





TWITTER **AIRFLOW PROJECT**

by Arjunan K



Introducing **ABOUT ME**

Mathematics Graduate from a non CS background. Passion for problem solving brought me here. Willing to learn and work.



Arjunan K
Data Guy



LIST OF CONTENTS

In this presentation I will be going through all the steps in building a Twitter Airflow Data Pipeline from Scratch.

01

OVERVIEW

03

APACHE AIRFLOW

05

AWS S3

02

TWITTER API

04

AWS EC2

06

ENDNOTE

TWITTER DATA PIPELINE



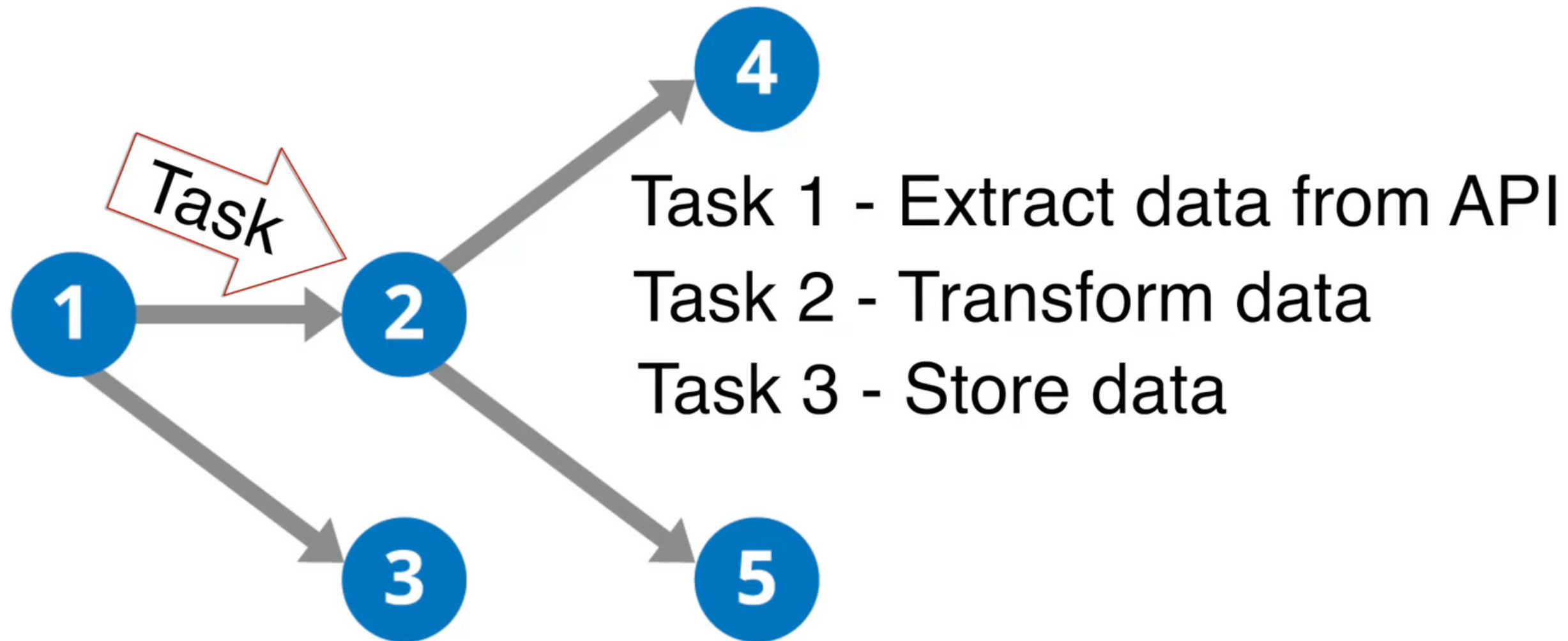
TWITTER API

- First, we need to extract data from a Twitter user.
- Official Twitter API to fetch the data.
- Tweepy library in Python to access the API.
- Pandas to clean the data and store it as a CSV file.
- For the code visit the Github below.
- github.com/arjunan-k/Twitter_Pipeline/blob/main/twitter_api.py

APACHE AIRFLOW

- Apache Airflow is an Open Source workflow platform.
- It is used to build data pipelines.
- Workflow Orchestration Tool
- Build, Schedule and Monitor data pipelines.
- Workflow - Sequence of every task.
- Airflow it is defined as Cyclic graph.
- In this we can set several Tasks.

DAG - DIRECTED ACYCLIC GRAPH



OPERATORS

- In order to build the task, we can use Operator.
- It is a predefined template to build various tasks.
- There are various Operators available.
- For our simple program, we can use PythonOperator
- For code go to the following URL.
- github.com/arjunan-k/Twitter_Pipeline/blob/main/twitter_dag.py

SETTING AWS EC2

- Launch an EC2 instance, in AWS
- Select OS as Ubuntu
- Select t3.Medium as Instance Type for Airflow
- Set a key pair for login
- Allow network settings for HTTP and HTTPS
- Others you can set as default.

CONNECT AIRFLOW ON EC2

- We will connect EC2 to localhost using SSH
- SSH (SECURE SHELL)
- Then we need to run some commands given below.
- These commands set on the Ubuntu Server.
- `sudo apt-get update`
- `sudo apt install python3-pip`
- `sudo pip install apache-airflow`
- `sudo pip install pandas`
- `sudo pip install s3fs`
- `sudo pip install tweepy`

RUNNING THE AIRFLOW

- We can run Airflow by using the following command.
- airflow standalone
- Now note the Login Credential.
- Go to EC2
- Then edit Inbound rules and set All Traffic in security.
- Copy the public DNS from EC2.
- Append :8080 port to login to the Airflow Server.

CREATING A S3 IN AWS

- Go to S3 in AWS
- Click on Create Bucket
- Just give it a globally unique name
- Set others as default
- Hit Create bucket.
- Change the location of the stored CSV in twitter_etl.py
- `df.to_csv("s3://Type your Bucket Name/tweets.csv")`
- github.com/arjunan-k/Twitter_Pipeline/blob/main/twitter_etl.py

IMPORTING CODE ON AIRFLOW

- Connect to EC2 using SSH
- Run the following Commands
- `cd airflow`
- `sudo nano airflow.cfg`
- change the dags_folder as /twitter_dag
- `mkdir twitter_dag`
- `cd twitter_dag`
- Copy the codes to the server
- `sudo nano twitter_etl.py`
- `sudo nano twitter_dag.py`

NOW RUNNING THE AIRFLOW

- Run the Airflow
- If it is a success we can see the tweets.csv in our S3 bucket.
- Check Logs to see errors
- If access is denied.
- Click on EC2
- Click on Actions > Security > Modify IAM role
- IAM (Identity and Access Management)
- Click create role
- Give S3 and EC2 full access in the IAM role.
- Run again to see the result.



**THANKS
FOR WATCHING**

