data® expert.

Data Science & Machine Learning

Boas Vindas ao Módulo

Consultor: Murilo Mendonça

Murilo Mendonça

ML Engineer
@Ambev Tech

Eng. Mecânico

Fraunhofer Institute

Lactec

Ambev

Ambev Tech

Muri Tech

Tópicos deste módulo:

01

Conceitos de nuvem

02

Principais ferramentas de Data Science

03

Open Source Data Science

data® expert.

Data Science & Machine Learning

MDC02 - Conceitos de Nuvem

Consultor: Murilo Mendonça

Tópicos desta aula:

O que é "nuvem"?

Benefícios

Principais provedores

Serviços típicos

O que é "nuvem"?

A disponibilização de serviços
de computação

Armazenamento sob-demanda
e "infinitos".

Benefícios

Custo significativamente inferior a adquirir e manter on-premises.

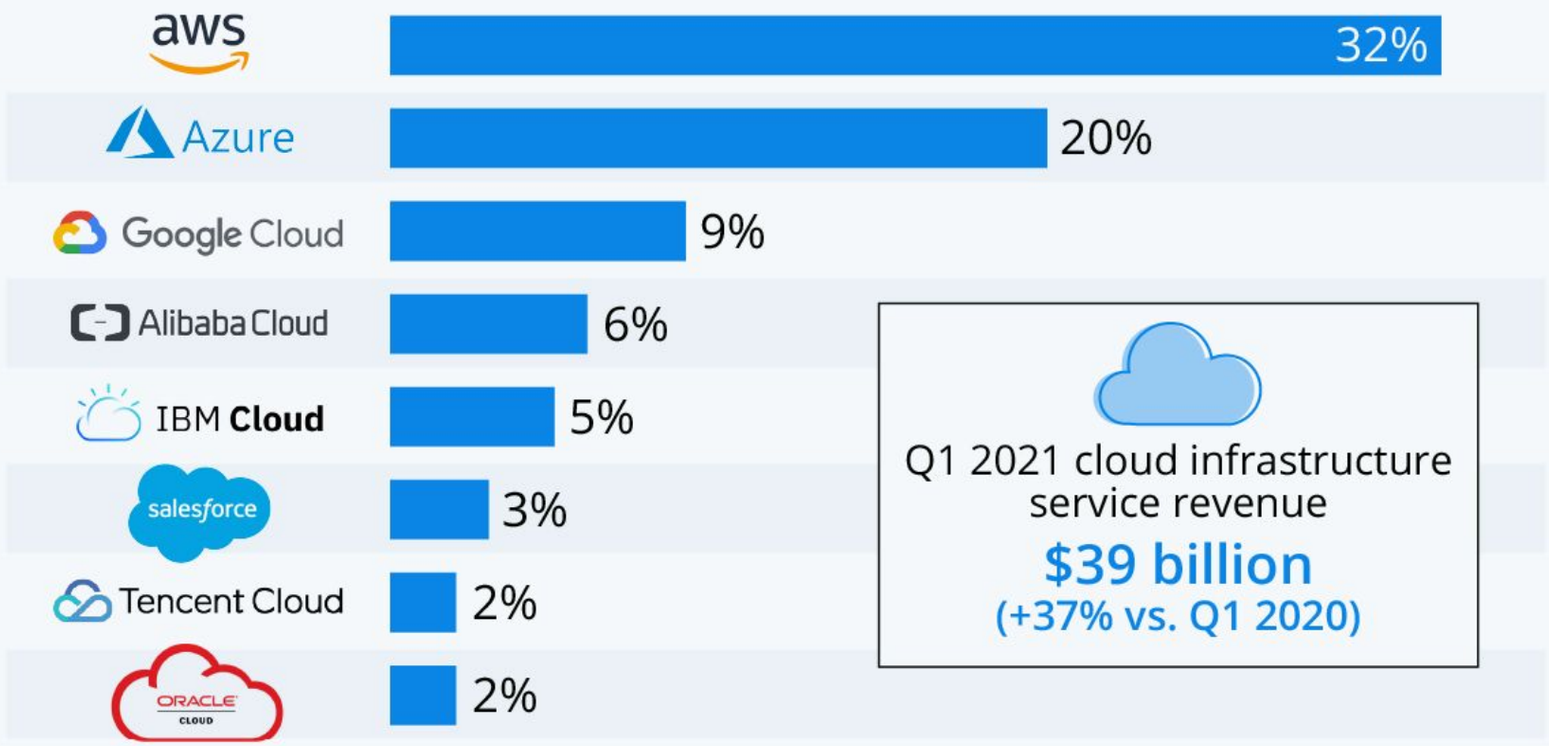
Diversos protocolos de segurança e outros serviços gerenciados.

Escalabilidade sob-demanda.

Principais Provedores

Amazon Leads $150-Billion Cloud Market

Worldwide market share of leading cloud infrastructure service providers in Q1 2021*



* includes platform as a service (PaaS) and infrastructure as a service (IaaS) as well as hosted private cloud services

Source: Synergy Research Group

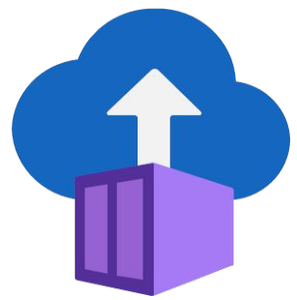
Serviços Típicos

Computação

VM

Kubernetes

Serverless



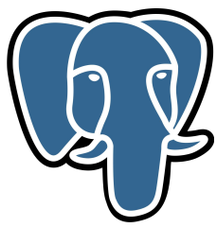
Spark

Serviços Típicos

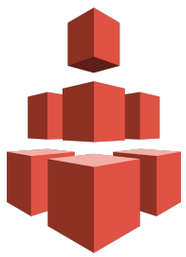
Armazenamento

Object

SQL e NoSQL



File System



Kubernetes

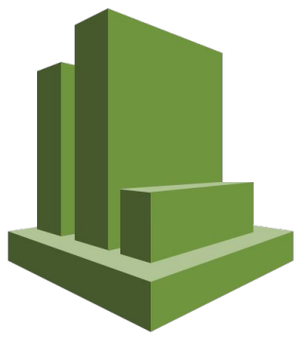
Serviços Típicos

Outros

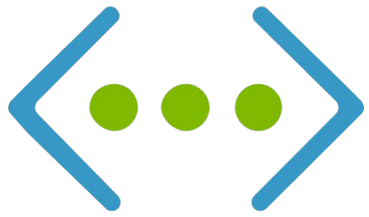
Orquestradores

API Gateway

Monitoramento



VNET



Key Vault

Resumo da Aula

O que é "nuvem"?

Benefícios

Principais provedores

Serviços típicos

data® expert.

Data Science & Machine Learning

MDC03 - Principais Ferramentas de Analytics

Consultor: Murilo Mendonça

O que veremos nesta aula:

01

Análise de Dados com SQL

02

Processamento de dados

03

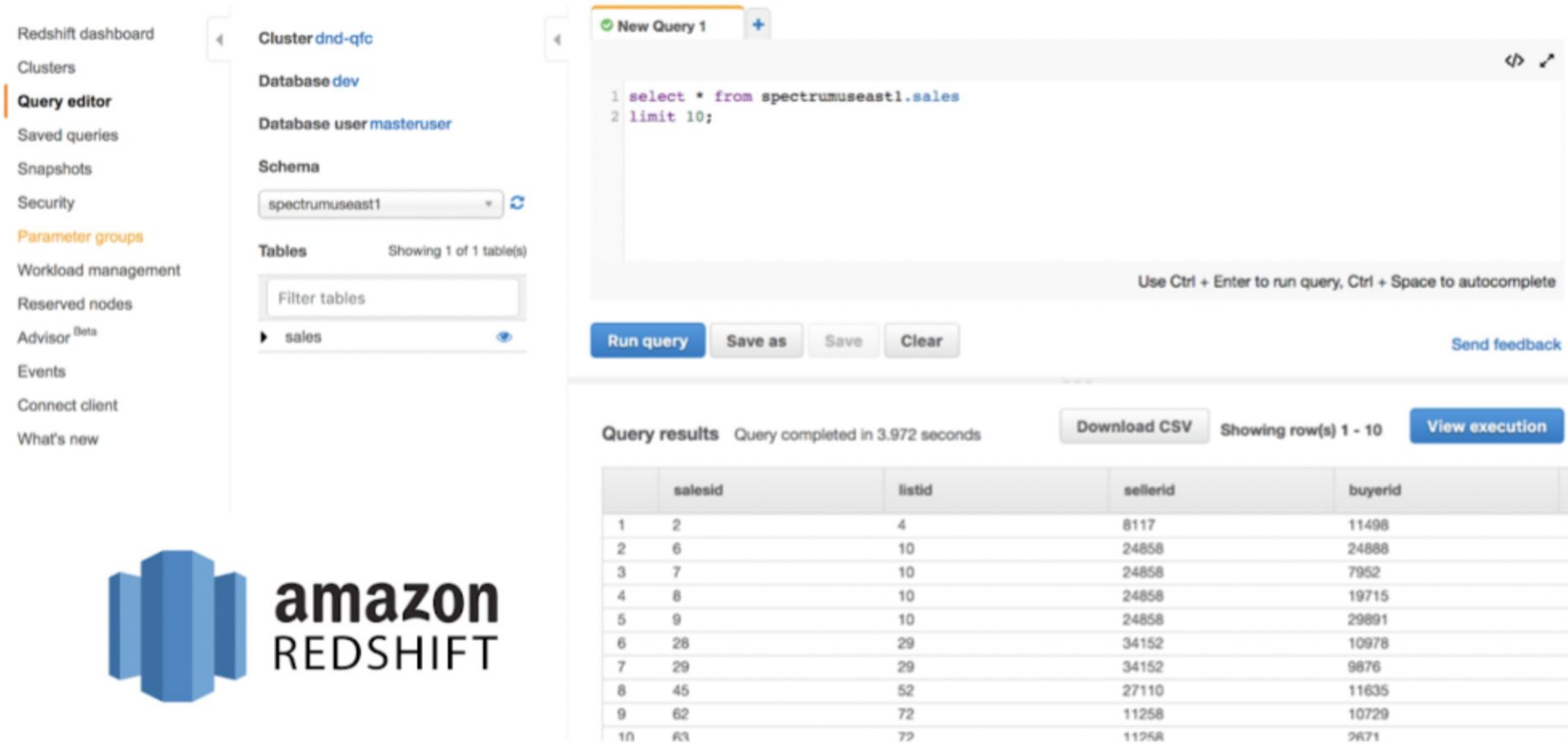
Machine Learning

04

Orquestradores

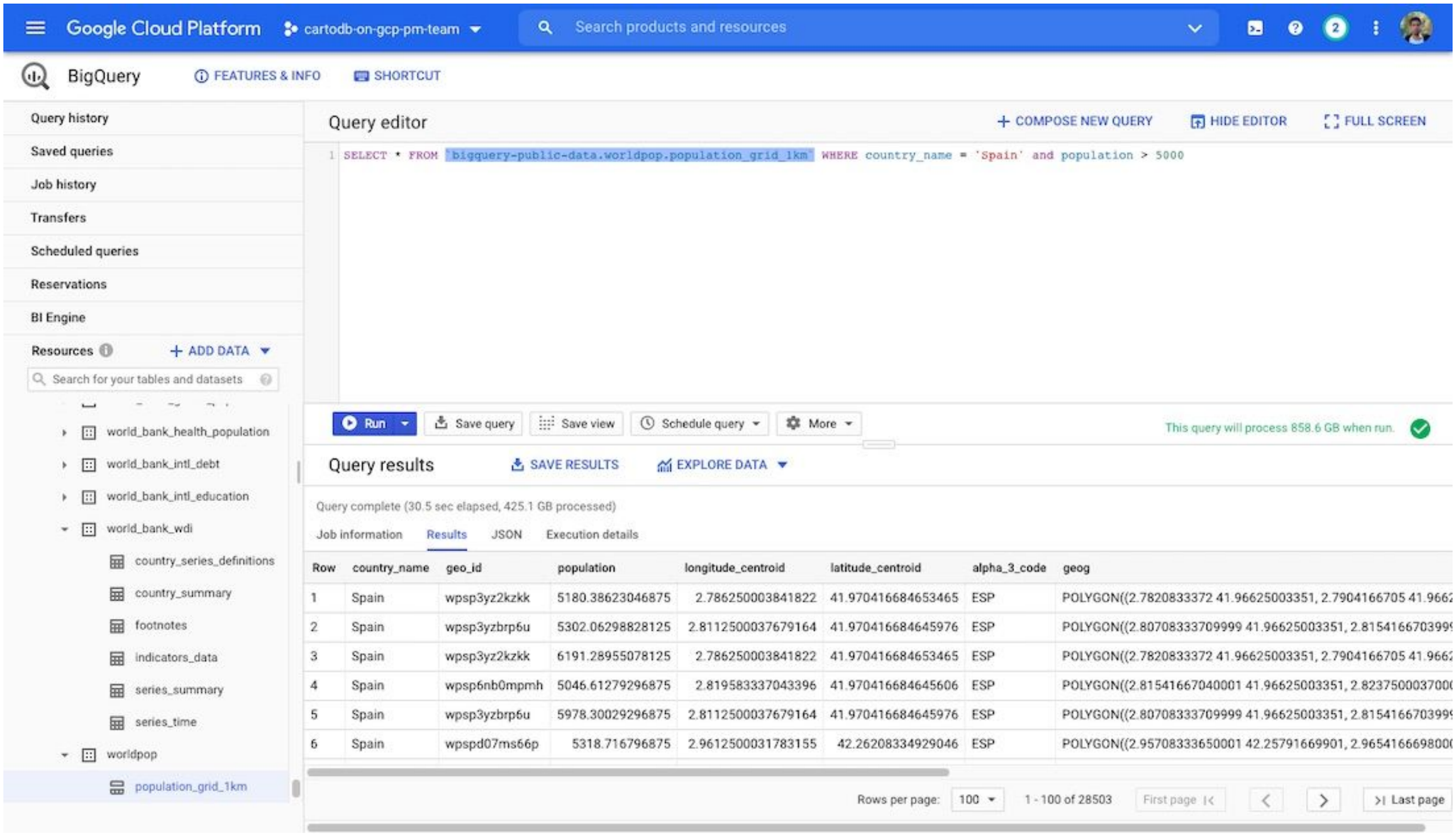
Análise de Dados (SQL)

Amazon REDSHIFT

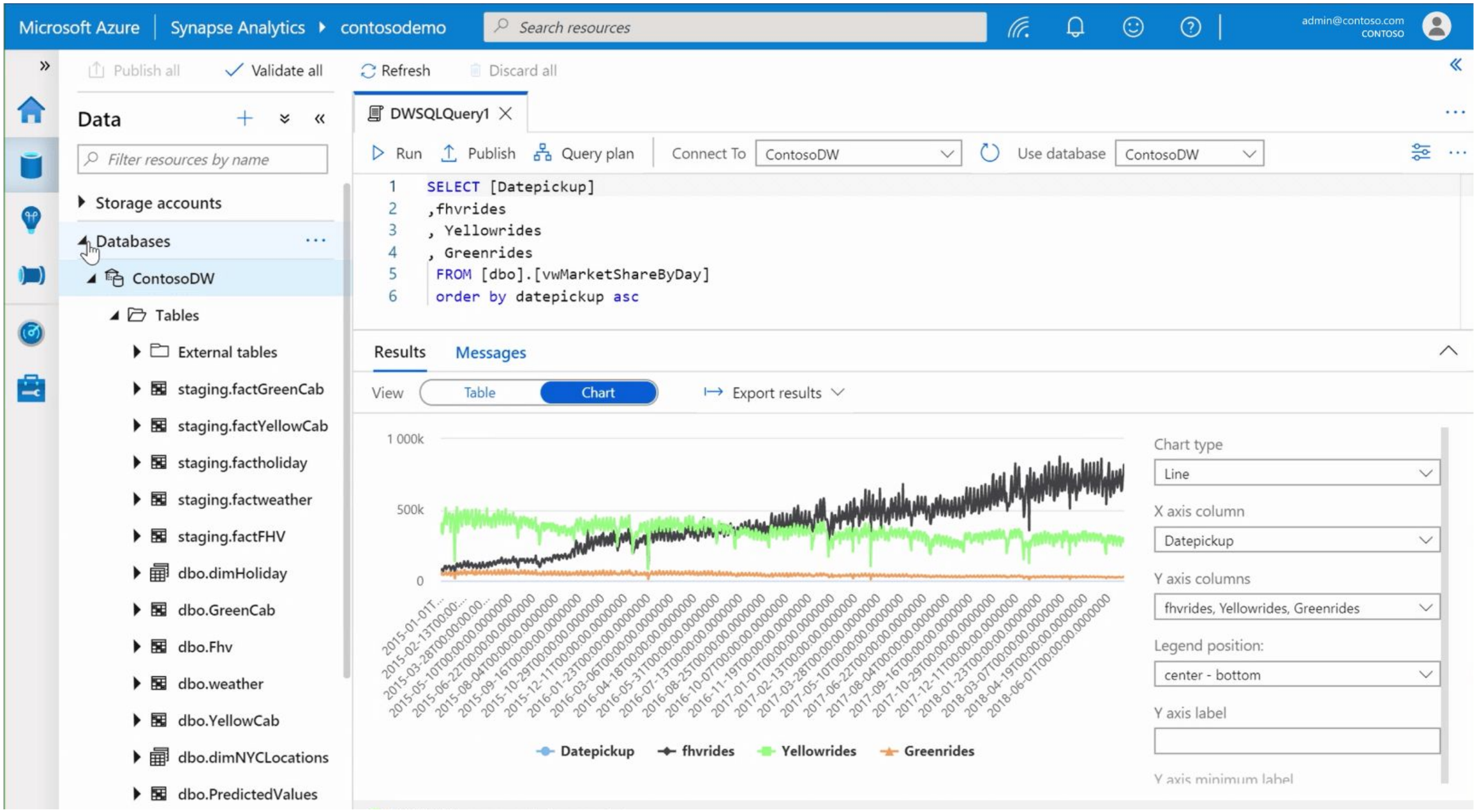


amazon REDSHIFT

Google Cloud Platform



Microsoft Azure



DataBricks

Cmd 7

```sql
%sql

SELECT
  join_key,
  sum(x.id),
  count()
FROM
  (SELECT
    x.id,
    y.join_key
  FROM
    table_x x
  CROSS JOIN
    table_y y
  on x.join_key = y.join_key)
GROUP BY join_key
```

▸ (2) Spark Jobs

⊞ Error in SQL statement: SparkException: Job 36 cancelled because Task 84820 in Stage 58 exceeded the maximum allowed ratio of input to output records (1 to 1060, max allowed 1 to 1000); this limit can be modified with configuration parameter spark.databricks.queryWatchdog.outputRatioThreshold

MetaBase

New question

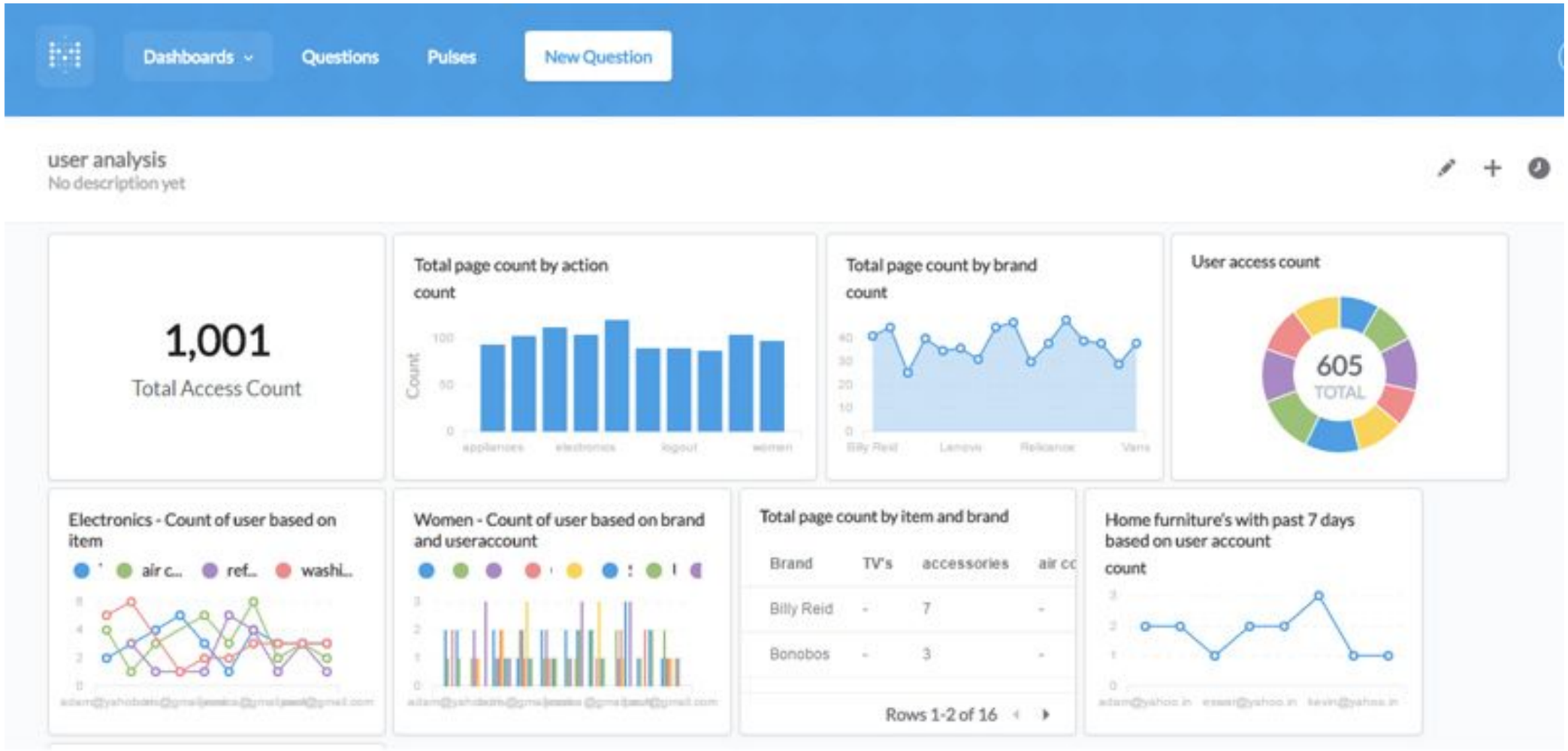
Save

Sample Dataset

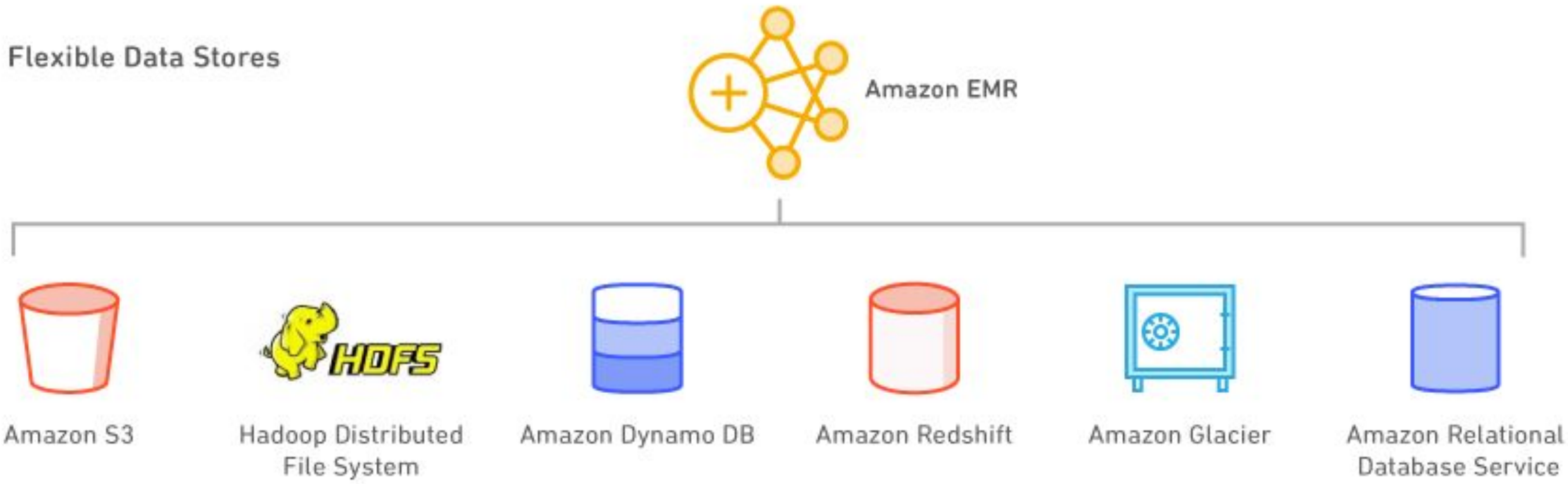
```
1  select
2      count(*) as "Total Orders",
3      CREATED_AT::date as "Order date"
4  from ORDERS
5  group by CREATED_AT
6  limit 10;
```

Total Orders	Order date
1	February 4, 2019, 12:00 AM
1	January 22, 2017, 12:00 AM
1	January 27, 2019, 12:00 AM

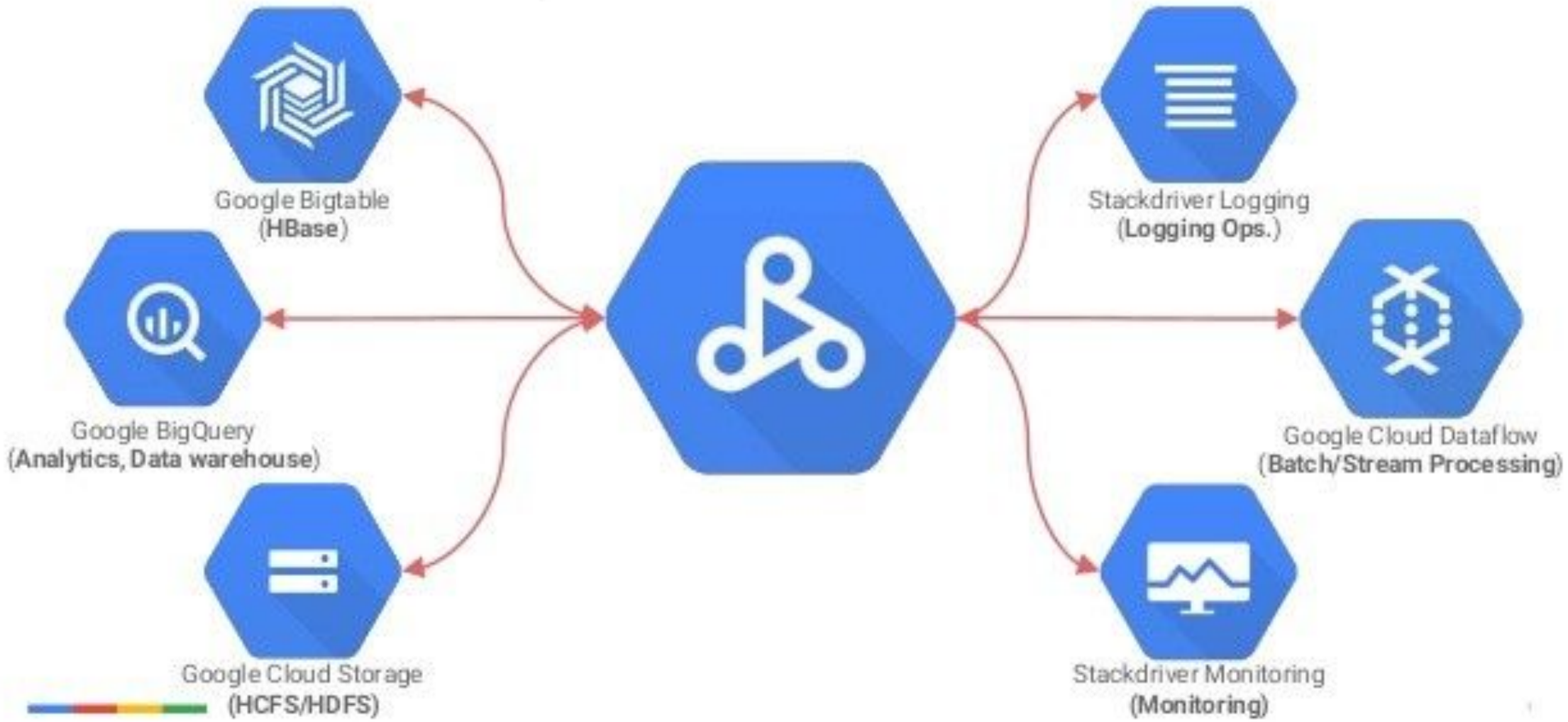
MetaBase

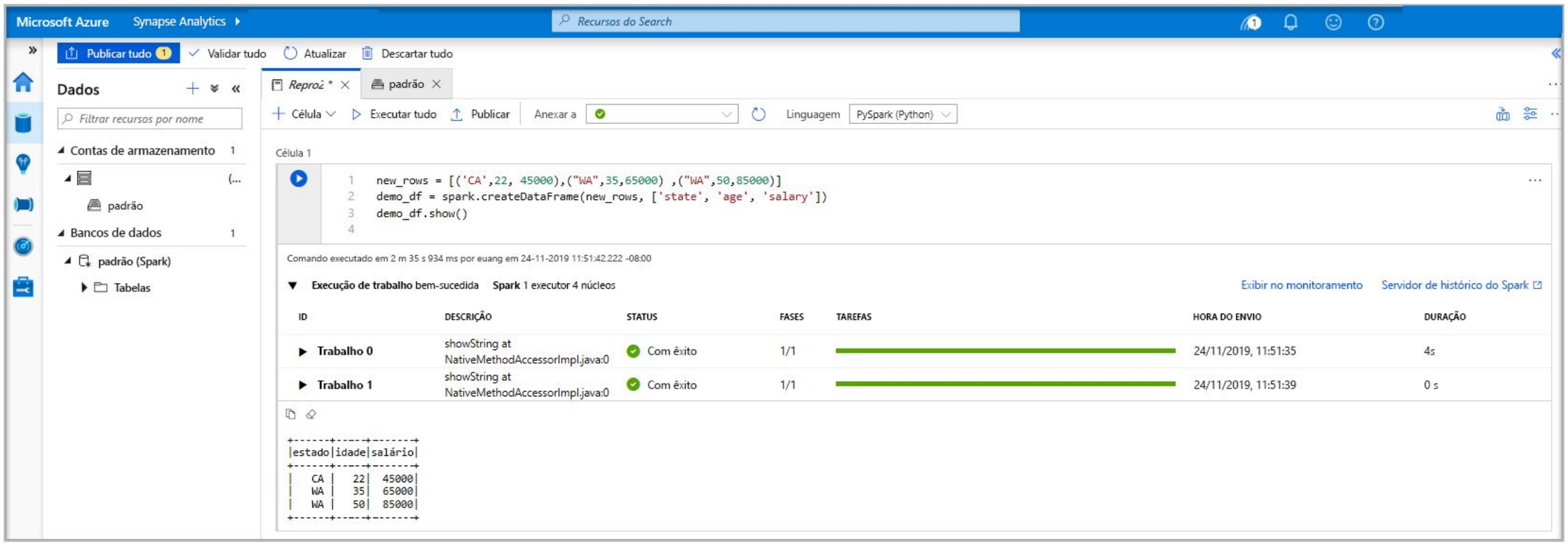

Dashboards
Questions
Pulses
New Question
user analysis
No description yet
1,001
Total Access Count
Total page count by action
count
Count
appliances
electronics
logout
women
Total page count by brand
count
Billy Reid
Lanvin
Reiss
Vans
User access count
605
TOTAL
Electronics - Count of user based on item
air c...
ref...
washi...
Women - Count of user based on brand and useraccount
Total page count by item and brand
Brand
TV's
accessories
Billy Reid
-
7
-
Bonobos
-
3
-
Rows 1-2 of 16
Home furniture's with past 7 days based on user account
count

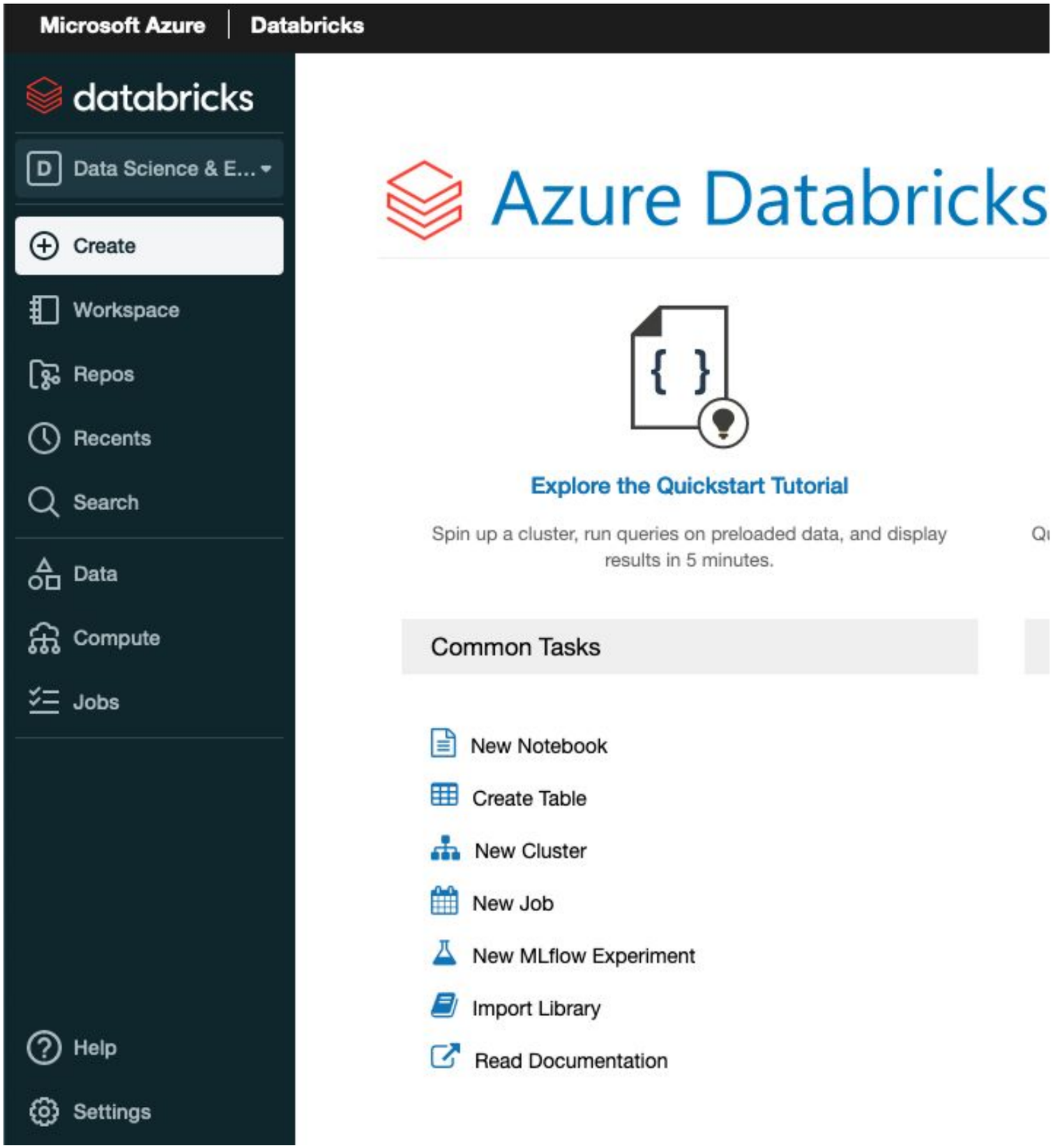
Processamento de Dados



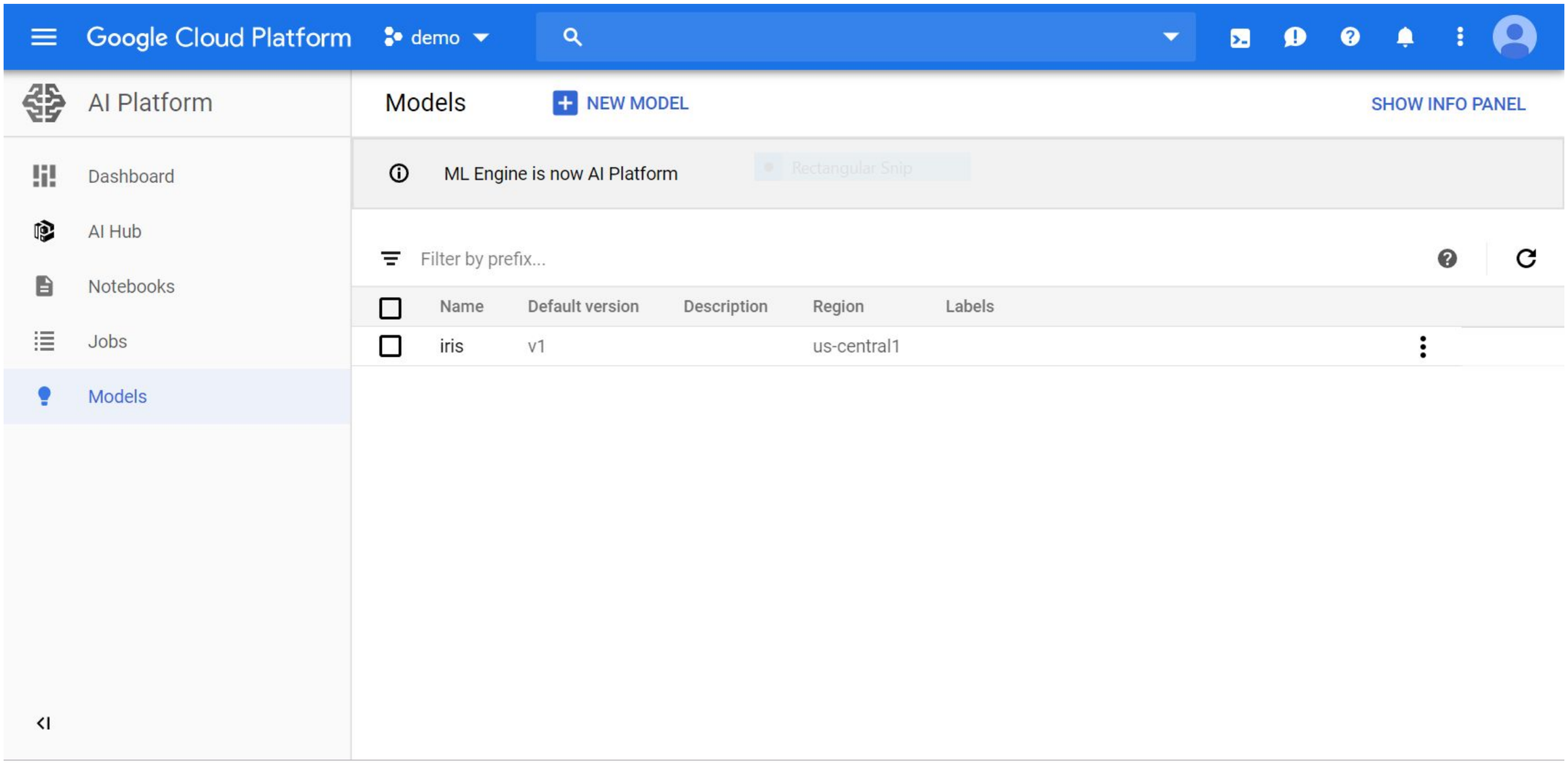
Where Cloud Dataproc fits into GCP

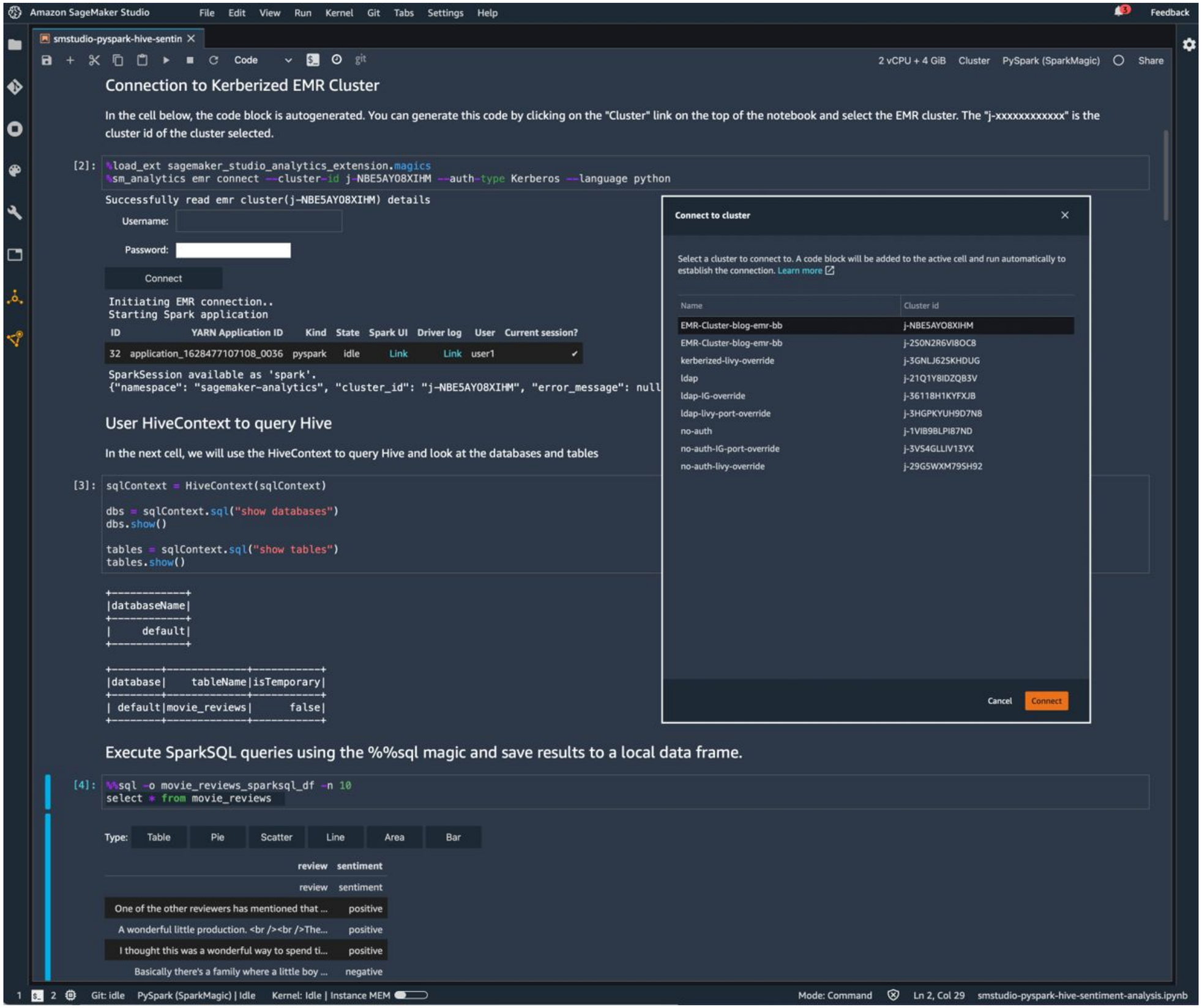



Microsoft Azure
Synapse Analytics
Recursos do Search
Publicar tudo 1
Validar tudo
Atualizar
Descartar tudo
Dados
Filtrar recursos por nome
Contas de armazenamento 1
padrão
Bancos de dados 1
padrão (Spark)
Tabelas
Reproz *
padrão
Célula
Executar tudo
Publicar
Anexar a
Linguagem
PySpark (Python)
Célula 1
1 new_rows = [('CA',22, 45000),("WA",35,65000) ,("WA",50,85000)]
2 demo_df = spark.createDataFrame(new_rows, ['state', 'age', 'salary'])
3 demo_df.show()
4
Comando executado em 2 m 35 s 934 ms por euang em 24-11-2019 11:51:42.222 -08:00
Execução de trabalho bem-sucedida Spark 1 executor 4 núcleos
Exibir no monitoramento
Servidor de histórico do Spark
ID
DESCRIÇÃO
STATUS
FASES
TAREFAS
HORA DO ENVIO
DURAÇÃO
Trabalho 0
showString at NativeMethodAccessorImpl.java:0
Com êxito
1/1
24/11/2019, 11:51:35
4s
Trabalho 1
showString at NativeMethodAccessorImpl.java:0
Com êxito
1/1
24/11/2019, 11:51:39
0 s
+------+----+------+
|estado|idade|salário|
+------+----+------+
CA	22	45000
WA	35	65000
WA	50	85000
+------+----+------+

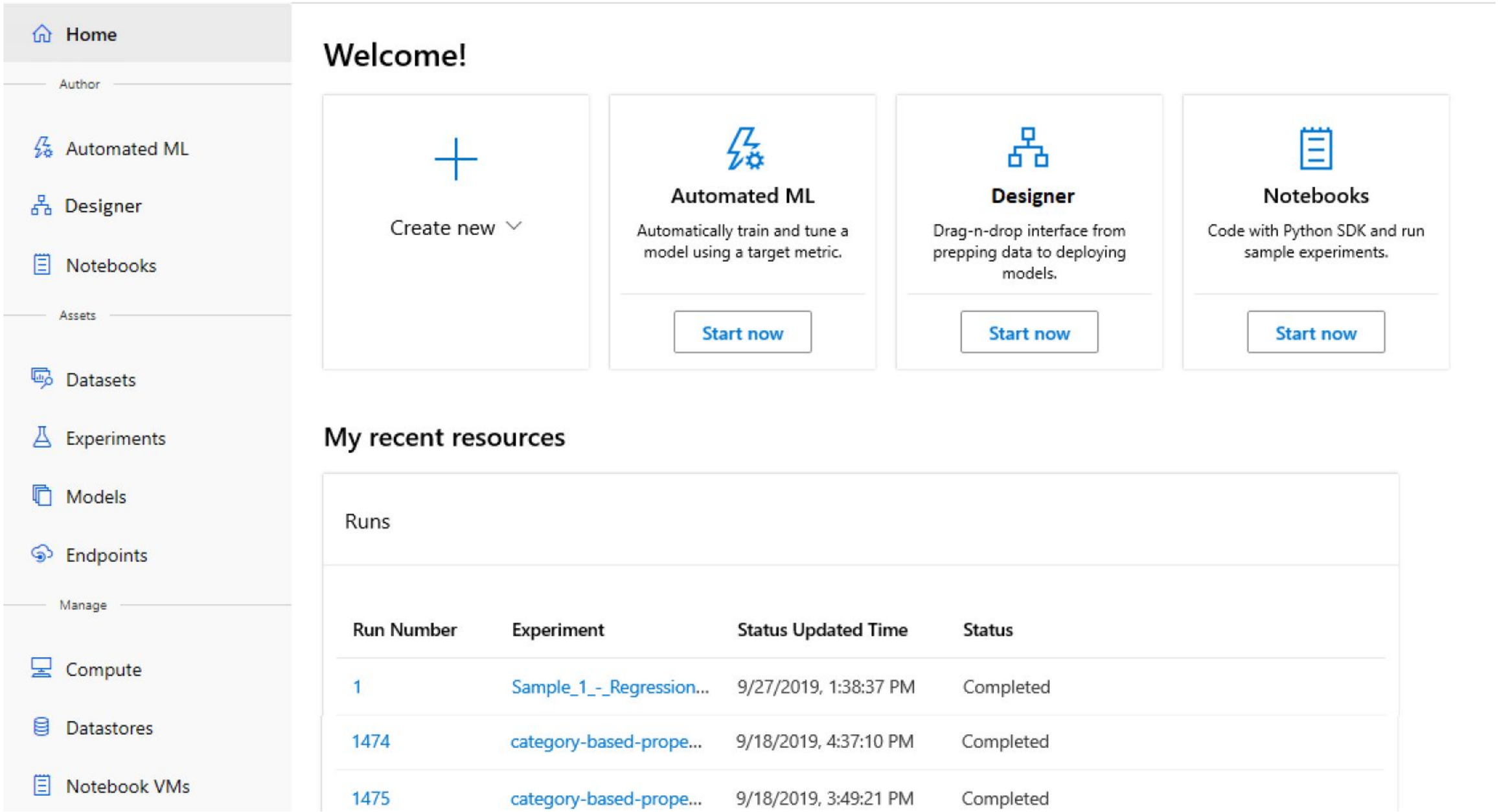

Microsoft Azure
Databricks
databricks
Data Science & E...
Create
Workspace
Repos
Recents
Search
Data
Compute
Jobs
Help
Settings
Azure Databricks
Explore the Quickstart Tutorial
Spin up a cluster, run queries on preloaded data, and display results in 5 minutes.
Common Tasks
New Notebook
Create Table
New Cluster
New Job
New MLflow Experiment
Import Library
Read Documentation

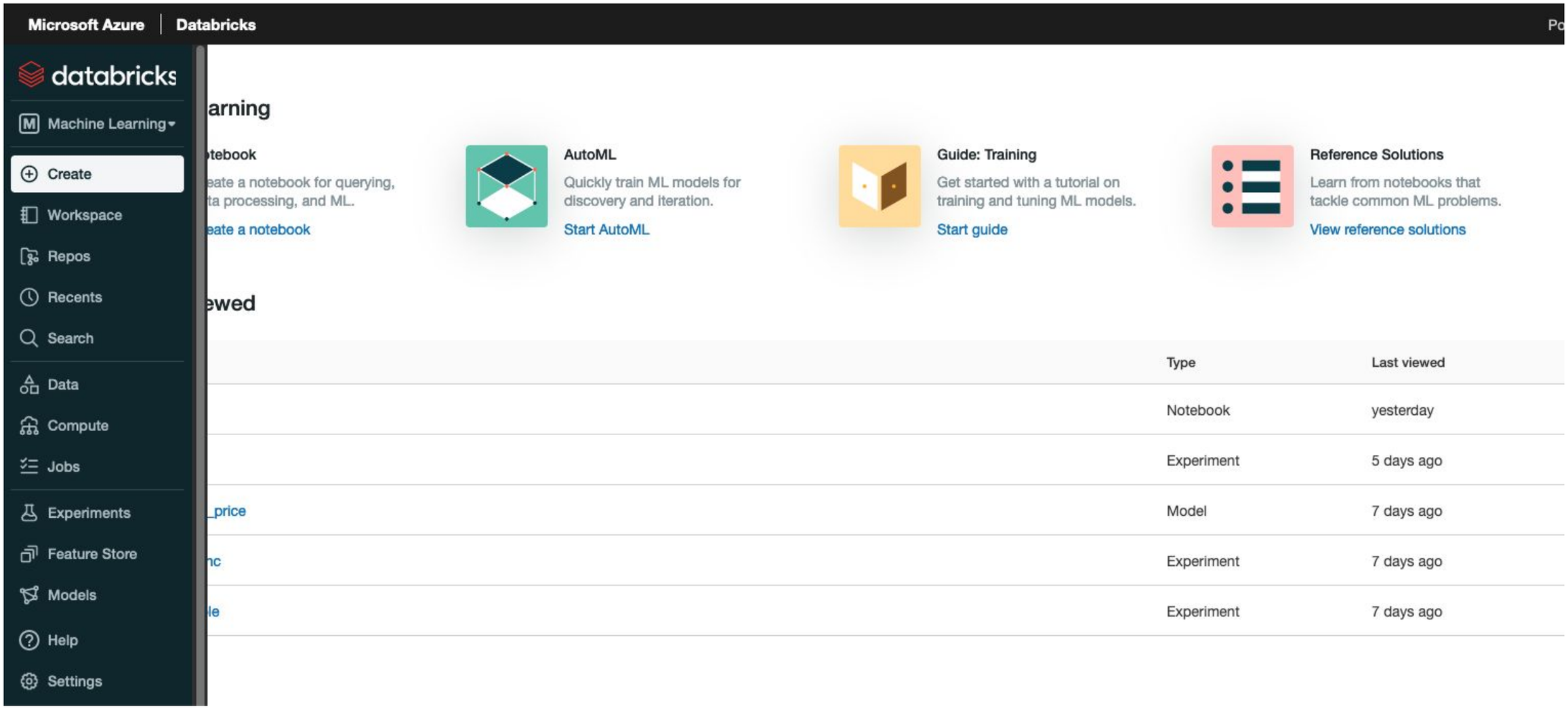
Machine Learning



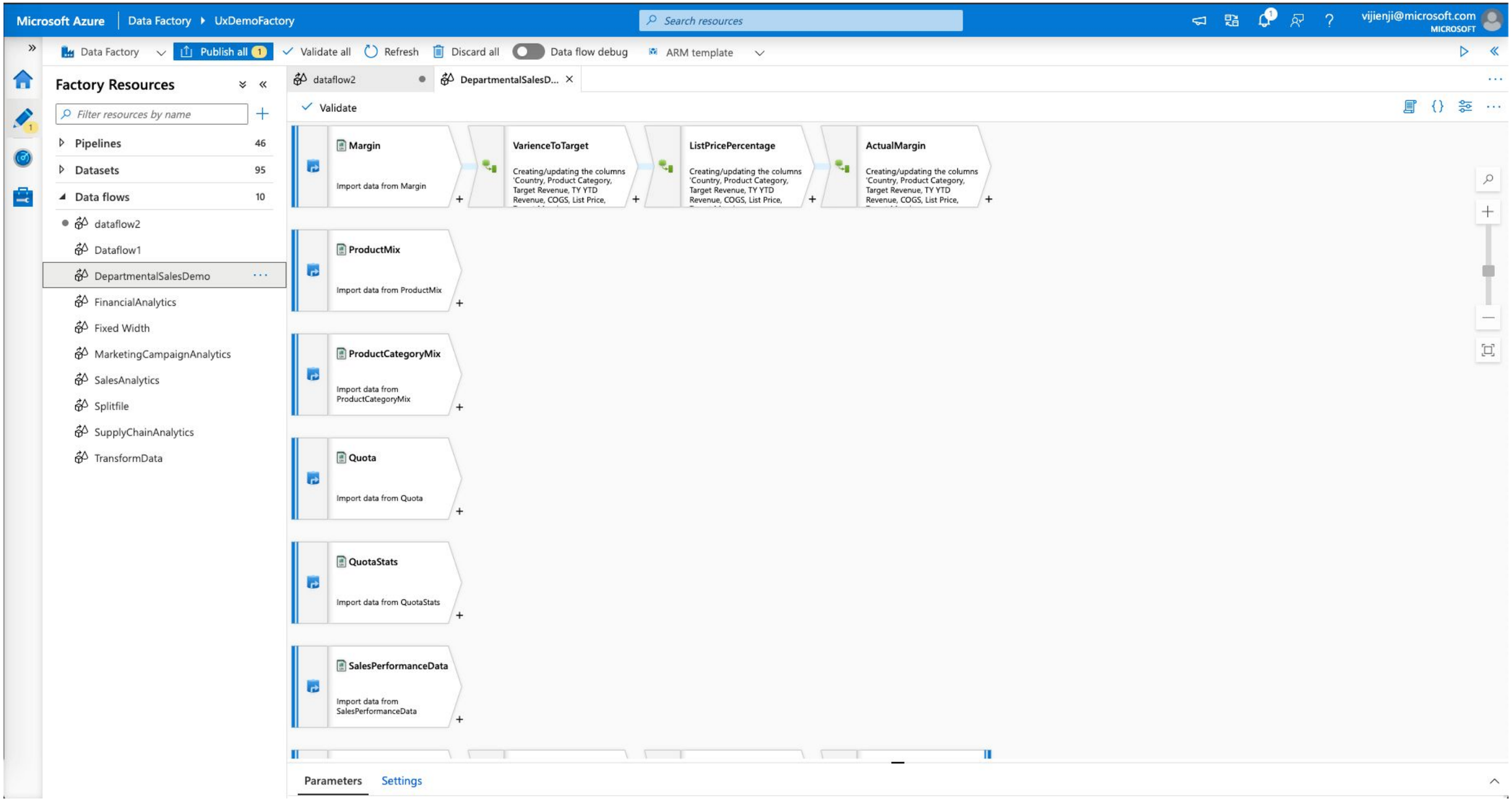

Amazon SageMaker Studio
File Edit View Run Kernel Git Tabs Settings Help
Feedback
smstudio-pyspark-hive-sentin
Code
2 vCPU + 4 GiB Cluster PySpark (SparkMagic) Share
Connection to Kerberized EMR Cluster
In the cell below, the code block is autogenerated. You can generate this code by clicking on the "Cluster" link on the top of the notebook and select the EMR cluster. The "j-xxxxxxxxxxxx" is the cluster id of the cluster selected.
[2]: %load_ext sagemaker_studio_analytics_extension.magics
%sm_analytics emr connect --cluster-id j-NBE5AYO8XIHM --auth-type Kerberos --language python
Successfully read emr cluster(j-NBE5AYO8XIHM) details
Username:
Password:
Connect
Initiating EMR connection..
Starting Spark application
ID YARN Application ID Kind State Spark UI Driver log User Current session?
32 application_1628477107108_0036 pyspark idle Link Link user1 ✔
SparkSession available as 'spark'.
{"namespace": "sagemaker-analytics", "cluster_id": "j-NBE5AYO8XIHM", "error_message": null
Connect to cluster
Select a cluster to connect to. A code block will be added to the active cell and run automatically to establish the connection. Learn more
Name
Cluster id
EMR-Cluster-blog-emr-bb j-NBE5AYO8XIHM
EMR-Cluster-blog-emr-bb j-2S0N2R6VI8OC8
kerberized-livy-override j-3GNLJ62SKHDUG
ldap j-21Q1Y8IDZQB3V
ldap-IG-override j-36118H1KYFXJB
ldap-livy-port-override j-3HGPKYUH9D7N8
no-auth j-1VIB9BLPI87ND
no-auth-IG-port-override j-3VS4GLLIV13YX
no-auth-livy-override j-29G5WXM79SH92
Cancel
Connect
User HiveContext to query Hive
In the next cell, we will use the HiveContext to query Hive and look at the databases and tables
[3]: sqlContext = HiveContext(sqlContext)
dbs = sqlContext.sql("show databases")
dbs.show()
tables = sqlContext.sql("show tables")
tables.show()
+------------+
|databaseName|
+------------+
| default|
+------------+
+--------+------------+-----------+
|database| tableName|isTemporary|
+--------+------------+-----------+
| default|movie_reviews| false|
+--------+------------+-----------+
Execute SparkSQL queries using the %%sql magic and save results to a local data frame.
[4]: %%sql -o movie_reviews_sparksql_df -n 10
select * from movie_reviews
Type: Table Pie Scatter Line Area Bar
review sentiment
review sentiment
One of the other reviewers has mentioned that ... positive
A wonderful little production.

The... positive
I thought this was a wonderful way to spend ti... positive
Basically there's a family where a little boy ... negative
1 2 Git: idle PySpark (SparkMagic) | Idle Kernel: Idle | Instance MEM
Mode: Command Ln 2, Col 29 smstudio-pyspark-hive-sentiment-analysis.ipynb


Home
Author
Automated ML
Designer
Notebooks
Assets
Datasets
Experiments
Models
Endpoints
Manage
Compute
Datastores
Notebook VMs
Welcome!
Create new
Automated ML
Automatically train and tune a model using a target metric.
Start now
Designer
Drag-n-drop interface from prepping data to deploying models.
Start now
Notebooks
Code with Python SDK and run sample experiments.
Start now
My recent resources
Runs
Run Number
Experiment
Status Updated Time
Status
1
Sample_1_-_Regression...
9/27/2019, 1:38:37 PM
Completed
1474
category-based-prope...
9/18/2019, 4:37:10 PM
Completed
1475
category-based-prope...
9/18/2019, 3:49:21 PM
Completed


Microsoft Azure
Databricks
databricks
Machine Learning
Create
Workspace
Repos
Recents
Search
Data
Compute
Jobs
Experiments
Feature Store
Models
Help
Settings
arning
tebook
eate a notebook for querying,
ta processing, and ML.
eate a notebook
AutoML
Quickly train ML models for discovery and iteration.
Start AutoML
Guide: Training
Get started with a tutorial on training and tuning ML models.
Start guide
Reference Solutions
Learn from notebooks that tackle common ML problems.
View reference solutions
ewed
Type
Last viewed
Notebook
yesterday
Experiment
5 days ago
_price
Model
7 days ago
nc
Experiment
7 days ago
le
Experiment
7 days ago

Orquestradores



Google Cloud Platform My GCloud Project

Cloud Composer API

Google Enterprise API

Manages Apache Airflow environments on Google Cloud Platform.

ENABLE TRY THIS API

OVERVIEW DOCUMENTATION

Overview

Manages Apache Airflow environments on Google Cloud Platform.

Additional details

Type: SaaS & APIs

Last updated: 7/22/21

Category: Google Enterprise APIs

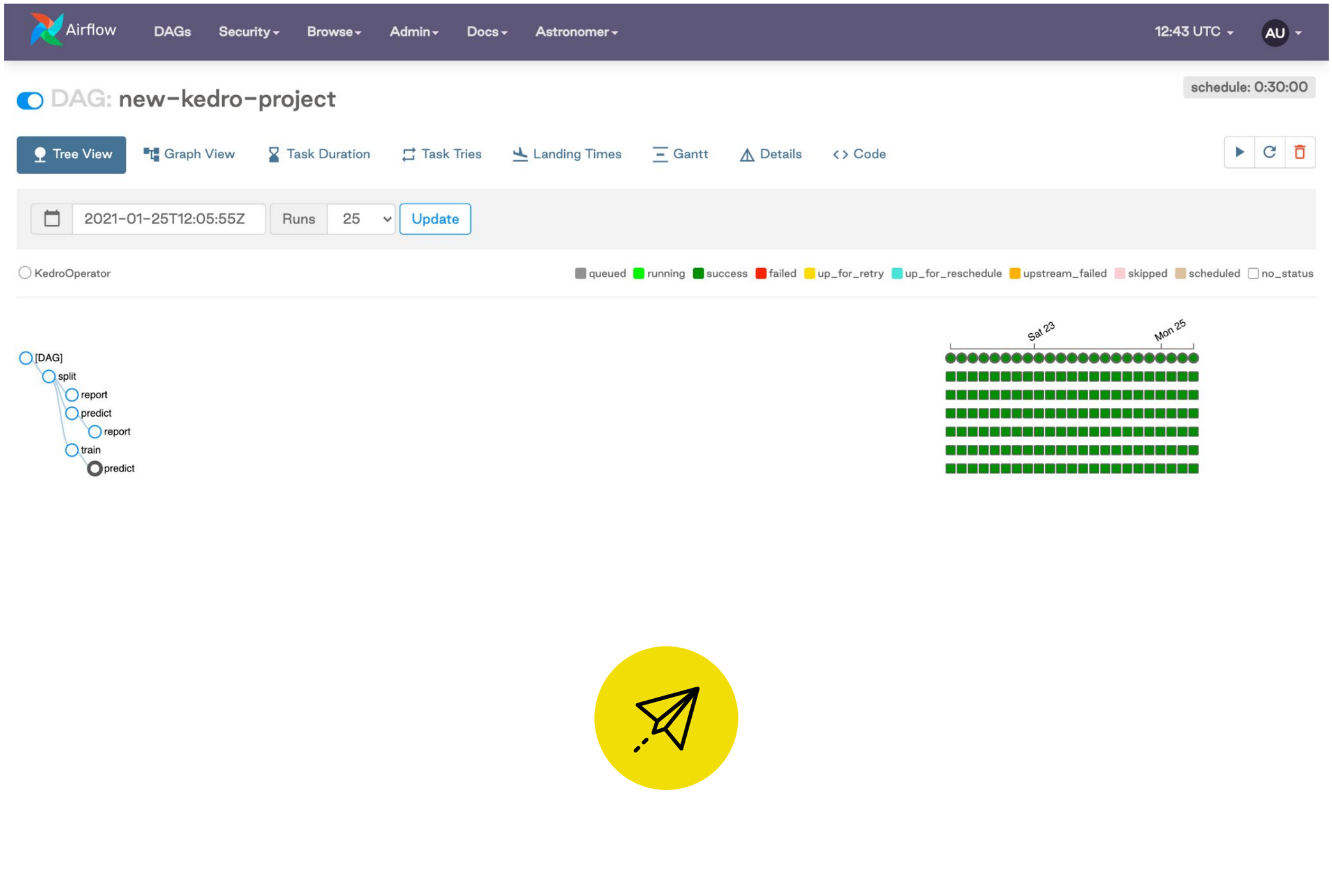
Service name: composer.googleapis.com

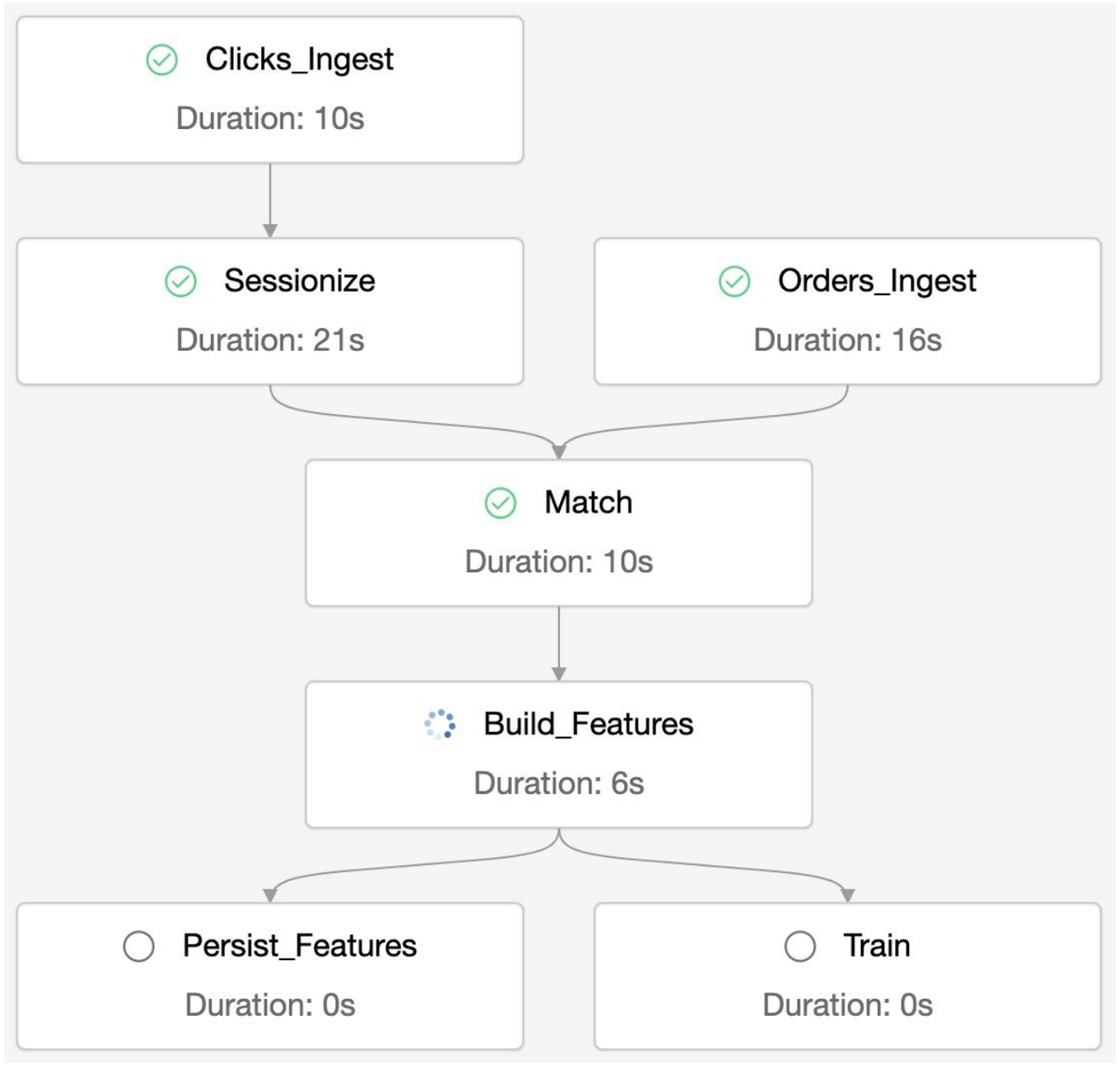
Airflow DAGs Security Browse Admin Docs 21:11 UTC RH

DAGs

All 26 Active 10 Paused 16 Filter DAGs by tag Search DAGs

	DAG	Owner	Runs	Schedule	Last Run	Recent Tasks	Actions	Links
on	example_bash_operator example example2	airflow	2	0 0 * * *	2020-10-26, 21:08:11	6	▶ ↻ 🗑	•••
on	example_branch_dop_operator_v3 example	airflow		*/1 * * * *			▶ ↻ 🗑	•••
off	example_branch_operator example example2	airflow	1	@daily	2020-10-23, 14:09:17	11	▶ ↻ 🗑	•••
on	example_complex example example2 example3	airflow	1 1	None	2020-10-26, 21:08:04	37 37	▶ ↻ 🗑	•••
on	example_external_task_marker_child	airflow	1	None	2020-10-26, 21:07:33	2	▶ ↻ 🗑	•••
on	example_external_task_marker_parent	airflow	1	None	2020-10-26, 21:08:34	1	▶ ↻ 🗑	•••
on	example_kubernetes_executor example example2	airflow		None			▶ ↻ 🗑	•••
on	example_kubernetes_executor_config example3	airflow	1	None	2020-10-26, 21:07:40	5	▶ ↻ 🗑	•••
on	example_nested_branch_dag example	airflow	1	@daily	2020-10-26, 21:07:37	9	▶ ↻ 🗑	•••
off	example_passing_params_via_test_command example	airflow		*/1 * * * *			▶ ↻ 🗑	•••


Airflow
DAGs
Security
Browse
Admin
Docs
Astronomer
12:43 UTC
AU
DAG: new-kedro-project
schedule: 0:30:00
Tree View
Graph View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
2021-01-25T12:05:55Z
Runs
25
Update
KedroOperator
queued
running
success
failed
up_for_retry
up_for_reschedule
upstream_failed
skipped
scheduled
no_status
[DAG]
split
report
predict
report
train
predict
Sat 23
Mon 25


Clicks_Ingest
Duration: 10s
Sessionize
Duration: 21s
Orders_Ingest
Duration: 16s
Match
Duration: 10s
Build_Features
Duration: 6s
Persist_Features
Duration: 0s
Train
Duration: 0s

Resumo da Aula

Análise de Dados com SQL

Processamento de dados

Machine Learning

Orquestradores

data® expert.

Data Science & Machine Learning

MDC04 - Data Sicence: Open Source

Consultor: Murilo Mendonça

Tópicos desta aula:

01

Treino de modelos

02

Ciclo de vida e pipelines

03

Mlflow

Treino de modelos

sklearn.base.BaseEstimator

class sklearn.base.BaseEstimator [source]

Base class for all estimators in scikit-learn.

Notes

All estimators should specify all the parameters that can be set at the class level in their `__init__` as explicit keyword arguments (no `*args` or `**kwargs`).

Methods

`get_params`([deep])	Get parameters for this estimator.
`set_params`(**params)	Set the parameters of this estimator.

get_params(*deep=True*) [source]

Get parameters for this estimator.

Parameters:	**deep : *bool, default=True*** If True, will return the parameters for this estimator and contained subobjects that are estimators.
Returns:	**params : *dict*** Parameter names mapped to their values.

set_params(***params*) [source]

Set the parameters of this estimator.

The method works on simple estimators as well as on nested objects (such as `Pipeline`). The latter have parameters of the form `<component>__<parameter>` so that it's possible to update each component of a nested object.

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentro
              metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

```
class MyModel(tf.keras.Model):
  def __init__(self):
    super(MyModel, self).__init__()
    self.conv1 = Conv2D(32, 3, activation='relu')
    self.flatten = Flatten()
    self.d1 = Dense(128, activation='relu')
    self.d2 = Dense(10, activation='softmax')

  def call(self, x):
    x = self.conv1(x)
    x = self.flatten(x)
    x = self.d1(x)
    return self.d2(x)
model = MyModel()

with tf.GradientTape() as tape:
  logits = model(images)
  loss_value = loss(logits, labels)
grads = tape.gradient(loss_value, model.trainable
optimizer.apply_gradients(zip(grads, model.traina
```

```python
import torch
import math


# Create Tensors to hold input and outputs.
x = torch.linspace(-math.pi, math.pi, 2000)
y = torch.sin(x)

# Prepare the input tensor (x, x^2, x^3).
p = torch.tensor([1, 2, 3])
xx = x.unsqueeze(-1).pow(p)

# Use the nn package to define our model and loss function.
model = torch.nn.Sequential(
    torch.nn.Linear(3, 1),
    torch.nn.Flatten(0, 1)
)
loss_fn = torch.nn.MSELoss(reduction='sum')

# Use the optim package to define an Optimizer that will update the weights of
# the model for us. Here we will use RMSprop; the optim package contains many other
# optimization algorithms. The first argument to the RMSprop constructor tells the
# optimizer which Tensors it should update.
learning_rate = 1e-3
optimizer = torch.optim.RMSprop(model.parameters(), lr=learning_rate)
for t in range(2000):
    # Forward pass: compute predicted y by passing x to the model.
    y_pred = model(xx)

    # Compute and print loss.
    loss = loss_fn(y_pred, y)
    if t % 100 == 99:
        print(t, loss.item())
```


CatBoost


DASK


LightGBM


XGBoost


fast.ai


PROPHET

Ciclo de vida e pipelines

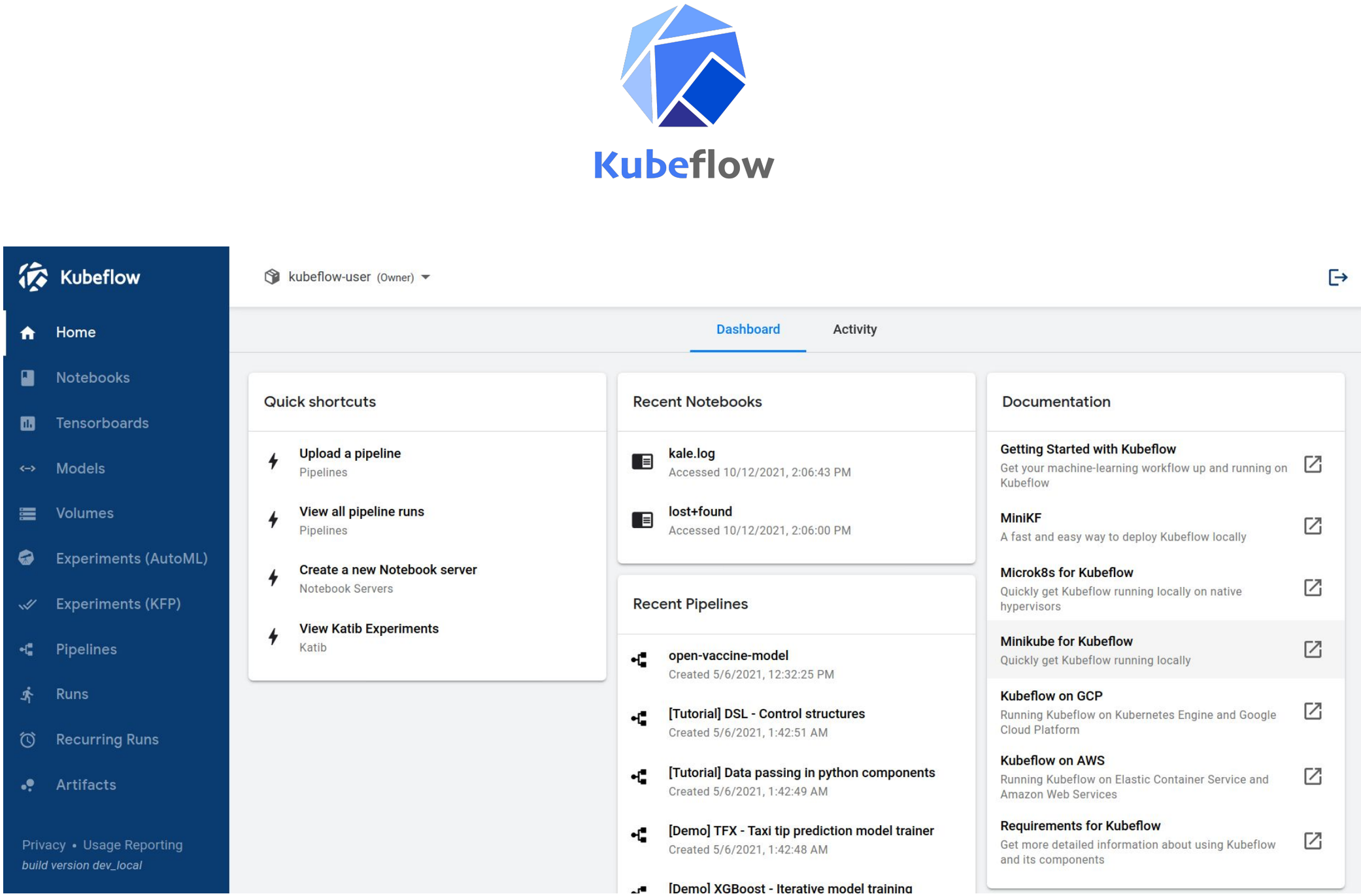
Blob Storage, and HDFS. See `dvc remote add` for more details and examples.

You can review the experiment results with `dvc exp show` and see these metrics and results in a nicely formatted table:

```
$ dvc exp show
```

Experiment	Created	loss	acc	train.epoch
workspace	-	0.23282	0.9152	10
7317bc6	Jul 18, 2021	-	-	10
└── 1a1d858 [exp-6dccf]	03:21 PM	0.23282	0.9152	10

```
          *
    +-------+
    | train |
    +-------+
```


Kubeflow
kubeflow-user (Owner)
Home
Notebooks
Tensorboards
Models
Volumes
Experiments (AutoML)
Experiments (KFP)
Pipelines
Runs
Recurring Runs
Artifacts
Privacy • Usage Reporting
build version dev_local
Dashboard
Activity
Quick shortcuts
Upload a pipeline
Pipelines
View all pipeline runs
Pipelines
Create a new Notebook server
Notebook Servers
View Katib Experiments
Katib
Recent Notebooks
kale.log
Accessed 10/12/2021, 2:06:43 PM
lost+found
Accessed 10/12/2021, 2:06:00 PM
Recent Pipelines
open-vaccine-model
Created 5/6/2021, 12:32:25 PM
[Tutorial] DSL - Control structures
Created 5/6/2021, 1:42:51 AM
[Tutorial] Data passing in python components
Created 5/6/2021, 1:42:49 AM
[Demo] TFX - Taxi tip prediction model trainer
Created 5/6/2021, 1:42:48 AM
Documentation
Getting Started with Kubeflow
Get your machine-learning workflow up and running on Kubeflow
MiniKF
A fast and easy way to deploy Kubeflow locally
Microk8s for Kubeflow
Quickly get Kubeflow running locally on native hypervisors
Minikube for Kubeflow
Quickly get Kubeflow running locally
Kubeflow on GCP
Running Kubeflow on Kubernetes Engine and Google Cloud Platform
Kubeflow on AWS
Running Kubeflow on Elastic Container Service and Amazon Web Services
Requirements for Kubeflow
Get more detailed information about using Kubeflow and its components

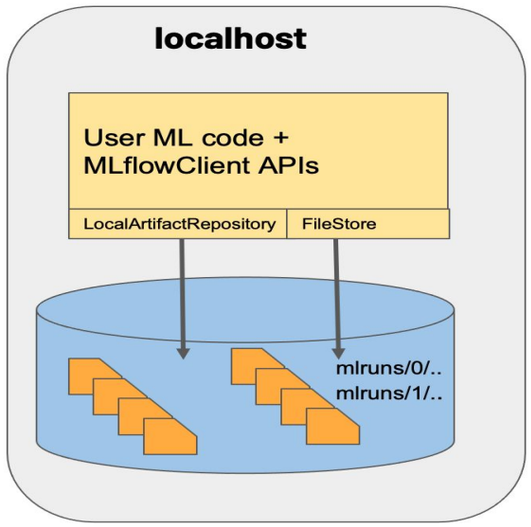

```
# Find the local path of the latest version IrisClassifier saved bundle
saved_path=$(bentoml get IrisClassifier:latest --print-location --quiet)


# Build docker image using saved_path directory as the build context, replace the
# {username} below to your docker hub account name
docker build -t {username}/iris_classifier_bento_service $saved_path

# Run a container with the docker image built and expose port 5000
docker run -p 5000:5000 {username}/iris_classifier_bento_service

# Push the docker image to docker hub for deployment
docker push {username}/iris_classifier_bento_service
```

Here's an example deployment you can create in a Kubernetes cluster using the docker image built above:

```
apiVersion: apps/v1 # for k8s versions before 1.9.0 use apps/v1beta2
kind: Deployment
metadata:
  name: iris_classifier
spec:
  selector:
    matchLabels:
      app: iris_classifier
  replicas: 3
  template:
    metadata:
      labels:
        app: iris_classifier
    spec:
      containers:
      - name: iris_classifier_bento_service
        image: {username}/iris_classifier_bento_service:latest
        ports:
        - containerPort: 5000
```

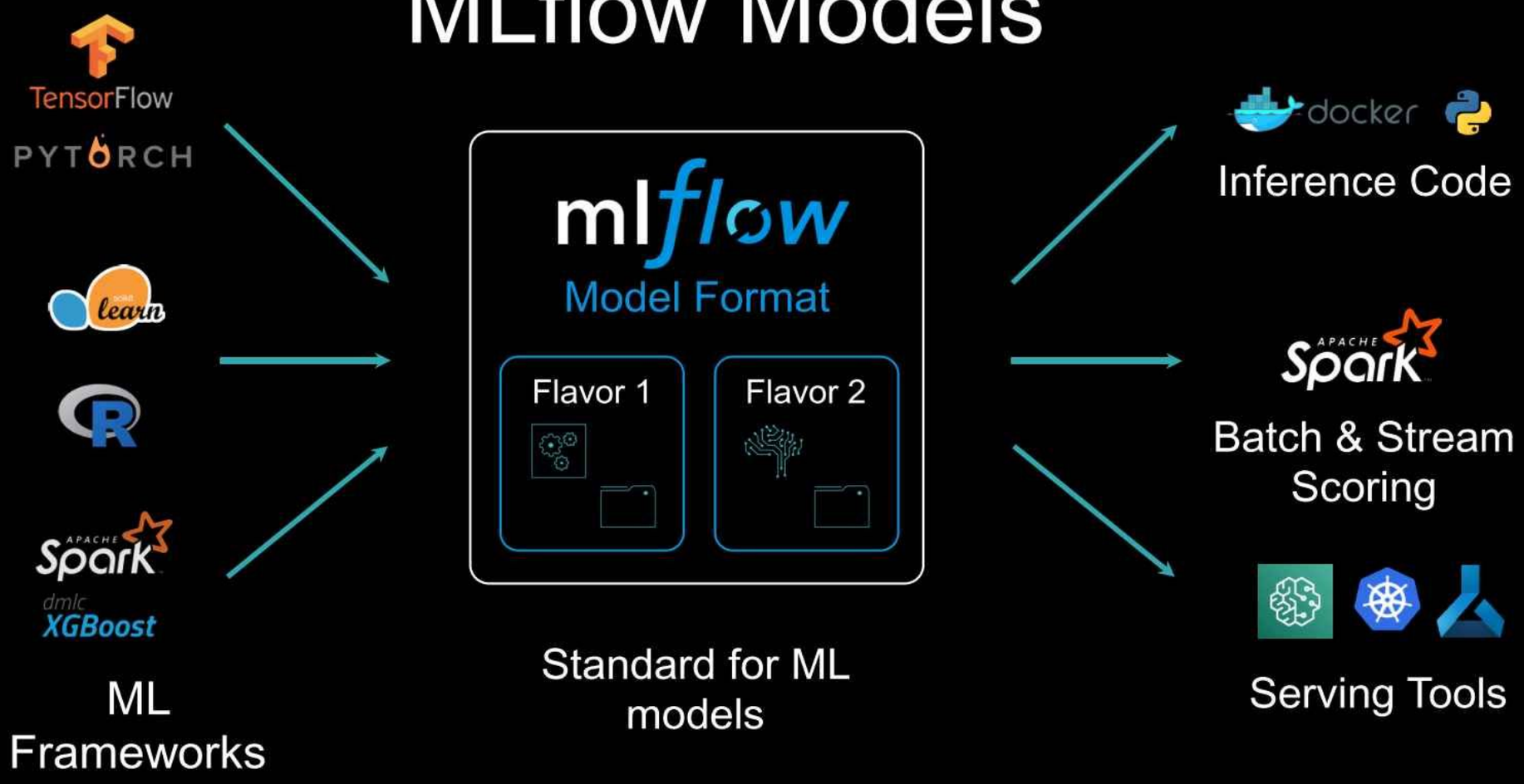
MLflow

			Parameters		Metrics		
	Start Time	User	mtry	trees	mae	rmse	rsq
	2020-08-16 10:55:54	mdneuzerling	5	700	0.156	0.259	0.663
	2020-08-13 17:02:13	mdneuzerling	3	500	0.169	0.286	0.521
	2020-08-13 15:41:07	mdneuzerling	2	100	0.15	0.246	0.698
	2020-08-13 15:40:59	mdneuzerling	2	1500	0.152	0.247	0.694
	2020-08-13 15:40:53	mdneuzerling	2	2000	0.152	0.246	0.698
	2020-08-13 15:39:19	mdneuzerling	2	1000	0.152	0.245	0.701
	2020-08-13 15:37:30	mdneuzerling	5	700	0.158	0.262	0.655
	2020-08-13 15:35:34	mdneuzerling	5	700	0.158	0.261	0.657
	2020-08-13 15:34:37	mdneuzerling	5	700	0.157	0.259	0.663



Scenario 1: MLflow on the localhost

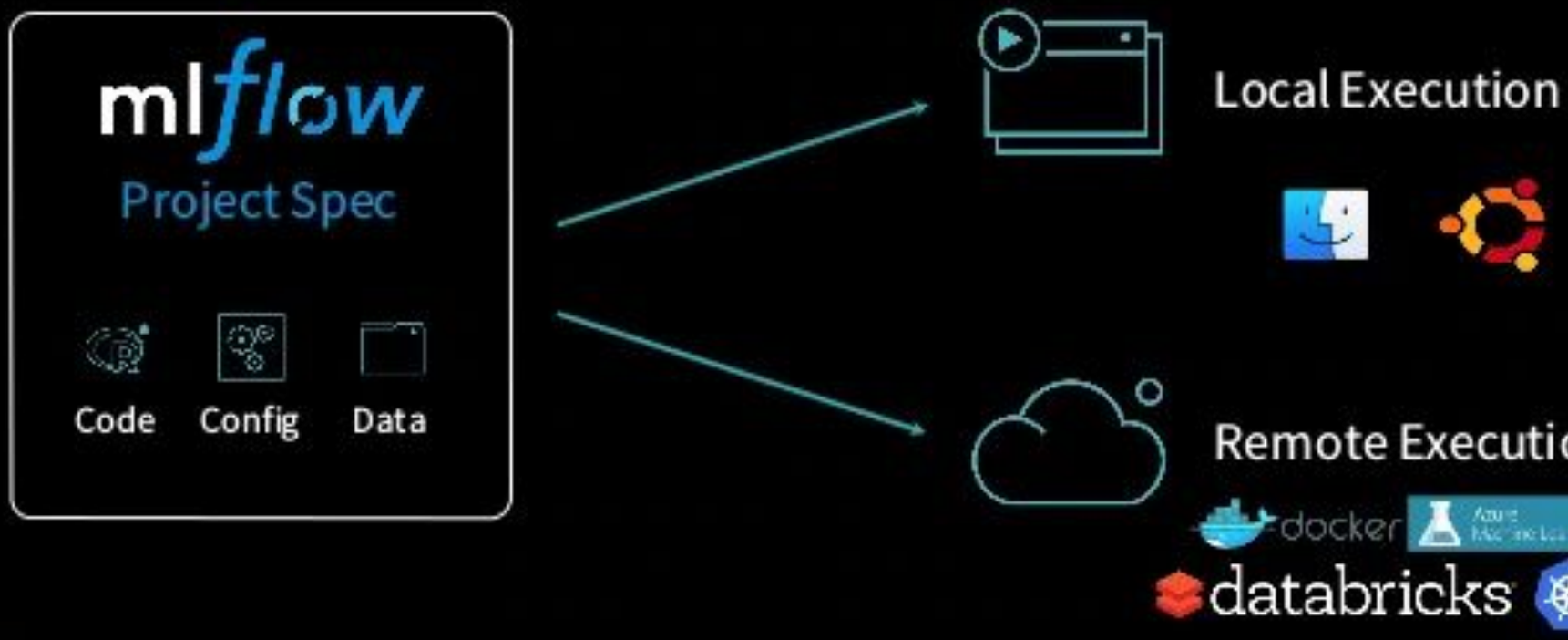
MLflow Models



databricks


```
mlflow models build-docker -m "runs:/some-run-uuid/my-model" -n "my-image-name"
```

MLflow Projects



mlflow GitHub Docs

Registered Models

search model name

Name	Latest Version	Staging	Production	Last Modified
Model A	Version 1	Version 1	–	2019-10-16 22:51:19
Model B	Version 1	–	–	2019-10-16 22:51:52

< 1 >

RESUMO

01

Treino de modelos

02

Ciclo de vida e pipelines

03

Mlflow

data® expert.

Data Science & Machine Learning

MDC06 - Resumo do Módulo

Consultor: Murilo Mendonça

O que vimos neste módulo:

01

Conceitos de nuvem

02

Principais ferramentas de Data Science

03

Open Source Data Science