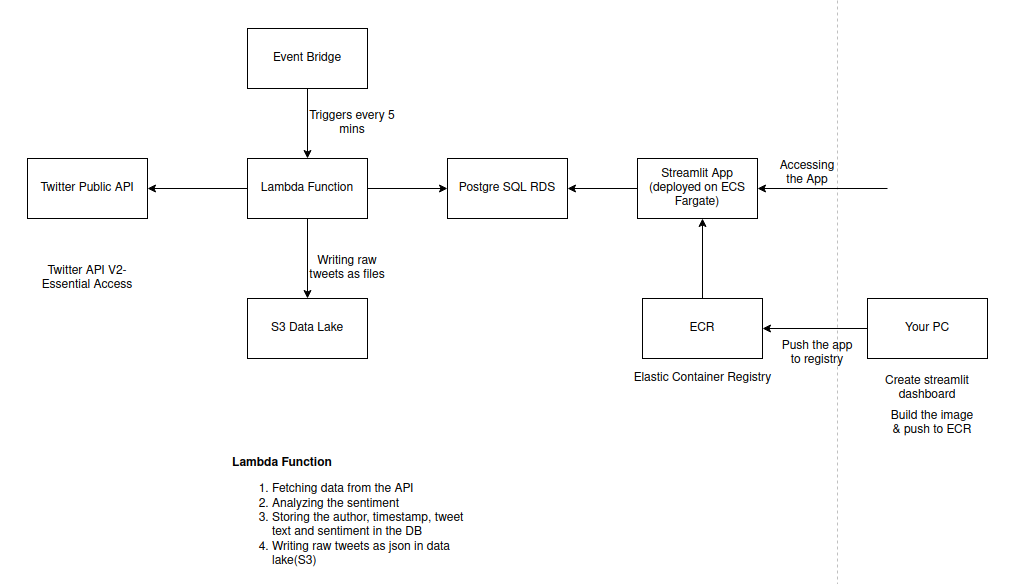
**ML & Containerization on AWS**

Tracking the tweets done by news agency ‘Reuters’ and analyzing its sentiment.

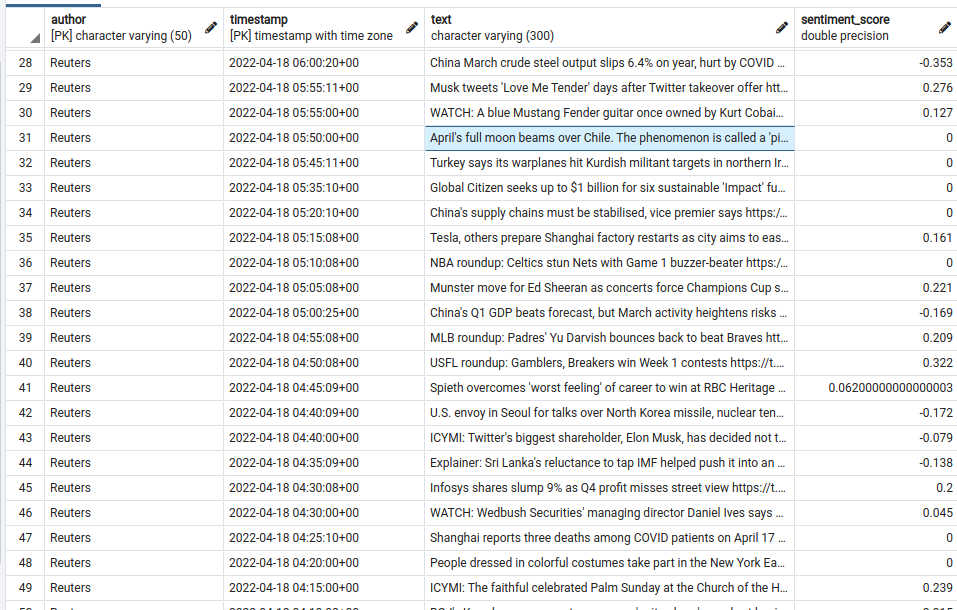


Using Twitter API V2 with Essential Access(most basic and free). First need to make Developer’s account in Twitter, can then generate unique Bearer Token using which API can be accessed.

AWS Lambda Function is performing the following tasks:

1. Fetching data from the API
2. Analyzing the sentiment
3. Storing the author, timestamp, tweet text and sentiment in the DB
4. Writing raw tweets as json in data lake(S3)

Using Postgres DB as the data warehouse as shown below:

Not all packages/libraries are available for use in the Lambda function. Hence, some layers have to be added to the function as we do by install packages in the environement.

Some layers can be found over the internet while rest can be created. Resources are mentioned below:

**Lambda klayers**

Layers to be added in Layers..ARN

Have to add layers in ARN for the packages being used in lambda

**Additional Resources:**

K-Layers documentation: https://github.com/keithrozario/Klayers

K-Layers commands you need to install: (for Twitter project in the course)

https://api.klayers.cloud/api/v1/layers/latest/us-east-1/pytz https://api.klayers.cloud/api/v1/layers/latest/us-east-1/nltk https://api.klayers.cloud/api/v1/layers/latest/us-east-1/pandas More info about the K-Layers in the GitHub readme: https://github.com/team-data-science/ML-on-AWS-1

**Adding layers**

https://medium.com/swlh/how-to-add-python-pandas-layer-to-aws-lambda-bab5ea7ced4f

Additional Resources:

**Makefile for the Layers:** https://github.com/team-data-science/ML-on-AWS-1/blob/main/src/lambda\_/make\_layer.sh

**Make custom layer for packages:**

Make\_layer.sh file present in \_lambda folder in twitter project

https://aws.amazon.com/premiumsupport/knowledge-center/lambda-layer-simulated-docker/

**Poetry:**

Using the Poetry dependencies. Poetry documentation: https://python-poetry.org/docs/basic-usage/

Poetry .toml file for the project: https://github.com/team-data-science/ML-on-AWS-1/blob/main/pyproject.toml

**Streamlit app deployment: on AWS ECS Fargate**

First create the image, push to AWS ECR(Elastic Container Registry) then deploy on Fargate.

With all files present

Install poetry

Create poetry environment and activate

Now add packages to poetry env: poetry add pandas

U will see 2 files getting created...one is .toml which will show the installed packages

Other is .lock

**AWS ECR**

Create ECR repo

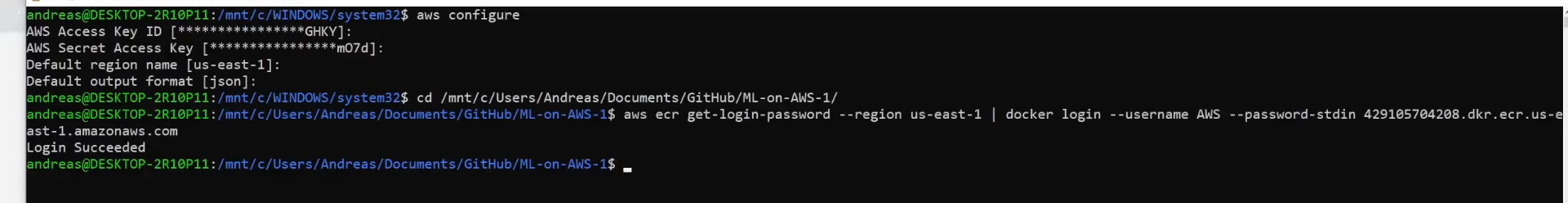
Make User group and add permissions like S3FullAccess, etc etc as per the requirement

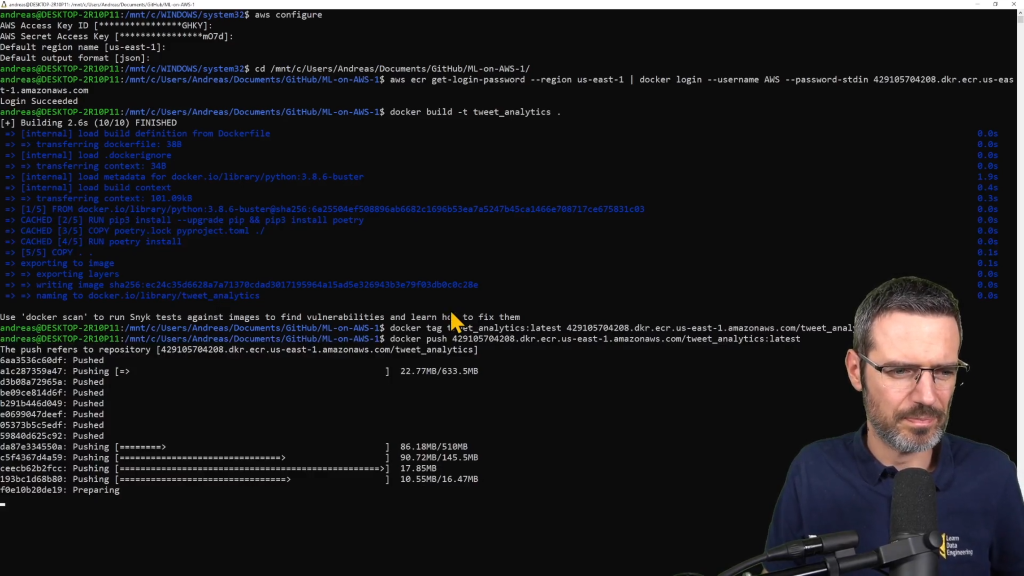
Make user in that user group...save secret key(visible once), access key and download the secret key csv file

**AWS CLI for pushing image**

Go to terminal, download and install AWS CLI(google the instructions)

Then configure AWS CLI ..now go to right directory where project is kept and Configure AWS CLI and login using ECR push repo instructions present in ECR repo. If issue in login(close and repeat)





Follow Docker push commands from ECR repo

**ECS Cluster**

Go to ECS, create cluster(can choose networking only...other are reqd. when u want cluster to run always)

**Add ECS Task**

Add role to this – go to IAM, choose service: ECS

Then Elastic Container Service Task

Attach policy- ECRAppRunnerServicepolicyForECRAccess, ECRFulllAccess.ClouddWatchFull Access

**Go to ECS**

Create Task Definition

Then Run Task...see while selecting Security group...select Edit...existing....default security group

**Now it will not run**

Because the default Security group doesn't have the inbound rule for the IP address of our cluster

Add Custom TCP, Anywhere, 8501 port to the Inbound Rules of the SG

Now use the public IP:8501 port

Streamlit app runs on 8501 port.