



# NFTcharts

Uncovering Trade Activity and Hot Trends

# Table of Contents



## Project outline

What this project is  
about



## Implementation

Result, approach,  
current product



## Tech Dive

Deep dive into tools  
used



## Summary

Limits, learnings,  
what's next

# Project Outline

# What is an NFT?

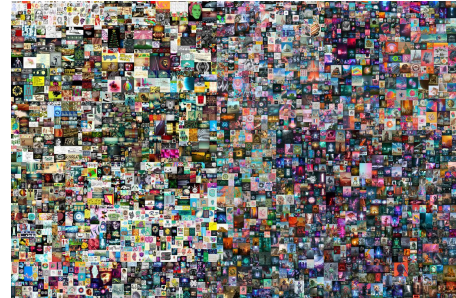


A non-fungible token is a unique and non-interchangeable unit of data stored on a blockchain, a form of digital ledger.

Can be in the form of photos, videos, and audio, etc.

# Fun Facts

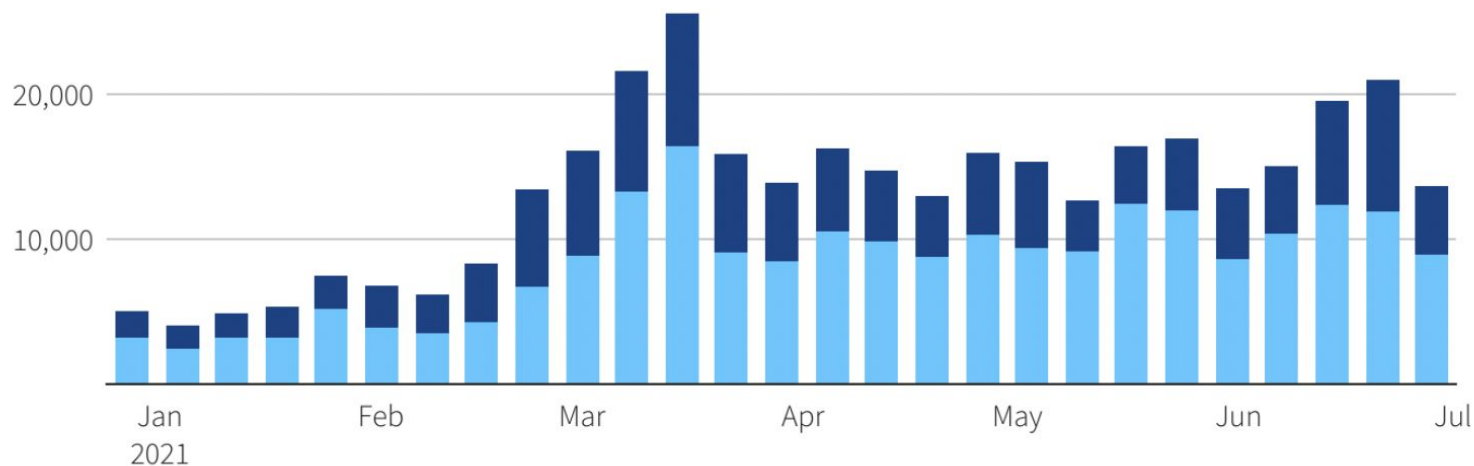
- The most expensive NFT ever sold was “The First 5000 Days” by Beeple for \$69.3 million – making it the most expensive virtual item ever traded on any platform in history.
- There are \$10-\$20 million worth of NFT sold in the blockchain every week.



# Weekly NFT buyers - NonFungible.com

Number of buyers on non-fungible tokens on the ethereum blockchain per week

● primary market ● secondary market



Note: Data only shows sales on the ethereum blockchain, which is used for the majority of NFT sales. Data does not include sales which took place "off-chain".

Source: NonFungible.com

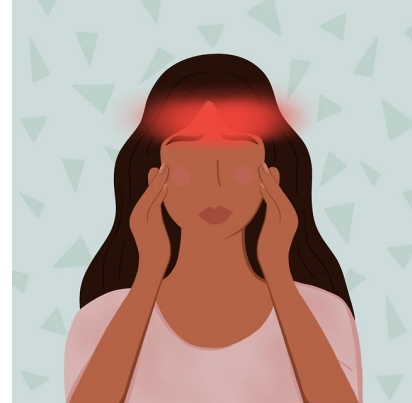
# Project Goals and Objectives

- **Uncover** the latest and hottest NFT collections
- **Provide insights** into which NFT was recently traded and its activity history



# Problem Space

- **Overwhelming** number of NFT projects in the market
- **Unable to keep track** of which collections are gaining popularity (aka “emerging”)
- **Unable to monitor** characteristics or trends of recent sales for emerging collections





# My motivation

- This is 100% a passion project
- Passionate about optimising life by generating automatic pull feeds
- Want to monitor NFT activity without spending too much time on it



# Implementation

# Personal Tech Objectives

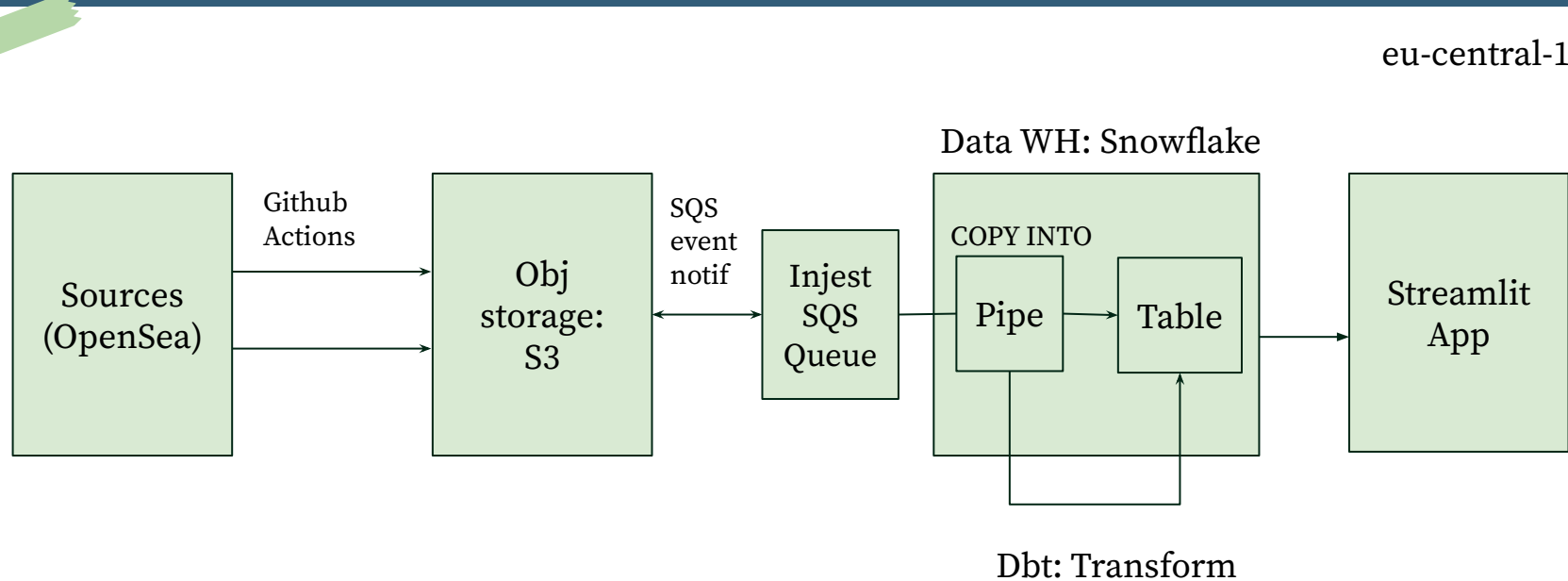
- Try as many tools as possible 🤗
- Determined to try **snowflake** + **ELT**
- Wanted some implementation of automated asynchronous queues
- Hoped for a **stable, visualisation** (grafana ... mode ... Streamlit!)
- Wanted to work with **json files** (not csv)
- Python #1 🥲

**Under Time Pressure:  
Use whatever works,  
cry later**

Don't do this



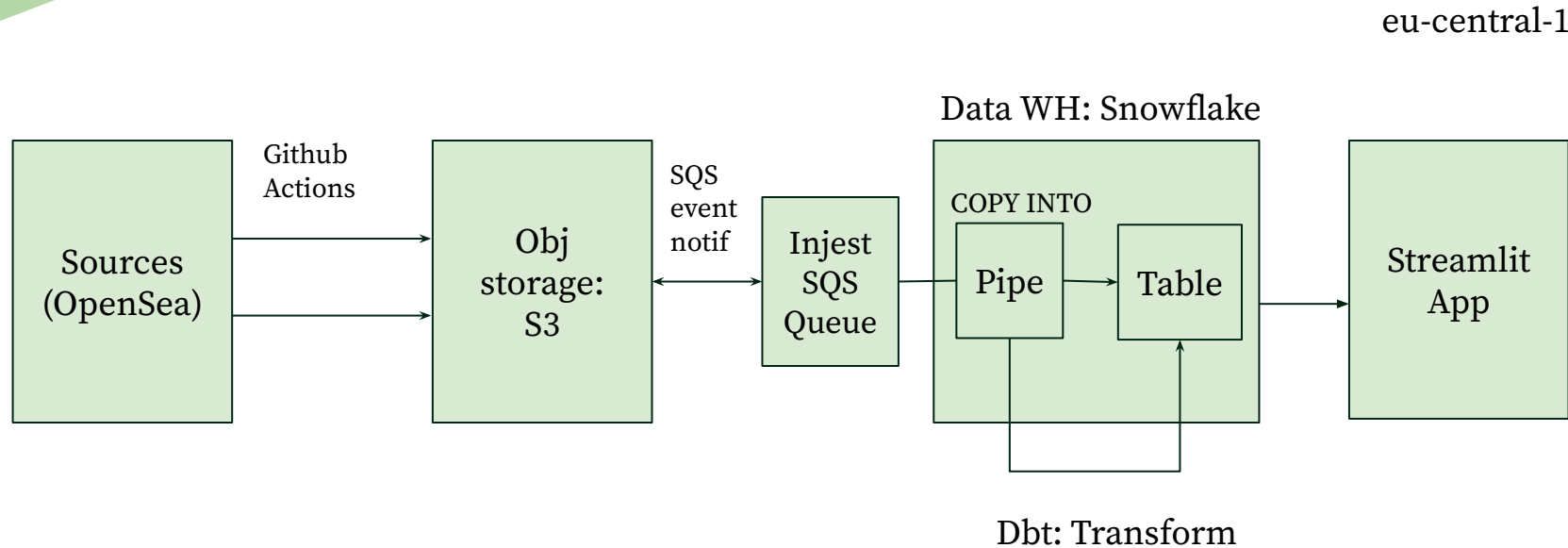
# Architecture Diagram



# Requirements.txt

```
beautifulsoup4==4.10.0  
boto3==1.20.23  
botocore==1.23.23  
cloudscraper==1.2.58  
pandas==1.3.3  
Pillow==8.4.0  
plotly==5.5.0  
requests==2.26.0  
snowflake==0.0.3  
snowflake_connector_python==2.6.2  
streamlit==1.0.0
```

# My Approach: Sequence of Implementation



[https://share.streamlit.i](https://share.streamlit.io/sni-c/final-project)  
[o/sni-c/final-project](https://share.streamlit.io/sni-c/final-project) :(

Currently down because... virtualenv python 3.9  
mac something not compatible with  
snowflake-connector something something



Your app is in the oven



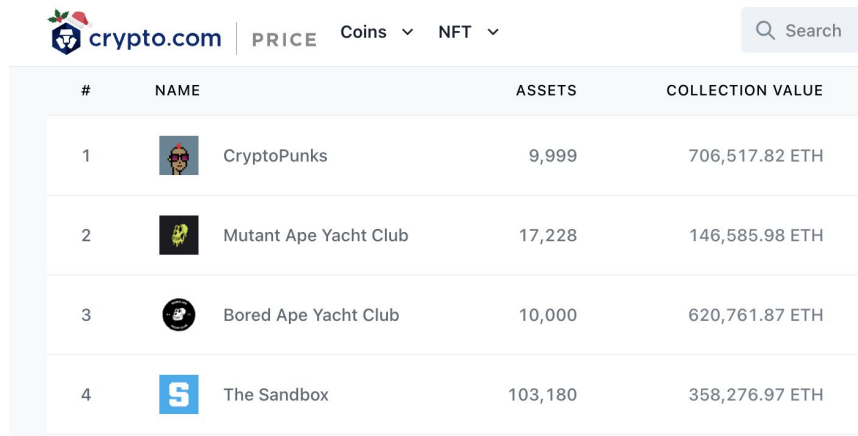
# Tech Dive

# ELT: Scraping + Github Actions





1. Scrape Top 10  
Collections: Crypto.com
2. Pull detailed collection  
data from OpenSea

Collection: 1x per 24 hour cron job

Sales: 1x per 15 min cron job

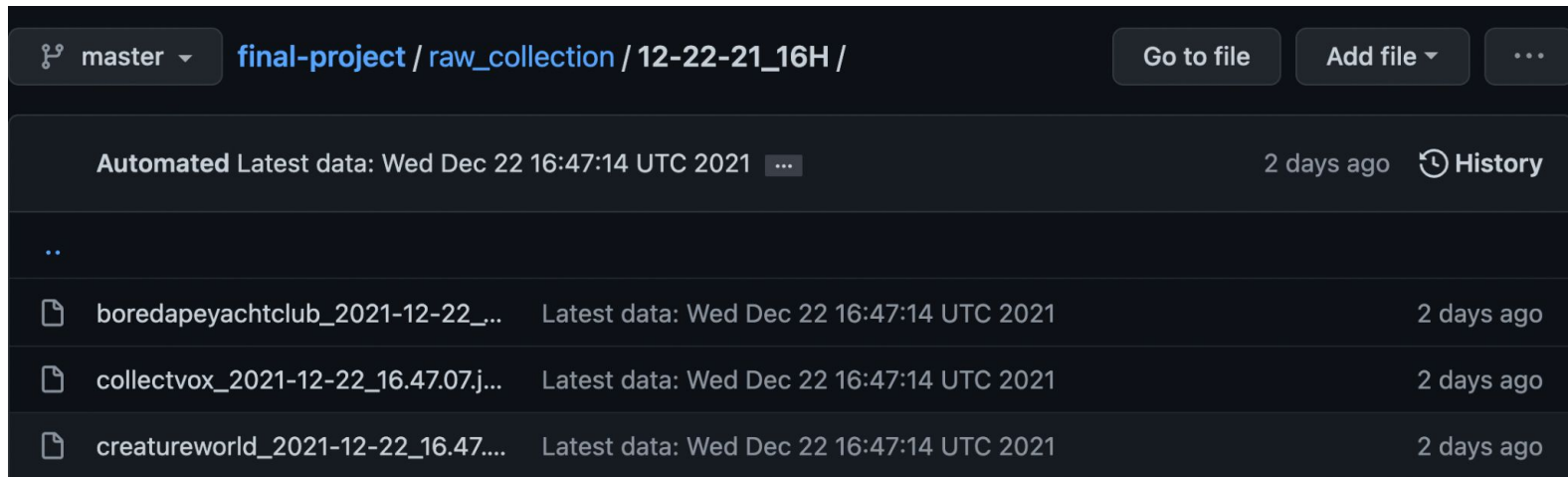


The screenshot shows the crypto.com website with a table of the top 10 NFT collections. The table has columns for rank (#), name, assets, and collection value. The collections listed are CryptoPunks, Mutant Ape Yacht Club, Bored Ape Yacht Club, and The Sandbox.

#	NAME	ASSETS	COLLECTION VALUE
1	 CryptoPunks	9,999	706,517.82 ETH
2	 Mutant Ape Yacht Club	17,228	146,585.98 ETH
3	 Bored Ape Yacht Club	10,000	620,761.87 ETH
4	 The Sandbox	103,180	358,276.97 ETH

# ELT: Scraping + Github Actions

Decisions made    Folder hierarchy: /json → /date and hour → /json file



The screenshot shows a GitHub repository interface for the 'final-project' repository, specifically the 'raw\_collection / 12-22-21\_16H /' directory. The interface includes a navigation bar with a 'master' branch selector, the repository path, and buttons for 'Go to file', 'Add file', and a menu. Below the navigation bar, there is a summary bar indicating 'Automated Latest data: Wed Dec 22 16:47:14 UTC 2021' with a '2 days ago' timestamp and a 'History' link. The main content area displays a list of files, each with a file icon, a truncated filename, a 'Latest data' timestamp, and an age indicator.

File Name	Latest data	Age
..		
boredapeyachtclub_2021-12-22_...	Latest data: Wed Dec 22 16:47:14 UTC 2021	2 days ago
collectvox_2021-12-22_16.47.07.j...	Latest data: Wed Dec 22 16:47:14 UTC 2021	2 days ago
creatureworld_2021-12-22_16.47....	Latest data: Wed Dec 22 16:47:14 UTC 2021	2 days ago

# ELT: Scraping + Github Actions

## Decisions made

- Upload files to s3 first > then upload to Github as backup
- `git pull` before `git add`

36 lines (34 sloc) | 996 Bytes

```
1  name: Scrape Top 10 Collection Stats Daily, upload to S3
2
3  on:
4    workflow_dispatch:
5    schedule:
6      - cron: '0 0 * * *'
7
8  jobs:
9    scheduled:
10     runs-on: ubuntu-latest
11     steps:
12       - name: Check out repo
13         uses: actions/checkout@v2
14       - name: Commit and Push Backups to Repo
15         run: |-
```

# ELT: Snowflake + Snowpipe

Load raw data with duplicates 🤔

Results Data Preview

← Open Histor

✓ Query ID SQL 903ms 9,600 rows

Filter result...



Copy

Columns ▾

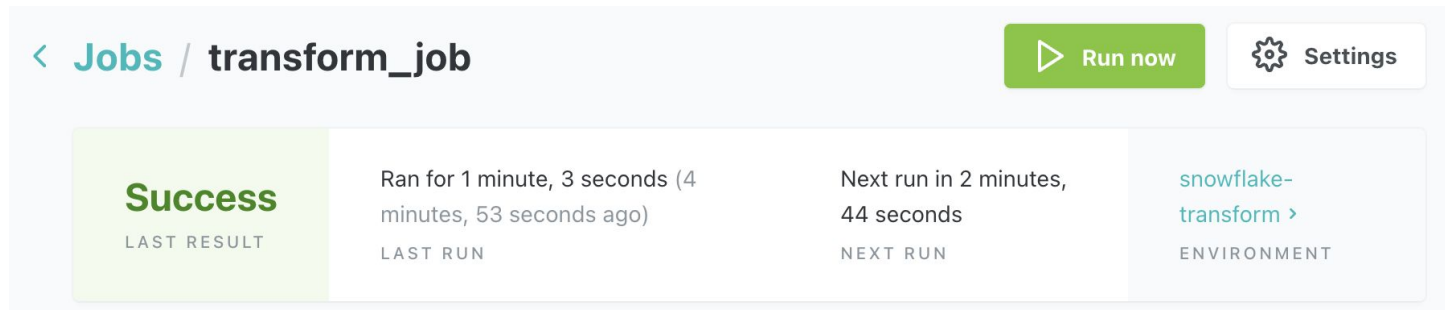
Row	RECENTSALE	CURRENT_TIMESTAMP
1	{ "collectionslug": "collectvox", "create_time": "2021-12-22_17..."	2021-12-22 17:40:40.868 +0000
2	{ "collectionslug": "collectvox", "create_time": "2021-12-22_17..."	2021-12-22 17:40:40.868 +0000

# ELT: Snowflake + Snowpipe

1. Create Snowflake external stage that references your S3 bucket.
2. **An S3 event notification** informs Snowpipe via an SQS queue that new files have been created and ready to load. Snowpipe copies files into a queue.
3. **A Snowflake-provided virtual warehouse** loads data from the queued files into the target table based on parameters defined in the **pipe**.

<https://docs.snowflake.com/en/user-guide/data-load-snowpipe-auto-s3.html>

# ELT: Materialised views with DBT



The screenshot shows the DBT Jobs interface for a job named 'transform\_job'. At the top left, there is a navigation link '< Jobs / transform\_job'. To the right of this are two buttons: a green 'Run now' button with a play icon, and a 'Settings' button with a gear icon. Below these buttons is a table with four columns. The first column, 'LAST RESULT', shows a green 'Success' status. The second column, 'LAST RUN', shows the job ran for 1 minute, 3 seconds (4 minutes, 53 seconds ago). The third column, 'NEXT RUN', shows the next run is in 2 minutes, 44 seconds. The fourth column, 'ENVIRONMENT', shows 'snowflake-transform' with a right arrow.

LAST RESULT	LAST RUN	NEXT RUN	ENVIRONMENT
Success	Ran for 1 minute, 3 seconds (4 minutes, 53 seconds ago)	Next run in 2 minutes, 44 seconds	snowflake-transform >

1 x 10 min cron job

<https://www.startdataengineering.com/post/build-a-simple-data-engineering-platform/>

# ELT: Materialised views with DBT

```
1 version: 2
2 models:
3   - name: stg_collection
4     columns:
5       - name: collectionslug
6         tests:
7           - not_null
8
9   - name: stg_sales
10    columns:
11      - name: collectionslug
12        tests:
13          - not_null
14      - name: id
15        tests:
16          - unique
17          - not_null
```

dbt .yaml file “tests” are helpful

Runs

dbt test

Enter

ready

dbt test

profiles

6s

dbt run

profiles

6s

dbt test

profiles

↓ Logs

Passed

4

0

0

0

0

09:22:30

6 seconds

RUN STATUS

PASS

WARN

FAIL

SKIPPED

QUEUED

START

DURATION

SYSTEM LOGS

> view logs

DETAILS

> not\_null\_stg\_collection\_collectionsug

> not\_null\_stg\_sales\_collectionsug

> not\_null\_stg\_sales\_id

> unique\_stg\_sales\_id



# ELT: Materialised views with DBT

## Materialised view!

Results Data Preview

← 0

✓ [Query ID](#) [SQL](#) 484ms 1,110 rows

Filter result...



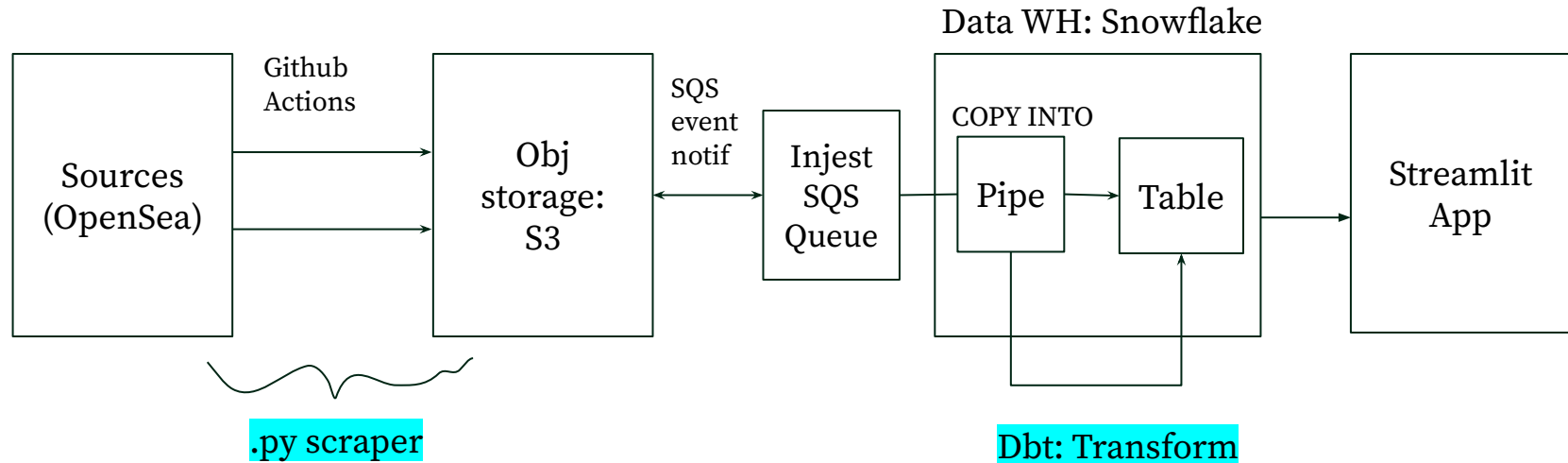
Copy

Colu

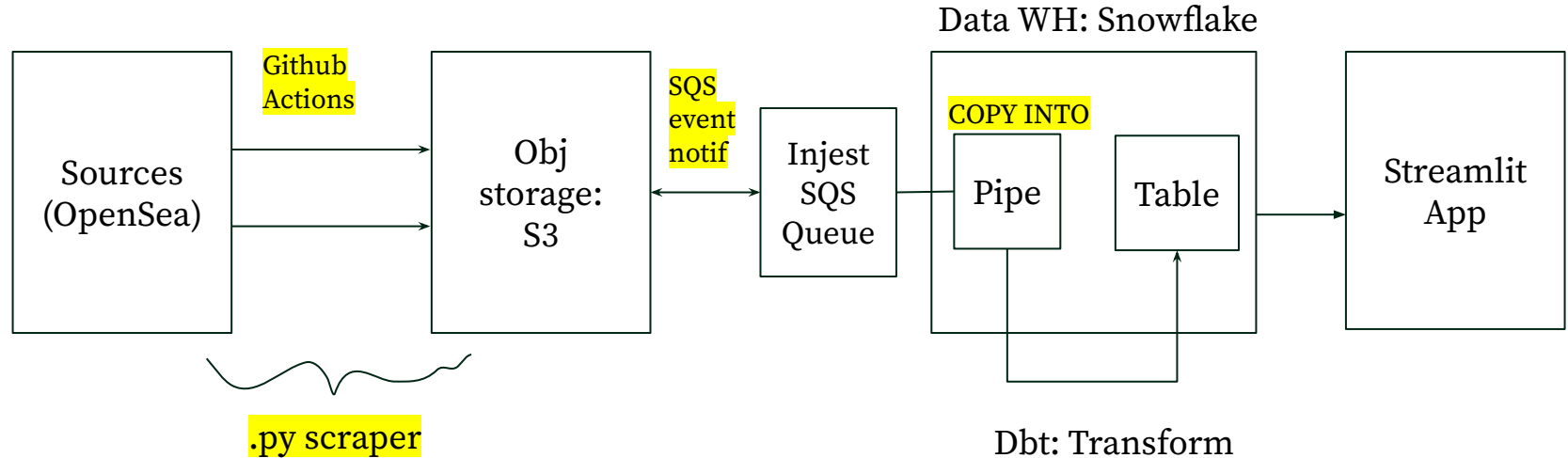
Row	CREATE_TIME	ID	URL	COLLECTIONSL	NAME	EVENT_TIMESTAMP	ETH_PRICE
1	2021-12-24_...	36112740	https://open...	collectvox	Rancher VO...	2021-12-23 ...	0.00012279...
2	2021-12-23_...	162554867	https://open...	punks-comic	PUNKS 2: X ...	2021-12-23 ...	0.00025302...
3	2021-12-22_...	31557874	https://open...	sandbox	LAND (43, 1...	2021-12-22 ...	0.00129351...
4	2021-12-23_...	150714021	https://open...	sandbox	LAND (-1, -1...	2021-12-23 ...	0.00129726...
5	2021-12-24_...	18216908	https://open...	sandbox	LAND (-177	2021-12-24	0.00149617

**Some thoughts that I have not  
found the answer to**

# (1) CD: Modifying code to pull the correct price field



## (2) What if I wanted to pull from more sources?



# Summary

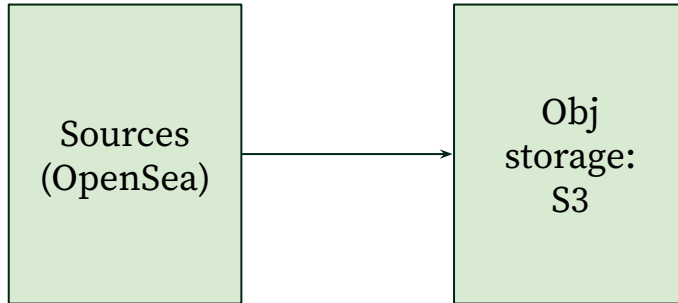
# Handling edge cases

- Make sure scraper always returns a collection, len 10

```
def scraperreturn(results):  
    assert(len(results) == 10)
```

# Handling edge cases

- Check source/destination



# Handling edge cases

- Make sure an image url exists

```
"image_url": "https://www.larvalabs.com/cryptop  
"image_preview_url": "https://www.larvalabs.com  
"image_thumbnail_url": null,  
"image_original_url": null,
```



# Limitations

- Only able to pull up to 50 recent sales per collection  
→ need to get creative around free APIs
- My implementation is heavily reliant on 3rd party tools  
→ how would I do it differently next time?

# Limitations

- 15 min cron job x 10 json files = Already reached 85% of AWS S3 Free Tier Limit → opportunity for compression?

Dear AWS Customer,

Your AWS account 012842063955 has exceeded 85% of the usage limit for one or more AWS Free Tier-eligible services for the month of December.

Product	AWS Free Tier Usage as of 12/23/2021	Usage Limit	AWS Free Tier Usage Limit
AmazonS3	1805.0 Requests	2000.0 Requests	2,000 Put, Copy, Post or List Requests of Amazon S3



# What's next?

Scrape more sources (build core engine)

Better ranking and presentation

Improve materialized view schema

Application beyond NFTs?

# If I had more time, I would...

- Create a setup for collaboration
  - Implement some Dev / Prod separation
- Create an environment for production “real world” development
  - Learn CI / CD better
- Reliability: Concerns with potential data loss
  - count rows in Snowflake vs S3?

# If I had more time, I would...

- Explore each tool in greater depth!



**Thanks!**  
**Any questions?**