How to do the same thing over and over again and yield different results

elastic

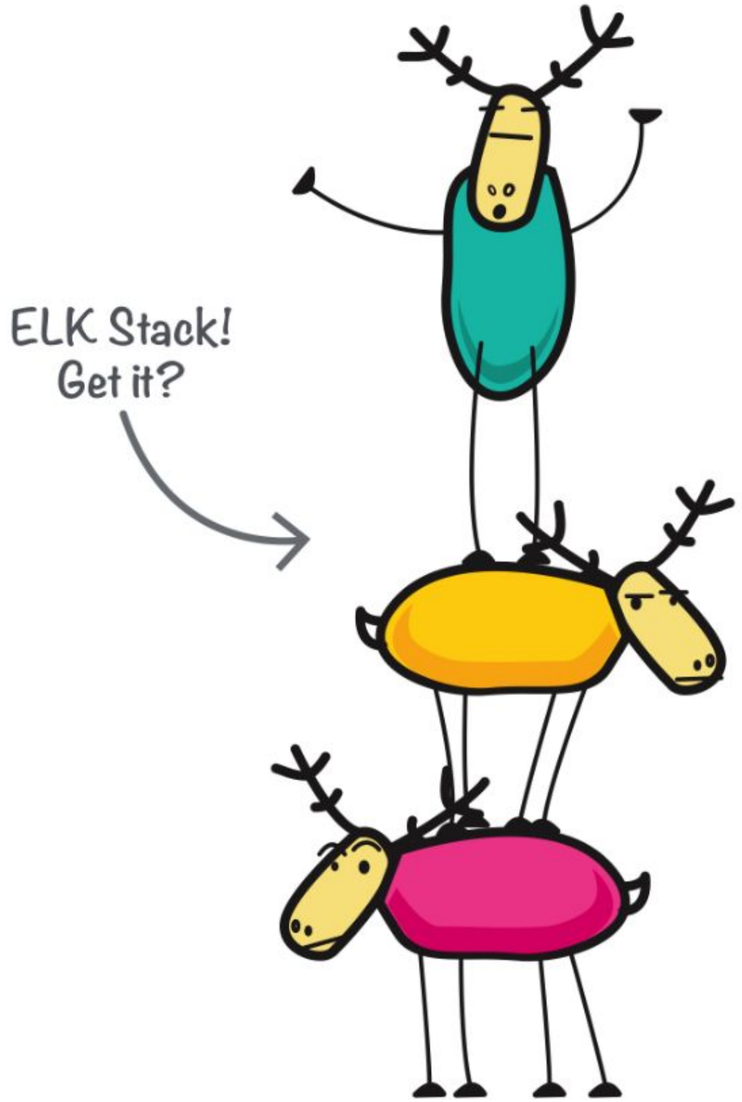
Jessica Garson

@JessicaGarson

@JessicaGarson@macaw.social

Senior Developer Advocate

elastic



E Elasticsearch

L Logstash

K Kibana

Add code link

elastic

Who is this talk for?

This talk is for anyone who is looking to get started with automation, wanting to learn more about scheduling services or anyone who is looking to get into developer relations.


My secret

Scheduling basics

elastic

Scheduling tasks is a fundamental aspect of programming

I've used a similar paradigm for many projects where I've had to run code at a specific time interval.

Technologies used

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

Cron jobs and serverless functions are both used for automating tasks.

What is a Cron Job?

A cron job is a command in Linux that schedules tasks to run in the future. Typically, it's used to set up jobs that happen on a regular basis, like weekly or monthly.

Cron jobs require a continuously running server and involve manual scaling and maintenance.

They are ideal for scheduled tasks such as backups or updates.

What’s a serverless function?

- Serverless functions are specialized, programmable functions typically hosted on a cloud provider's managed infrastructure.
- They perform specific tasks without requiring the user to maintain underlying servers.

Serverless functions operate in a cloud provider's managed environment, triggered by events rather than by schedule.

They automatically scale and require no server management.

What have I used this for in the past?

- Twitter / Mastodon bots
- How I solved my parking problem with Python
- Time based art

https://developer.x.com/en/docs/tutorials/how-to-create-a-twitter-bot-with-twitter-api-v2

https://youtu.be/pgDu37WtDhk

https://www.youtube.com/watch?v=OTQTS5aKh7s

elastic

What can go wrong?

- The code can go out of date
- The structure can change
- OMG my bot said something it shouldn’t

Cron syntax

elastic

Questions to ask yourself

- How often do you want something to happen?
- When do you want something to happen?
- What is the schedule you want this to run at?

```
# ┌──────────────── minute (0 - 59)
# │ ┌──────────────── hour (0 - 23)
# │ │ ┌──────────────── day of the month (1 - 31)
# │ │ │ ┌──────────────── month (1 - 12)
# │ │ │ │ ┌──────────────── day of the week (0 - 6) (Sunday to Saturday;
# │ │ │ │ │                                       7 is also Sunday on some systems)
# │ │ │ │ │
# │ │ │ │ │
# * * * * * command to execute
```

elastic

Commonly used syntax

* * * * *
Every minute

0 8 * * WED
At 8:00am on Wednesdays

0 */12 * * *
Every 12 hours

elastic

A resource I like

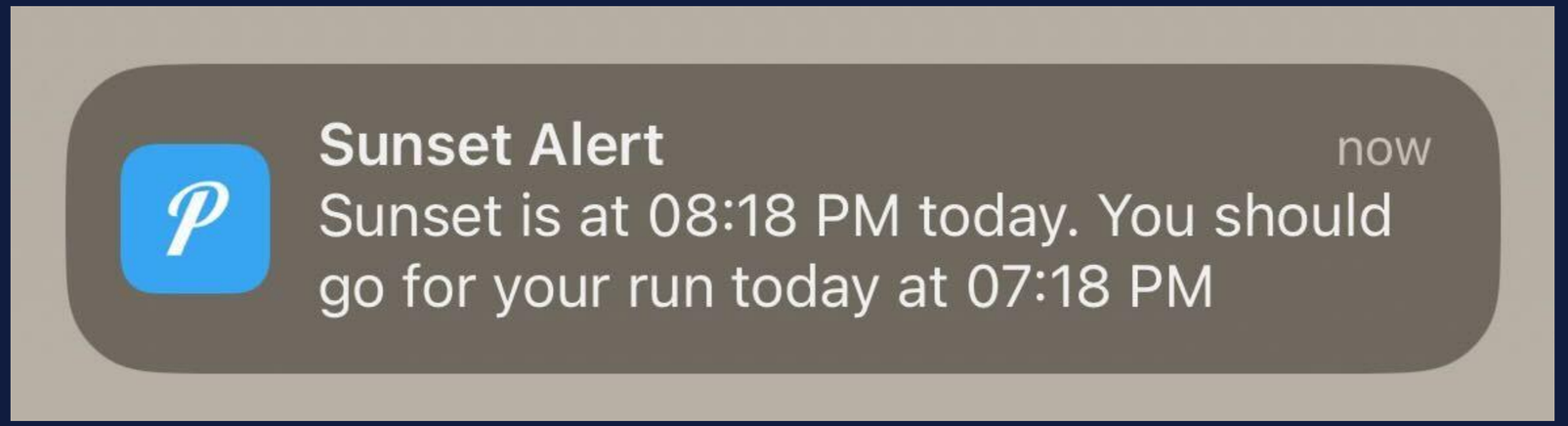
crontab.guru

elastic

When should I go for my run today?

elastic

I built a solution to let let me know what time I should go for a run.



elastic

This code is deployed using a Render cron job from a private GitHub repository.

https://docs.render.com/cronjobs

elastic

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!

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

Schema definition

Each index has a mapping that defines the structure of the documents, specifying field types and how they should be indexed and stored.

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

elastic

Scalability and Performance

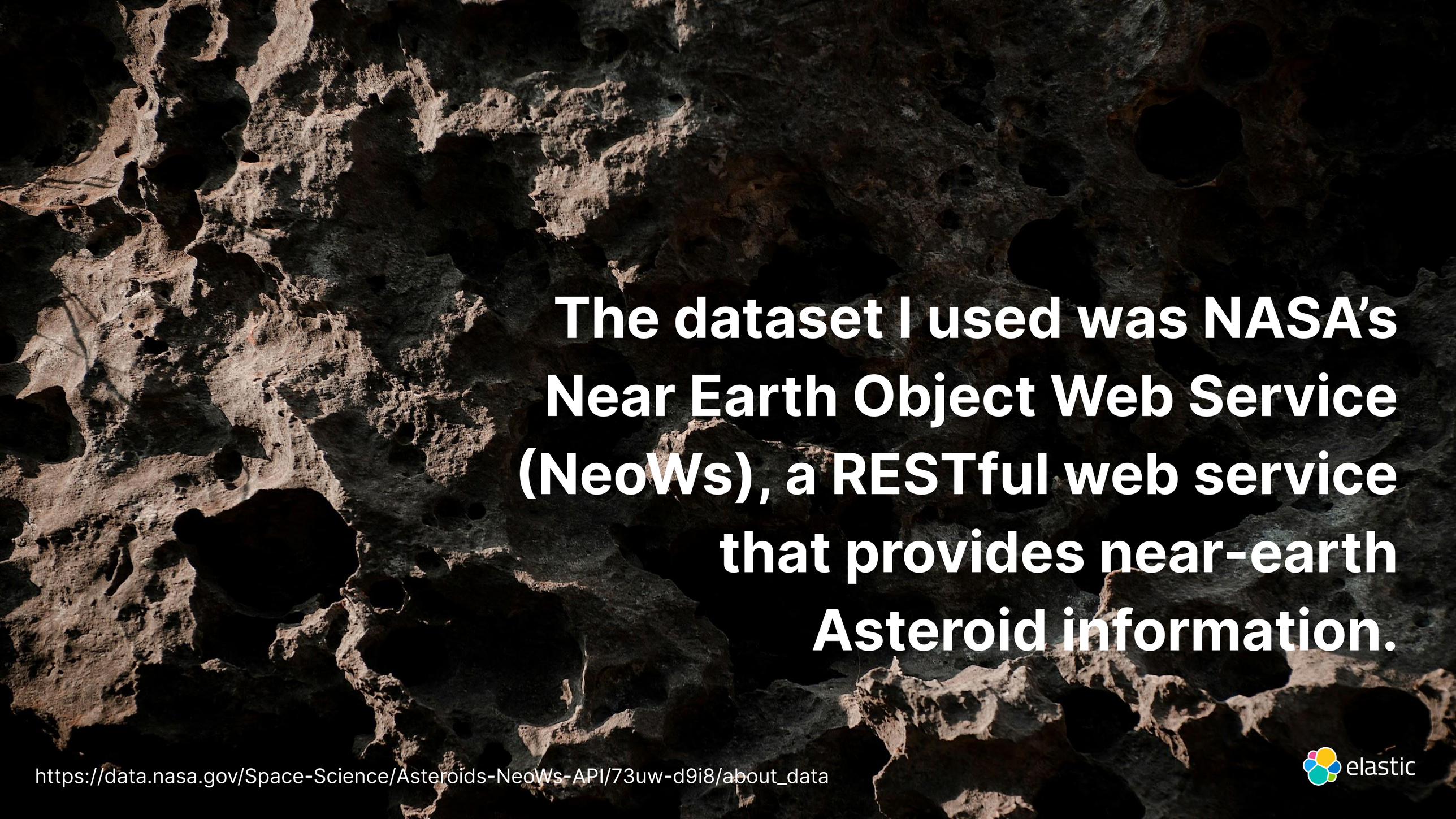
Elasticsearch indexes are designed to handle large volumes of data, offering fast search and retrieval capabilities through distributed architecture and advanced indexing techniques

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

elastic

Why use a time based solution for this problem?

Maintaining up-to-date data is crucial, especially when dealing with frequently changing dynamic datasets.


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

Let's take a look at the dataset in Postman

https://www.postman.com/

elastic

What you need to get started

- Python
- 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

You will need to have data loaded in your index already.

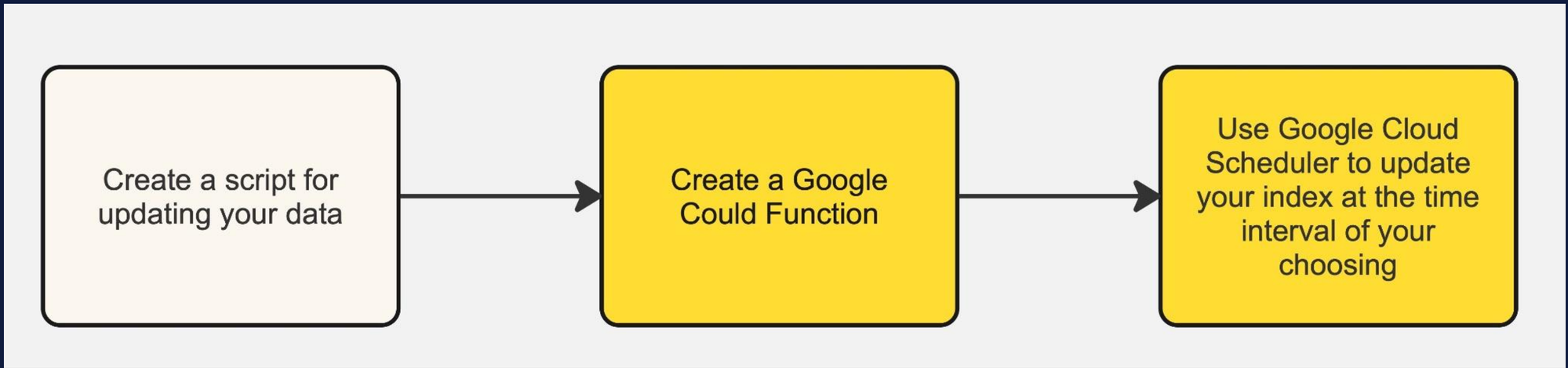
You can also update your script to create an index if one doesn’t already exist.

https://github.com/elastic/elasticsearch-labs/blob/main/supporting-blog-content/keeping-your-index-current/local_testing.ipynb

elastic


GCP

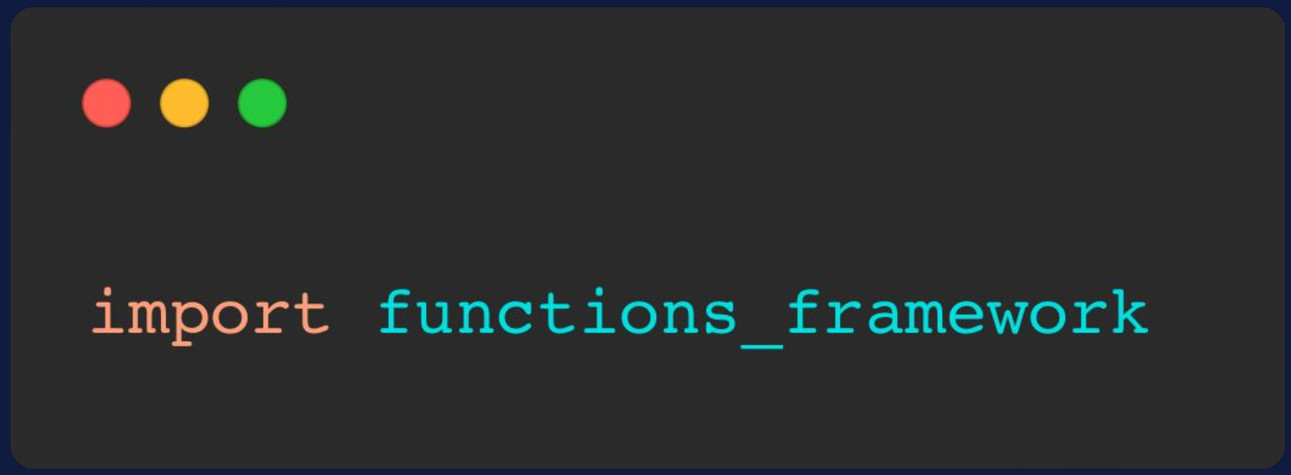
Solution overview



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

Import statement



elastic

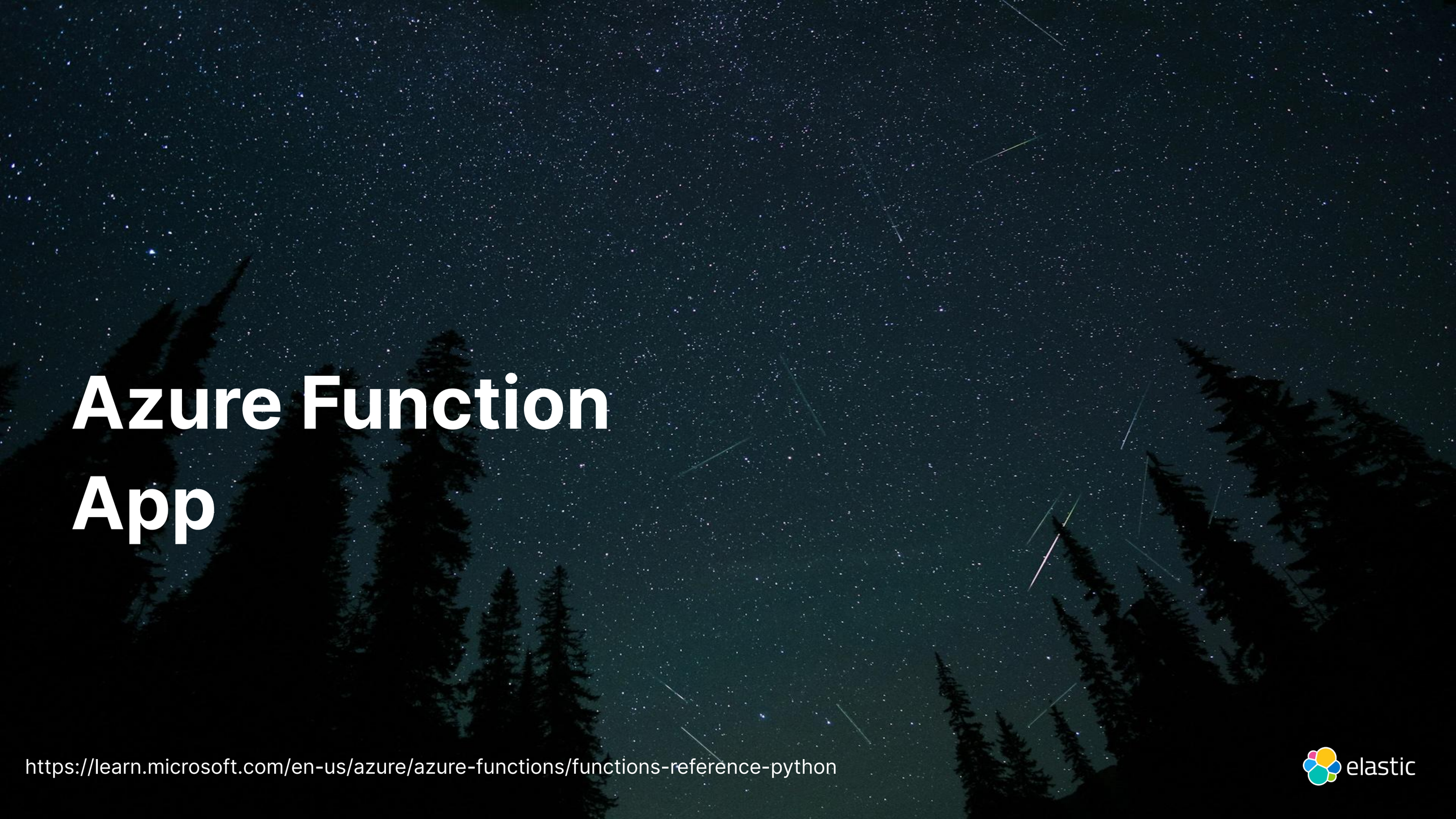
Like a main function

```python
@functions_framework.cloud_event
def hello_pubsub(cloud_event):
    index_name = "asteroid_data_set"
    es = connect_to_elastic()
    last_update_date = updated_last(es, index_name)
    print(last_update_date)
    response = connect_to_nasa(last_update_date)
    print(response)
    df = create_df(response)
    try:
        if df is None:
            raise ValueError("DataFrame is None. There may be a problem.")
        update_new_data(df, es, last_update_date, index_name)
        print(updated_last(es, index_name))
    except Exception as e:
        print(f"An error occurred: {e}")
```

elastic

Demo

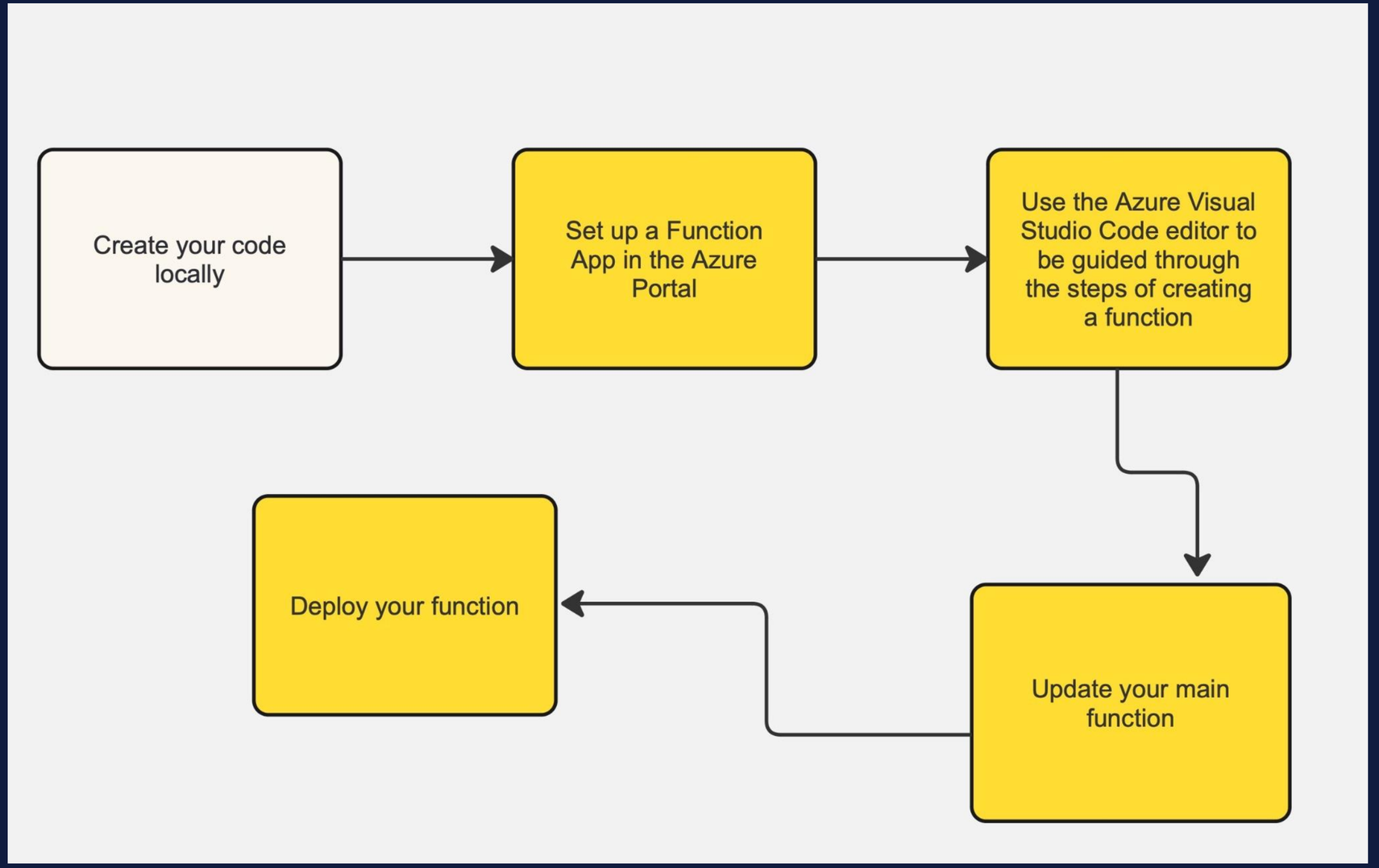
- Show configuration of the function
- Show the code
- Trigger a run
- Check in Elastic



https://learn.microsoft.com/en-us/azure/azure-functions/functions-reference-python

elastic

Solution overview



elastic

When creating function apps I usually start with the timed function template and adjust from there

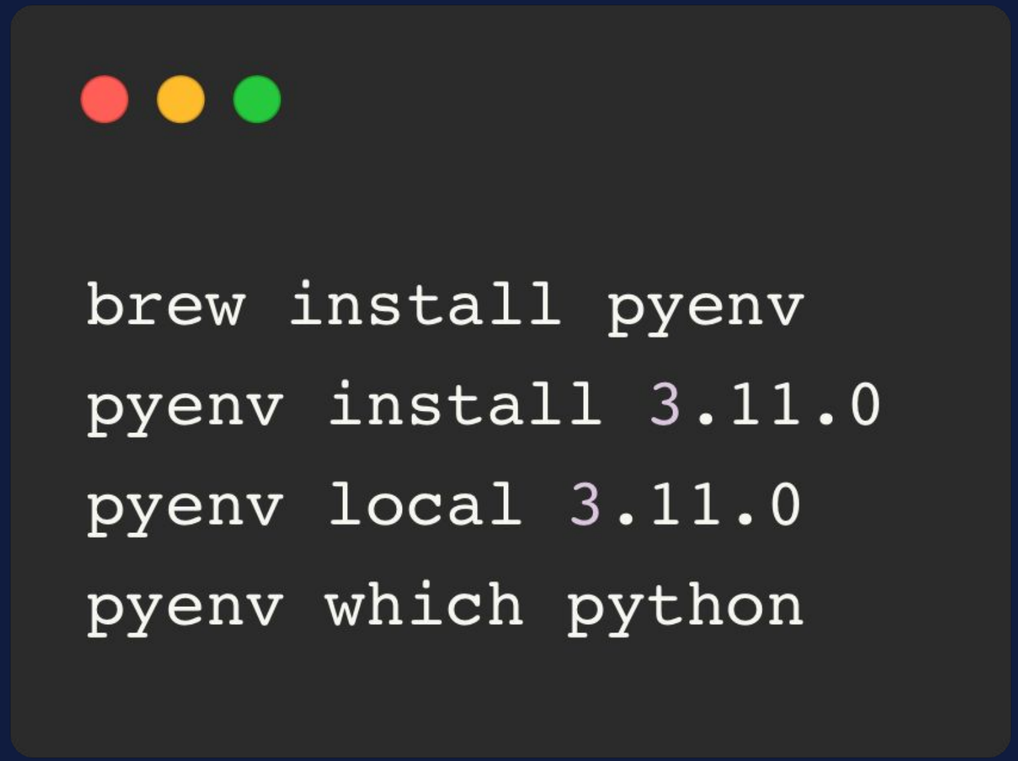
https://learn.microsoft.com/en-us/azure/azure-functions/functions-bindings-timer?tabs=python-v2

elastic

The most recent version of Python that’s supported is Python 3.11

elastic

I had to use pyenv to use Python 3.11



elastic

The cron syntax is slightly different

{second} {minute} {hour} {day} {month} {day of week}

Every morning at 9:30am

0 30 9 * * *

I used the Visual Studio Code Extension to create a function from my editor

1. Create a function
2. Select the template for timed trigger
3. Set the time interval
4. Paste the link to your Python path
5. Update function_app.py to your own code
6. Update your requirements.txt
7. Deploy your app

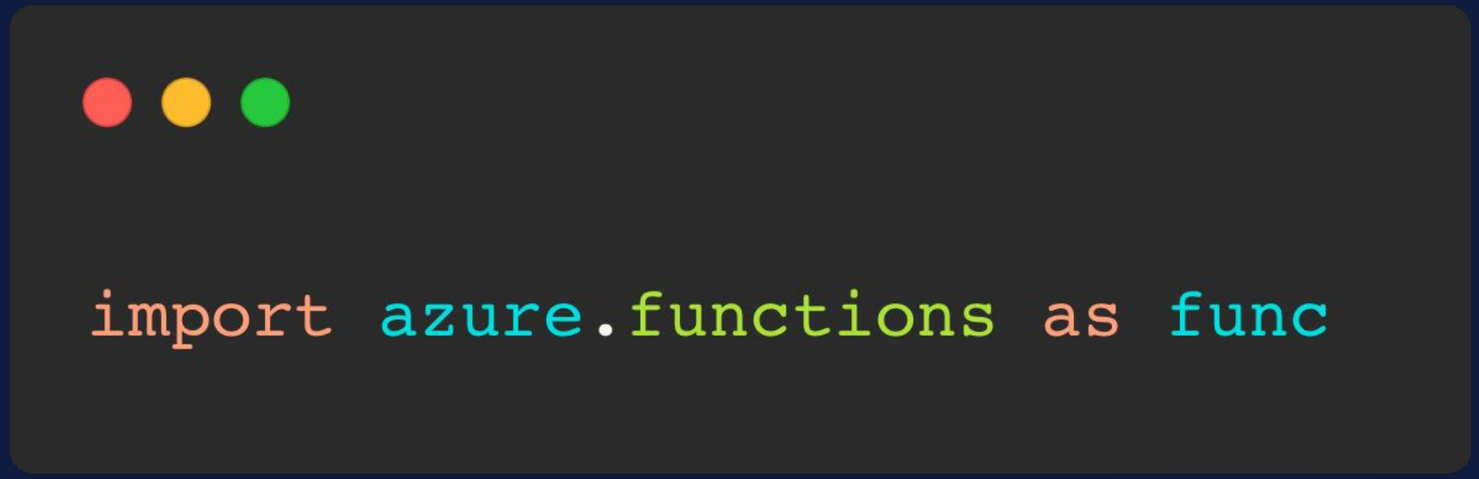
https://learn.microsoft.com/en-us/azure/azure-functions/functions-develop-vs-code?tabs=node-v4%2Cpython-v2

elastic

Inside of Azure console you can update your environment variables and test your code

elastic

Import statement



elastic

```python
@app.timer_trigger(schedule="0 30 9 * * *", arg_name="myTimer", run_on_startup=False,
use_monitor=False)
def main(myTimer: func.TimerRequest) -> None:
    if myTimer.past_due:
        logging.info('The timer is past due!')
    logging.info('Python timer trigger function started execution.')

    index_name = "python-azure-asteroid"
    es = connect_to_elastic()
    last_update_date = updated_last(es, index_name)
    response = connect_to_nasa(last_update_date)
    df = create_df(response)
    try:
        update_new_data(df, es, last_update_date, index_name)
        last_update_date = updated_last(es, index_name)
        logging.info(f'Updated last date: {last_update_date}')
    except Exception as e:
        logging.error(f"An error occurred: {e}")
```

elastic

Demo

- Show configuration of the function
- Show the code
- Trigger a run
- Check in Elastic

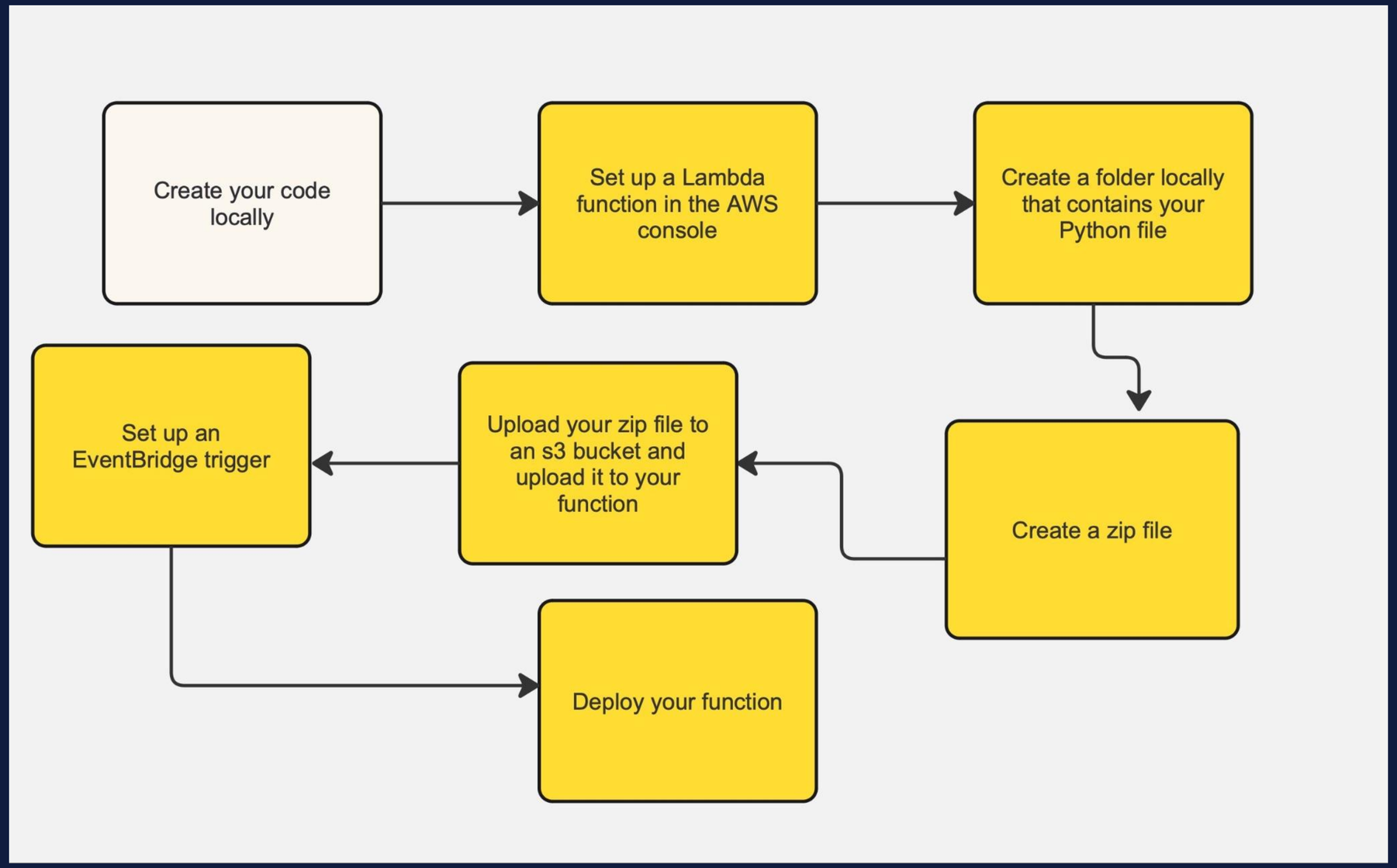
elastic

AWS Lambda

https://docs.aws.amazon.com/lambda/latest/dg/lambda-python.html

elastic

Solution overview



elastic

Cron scheduling is slightly different

cron(minutes hours day-of-month month day-of-week)

I used EventBridge for scheduling.

Everyday at 10am
0 10 * * ? *

https://docs.aws.amazon.com/eventbridge/latest/userguide/eb-scheduled-rule-pattern.html

No separate import statement is needed

elastic

```python
def lambda_handler(event, context):
    logging.basicConfig(level=logging.INFO)
    logging.info('Lambda function execution started.')

    index_name = "python-aws-asteroid"
    es = connect_to_elastic()
    last_update_date = updated_last(es, index_name)
    response = connect_to_nasa(last_update_date)
    df = create_df(response)

    try:
        update_new_data(df, es, last_update_date, index_name)
        last_update_date = updated_last(es, index_name)
        logging.info(f'Updated last date: {last_update_date}')
    except Exception as e:
        logging.error(f"An error occurred: {e}")
```

elastic

So this will be the same as the others, and you can create a requirements.txt file, and the required packages will be installed during the deployment process?

No 😔

Step 1: Create a virtual environment and install the required packages

```
cd aws-pyohio/
python3.12 -m venv my_virtual_env
source ./my_virtual_env/bin/activate
pip install requests elasticsearch boto3
deactivate
```

https://docs.aws.amazon.com/lambda/latest/dg/python-package.html

elastic

Step 2: Create a zip file to upload

```
cd my_virtual_env/lib/python3.12/site-packages
zip -r ../../../../my_deployment_package.zip .
cd ../../../../
zip my_deployment_package.zip lambda_function.py
```

https://docs.aws.amazon.com/lambda/latest/dg/python-package.html

elastic

Why is pandas missing?

elastic

Oh no!


```
{
  "errorMessage": "Unable to import module 'lambda_function': Unable to import required
dependencies:\nnumpy: Error importing numpy: you should not try to import numpy from\n        its source
directory; please exit the numpy source tree, and relaunch\n        your python interpreter from
there.",
  "errorType": "Runtime.ImportModuleError",
  "requestId": "509a605e-5987-42a3-9098-1276c19c6365",
  "stackTrace": []
}
```


elastic


elastic

I was able to get this working by installing the AWS SDK Pandas layer to my function:

AWSSDKPandas-Python312

https://aws-sdk-pandas.readthedocs.io/en/stable/layers.html

elastic

Demo

- Show configuration of the function
- Show the code
- Trigger a run
- Check in Elastic

elastic


Key takeaways

	Render	GCP	Azure	AWS
Advantages	Really easy to deploy and configure	Similar to how I write code locally	Scheduling happens in the in the code itself.	Widely used in the community
Drawbacks	Might not scale as well for larger projects	Not as widely used.	The process of creating a function took me the longest	You need to create a deployment zip.

- Be sure to check out the pricing for each technologies used
- The timezone is usually UTC
- The cron syntax might be slightly different depending on the technology you are using
- Some services alert you when your scheduled job fails but not all. You may have to set up your own alerting.

Next steps

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.

elastic

Be mindful of data security.

elastic

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

What can you schedule?

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

Let us know if you have any questions on our Discuss forums and the community Slack channel.

https://discuss.elastic.co/
https://communityinviter.com/apps/elasticstack/elastic-community

elastic


Thank you!
elastic