# FEED THE DATA SCIENTISTS

SENTIMENT-ANALYSIS-PYTHON

# A CREATIVE CLOUD SOLUTION FOR DATA SCIENTISTS & OTHER COMPANIES

https://huggingface.co/blog/sentiment-analysis-python

The model into the container is the fourth section of this paper:

## 4. Analyzing Tweets with Sentiment Analysis and Python

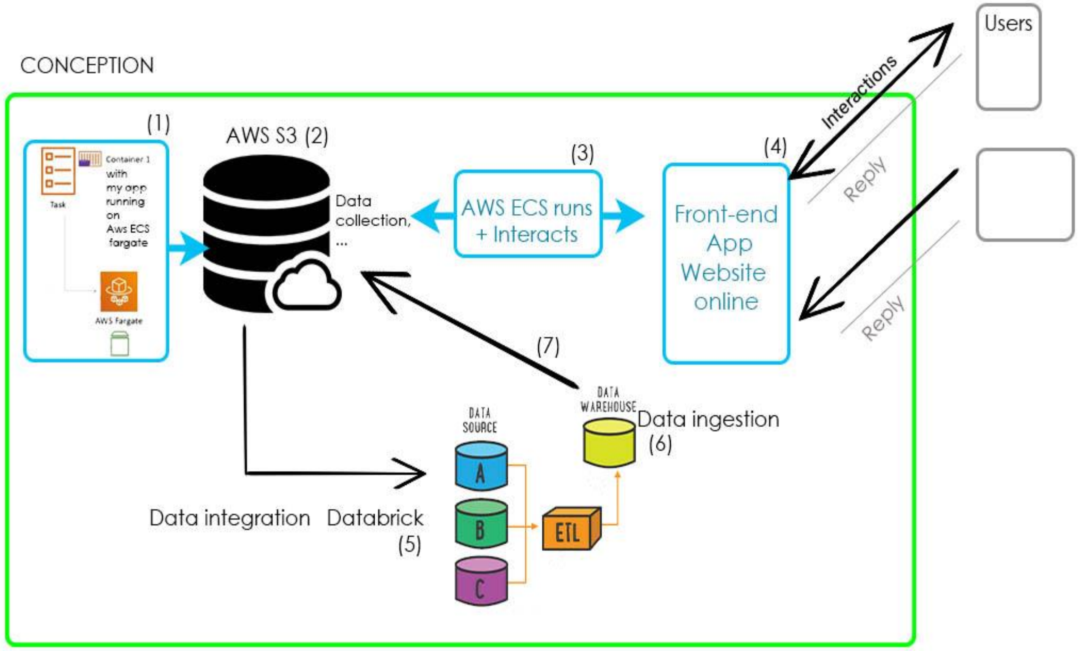
The model within the container is the fourth section of this paper:

Codes run into Vscode on a computer for the test phase (as you could see on a following screenshot) in relation to Anaconda prompt librairies.

A "X" developer account allows me the access of the APIs. We could imagine as it's explained in the link from Huggingface, being a multinational Films producer wishing an analyze and to know the public opinion about the release of a movie, or being Social Data Scientists;

We also might be working for the French President and make an assessment of his last decision making and what the general opinion thinks on it...

The provided code is then containerized (Docker) and it runs by aws fargate cloud service. Users here, are people using the X application with a freedom of speech.



# HOW TO PROCESS? STEP BY STEP

1/ Creation of codes using Python Environments in Visual Studio Code (Vscode) with the X APIs (X developer account) https://huggingface.co/blog/sentiment-analysis-python

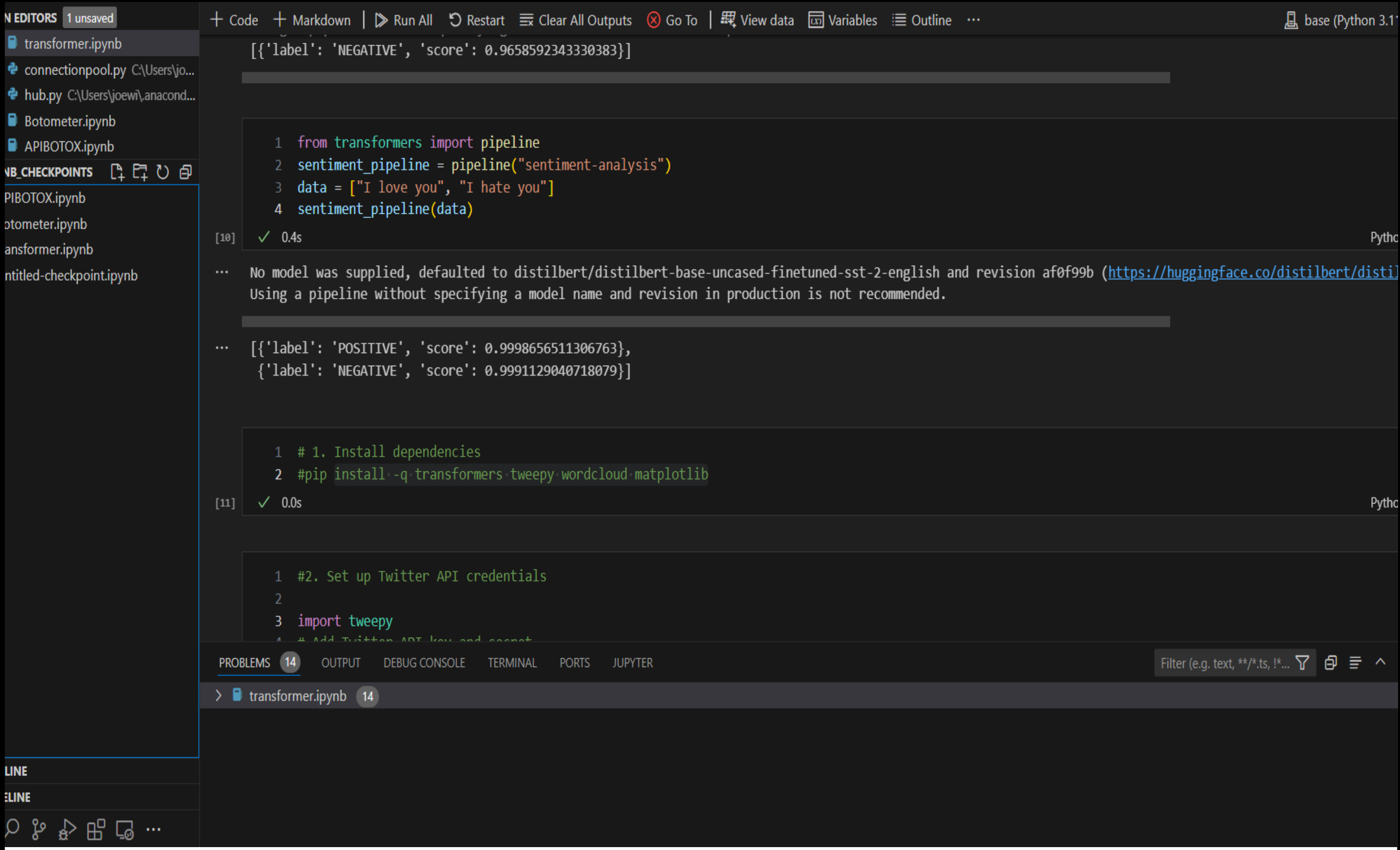
2/ Containerization of the model (Docker) after test

2/ AWS ECS Fargate + S3 services (creation account)

3/ Databricks account

4/ Terraform Installation, creation of the compliant files, interactions between the different types of "modules" <->

5/ Databricks Integration + Ingestion ETL



```
[{'label': 'NEGATIVE', 'score': 0.9658592343330383}]
```

```
from transformers import pipeline
sentiment_pipeline = pipeline("sentiment-analysis")
data = ["I love you", "I hate you"]
sentiment_pipeline(data)
```

[10] ✓ 0.4s

```
No model was supplied, defaulted to distilbert/distilbert-base-uncased-finetuned-sst-2-english and revision af0f99b (https://huggingface.co/distilbert/disti
Using a pipeline without specifying a model name and revision in production is not recommended.
```

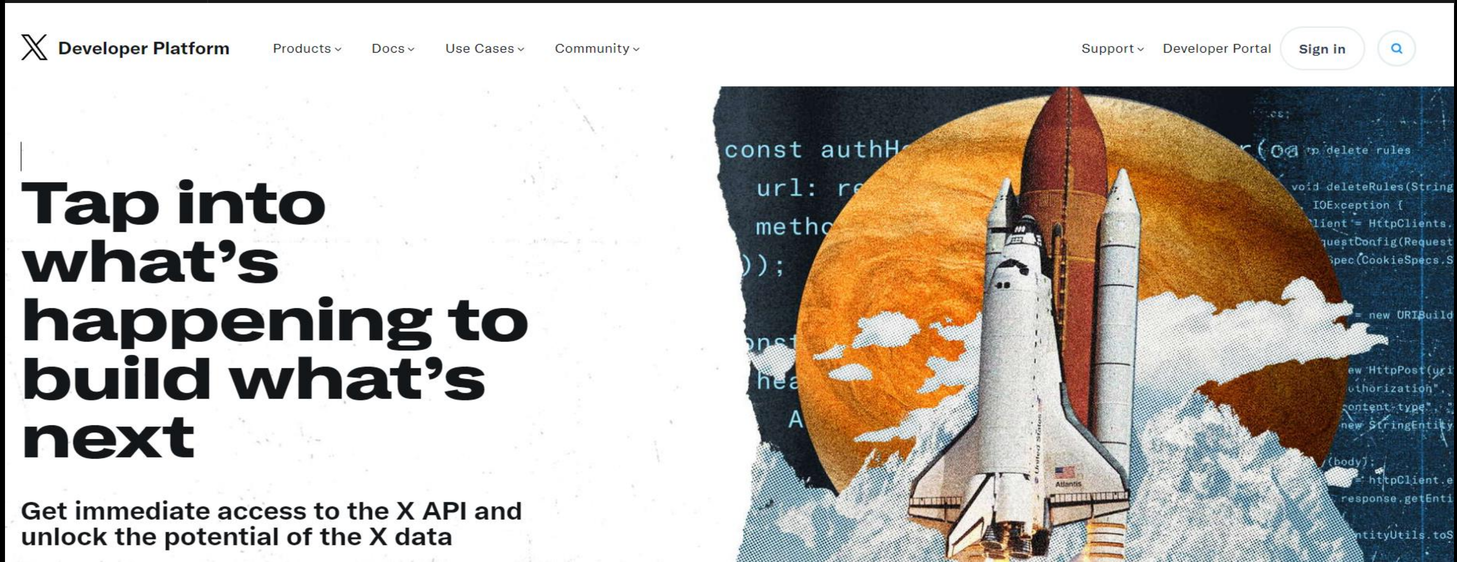
```
[{'label': 'POSITIVE', 'score': 0.9998656511306763},
 {'label': 'NEGATIVE', 'score': 0.9991129040718079}]
```

```
# 1. Install dependencies
#pip install -q transformers tweepy wordcloud matplotlib
```

[11] ✓ 0.0s

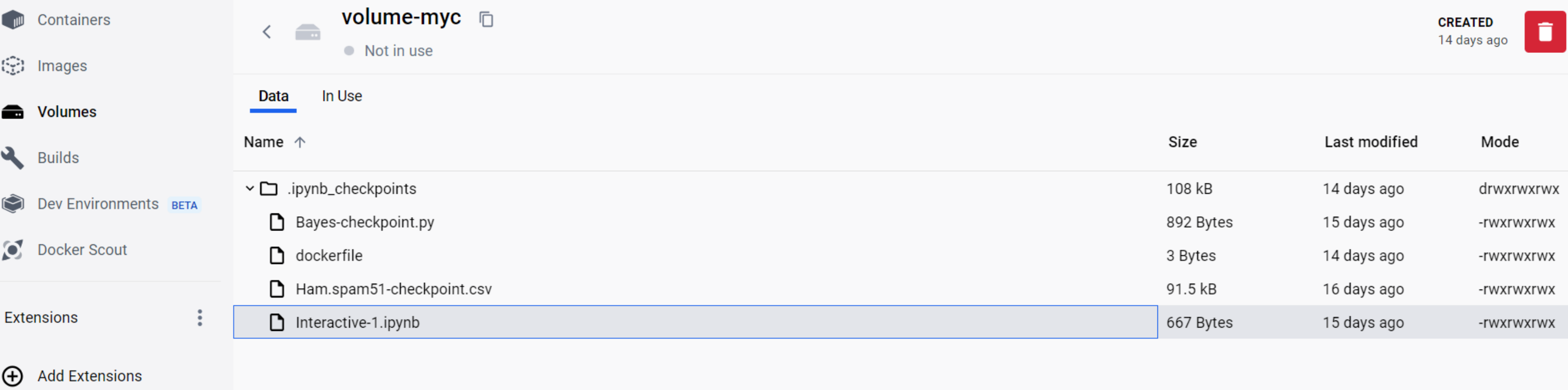
```
#2. Set up Twitter API credentials

import tweepy
# Add Twitter API key and secret
```



Tap into what’s happening to build what’s next

Get immediate access to the X API and unlock the potential of the X data

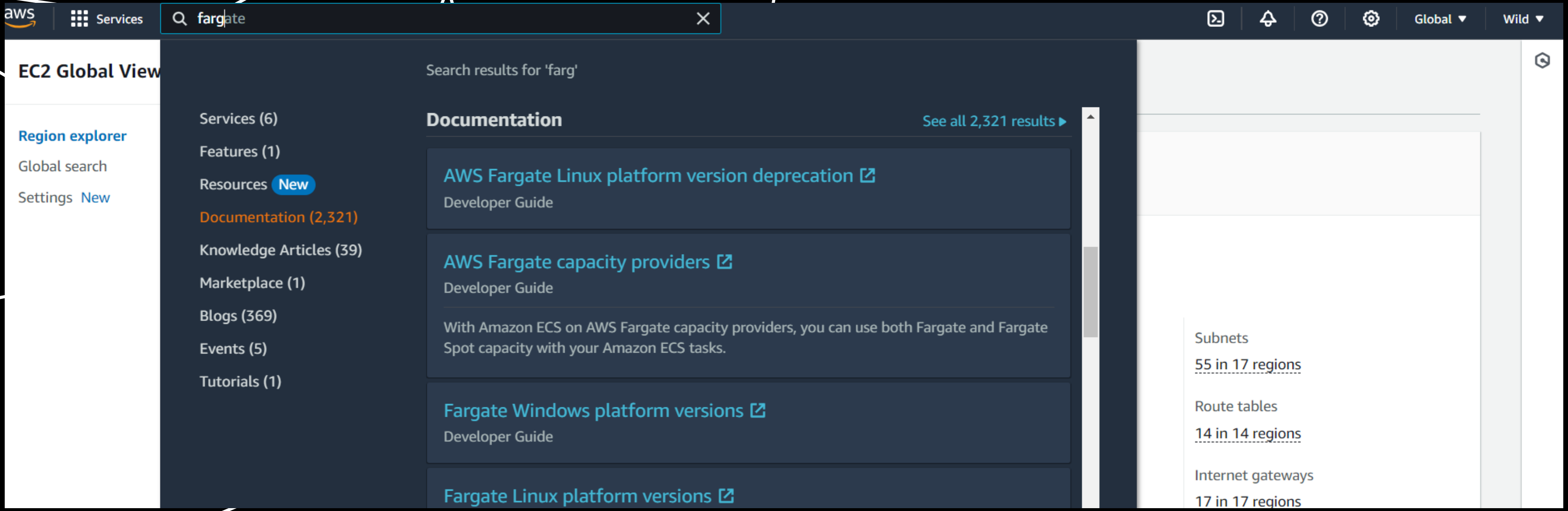


volume-myc

Not in use

CREATED
14 days ago

| Name ↑ | Size | Last modified | Mode |
|---|---|---|---|
| .ipynb_checkpoints | 108 kB | 14 days ago | drwxrwxrwx |
| Bayes-checkpoint.py | 892 Bytes | 15 days ago | -rwxrwxrwx |
| dockerfile | 3 Bytes | 14 days ago | -rwxrwxrwx |
| Ham.spam51-checkpoint.csv | 91.5 kB | 16 days ago | -rwxrwxrwx |
| Interactive-1.ipynb | 667 Bytes | 15 days ago | -rwxrwxrwx |



fargate

Search results for 'farg'

Documentation

See all 2,321 results ▸

AWS Fargate Linux platform version deprecation
Developer Guide

AWS Fargate capacity providers
Developer Guide

With Amazon ECS on AWS Fargate capacity providers, you can use both Fargate and Fargate Spot capacity with your Amazon ECS tasks.

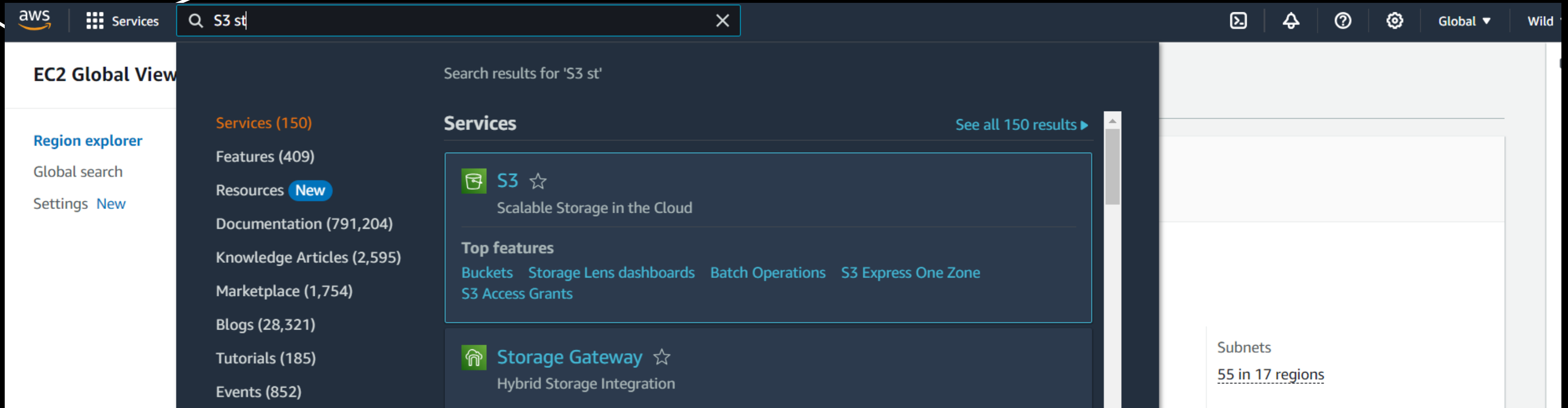
Fargate Windows platform versions
Developer Guide

Fargate Linux platform versions

Subnets
55 in 17 regions

Route tables
14 in 14 regions

Internet gateways
17 in 17 regions



S3 st

Search results for 'S3 st'

Services

See all 150 results ▸

S3 ☆
Scalable Storage in the Cloud

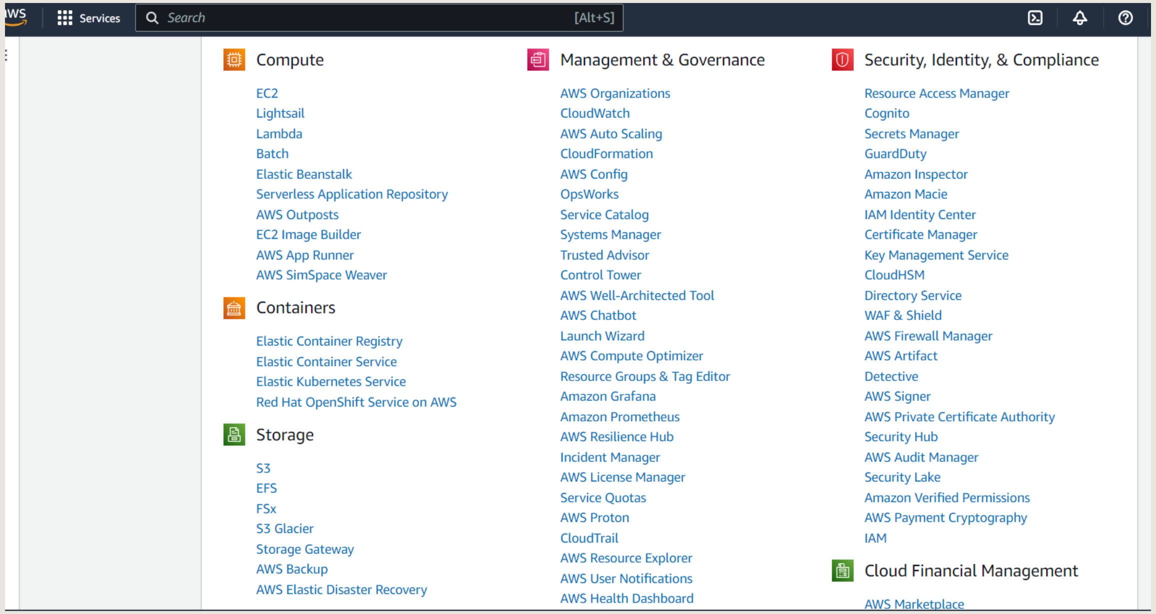
Top features
Buckets Storage Lens dashboards Batch Operations S3 Express One Zone S3 Access Grants

Storage Gateway ☆
Hybrid Storage Integration

Subnets
55 in 17 regions

# AWS SERVICES

- https://us-east-1.console.aws.amazon.com/console/services?region=us-east-1



Compute
- EC2
- Lightsail
- Lambda
- Batch
- Elastic Beanstalk
- Serverless Application Repository
- AWS Outposts
- EC2 Image Builder
- AWS App Runner
- AWS SimSpace Weaver

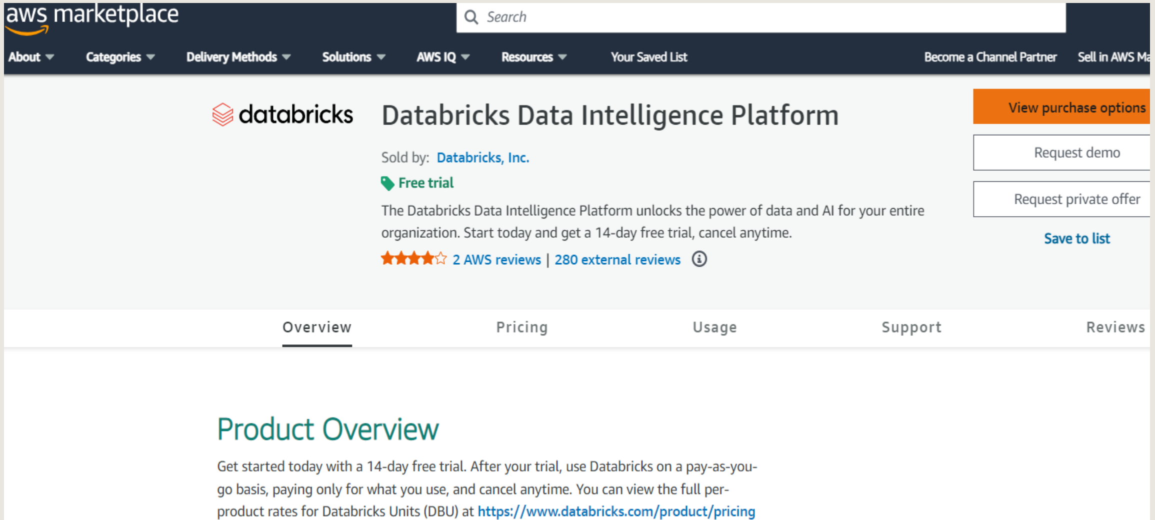
Containers
- Elastic Container Registry
- Elastic Container Service
- Elastic Kubernetes Service
- Red Hat OpenShift Service on AWS

Storage
- S3
- EFS
- FSx
- S3 Glacier
- Storage Gateway
- AWS Backup
- AWS Elastic Disaster Recovery

Management & Governance
- AWS Organizations
- CloudWatch
- AWS Auto Scaling
- CloudFormation
- AWS Config
- OpsWorks
- Service Catalog
- Systems Manager
- Trusted Advisor
- Control Tower
- AWS Well-Architected Tool
- AWS Chatbot
- Launch Wizard
- AWS Compute Optimizer
- Resource Groups & Tag Editor
- Amazon Grafana
- Amazon Prometheus
- AWS Resilience Hub
- Incident Manager
- AWS License Manager
- Service Quotas
- AWS Proton
- CloudTrail
- AWS Resource Explorer
- AWS User Notifications
- AWS Health Dashboard

Security, Identity, & Compliance
- Resource Access Manager
- Cognito
- Secrets Manager
- GuardDuty
- Amazon Inspector
- Amazon Macie
- IAM Identity Center
- Certificate Manager
- Key Management Service
- CloudHSM
- Directory Service
- WAF & Shield
- AWS Firewall Manager
- AWS Artifact
- Detective
- AWS Signer
- AWS Private Certificate Authority
- Security Hub
- AWS Audit Manager
- Security Lake
- Amazon Verified Permissions
- AWS Payment Cryptography
- IAM

Cloud Financial Management
- AWS Marketplace



Databricks Data Intelligence Platform

Sold by: Databricks, Inc.

Free trial

The Databricks Data Intelligence Platform unlocks the power of data and AI for your entire organization. Start today and get a 14-day free trial, cancel anytime.

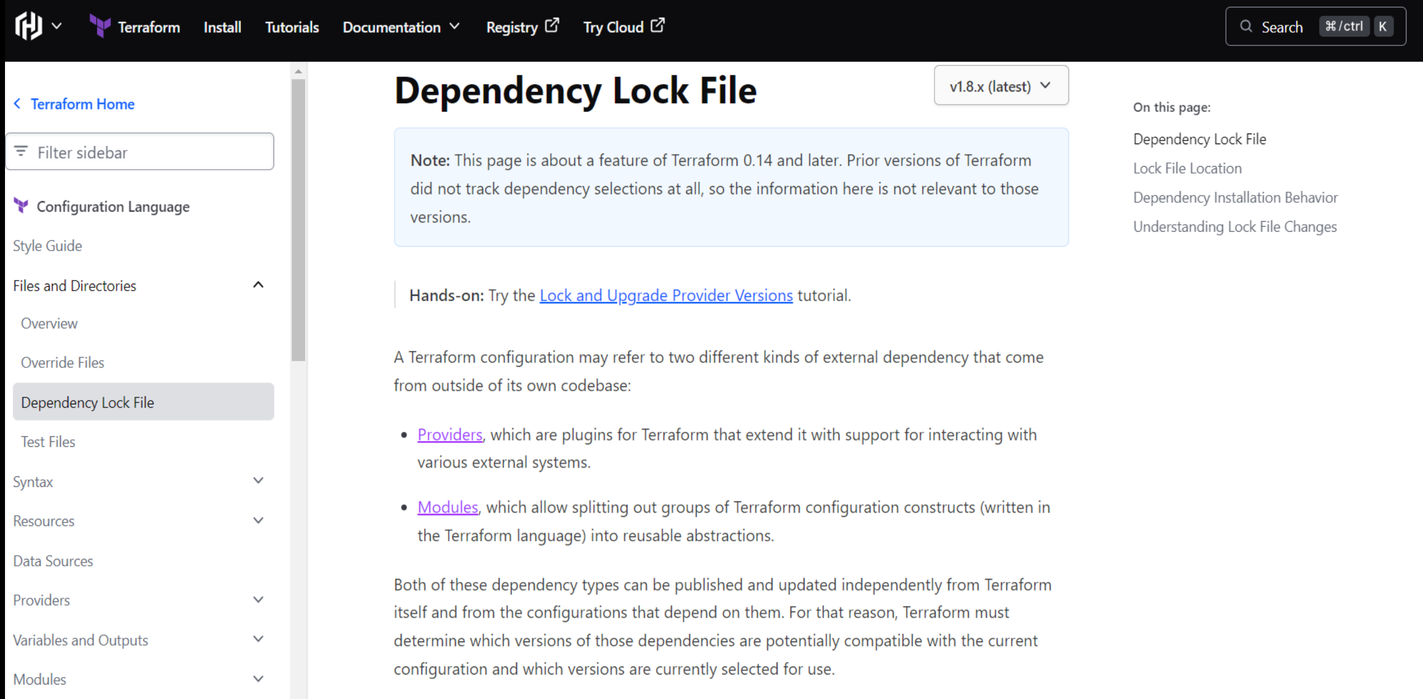
★★★★☆ 2 AWS reviews | 280 external reviews

Product Overview

Get started today with a 14-day free trial. After your trial, use Databricks on a pay-as-you-go basis, paying only for what you use, and cancel anytime. You can view the full per-product rates for Databricks Units (DBU) at https://www.databricks.com/product/pricing

# TERRAFORM FILES FOR STRUCTURE

- **provider.tf** – containing the terraform block, s3 backend definition, provider configurations, and aliases.
- **main.tf** – containing the resource blocks which define the resources to be created in the target cloud platform.
- **variables.tf** – containing the variable declarations used in the resource blocks.
- **output.tf** – containing the output that needs to be generated on successful completion of "apply" operation.
- *.tfvars – containing the environment-specific default values of variables.
- More information (https://spacelift.io/blog/terraform-files)



## Dependency Lock File

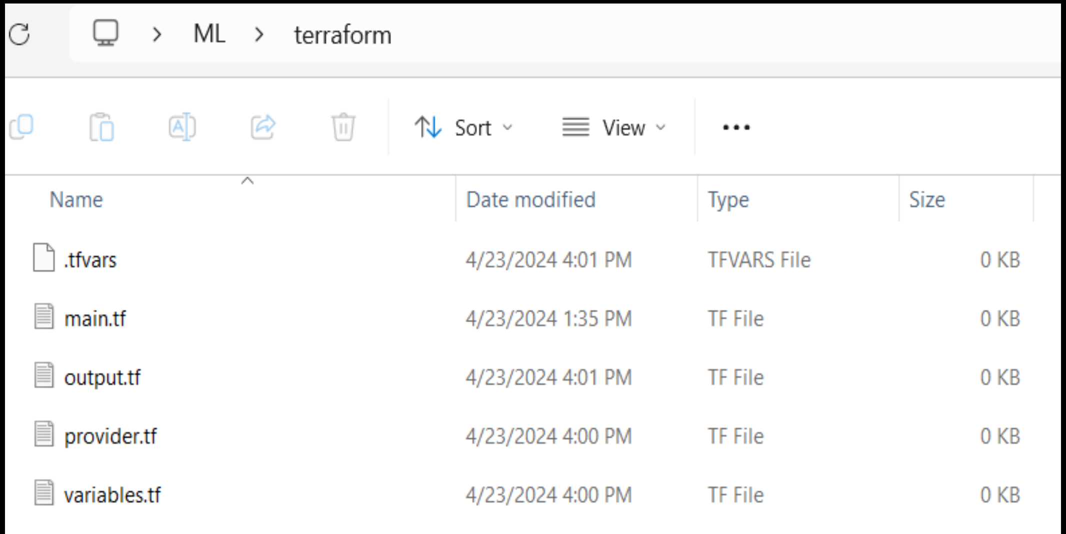
**Note:** This page is about a feature of Terraform 0.14 and later. Prior versions of Terraform did not track dependency selections at all, so the information here is not relevant to those versions.

**Hands-on:** Try the Lock and Upgrade Provider Versions tutorial.

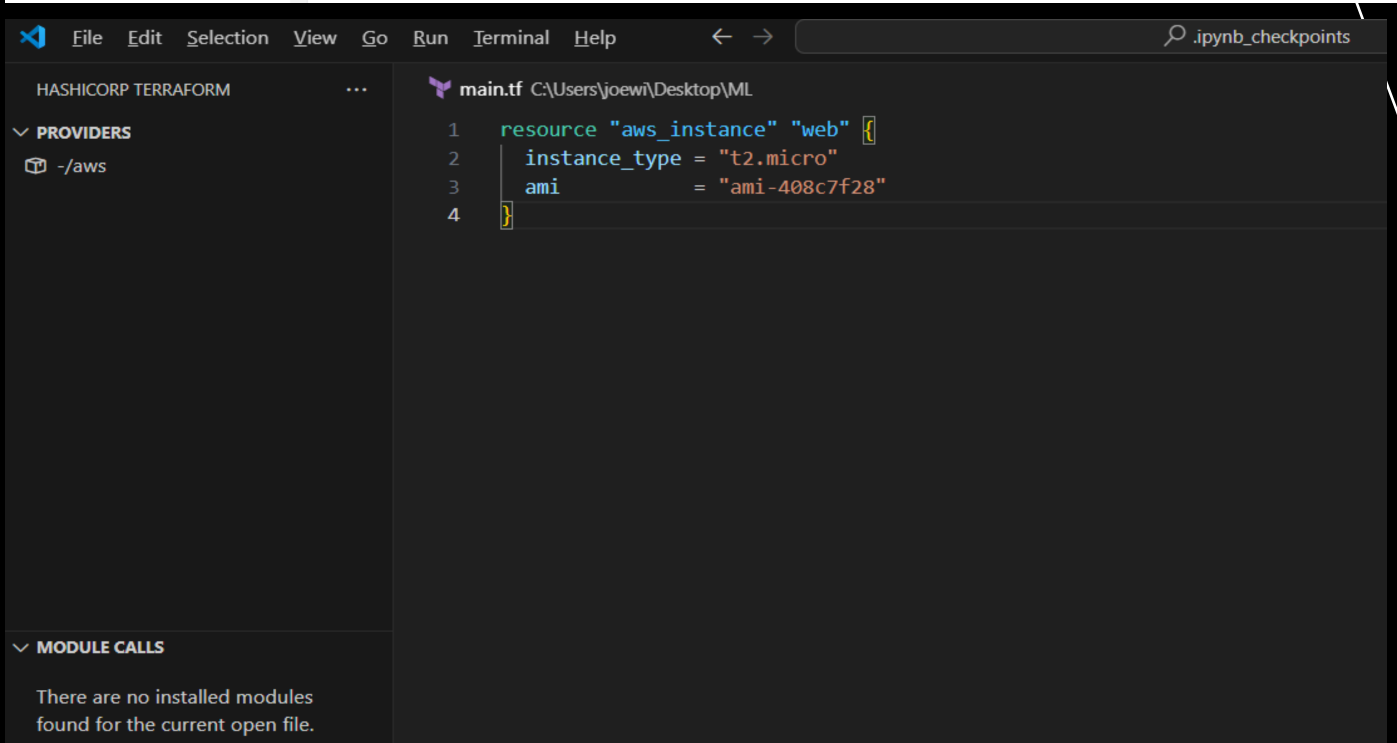
A Terraform configuration may refer to two different kinds of external dependency that come from outside of its own codebase:

- Providers, which are plugins for Terraform that extend it with support for interacting with various external systems.
- Modules, which allow splitting out groups of Terraform configuration constructs (written in the Terraform language) into reusable abstractions.

Both of these dependency types can be published and updated independently from Terraform itself and from the configurations that depend on them. For that reason, Terraform must determine which versions of those dependencies are potentially compatible with the current configuration and which versions are currently selected for use.



| Name | Date modified | Type | Size |
|---|---|---|---|
| .tfvars | 4/23/2024 4:01 PM | TFVARS File | 0 KB |
| main.tf | 4/23/2024 1:35 PM | TF File | 0 KB |
| output.tf | 4/23/2024 4:01 PM | TF File | 0 KB |
| provider.tf | 4/23/2024 4:00 PM | TF File | 0 KB |
| variables.tf | 4/23/2024 4:00 PM | TF File | 0 KB |



There are no installed modules found for the current open file.

main.tf C:\Users\joewi\Desktop\ML

```
resource "aws_instance" "web" {
  instance_type = "t2.micro"
  ami           = "ami-408c7f28"
}
```

At that point in the process, we could:

Have a code series provided by https://huggingface.co/ as Transformers and be able to define sentiments according to text interactions (comments) on Twitter (X) in response to a post made by our company, utilizing the API of X's developer space.

We should aim to minimize spending on AWS services for the "run" process of the container by using AWS ECS Fargate and S3 storage for gathering returned data from the frontend application.

We would also manage all this infrastructure "easily" with Terraform in VSCode, ensuring that elements are correctly assigned according to different files and their purposes.

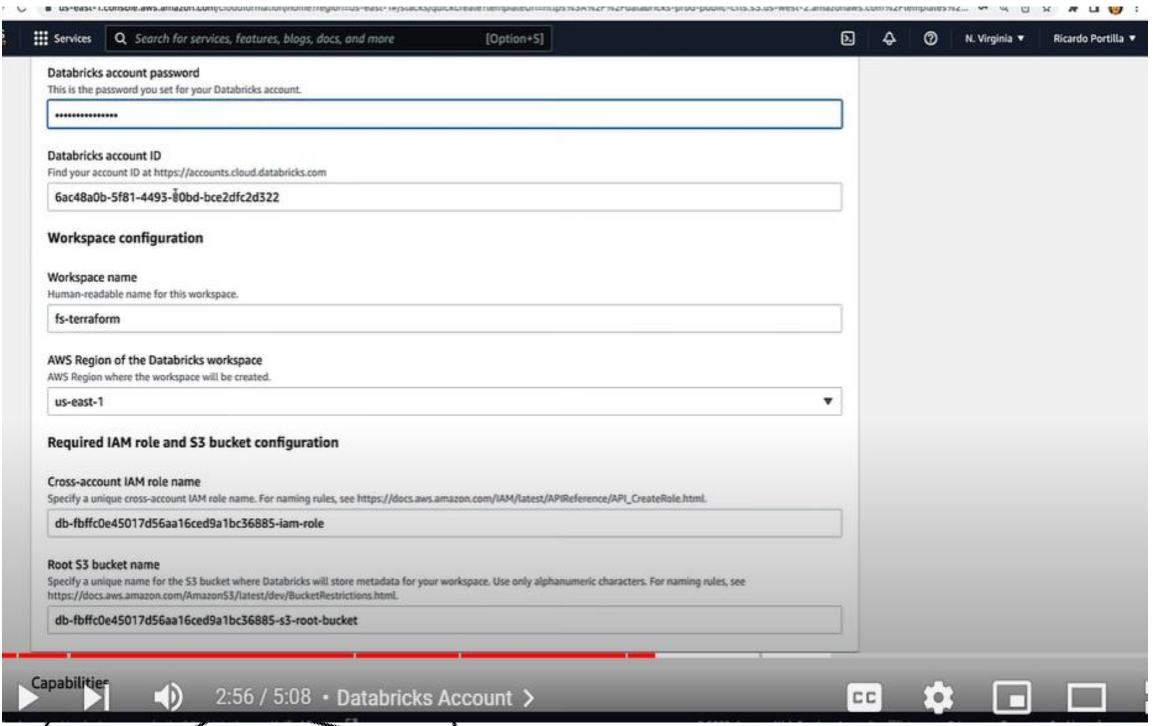
The model is capable of providing our company with a clear report of the situation. We can either store all of them in the data storage or sort and list them, allowing us to review past events in order to avoid making bad choices again.

**"Those who control the present, control the past and those who control the past control the future.”**
**― George Orwell, 1984**

## DATABRICKS ETL PIPELINES

[(*) https://youtu.be/gEDS5DOUgY8?feature=shared](https://youtu.be/gEDS5DOUgY8?feature=shared)

(*)



Databricks account password
This is the password you set for your Databricks account.
•••••••••••••••

Databricks account ID
Find your account ID at https://accounts.cloud.databricks.com
6ac48a0b-5f81-4493-80bd-bce2dfc2d322

**Workspace configuration**

Workspace name
Human-readable name for this workspace.
fs-terraform

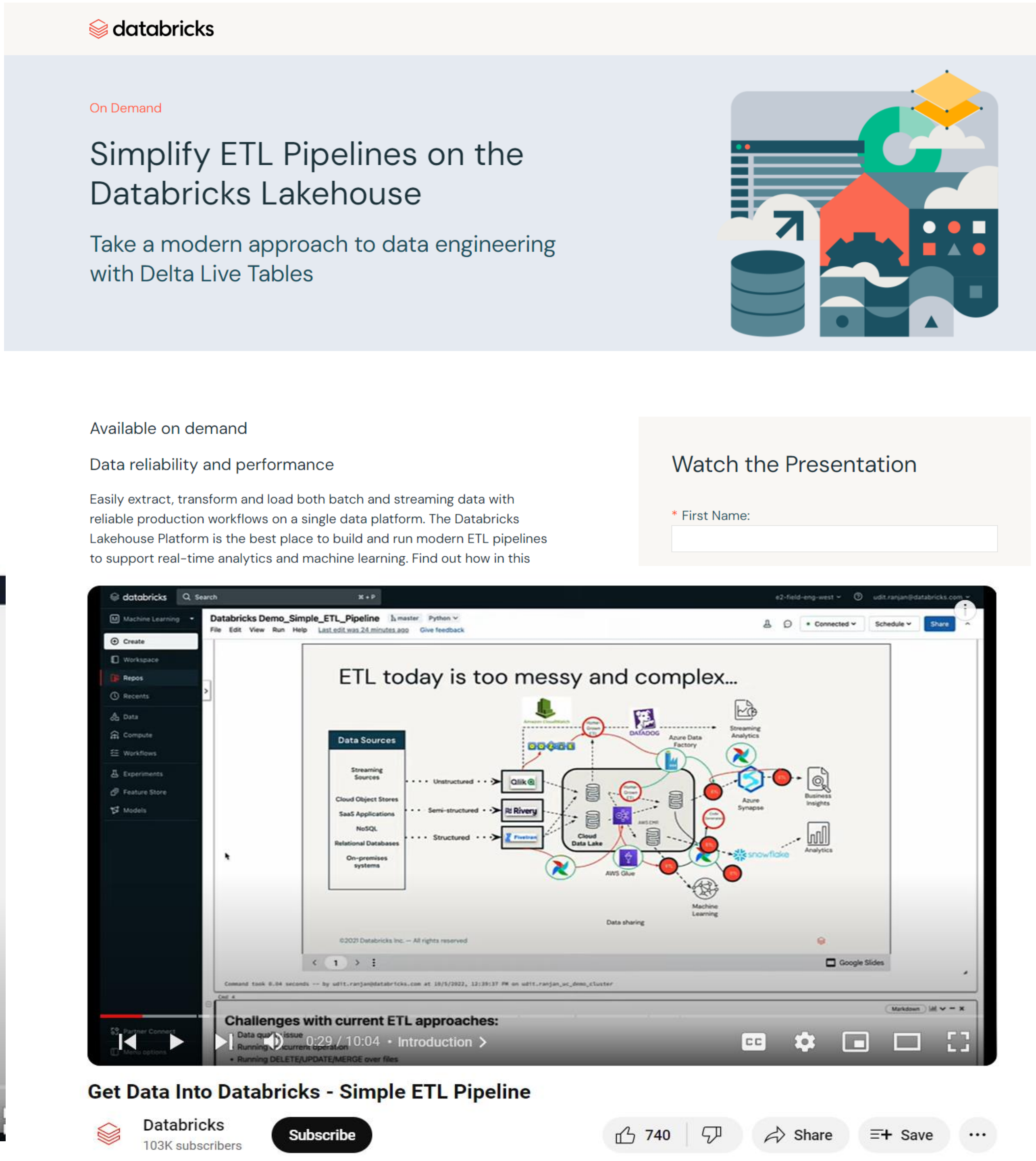
AWS Region of the Databricks workspace
AWS Region where the workspace will be created.
us-east-1

**Required IAM role and S3 bucket configuration**

Cross-account IAM role name
Specify a unique cross-account IAM role name. For naming rules, see https://docs.aws.amazon.com/IAM/latest/APIReference/API_CreateRole.html.
db-fbffc0e45017d56aa16ced9a1bc36885-iam-role

Root S3 bucket name
Specify a unique name for the S3 bucket where Databricks will store metadata for your workspace. Use only alphanumeric characters. For naming rules, see https://docs.aws.amazon.com/AmazonS3/latest/dev/BucketRestrictions.html.
db-fbffc0e45017d56aa16ced9a1bc36885-s3-root-bucket

Services
Search for services, features, blogs, docs, and more
[Option+S]
N. Virginia ▾
Ricardo Portilla ▾
Capabilities
2:56 / 5:08 • Databricks Account ›



databricks

On Demand

Simplify ETL Pipelines on the Databricks Lakehouse

Take a modern approach to data engineering with Delta Live Tables

Available on demand

Data reliability and performance

Easily extract, transform and load both batch and streaming data with reliable production workflows on a single data platform. The Databricks Lakehouse Platform is the best place to build and run modern ETL pipelines to support real-time analytics and machine learning. Find out how in this

Watch the Presentation

* First Name:

Databricks Demo_Simple_ETL_Pipeline

ETL today is too messy and complex...

©2021 Databricks Inc. — All rights reserved

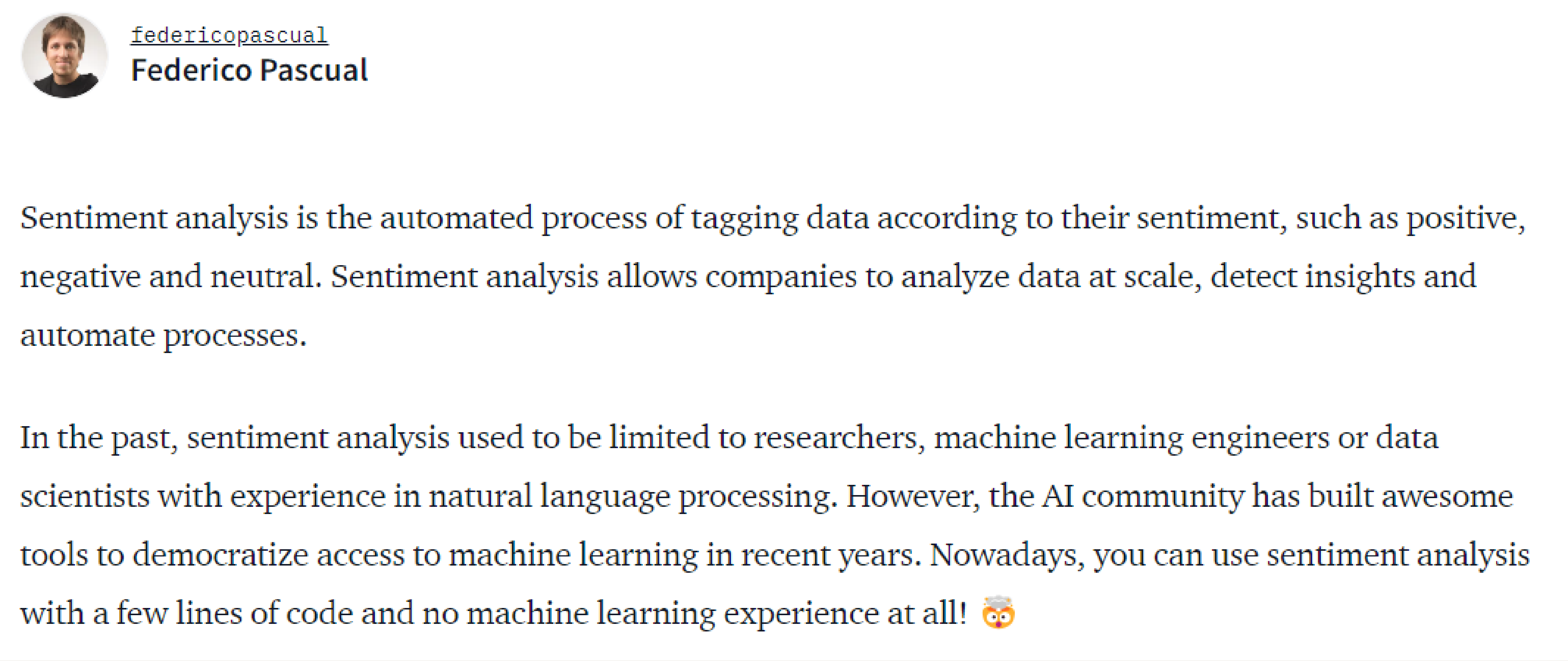
Command took 0.04 seconds -- by udit.ranjan@databricks.com at 10/5/2022, 12:39:17 PM on udit.ranjan_uc_demo_cluster

**Challenges with current ETL approaches:**
- Data quality issue
- Running concurrent operation
- Running DELETE/UPDATE/MERGE over files

Get Data Into Databricks - Simple ETL Pipeline

Databricks
103K subscribers

# GROWTH STRATEGY



federicopascual

Federico Pascual

Sentiment analysis is the automated process of tagging data according to their sentiment, such as positive, negative and neutral. Sentiment analysis allows companies to analyze data at scale, detect insights and automate processes.

In the past, sentiment analysis used to be limited to researchers, machine learning engineers or data scientists with experience in natural language processing. However, the AI community has built awesome tools to democratize access to machine learning in recent years. Nowadays, you can use sentiment analysis with a few lines of code and no machine learning experience at all! 🤯

https://huggingface.co/blog/sentiment-analysis-python

JOE WILD