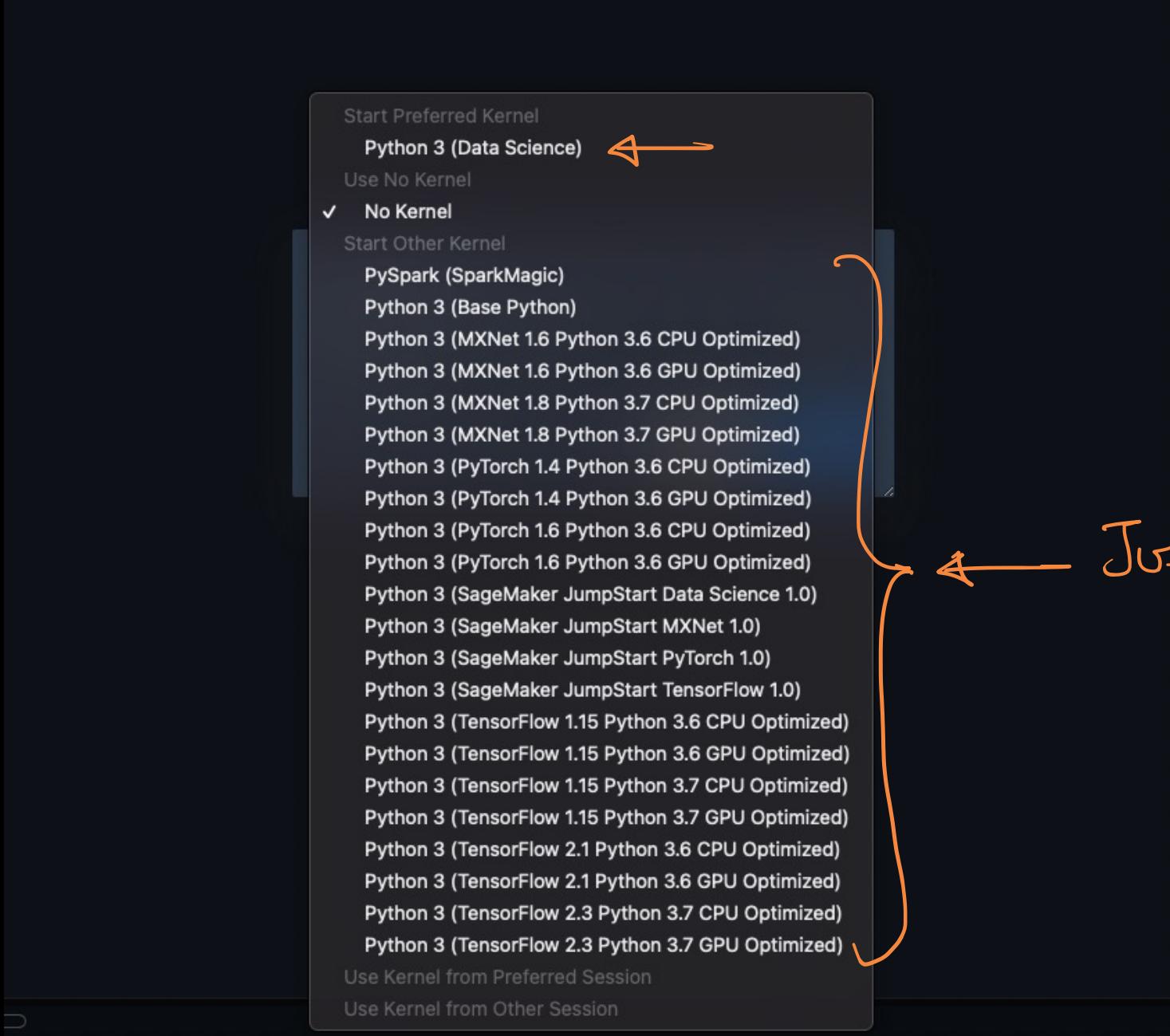
Select Kernel

Select kernel for: "Untitled.ipynb"

No Kernel

Cancel

Select



name

default EC2

kernel

The screenshot shows a Jupyter Notebook interface with the following details:

- Launcher** tab is selected.
- The notebook file is **Untitled.ipynb**.
- The kernel configuration is set to **2 vCPU + 4 GiB Python 3 (Data Science)**.
- The code cell [1] contains Python code to handle SageMaker version compatibility and role setup:

```
[1]: import boto3
import sagemaker
from sagemaker import get_execution_role
import sys
import IPython

if int(sagemaker.__version__.split('.')[0]) == 2:
    print("Installing previous SageMaker Version and restarting the kernel")
    !{sys.executable} -m pip install sagemaker==1.72.0
    IPython.Application.instance().kernel.do_shutdown(True)

else:
    print("Version is good")

role = get_execution_role()
sess = sagemaker.Session()
region = boto3.session.Session().region_name
print("Region = {}".format(region))
sm = boto3.Session().client('sagemaker')
```

SageMaker Version

```
[1]: import boto3
      import sagemaker
      from sagemaker import get_execution_role
      import sys
      import IPython

      if int(sagemaker.__version__.split('.')[0]) == 2:
          print("Installing previous SageMaker Version and restarting the kernel")
          !{sys.executable} -m pip install sagemaker==1.72.0
          IPython.Application.instance().kernel.do_shutdown(True)

      else:
          print("Version is good")

      role = get_execution_role()
      sess = sagemaker.Session()
      region = boto3.session.Session().region_name
      print("Region = {}".format(region))
      sm = boto3.Session().client('sagemaker')
```

For AWS admin

→ Show output on Notebook

```
[1]: # import libraries
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import os
from time import sleep, gmtime, strftime
import json
import time

[2]: # Sagemaker-Experiments to track various training experiments
!pip install sagemaker-experiments
from sagemaker.analytics import ExperimentAnalytics
from smexperiments.experiment import Experiment
from smexperiments.trial import Trial
from smexperiments.trial_component import TrialComponent
from smexperiments.tracker import Tracker
```

S3-bucket & folders

```
# S3 bucket & folders to store all project files
rawbucket= sess.default_bucket() # Alternatively you can use our custom bucket here.

prefix = 'sagemaker-modelmonitor' # use this prefix to store all files pertaining to this workshop.

dataprefix = prefix + '/data'
traindataprefix = prefix + '/train_data'
testdataprefix = prefix + '/test_data'
testdatanolabelprefix = prefix + '/test_data_no_label'
trainheaderprefix = prefix + '/train_headers'
```

INFO:sagemaker:Created S3 bucket: sagemaker-us-west-2-047419640029

AWS homepage

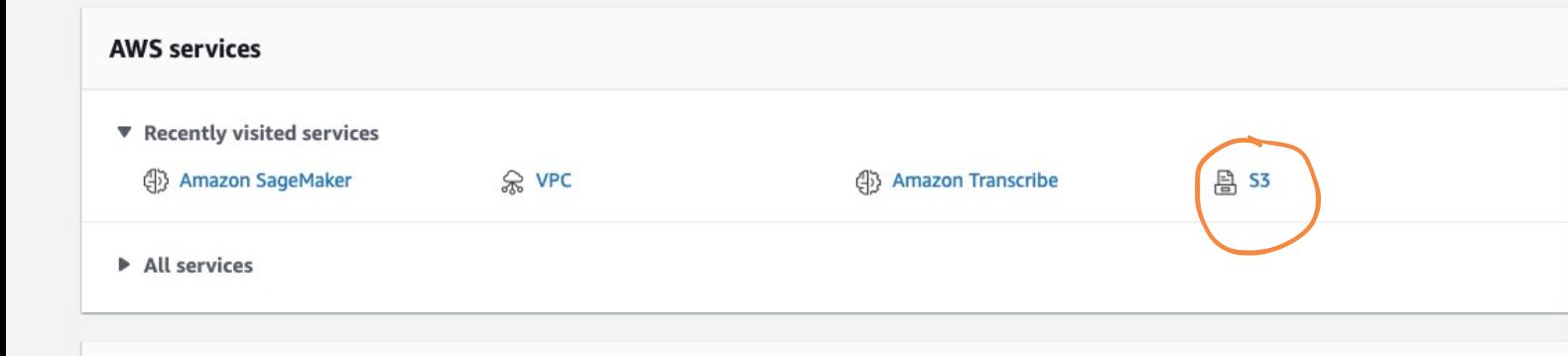
AWS Management Console

AWS services

▼ Recently visited services

- Amazon SageMaker
- VPC
- Amazon Transcribe
- S3

► All services



Amazon S3

► Account snapshot

Storage lens provides visibility into storage usage and activity trends. [Learn more](#)

[View Storage Lens dashboard](#)

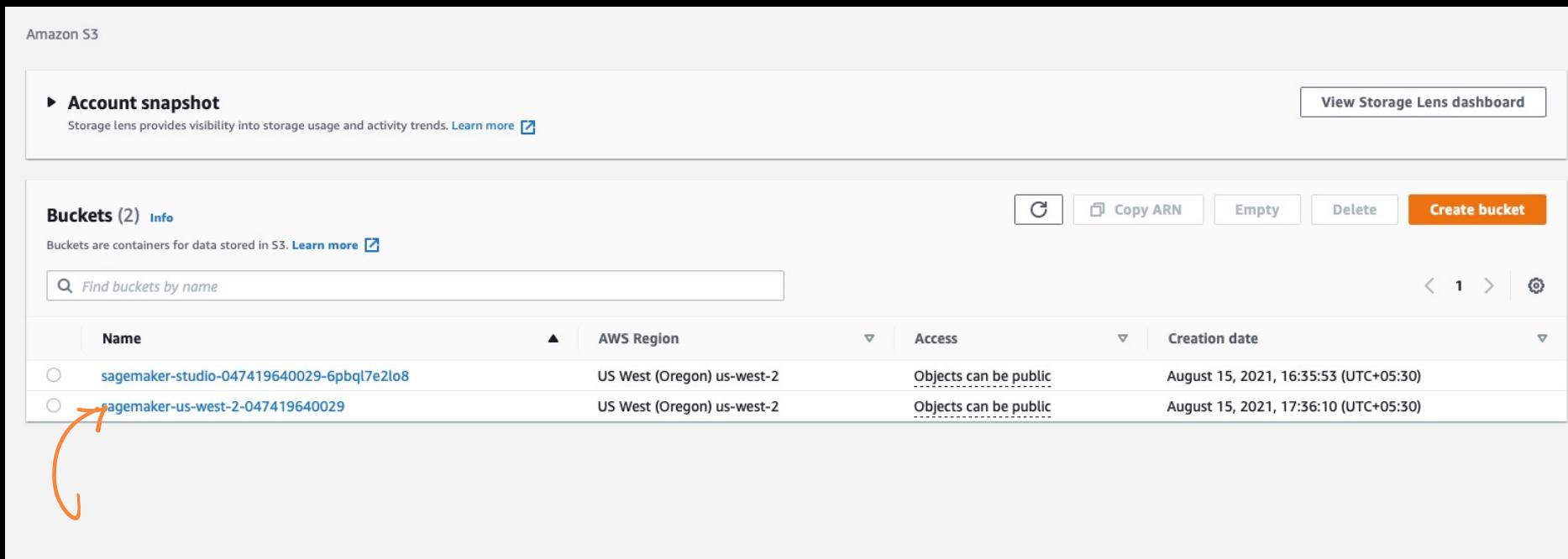
Buckets (2) [Info](#)

Buckets are containers for data stored in S3. [Learn more](#)

Find buckets by name

[Create bucket](#)

Name	AWS Region	Access	Creation date
sagemaker-studio-047419640029-6pbql7e2lo8	US West (Oregon) us-west-2	Objects can be public	August 15, 2021, 16:35:53 (UTC+05:30)
sagemaker-us-west-2-047419640029	US West (Oregon) us-west-2	Objects can be public	August 15, 2021, 17:36:10 (UTC+05:30)



Download data from web:

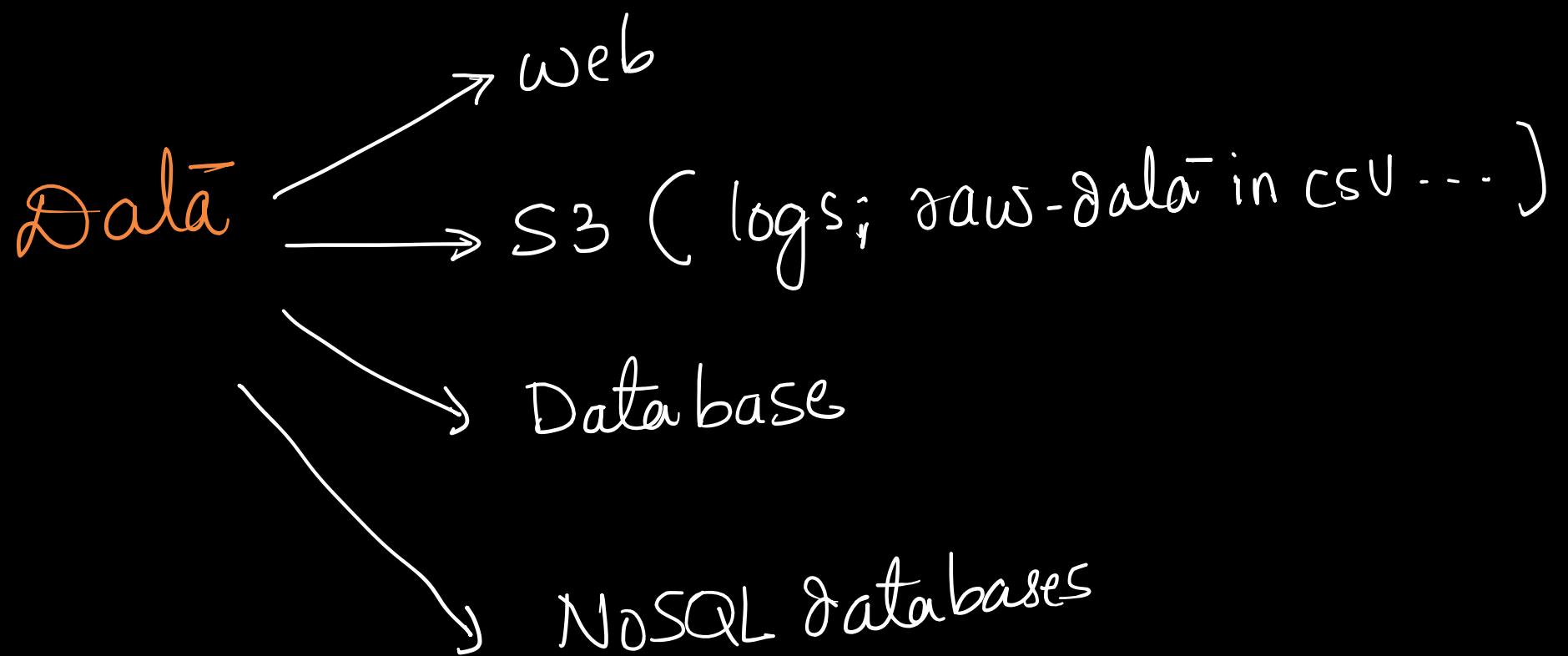
```
# download data using wget
! wget https://archive.ics.uci.edu/ml/machine-learning-databases/00350/default%20of%20credit%20card%20clients.xls
data = pd.read_excel('default of credit card clients.xls', header=1)
data = data.drop(columns = ['ID'])
data.head()
```

```
--2021-08-15 12:09:38-- https://archive.ics.uci.edu/ml/machine-learning-databases/00350/default%20of%20credit%20card%20clients.xls
Resolving archive.ics.uci.edu (archive.ics.uci.edu)... 128.195.10.252
Connecting to archive.ics.uci.edu (archive.ics.uci.edu)|128.195.10.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 5539328 (5.3M) [application/x-httdp-php]
Saving to: 'default of credit card clients.xls'

default of credit c 100%[=====>] 5.28M 20.4MB/s in 0.3s

2021-08-15 12:09:39 (20.4 MB/s) - 'default of credit card clients.xls' saved [5539328/5539328]
```

	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	PAY_5	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
0	20000	2	2	1	24	2	2	-1	-1	-2	...	0	0	0	0	0
1	120000	2	2	2	26	-1	2	0	0	0	...	3272	3455	3261	0	1
2	90000	2	2	2	34	0	0	0	0	0	...	14331	14948	15549	1518	1
3	50000	2	2	1	37	0	0	0	0	0	...	28314	28959	29547	2000	2
4	50000	1	2	1	57	-1	0	-1	0	0	...	20940	19146	19131	2000	36



Refer:

<https://docs.aws.amazon.com/sagemaker/latest/dg/data-wrangler-import.html>

:

[Data Wrangler]

```
[9]: # rename last col as "Label" and move it to the first column for Xgboost formatting  
data.rename(columns={"default payment next month": "Label"}, inplace=True)  
lbl = data.Label  
data = pd.concat([lbl, data.drop(columns=['Label'])], axis = 1)  
data.head()
```

```
[9]:
```

	Label	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT3	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_-
0	1	20000	2	2	1	24	2	2	-1	-1	...	689	0	0	0	0
1	1	120000	2	2	2	26	-1	2	0	0	...	2682	3272	3455	3261	
2	0	90000	2	2	2	34	0	0	0	0	...	13559	14331	14948	15549	
3	0	50000	2	2	1	37	0	0	0	0	...	49291	28314	28959	29547	
4	0	50000	1	2	1	57	-1	0	-1	0	...	35835	20940	19146	19131	

5 rows × 24 columns

default next month or not

```
# data to S3
if not os.path.exists('rawdata/rawdata.csv'):
    mkdir rawdata
    data.to_csv('rawdata/rawdata.csv', index=None)
else:
    pass

# Upload the raw dataset
raw_data_location = sess.upload_data('rawdata', bucket=rawbucket, key_prefix=dataprefix)
print(raw_data_location)

s3://sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/data
```

} → disk on EC2 instance

to S3 ↗

Data Preprocessing

```
# Data Preprocessing using SkLearn
# Use scikit-learn processing container (Docker) for more popular a EC2 instance
# INsatnce types and pricing: https://aws.amazon.com/sagemaker/pricing/
from sagemaker.sklearn.processing import SKLearnProcessor
sklearn_processor = SKLearnProcessor(framework_version='0.20.0',
                                      role=role,
                                      instance_type='ml.c5.xlarge',
                                      instance_count=1)
```

EC2 - instance type

<https://aws.amazon.com/sagemaker/pricing/>

Compute-optimized: 16 vCPU, 32 GB RAM

```
%%writefile preprocessing.py

import argparse
import os
import warnings

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.exceptions import DataConversionWarning
from sklearn.compose import make_column_transformer

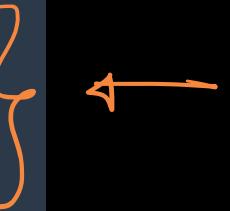
warnings.filterwarnings(action='ignore', category=DataConversionWarning)

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument('--train-test-split-ratio', type=float, default=0.3)
    parser.add_argument('--random-split', type=int, default=0)
    args, _ = parser.parse_known_args()
```

EC2 local-disk

[15]: ! ls -l

```
total 5448
-rw-r--r-- 1 root root  25476 Aug 15 12:27 Untitled.ipynb
-rw-r--r-- 1 root root 5539328 Jan 26  2016 'default of credit card clients.xls'
-rw-r--r-- 1 root root   3035 Aug 15 12:27 preprocessing.py
drwxr-xr-x 2 root root  6144 Aug 15 12:18 rawdata
```



Must-do : [WHY?]

```
# Copy the preprocessing code over to the s3 bucket
codeprefix = prefix + '/code'
codeupload = sess.upload_data('preprocessing.py', bucket=rawbucket, key_prefix=codeprefix)
print(codeupload)

s3://sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/code/preprocessing.py
```

```
: # S3 paths to store train and test data after processing
train_data_location = rawbucket + '/' + traindataprefix
test_data_location = rawbucket + '/' + testdataprefix
print("Training data location = {}".format(train_data_location))
print("Test data location = {}".format(test_data_location))
```

```
Training data location = sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/train_data
Test data location = sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/test_data
```

Start sklearn Processing Job:

```
from sagemaker.processing import ProcessingInput, ProcessingOutput

sklearn_processor.run(code=codeupload,
                      inputs=[ProcessingInput(
                          source=raw_data_location,
                          destination='/opt/ml/processing/input')],
                      outputs=[ProcessingOutput(output_name='train_data',
                                                source='/opt/ml/processing/train',
                                                destination='s3://'+ train_data_location),
                               ProcessingOutput(output_name='test_data',
                                                source='/opt/ml/processing/test',
                                                destination="s3://" + test_data_location),
                               ProcessingOutput(output_name='train_data_headers',
                                                source='/opt/ml/processing/train_headers',
                                                destination="s3://" + rawbucket + '/' + prefix + '/train_headers'
                               ),
                               arguments=['--train-test-split-ratio', '0.2']
                           )
preprocessing_job_description = sklearn_processor.jobs[-1].describe()

output_config = preprocessing_job_description['ProcessingOutputConfig']
for output in output_config['Outputs']:
    if output['OutputName'] == 'train_data':
        preprocessed_training_data = output['S3Output']['S3Uri']
    if output['OutputName'] == 'test_data':
        preprocessed_test_data = output['S3Output']['S3Uri']
```

How to
understand
this
code?

```
INFO:sagemaker:Creating processing-job with name sagemaker-scikit-learn-2021-08-15-12-34-15-337
```

```
Job Name: sagemaker-scikit-learn-2021-08-15-12-34-15-337
Inputs: [{"InputName": "input-1", "S3Input": {"S3Uri": "s3://sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/data", "LocalPath": "/opt/ml/processing/input", "S3DataType": "S3Prefix", "S3InputMode": "File", "S3DataDistributionType": "FullyReplicated", "S3CompressionType": "None"}}, {"InputName": "code", "S3Input": {"S3Uri": "s3://sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/code/preprocessing.py", "LocalPath": "/opt/ml/processing/input/code", "S3DataType": "S3Prefix", "S3InputMode": "File", "S3DataDistributionType": "FullyReplicated", "S3CompressionType": "None"}}]
Outputs: [{"OutputName": "train_data", "S3Output": {"S3Uri": "s3://sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/train_data", "LocalPath": "/opt/ml/processing/train", "S3UploadMode": "EndOfJob"}}, {"OutputName": "test_data", "S3Output": {"S3Uri": "s3://sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/test_data", "LocalPath": "/opt/ml/processing/test", "S3UploadMode": "EndOfJob"}}, {"OutputName": "train_data_headers", "S3Output": {"S3Uri": "s3://sagemaker-us-west-2-047419640029/sagemaker-modelmonitor/train_headers", "LocalPath": "/opt/ml/processing/train_headers", "S3UploadMode": "EndOfJob"}}]
.....
```

Sagemaker Homepage

The screenshot shows the Amazon SageMaker homepage with a sidebar and a main content area. The sidebar on the left lists various services under 'Amazon SageMaker Studio' such as Dashboard, Search, Images, Ground Truth, Notebook, Processing, Training, Inference, Edge Manager, Augmented AI, and AWS Marketplace. The main content area has four main sections: Prepare, Build, Train & Tune, and Deploy & Manage. Under 'Recent activity', there are five categories: Ground Truth, Notebook, Training, Inference, and Processing. The 'Processing' section shows one job as 'Running' and another as 'Created'. A red arrow points from the top right towards the 'Running' job.

Recent activity within the Last 7 days

Category	Activity	Status
Ground Truth	Labeling jobs	No recent activity.
Notebook	Notebook instances	No recent activity.
Training	Training jobs	No recent activity.
Inference	Models	No recent activity.
Processing	Processing jobs	1 Running 1 Created

Processing jobs



Actions ▾

Create processing job

 Search processing jobs

< 1 > ⚙

Name	ARN	Creation time	Duration	Status
sagemaker-scikit-learn-2021-08-15-12-34-15-337	arn:aws:sagemaker:us-west-2:047419640029:processing-job/sagemaker-scikit-learn-2021-08-15-12-34-15-337	Aug 15, 2021 12:34 UTC	3 minutes	InProgress

[Amazon SageMaker](#) > [Processing jobs](#) > sagemaker-scikit-learn-2021-08-15-12-34-15-337

sagemaker-scikit-learn-2021-08-15-12-34-15-337

Job settings

Job name	Status	IAM role ARN	Processing time (seconds)
sagemaker-scikit-learn-2021-08-15-12-34-15-337	InProgress	arn:aws:iam::047419640029:role/service-role/AmazonSageMaker-ExecutionRole-20210815T163173	-
ARN	Creation time		
arn:aws:sagemaker:us-west-2:047419640029:processing-job/sagemaker-scikit-learn-2021-08-15-12-34-15-337	Aug 15, 2021 12:34 UTC		
	Last modified time		
	Aug 15, 2021 12:34 UTC		

App specification

Processing image	Instance count	Additional volume size (GB)	Volume encryption key
246618743249.dkr.ecr.us-west-2.amazonaws.com/sagemaker-scikit-learn:0.20.0-cpu-py3	1	30	-
Instance type		Maximum runtime (seconds)	
ml.c5.xlarge		86400	
Entrypoint			
python3			

Notebook output

```
.....  
/miniconda3/lib/python3.7/site-packages/sklearn/externals/joblib/externals/cloudpickle/cloudpickle.py:47: DeprecationWarning: the imp module is deprecated in favour of importlib; see the module's documentation for alternative uses  
    import imp  
Received arguments Namespace(random_split=0, train_test_split_ratio=0.2)  
Reading input data from /opt/ml/processing/input/rawdata.csv  
Running preprocessing and feature engineering transformations  
Train data shape after preprocessing: (24000, 23)  
Test data shape after preprocessing: (6000, 23)  
Saving training features to /opt/ml/processing/train/train_data.csv  
Complete  
Save training data with headers to /opt/ml/processing/train_headers/train_data_with_headers.csv  
Saving test features to /opt/ml/processing/test/test_data.csv  
Complete
```



Amazon S3 > **sagemaker-us-west-2-047419640029** > sagemaker-modelmonitor/

sagemaker-modelmonitor/

Objects (5)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in this bucket.

Find objects by prefix

<input type="checkbox"/>	Name	Type	Last
<input type="checkbox"/>	code/	Folder	-
<input type="checkbox"/>	data/	Folder	-
<input type="checkbox"/>	test_data/	Folder	-
<input type="checkbox"/>	train_data/	Folder	-
<input type="checkbox"/>	train_headers/	Folder	-

Next:

- Training Experiments (Xgboost)
- Deploy for Batch-processing
- Deploy for live-inference (REST APIs)
- Model-monitoring

Note: Always delete & cleanup to avoid costs.

https://docs.aws.amazon.com/sagemaker/latest/dg/gs-studio-delete-domain.html?icmpid=docs_sagemaker_console_studio

Q&A