
Elasticsearch Essentials:
Data Loading with Python
for Interplanetary Insights
elastic

Code and slides

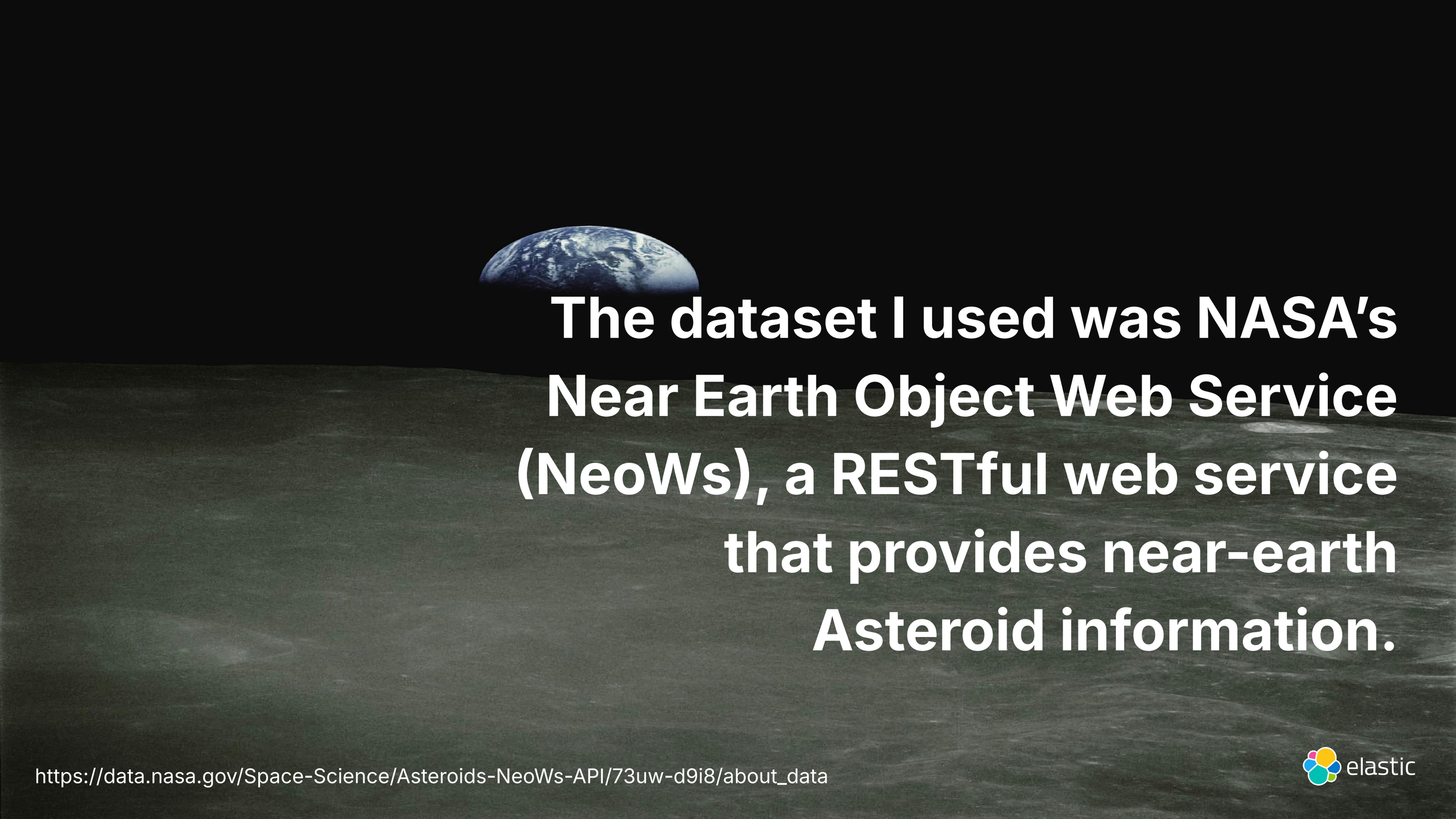
https://github.com/JessicaGarson/Elasticsearch-Essentials-Data-Loading-with-Python-for-Interplanetary-Insights

elastic

This talk will walk you through how to use the Python client as a toolkit 🧰

https://www.elastic.co/guide/en/elasticsearch/client/python-api/current/index.html

elastic



The dataset I used was NASA's Near Earth Object Web Service (NeoWs), a RESTful web service that provides near-earth Asteroid information.

https://data.nasa.gov/Space-Science/Asteroids-NeoWs-API/73uw-d9i8/about_data

elastic

What is an index?

An index inside Elasticsearch is a data structure where you can store your data in documents.

https://www.elastic.co/blog/what-is-an-elasticsearch-index

elastic

What you need to get started

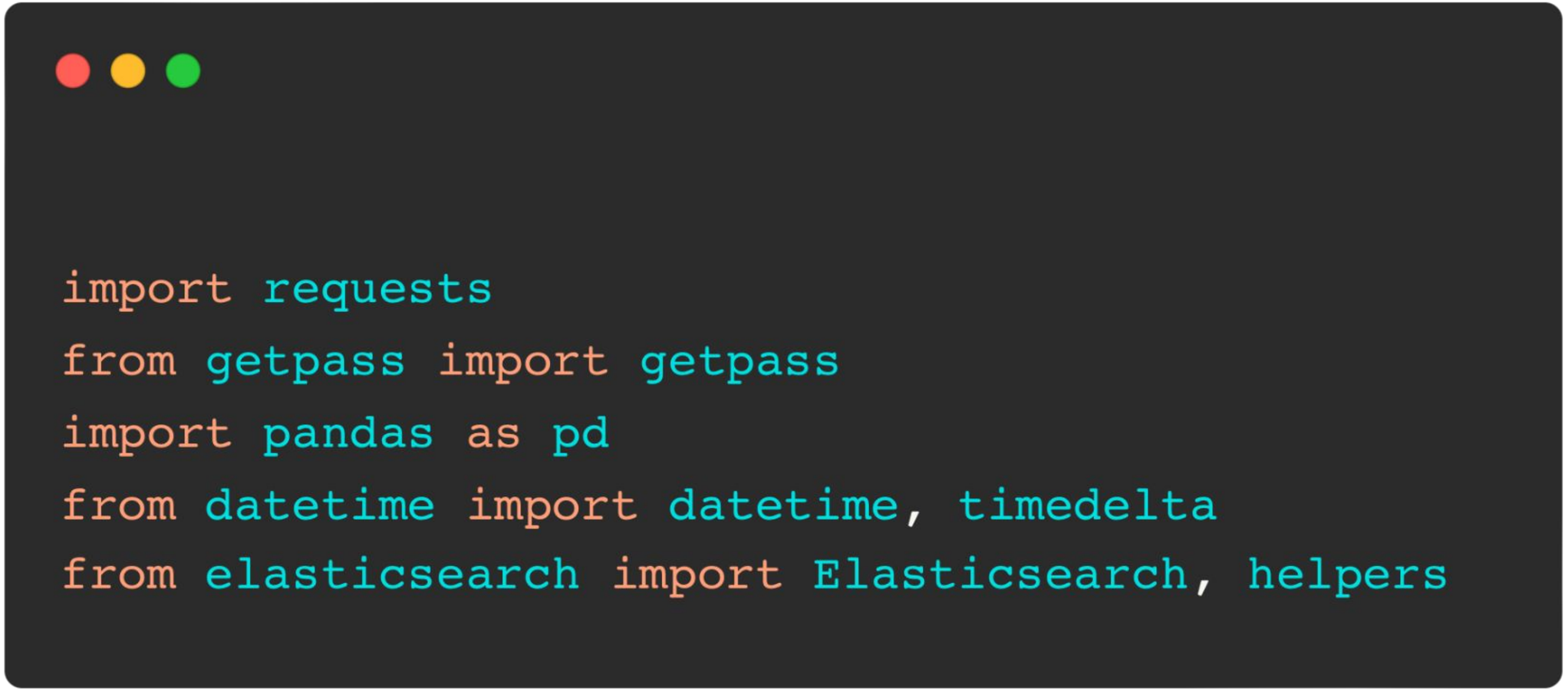
- Python > 3.8
- A NASA API key
- Elasticsearch > 8.x

https://wiki.python.org/moin/BeginnersGuide/Download
https://api.nasa.gov/
https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html

elastic

elastic

Updating our data once



```
import requests
from getpass import getpass
import pandas as pd
from datetime import datetime, timedelta
from elasticsearch import Elasticsearch, helpers
```

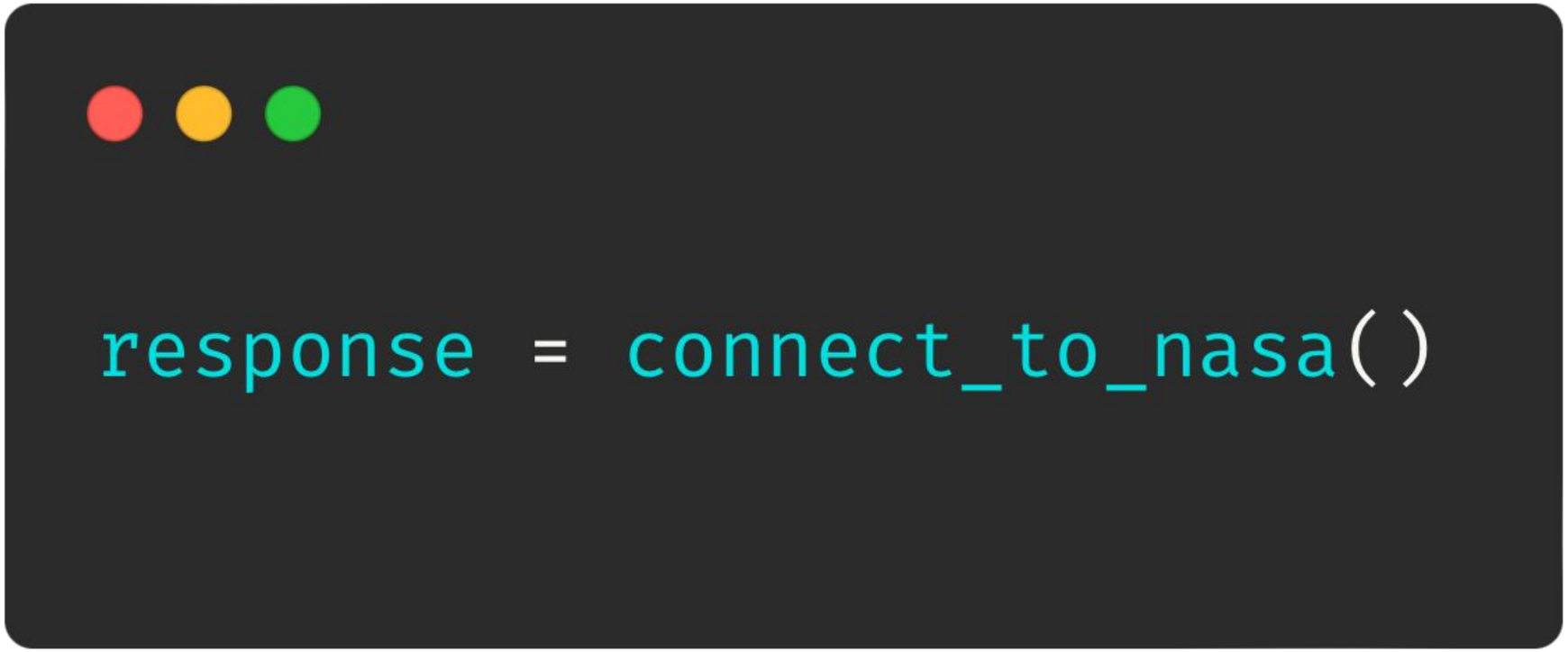
elastic

```
def connect_to_nasa():
    url = "https://api.nasa.gov/neo/rest/v1/feed"
    nasa_api_key = getpass("NASA API Key: ")
    today = datetime.now()
    params = {
        "api_key": nasa_api_key,
        "start_date": today - timedelta(days=7),
        "end_date": datetime.now(),
    }
    return requests.get(url, params).json()
```

elastic

```
def connect_to_nasa():
    url = "https://api.nasa.gov/neo/rest/v1/feed"
    nasa_api_key = getpass("NASA API Key: ")
    today = datetime.now()
    params = {
        "api_key": nasa_api_key,
        "start_date": today - timedelta(days=7),
        "end_date": datetime.now(),
    }
    return requests.get(url, params).json()
```

elastic



elastic

elastic

	id	neo_reference_id	name	nasa_jpl_url	absolute_magnitude_h
0	2434786	2434786	434786 (2006 PW)	https://ssd.jpl.nasa.gov/tools/sbdb_lookup.htm...	18.16
1	3756789	3756789	(2016 PO)	https://ssd.jpl.nasa.gov/tools/sbdb_lookup.htm...	25.10
2	3781330	3781330	(2017 QX35)	https://ssd.jpl.nasa.gov/tools/sbdb_lookup.htm...	22.24
3	54203000	54203000	(2018 UY37)	https://ssd.jpl.nasa.gov/tools/sbdb_lookup.htm...	22.53
4	54317182	54317182	(2022 UG2)	https://ssd.jpl.nasa.gov/tools/sbdb_lookup.htm...	27.11

```python
def connect_to_elastic():
    elasticsearch_endpoint = getpass("Host Address: ")
    elasticsearch_api_key = getpass("Elastic API Key: ")
    return Elasticsearch(hosts=elasticsearch_endpoint, api_key=elasticsearch_api_key)

es = connect_to_elastic()
```

elastic

```python
def connect_to_elastic():
    elasticsearch_endpoint = getpass("Host Address: ")
    elasticsearch_api_key = getpass("Elastic API Key: ")
    return Elasticsearch(hosts=elasticsearch_endpoint, api_key=elasticsearch_api_key)

es = connect_to_elastic()
```

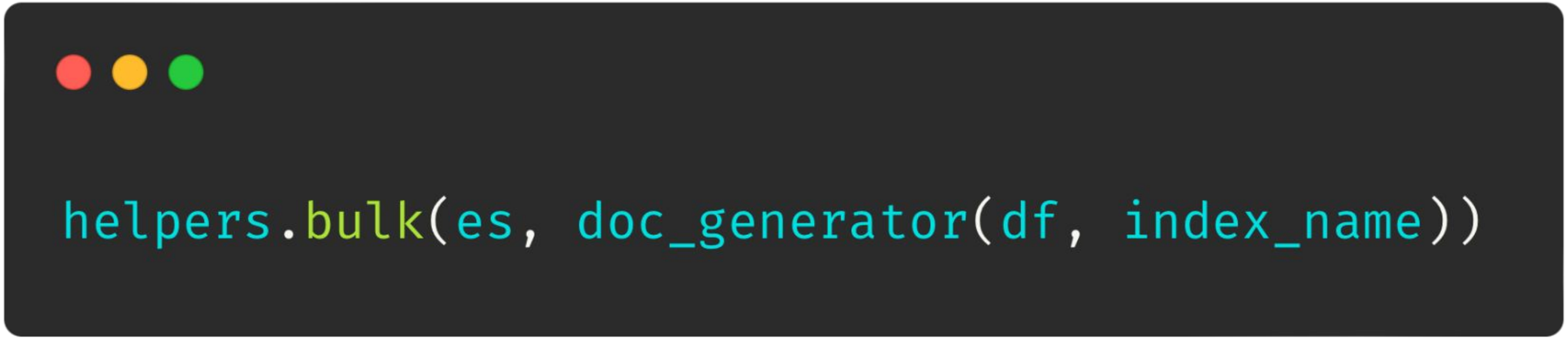
elastic

```
index_name = "all_things_open"
es.indices.create(index=index_name)
```

```
def doc_generator(df, index_name):
    for index, document in df.iterrows():
        yield {
            "_index": index_name,
            "_id": f"{document['id']}",
            "_source": document.to_dict(),
        }
```

elastic

```
def doc_generator(df, index_name):
    for index, document in df.iterrows():
        yield {
            "_index": index_name,
            "_id": f"{document['id']}",
            "_source": document.to_dict(),
        }
```



elastic

Parallel Bulk

parallel_bulk() is a tool that enhances the bulk() API by using multiple threads to process tasks simultaneously.

```
for success, info in helpers.parallel_bulk(es, doc_generator(df, 'all_things_open'), thread_count=4):
    if not success:
        print('A document failed:', info)
```

I had a problem

When I first started working with Elasticsearch my data in my index would quickly go out of date.

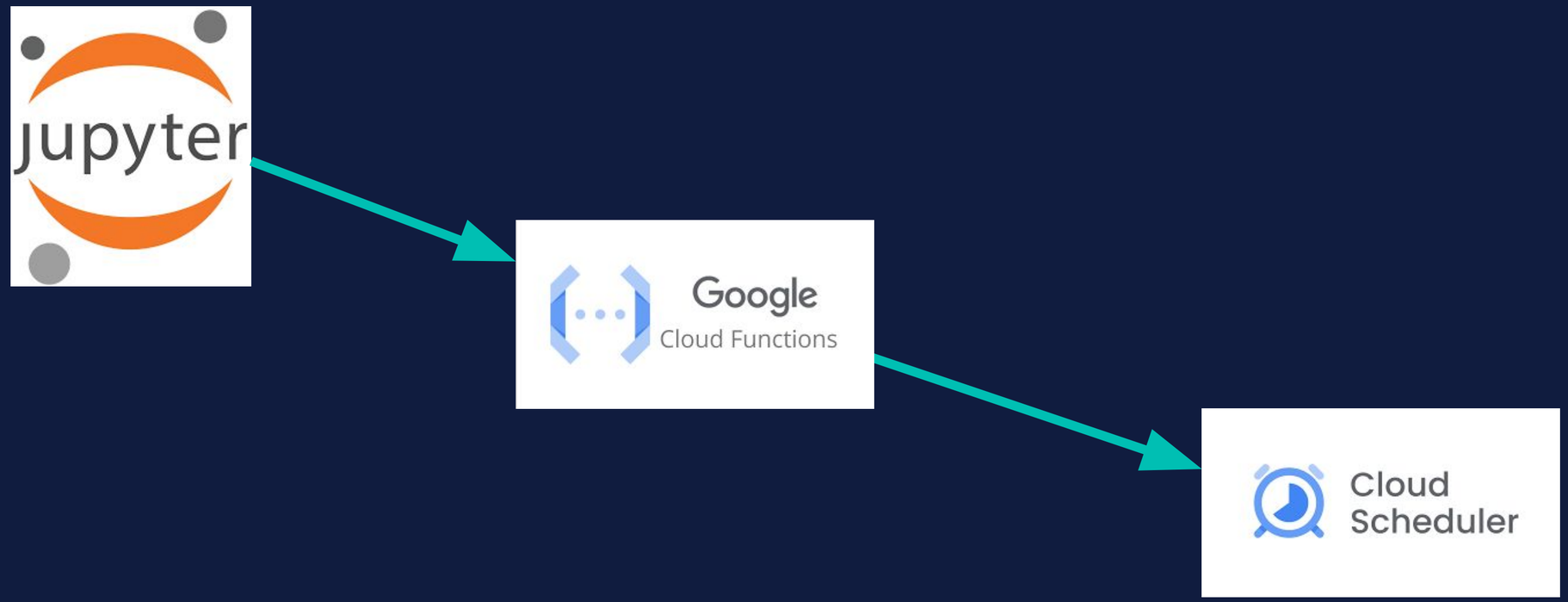

Oh wait, I know how
to solve this!

You can do this a lot of different ways

- Python scheduling libraries like `APScheduler` or `schedule`
- GCP Cloud Functions and Cloud Scheduler
- AWS Lambda
- Azure Function Apps
- Railway cron jobs
- Render cron jobs
- GitHub Actions


Using Google
Cloud Platform
elastic

Solution overview



https://cloud.google.com/scheduler
https://cloud.google.com/functions
https://www.elastic.co/search-labs/blog/keeping-your-elasticsearch-index-current-with-python-and-google-cloud-platform-functions

elastic

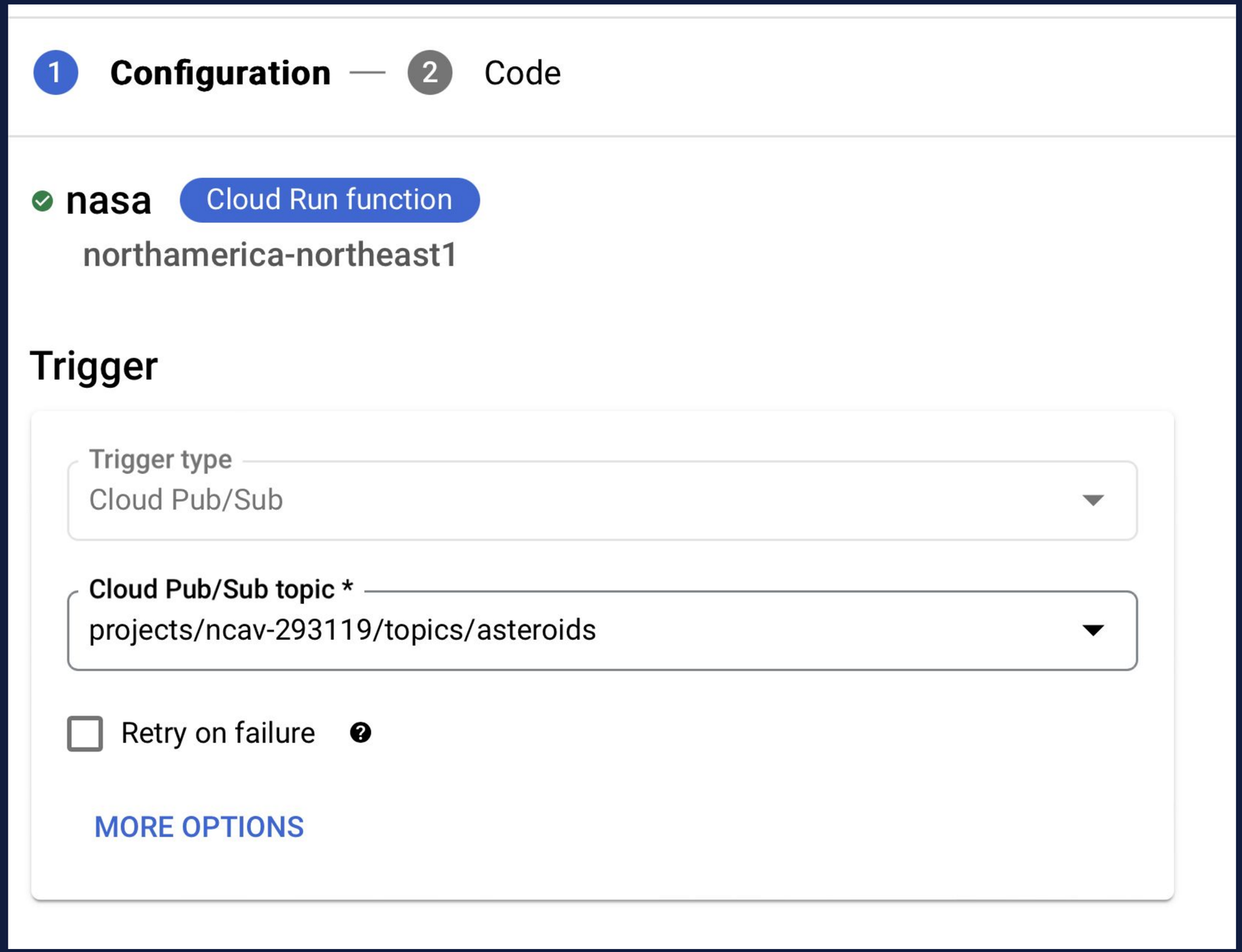
Step 1

Filter Filter functions

		Environment	Name ↑	Last deployed	Region	Recommendation	Trigger	Runtime	Memory allocated	Executed function
☐	✓	Cloud Run function (1st gen)	function-1	May 12, 2022, 2:07:38 PM	us-central1		Topic: Cats	Python 3.9	256 MB	hello_pubsub
☐	✓	Cloud Run function (1st gen)	function-2	Feb 14, 2024, 5:26:24 PM	us-central1		Topic: Celtics	Python 3.9	256 MB	hello_pubsub
☐	✓	Cloud Run function	function-3	Feb 15, 2024, 5:24:10 PM	northamerica-northeast1		HTTP	Python 3.12	256 MB	hello_pubsub
☐	✓	Cloud Run function	nasa	Jul 26, 2024, 8:39:17 PM	northamerica-northeast1		Topic: asteroids	Python 3.12	256 MiB	hello_pubsub

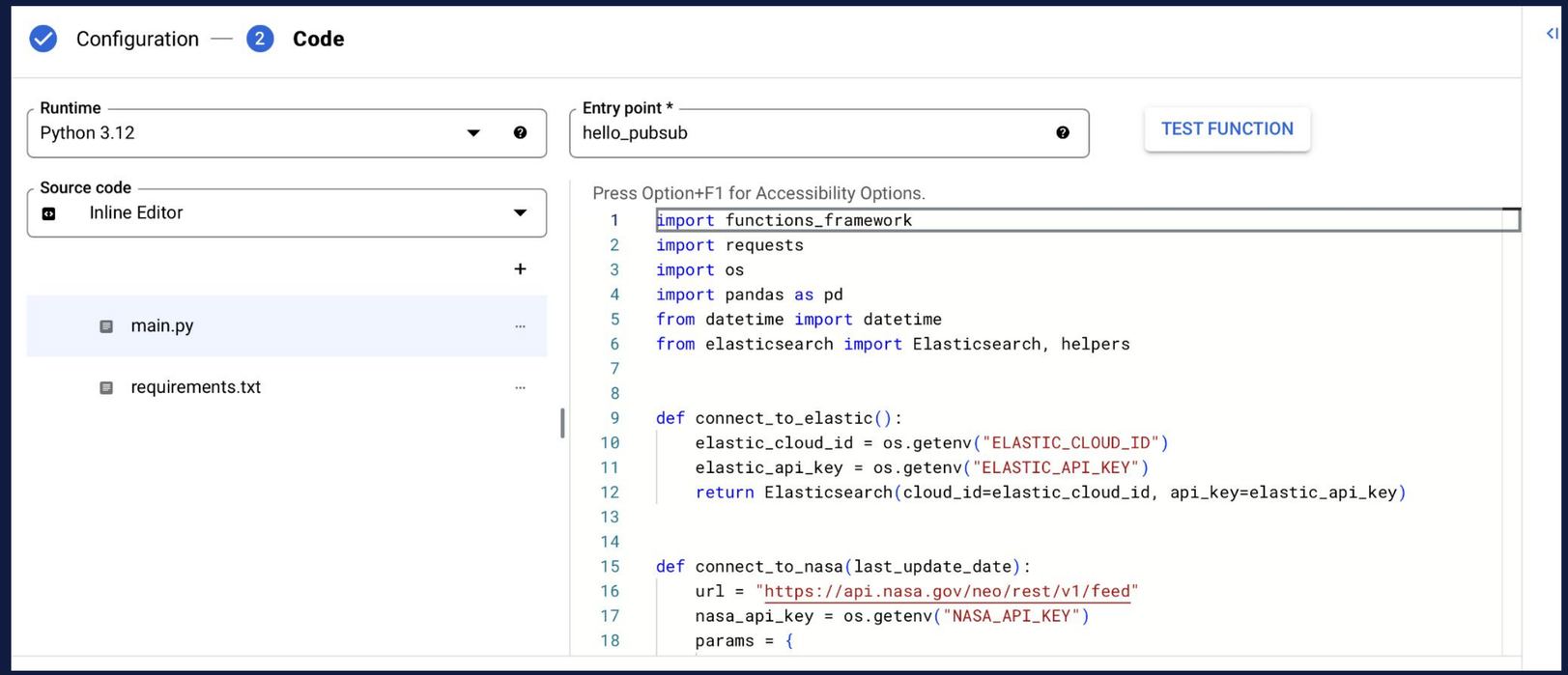
elastic

Step 2

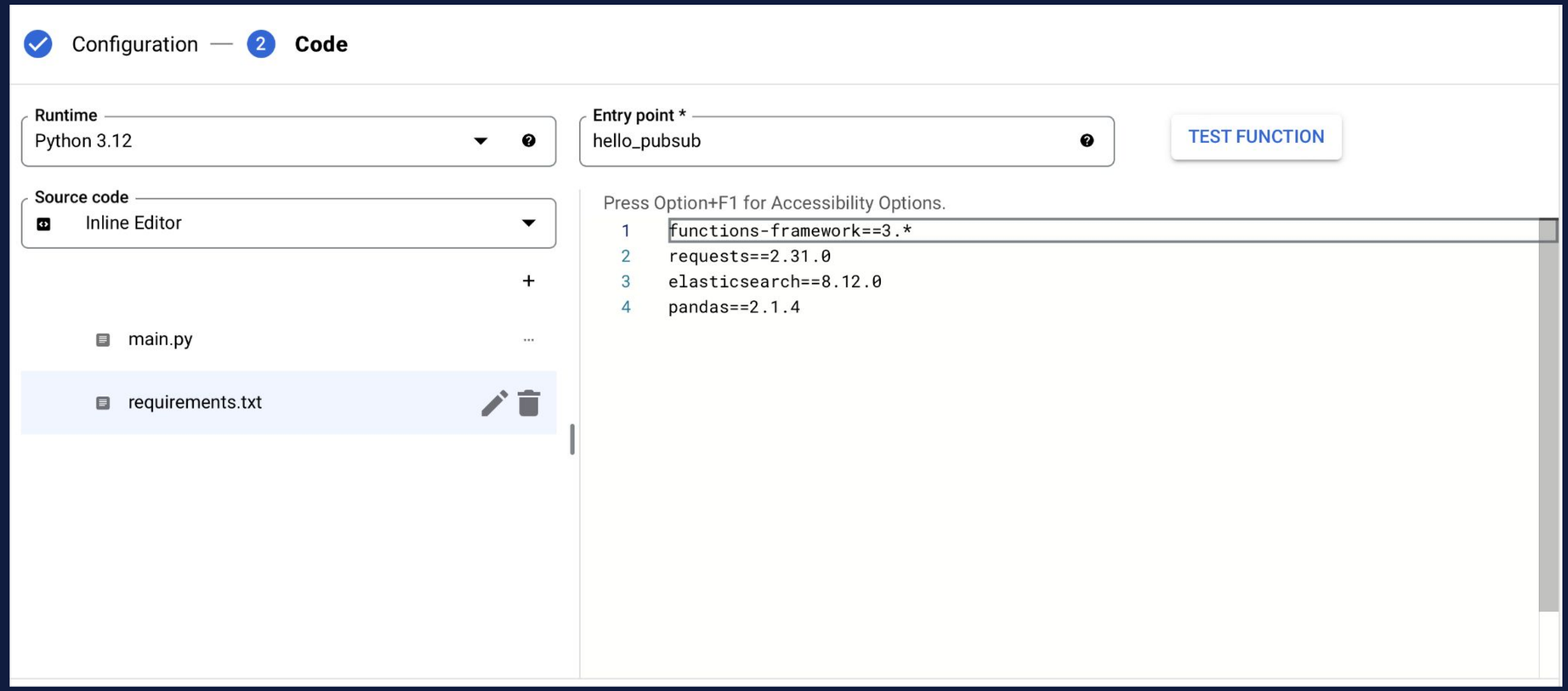


elastic

Step 3

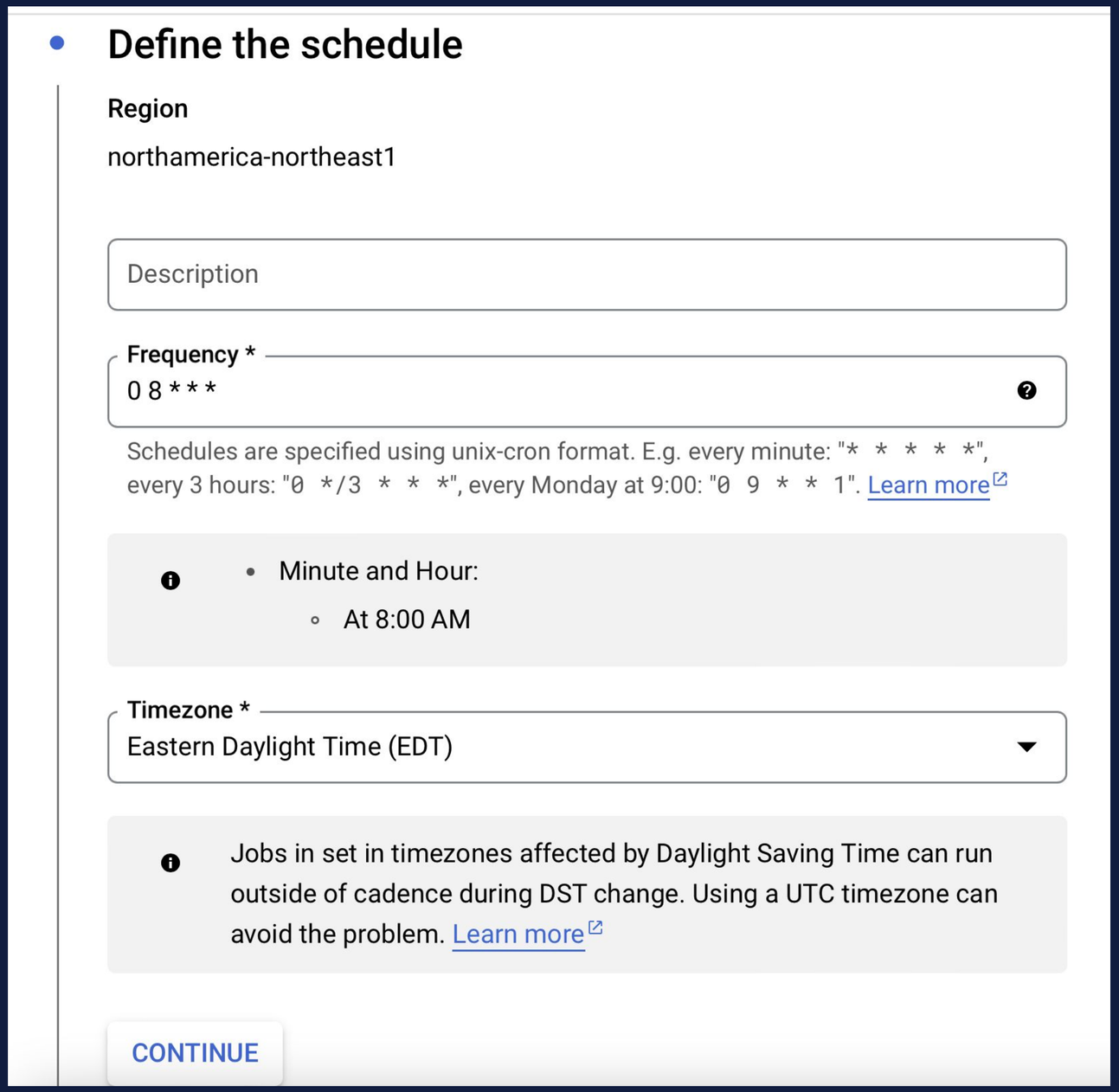

Configuration
2
Code
Runtime
Python 3.12
Entry point *
hello_pubsub
TEST FUNCTION
Source code
Inline Editor
main.py
requirements.txt
Press Option+F1 for Accessibility Options.
import functions_framework
import requests
import os
import pandas as pd
from datetime import datetime
from elasticsearch import Elasticsearch, helpers
def connect_to_elastic():
elastic_cloud_id = os.getenv("ELASTIC_CLOUD_ID")
elastic_api_key = os.getenv("ELASTIC_API_KEY")
return Elasticsearch(cloud_id=elastic_cloud_id, api_key=elastic_api_key)
def connect_to_nasa(last_update_date):
url = "https://api.nasa.gov/neo/rest/v1/feed"
nasa_api_key = os.getenv("NASA_API_KEY")
params = {

Step 4



elastic

Step 5



elastic

Step 6

SCHEDULER JOBS | APP ENGINE CRON JOBS

Filter Filter jobs

Name	Status of last execution	Region	State	Description	Frequency	Target	Last run	Next run
Cats	Success	us-central1	Enabled	Daily Cat Facts	0 */12 * * * (America/Detroit)	Topic : projects/ncav-293119/topics/Cats	Oct 27, 2024, 12:00:00 PM	Oct 28, 2024, 12:00:00 AM
Celtics_UPdate	Success	northamerica-northeast1	Enabled	Celtics_Sched	0 8 * * * (America/New_York)	Topic : projects/ncav-293119/topics/Celtics	Oct 27, 2024, 8:00:00 AM	Oct 28, 2024, 8:00:00 AM
nasa	Success	northamerica-northeast1	Enabled		0 8 * * * (America/New_York)	Topic : projects/ncav-293119/topics/asteroids	Oct 27, 2024, 8:00:00 AM	Oct 28, 2024, 8:00:00 AM

elastic

Using an ingest pipeline for Elasticsearch can be a natural next step for optimizing data uploads into an index, mainly if you deal with large volumes or complex data transformations.

https://www.elastic.co/guide/en/elasticsearch/reference/current/ingest.html

elastic

Depending on the size and frequency of data updates, consider batching data to reduce the number of API calls and to enhance performance.

Let me know if this talk inspires you to build anything. I'm @JessicaGarson on most platforms.

My email is jess.garson@elastic.co

elastic

Code and slides

https://github.com/JessicaGarson/Elasticsearch-Essentials-Data-Loading-with-Python-for-Interplanetary-Insights

elastic

RESOURCES FOR DEVELOPERS | BY DEVELOPERS LIKE YOU!

Search Labs

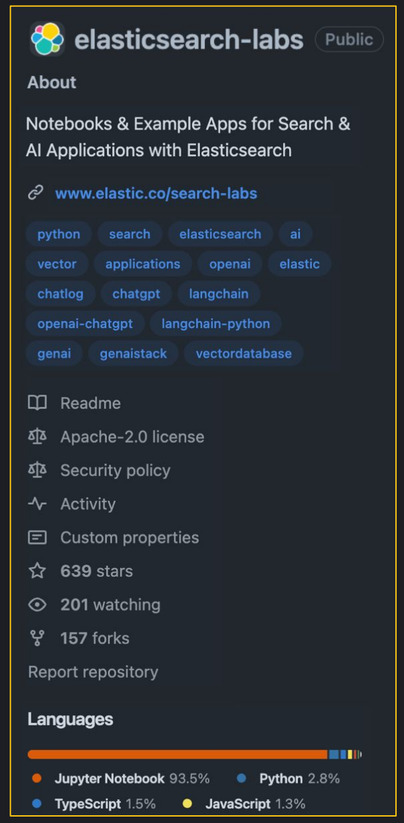
ML Research December 1, 2023

RAG evaluation metrics: A journey through metrics

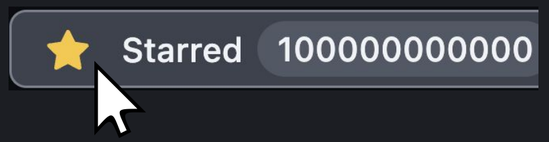
Explore RAG evaluation metrics like BLEU score, ROUGE score, PPL, BARTScore, and more. Discover...

By: **Quentin Herreros**, **Thomas Veasey** and **Thanos Papaoikonomou**

elastic.co/**search-labs**



github.com/elastic/**elasticsearch-labs**



Tutorials
GitHub Example Code
Integrations
Blogs

elastic


Thank you!
elastic