



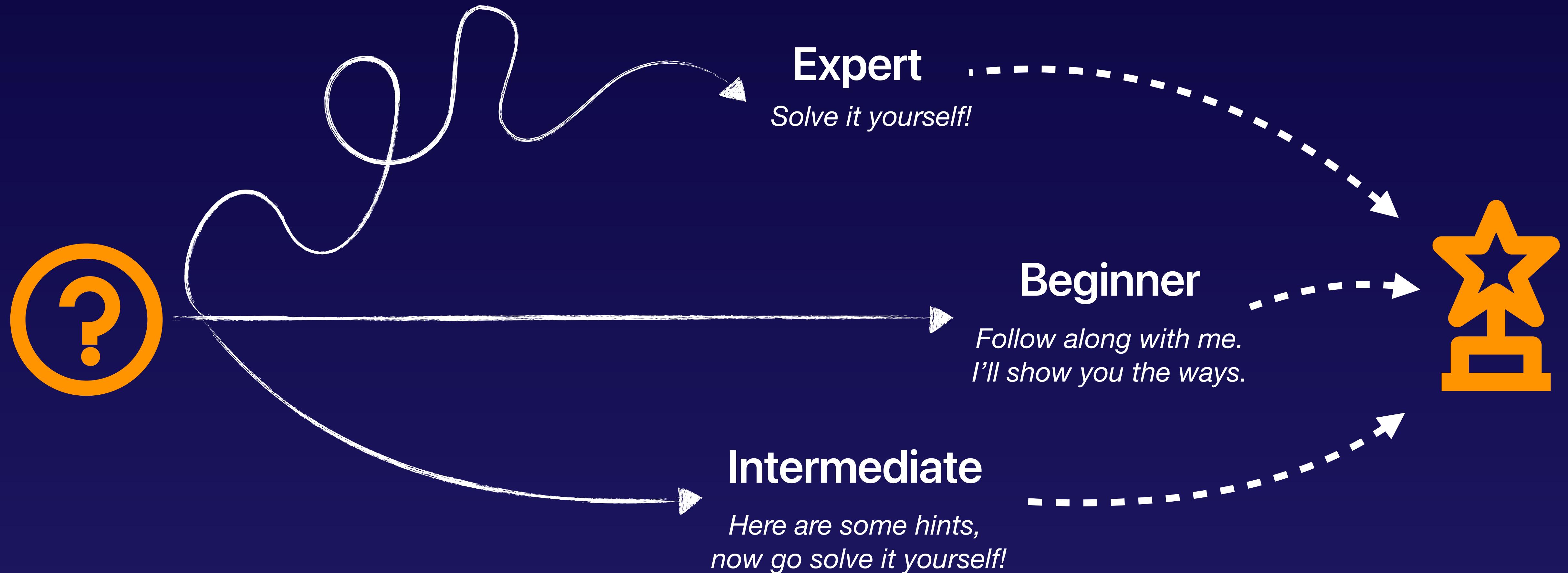
Streaming Data Collection Lab



Brock Tubre

INSTRUCTOR

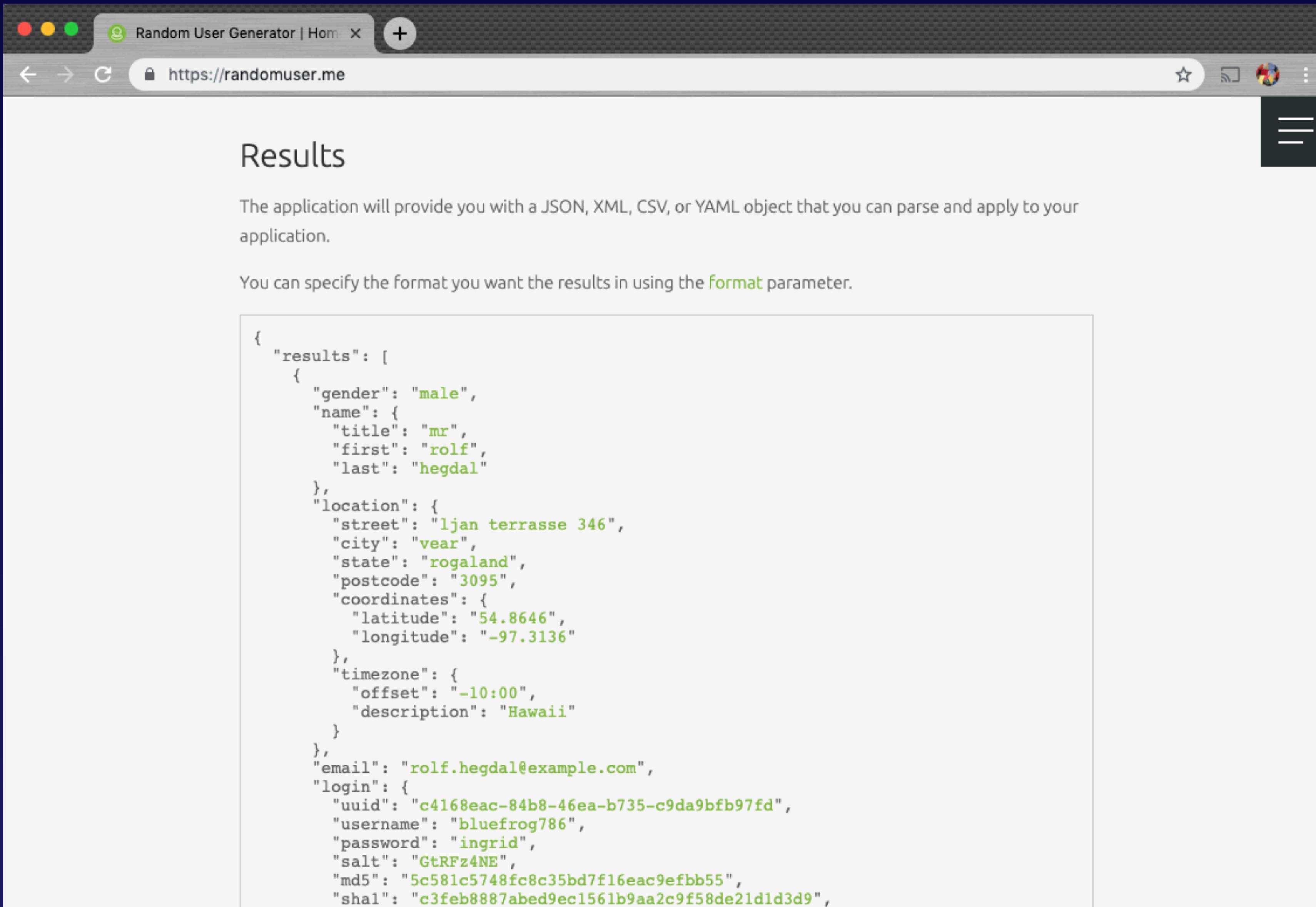
Choose Your Path



Use Case

You work for a company who has thousands of users interacting with the company's application. You have been tasked with capturing real-time data about the users for a marketing campaign. You need to capture information like name, age, gender, and location for users who are 21 and older.





The application will provide you with a JSON, XML, CSV, or YAML object that you can parse and apply to your application.

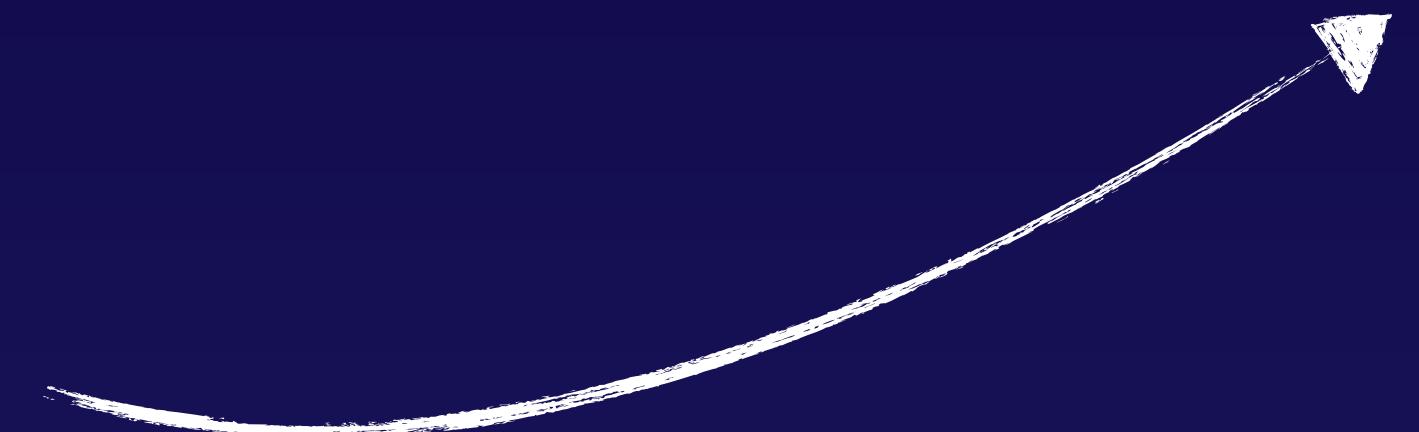
You can specify the format you want the results in using the [format](#) parameter.

```
{
  "results": [
    {
      "gender": "male",
      "name": {
        "title": "mr",
        "first": "rolf",
        "last": "hegdal"
      },
      "location": {
        "street": "1jan terrasse 346",
        "city": "vear",
        "state": "rogaland",
        "postcode": "3095",
        "coordinates": {
          "latitude": "54.8646",
          "longitude": "-97.3136"
        },
        "timezone": {
          "offset": "-10:00",
          "description": "Hawaii"
        }
      },
      "email": "rolf.hegdal@example.com",
      "login": {
        "uuid": "c4168eac-84b8-46ea-b735-c9da9bfb97fd",
        "username": "bluefrog786",
        "password": "ingrid",
        "salt": "GtRFz4NE",
        "md5": "5c581c5748fc8c35bd7f16eac9efbb55",
        "sha1": "c3feb8887abed9ec1561b9aa2c9f58de21d1d3d9"
      }
    }
  ]
}
```

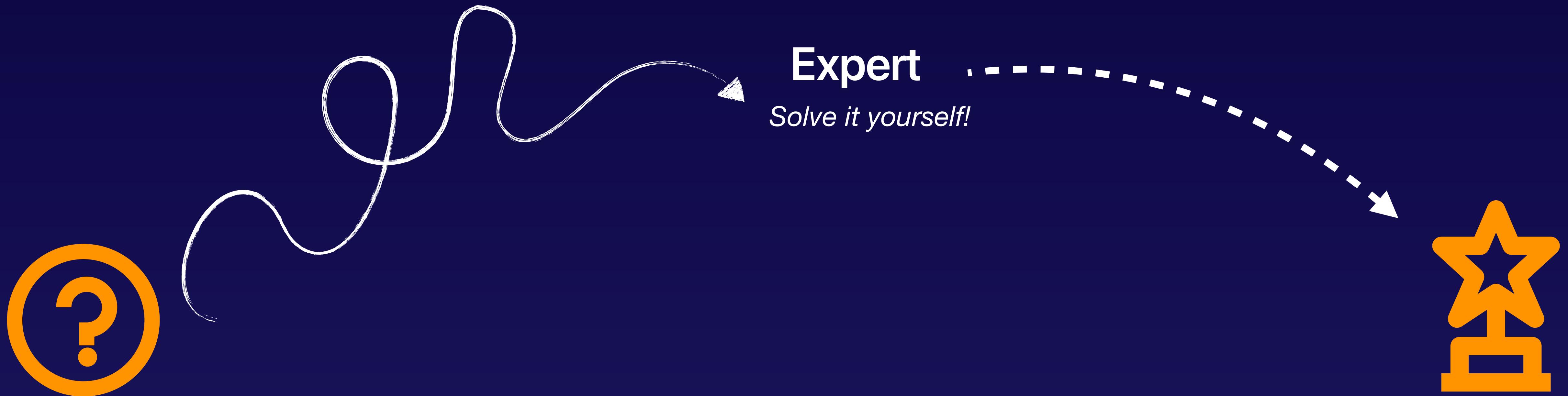
- **JSON files**
- **Stored in S3**
- **First Name**
- **Last Name**
- **Age (≥ 21)**
- **Gender**
- **Latitude**
- **Longitude**

Final results?

- JSON files
- Stored in S3
- First Name
- Last Name
- Age (≥ 21)
- Gender
- Latitude
- Longitude



```
{  
    "FIRST": "gladimir",  
    "LAST": "silveira",  
    "AGE": 40,  
    "GENDER": "male",  
    "LATITUDE": -72.6634,  
    "LONGITUDE": 54.8243  
}  
{  
    "FIRST": "aquino",  
    "LAST": "peixoto",  
    "AGE": 44,  
    "GENDER": "male",  
    "LATITUDE": -71.4789,  
    "LONGITUDE": -124.9399  
}  
{  
    "FIRST": "elisete",  
    "LAST": "ferreira",  
    "AGE": 42,  
    "GENDER": "male",  
    "LATITUDE": -72.8232,  
    "LONGITUDE": 131.9257  
}  
...
```



Using PutRecord

Create a simple python program that simulates streaming data. Call the PutRecord to send data to Kinesis Streams for ingestion.

Using PutRecord

```
import requests
import boto3
import uuid
import time
import random
import json

client = boto3.client('kinesis', region_name='<INSERT_YOUR_REGION>')
partition_key = str(uuid.uuid4())

while True:
    r = requests.get('https://randomuser.me/api/?exc=login')
    data = json.dumps(r.json())
    client.put_record(
        StreamName='<INSERT_YOUR_STREAM_NAME>',
        Data=data,
        PartitionKey=partition_key)
    time.sleep(random.uniform(0, 1))
```

Using PutRecord

```
import requests
import boto3
import uuid
import time
import random
import json

client = boto3.client('kinesis', region_name='<INSERT_YOUR_REGION>')
partition_key = str(uuid.uuid4())

while True:
    r = requests.get('https://randomuser.me/api/?exc=login')
    data = json.dumps(r.json())
    client.put_record(
        StreamName='<INSERT_YOUR_STREAM_NAME>',
        Data=data,
        PartitionKey=partition_key)
    time.sleep(random.uniform(0, 1))
```

Using PutRecord

```
import requests
import boto3
import uuid
import time
import random
import json

client = boto3.client('kinesis', region_name='<INSERT_YOUR_REGION>')
partition_key = str(uuid.uuid4())

while True:
    r = requests.get('https://randomuser.me/api/?exc=login')
    data = json.dumps(r.json())
    client.put_record(
        StreamName='<INSERT_YOUR_STREAM_NAME>',
        Data=data,
        PartitionKey=partition_key)
    time.sleep(random.uniform(0, 1))
```

Using PutRecord

```
import requests
import boto3
import uuid
import time
import random
import json

client = boto3.client('kinesis', region_name='<INSERT_YOUR_REGION>')
partition_key = str(uuid.uuid4())

while True:
    r = requests.get('https://randomuser.me/api/?exc=login')
    data = json.dumps(r.json())
    client.put_record(
        StreamName='<INSERT_YOUR_STREAM_NAME>',
        Data=data,
        PartitionKey=partition_key)
    time.sleep(random.uniform(0, 1))
```

Using PutRecord

```
import requests
import boto3
import uuid
import time
import random
import json

client = boto3.client('kinesis', region_name='<INSERT_YOUR_REGION>')
partition_key = str(uuid.uuid4())

while True:
    r = requests.get('https://randomuser.me/api/?exc=login')
    data = json.dumps(r.json())
    client.put_record(
        StreamName='<INSERT_YOUR_STREAM_NAME>',
        Data=data,
        PartitionKey=partition_key)
    time.sleep(random.uniform(0, 1))
```

Using PutRecord

```
import requests
import boto3
import uuid
import time
import random
import json

client = boto3.client('kinesis', region_name='<INSERT_YOUR_REGION>')
partition_key = str(uuid.uuid4())

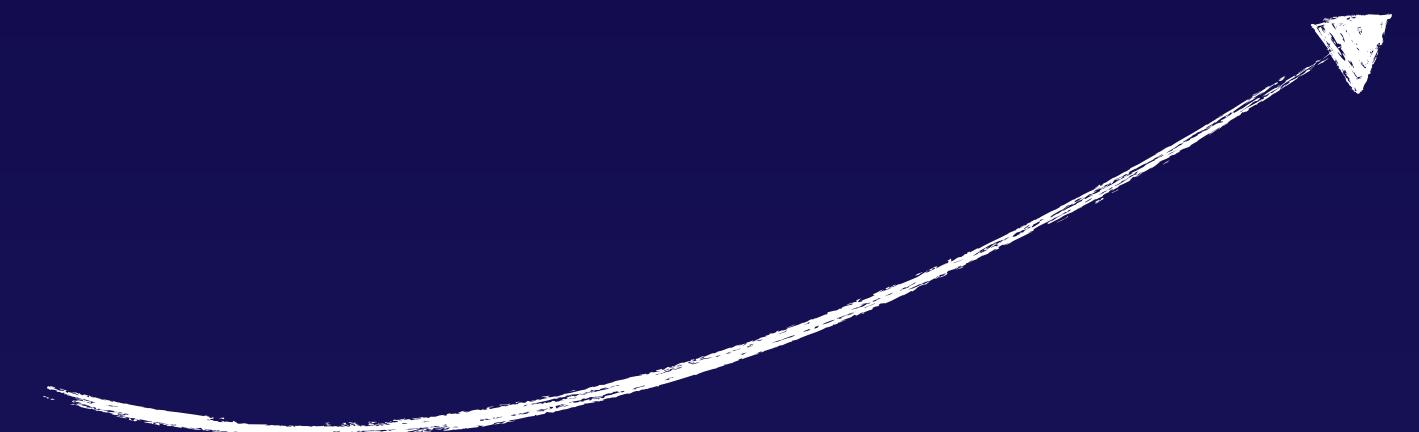
while True:
    r = requests.get('https://randomuser.me/api/?exc=login')
    data = json.dumps(r.json())
    client.put_record(
        StreamName='<INSERT_YOUR_STREAM_NAME>',
        Data=data,
        PartitionKey=partition_key)
    time.sleep(random.uniform(0, 1))
```

Transform and Load into S3

Find a way to transform the streaming data and output the results onto S3.

Final results?

- **JSON files**
- **Stored in S3**
- **First Name**
- **Last Name**
- **Age (> 21)**
- **Gender**
- **Latitude**
- **Longitude**

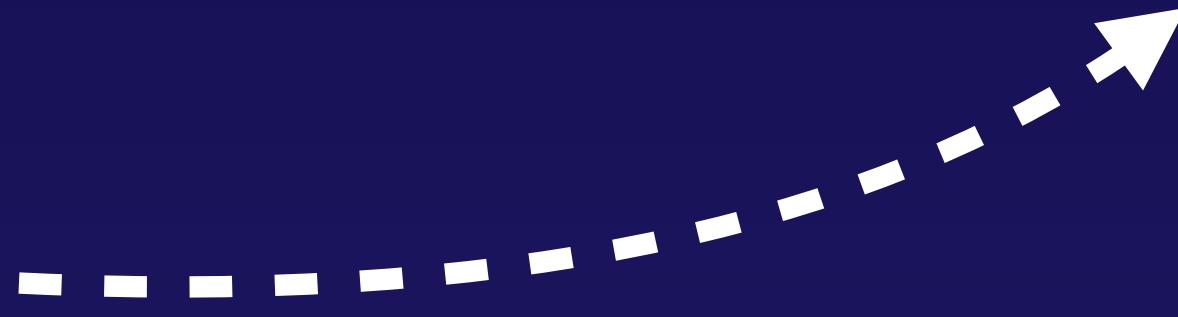


```
{  
  "FIRST": "gladimir",  
  "LAST": "silveira",  
  "AGE": 40,  
  "GENDER": "male",  
  "LATITUDE": -72.6634,  
  "LONGITUDE": 54.8243  
}  
{  
  "FIRST": "aquino",  
  "LAST": "peixoto",  
  "AGE": 44,  
  "GENDER": "male",  
  "LATITUDE": -71.4789,  
  "LONGITUDE": -124.9399  
}  
{  
  "FIRST": "elisete",  
  "LAST": "ferreira",  
  "AGE": 42,  
  "GENDER": "male",  
  "LATITUDE": -72.8232,  
  "LONGITUDE": 131.9257  
}  
...
```



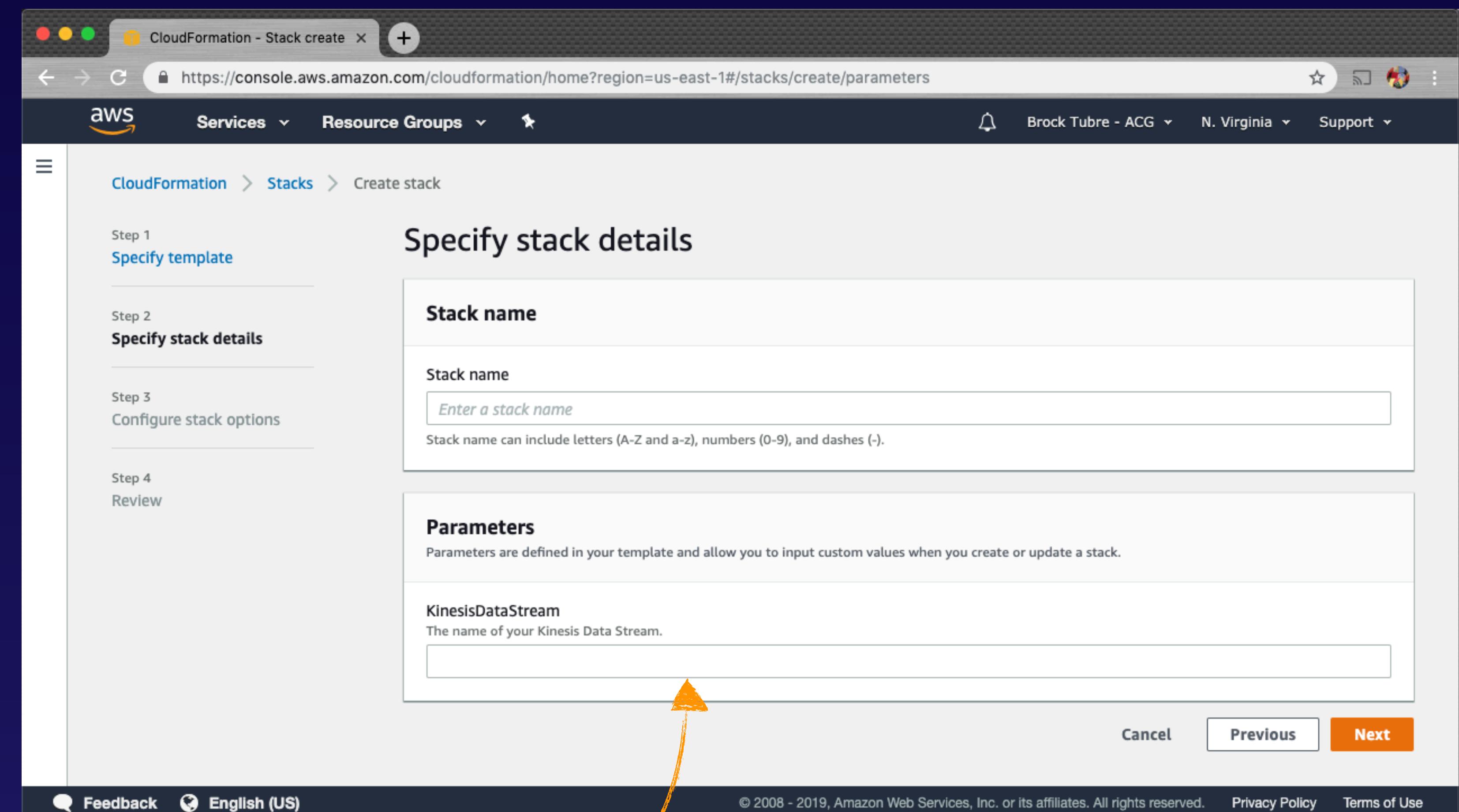
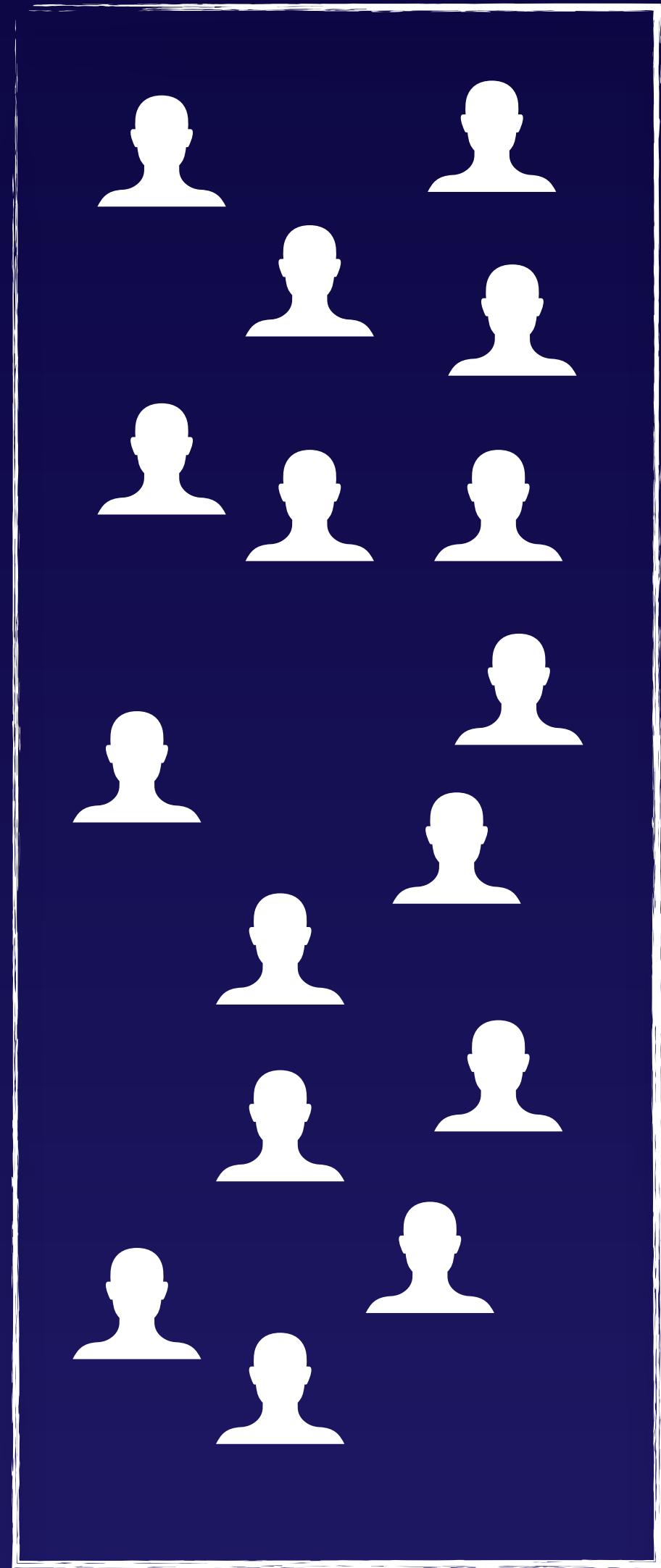
Intermediate

*Here are some hints,
now go solve it yourself!*



Streaming Data Collection Lab

Data Producer

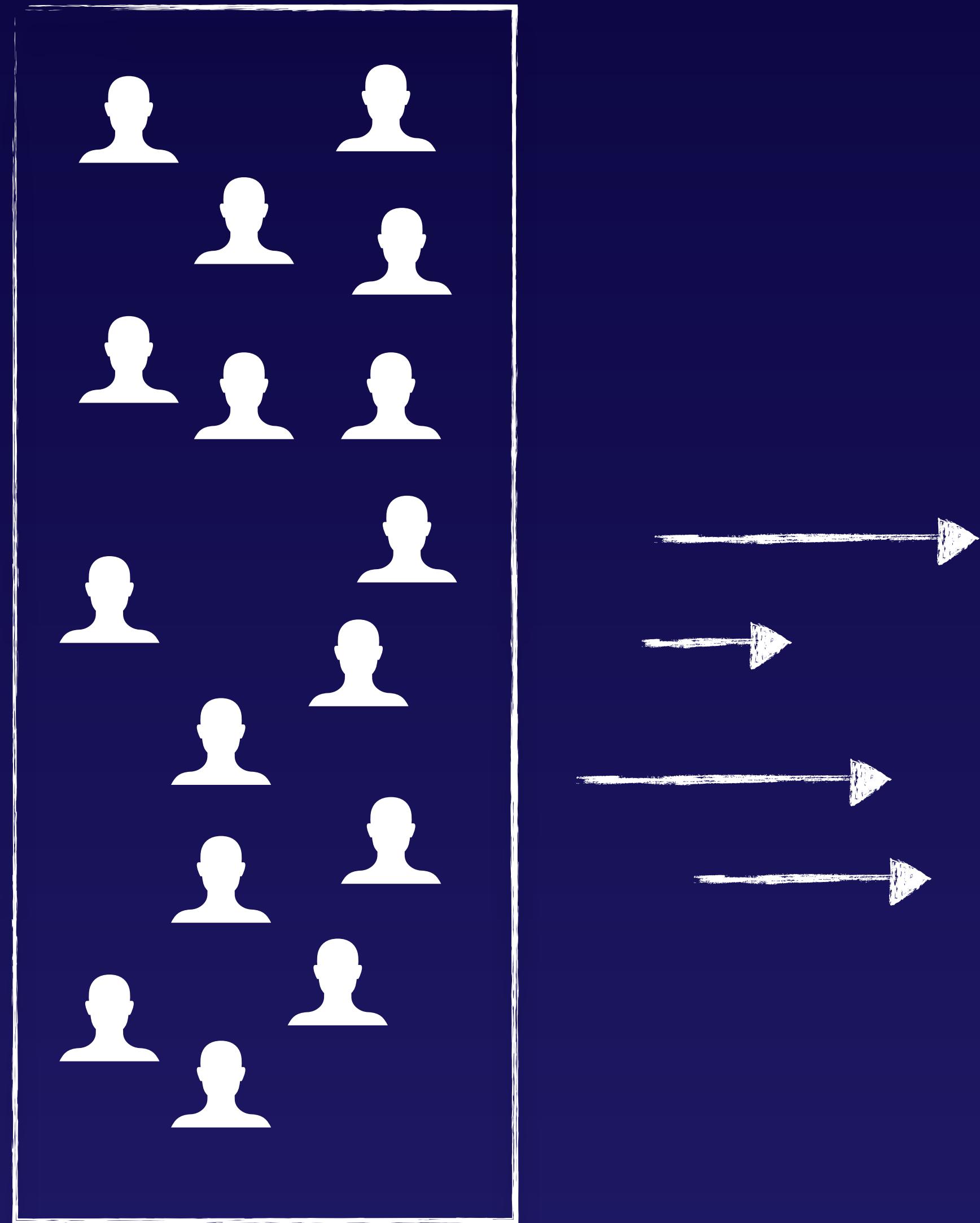


Screenshot of the AWS CloudFormation 'Specify stack details' step. The page shows four steps: Step 1 (Specify template), Step 2 (Specify stack details), Step 3 (Configure stack options), and Step 4 (Review). The 'Specify stack details' section contains fields for 'Stack name' and 'Parameters'. The 'Parameters' section includes a field for 'KinesisDataStream' with the placeholder text 'The name of your Kinesis Data Stream.' An orange arrow points from this field to the text 'Insert your stream name here.' located below the diagram.

*Insert your stream
name here.*

Streaming Data Collection Lab

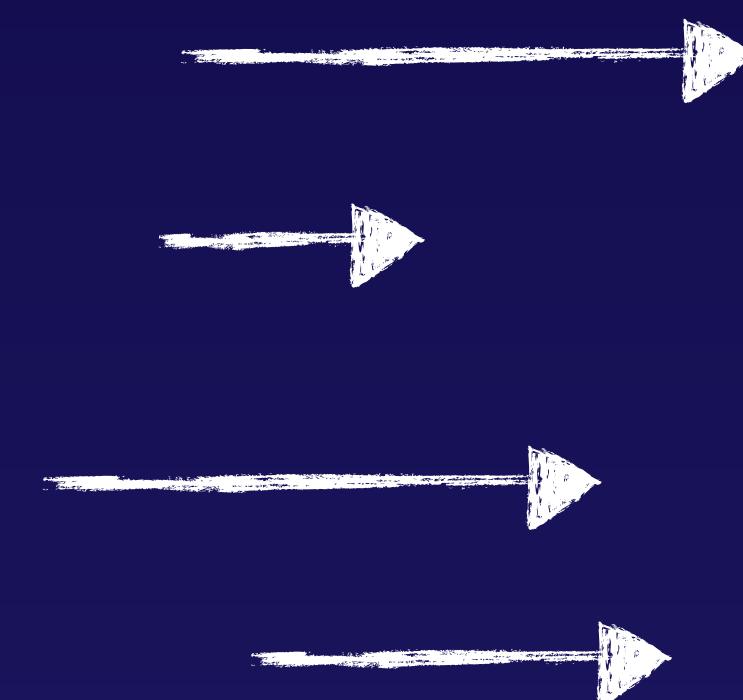
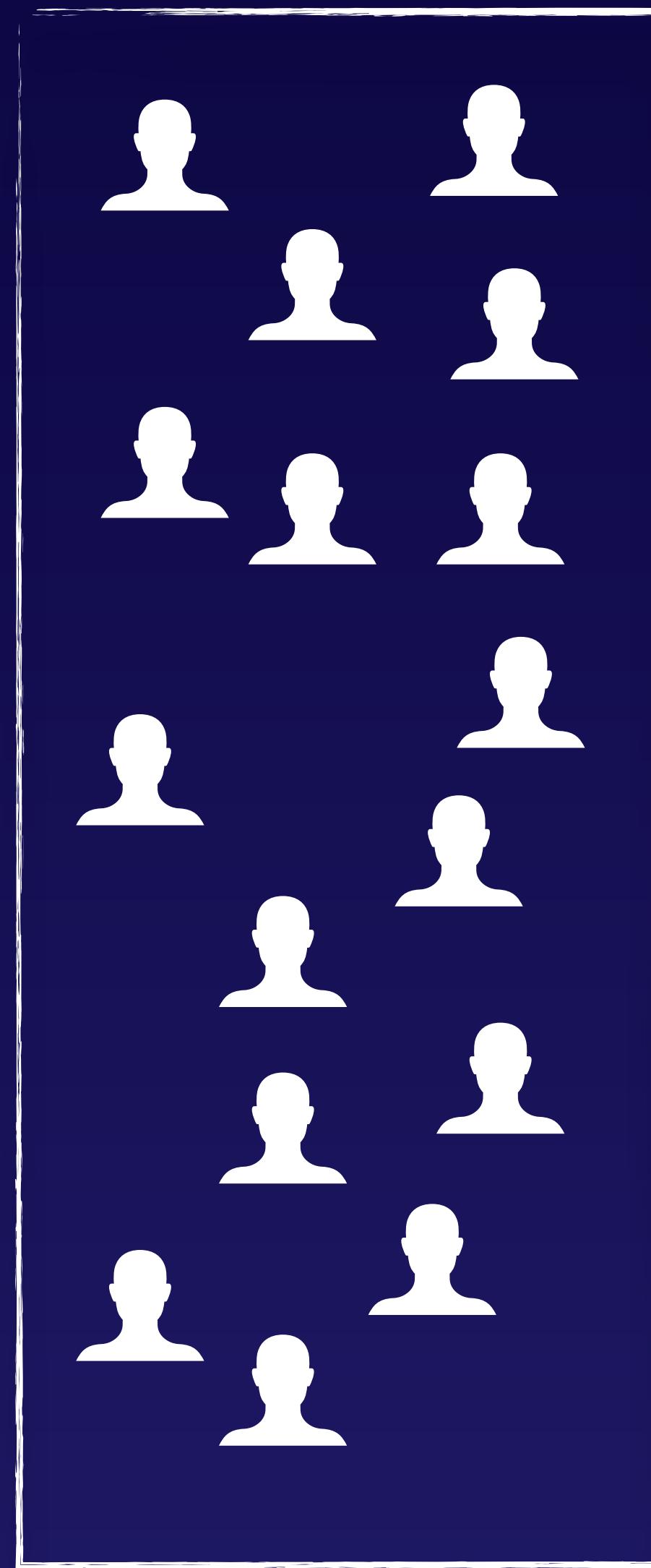
Data Producer



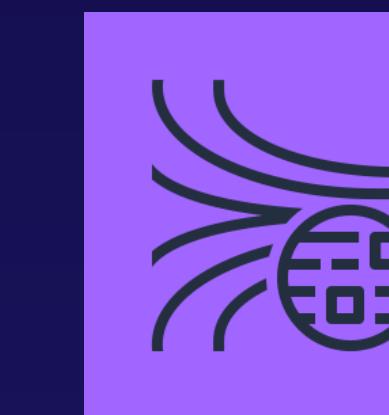
Kinesis Data Streams

Streaming Data Collection Lab

Data Producer



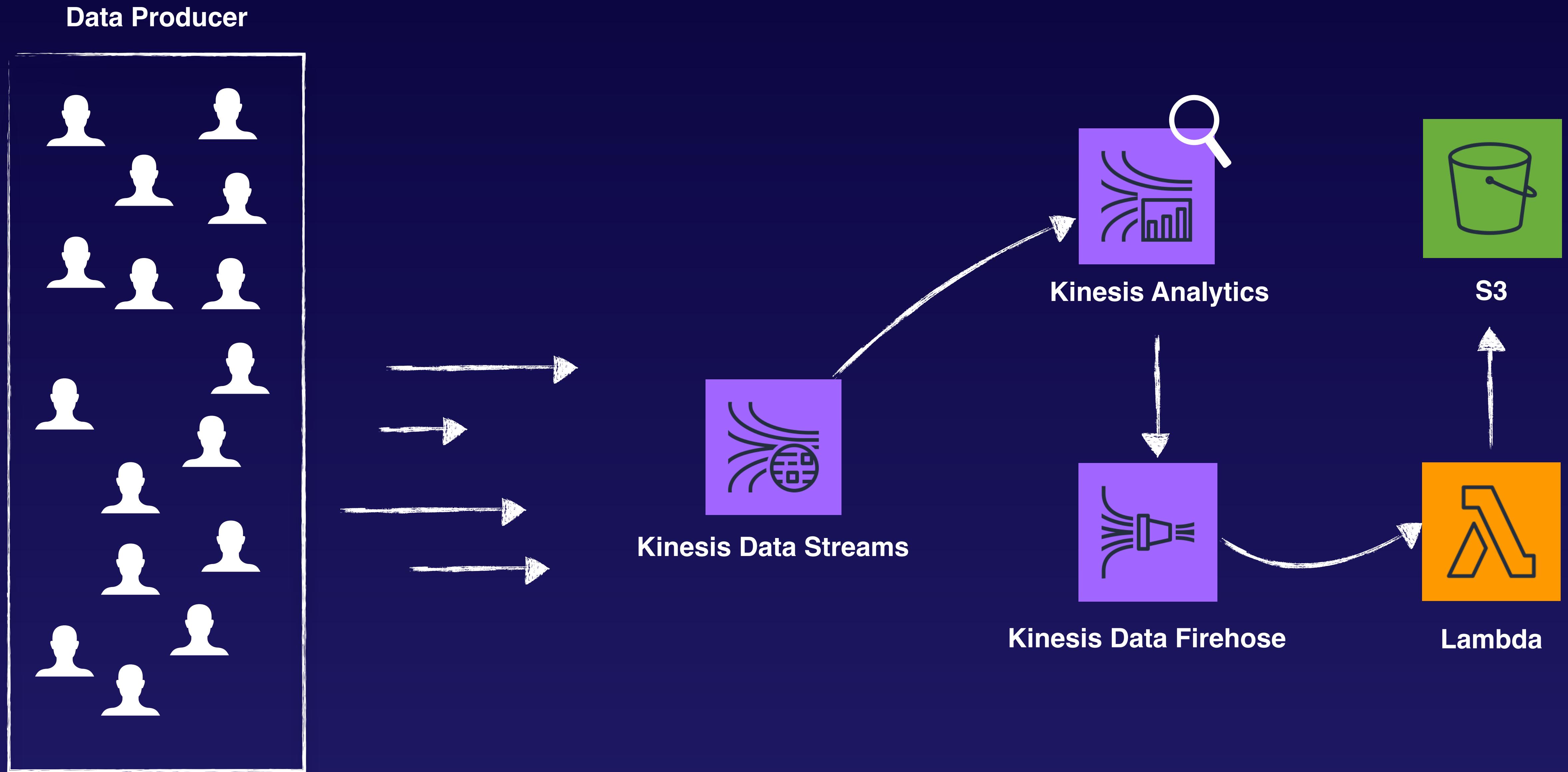
Kinesis Data Streams



Kinesis Analytics



Streaming Data Collection Lab





Beginner

*Follow along with me.
I'll show you the ways.*

