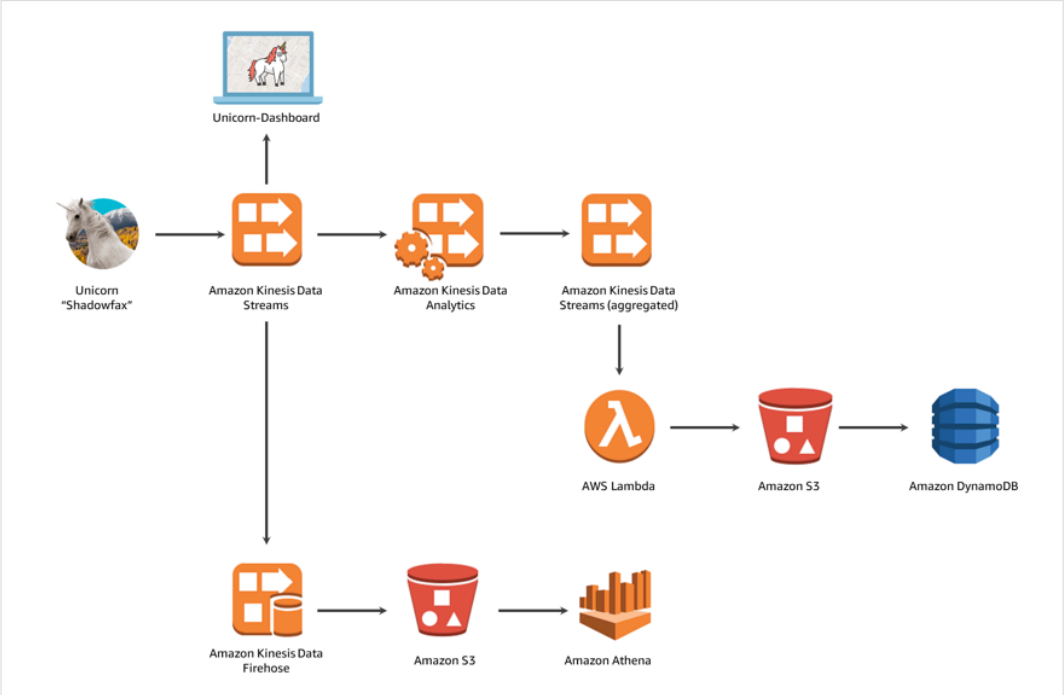
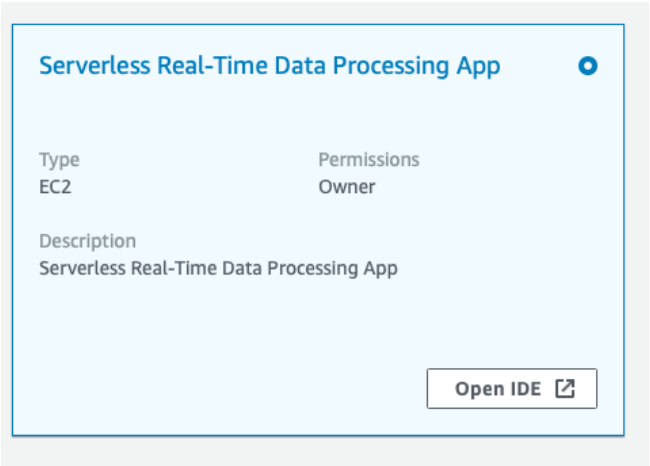
Build a Serverless Real-Time Data Processing App

Intro

Application Architecture



Step 1: Set up your AWS Cloud9 IDE



```
bash - "ip-172-31×    Immediate  ×
ec2-user:~/environment $ aws sts get-caller identiy
ec2-user:~/environment $ aws sts get-caller identiy
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: argument operation: Invalid choice, valid choices are:

assume-role                              | assume-role-with-saml
assume-role-with-web-identity            | decode-authorization-message
get-caller-identity                      | get-federation-token
get-session-token                        | help
ec2-user:~/environment $
```

Step 2: Set up the Command Line Clients

Download and unpack the command line clients by running the following command in the Cloud9 terminal:

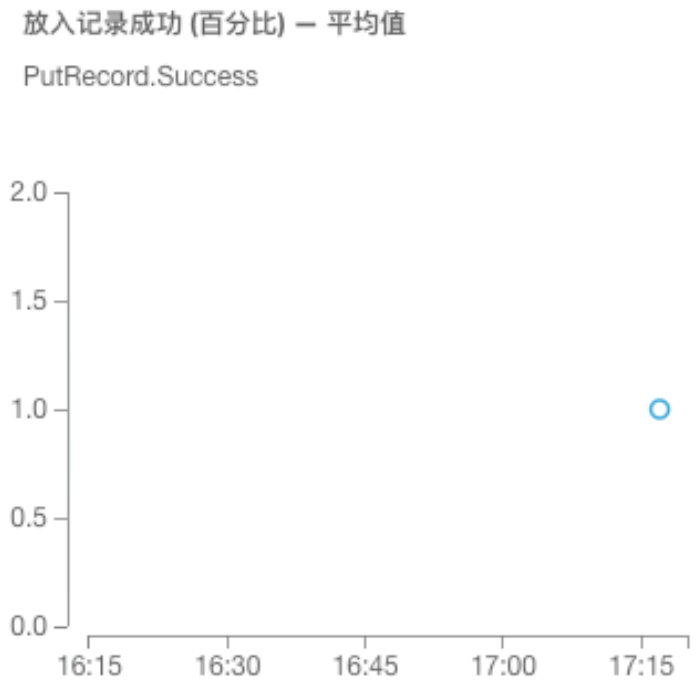
Build a data stream

In this module, you'll create an Amazon Kinesis stream to collect and store sensor data from our unicorn fleet.

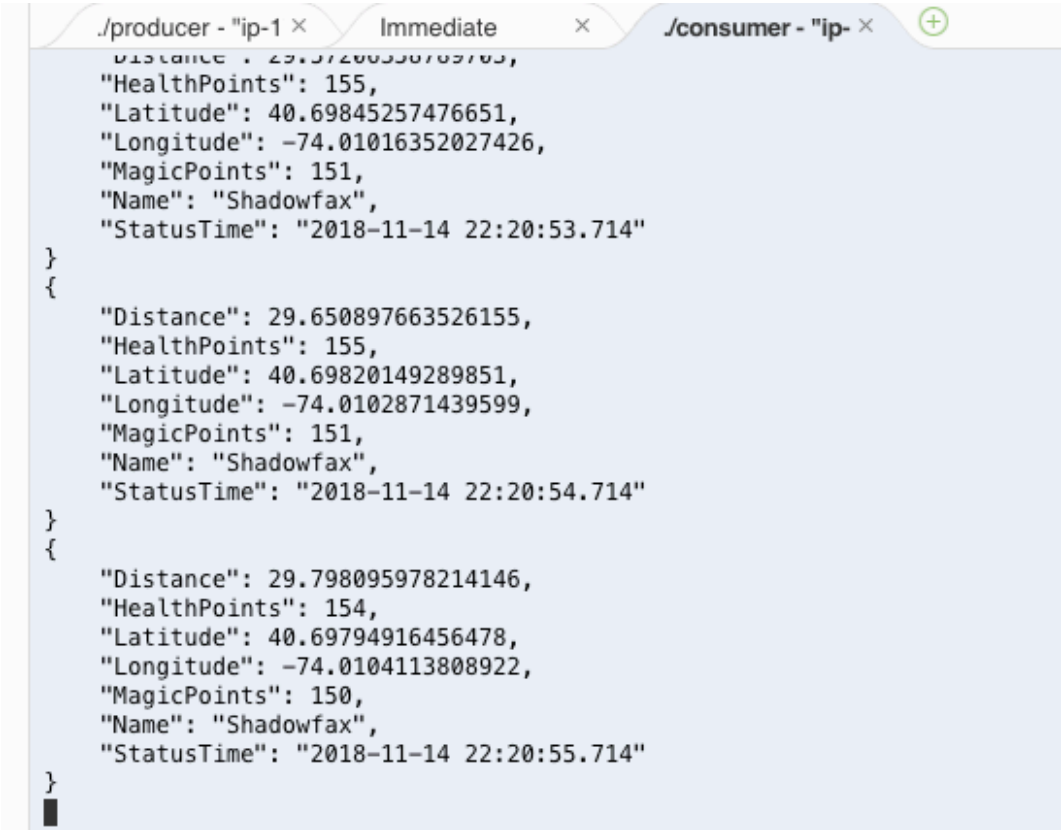
```
get-session-token                        | help
ec2-user:~/environment $ curl -s curl -s https://dataprocessing.wildrydes.com/client/client.tar | tar -xv
producer
consumer
ec2-user:~/environment $
```

Step 1: Create an Amazon Kinesis stream

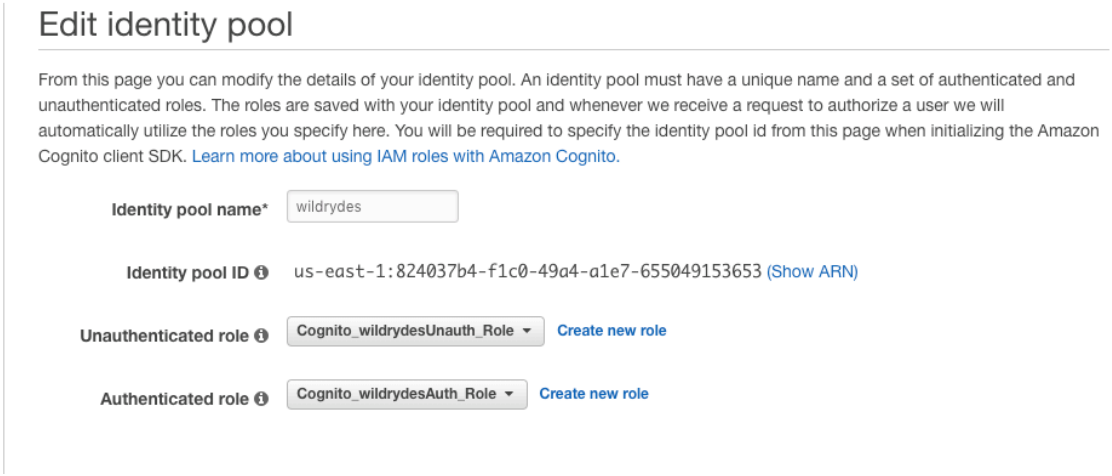
Step 2: Produce messages into the stream



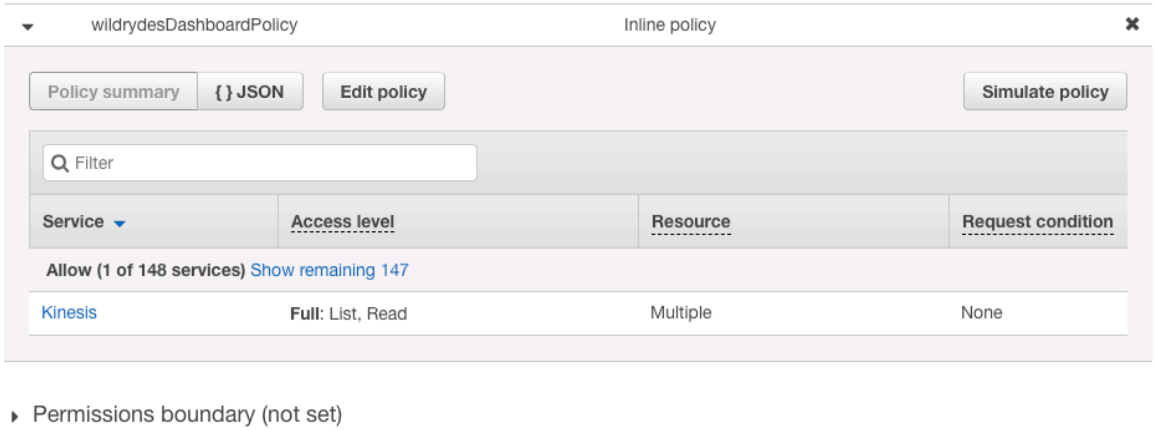
Step 3: Read messages from the stream



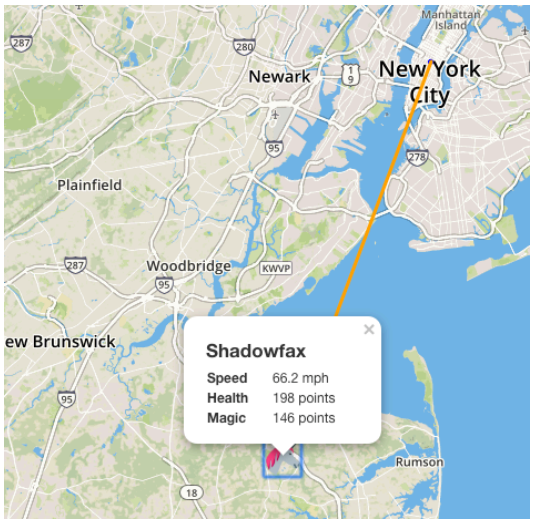
Step 4: Create an identity poll for the unicorn dashboard



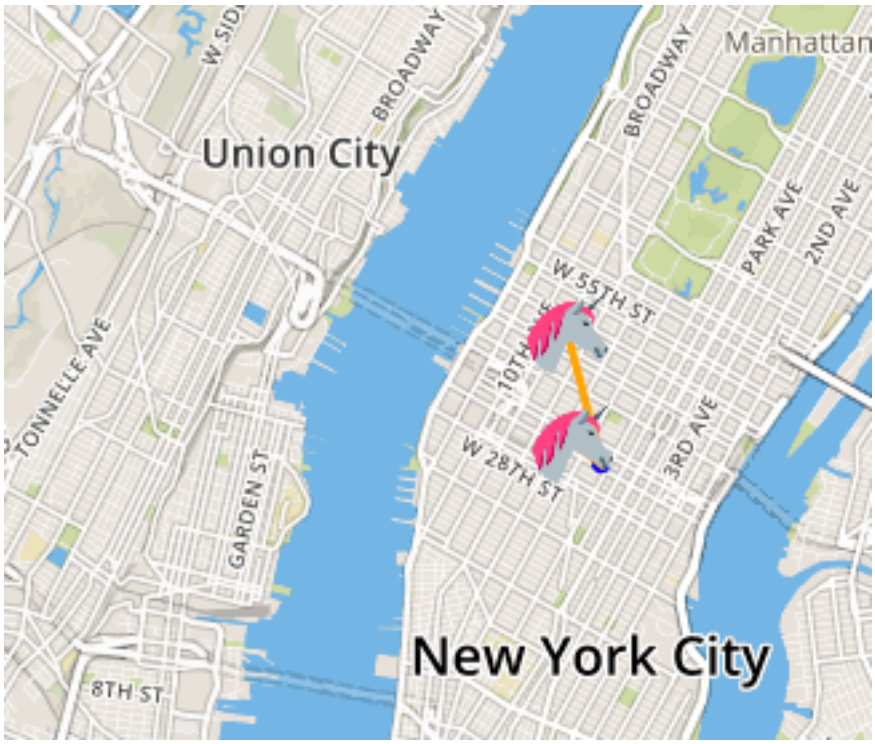
Step 5: Grant the unauthenticated role access to the stream



Step 6: View unicorn status on the dashboard



Step 7: Experiment with the producer



Aggregate data

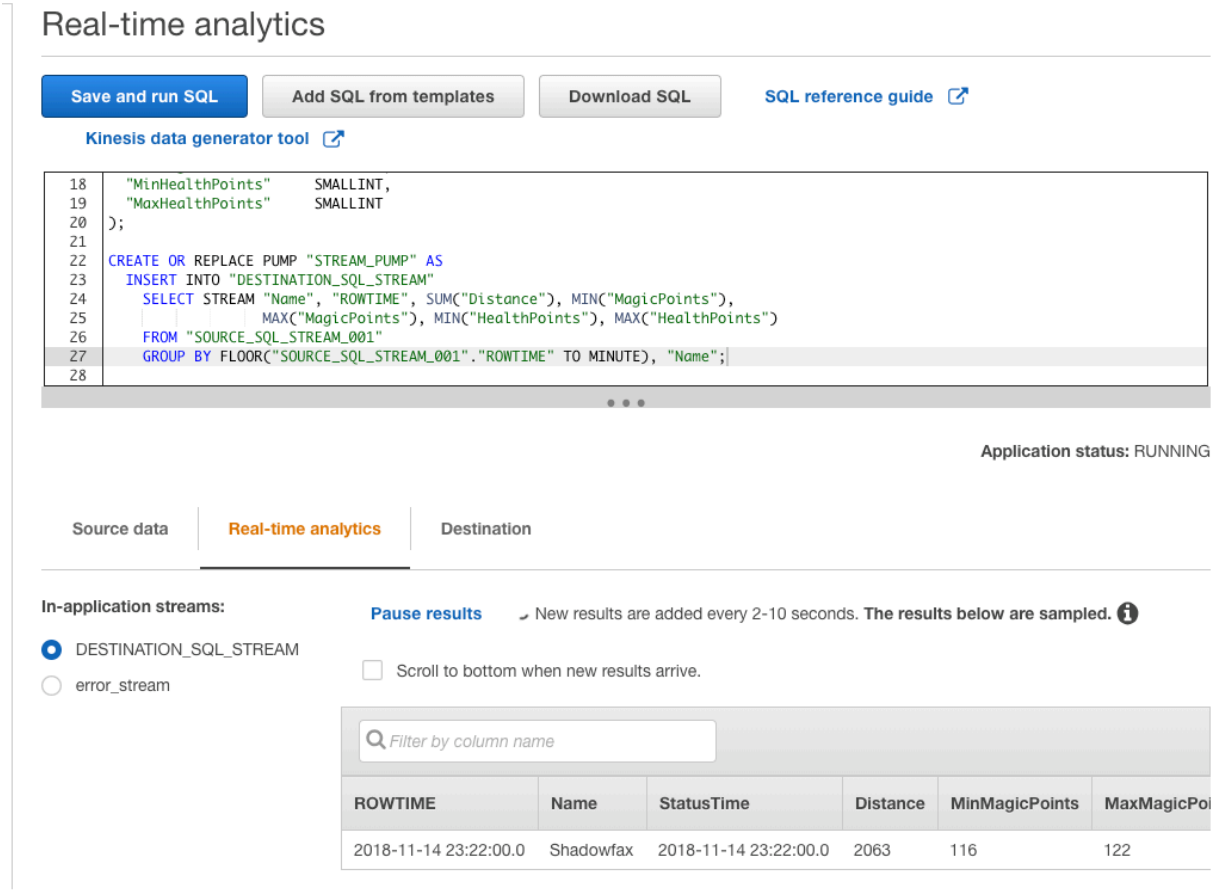
In this module, you'll create an Amazon Kinesis Data Analytics application to aggregate sensor data from the unicorn fleet in real-time.

Step 1: Create an Amazon Kinesis stream

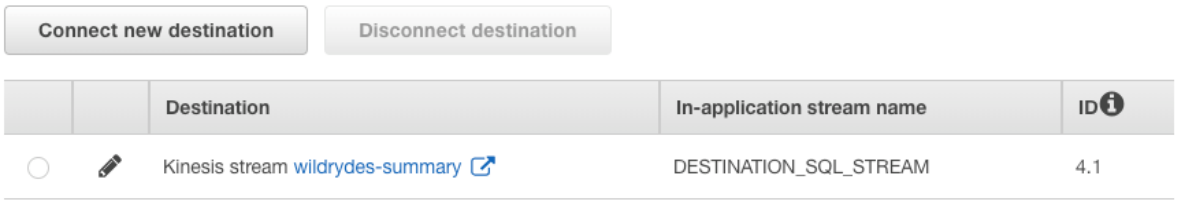
☐	wildrydes-summary	1	Active	0

Step 2: Create an Amazon Kinesis Data Analytics application

Real-time analytics



Connect to a destination



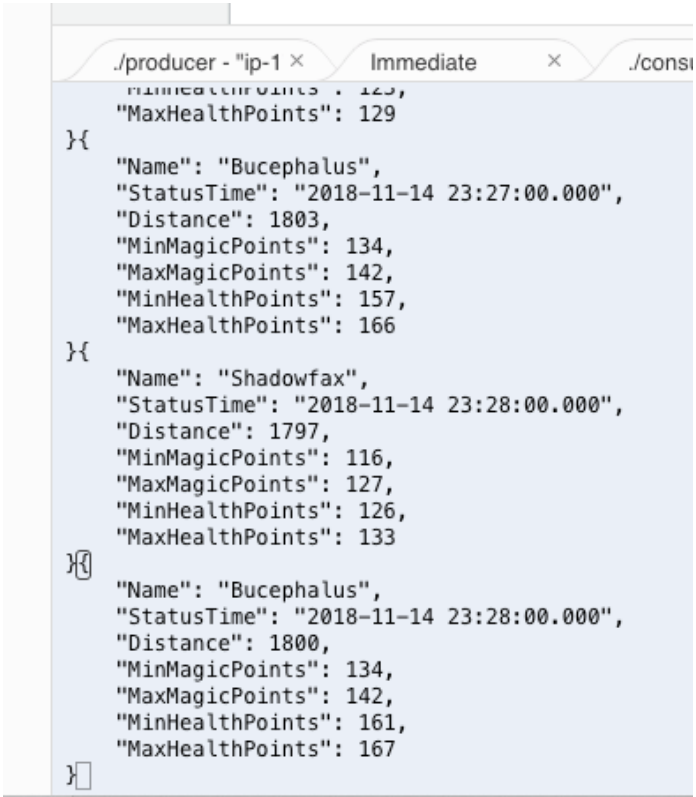
Step 3: Read messages from the stream

Use the command line consumer to view messages from the Kinesis stream to see the aggregated data being sent every minute.

```
ec2-user:~/environment $ ./consumer -stream wildrydes-summary
{
    "Name": "Shadowfax",
    "StatusTime": "2018-11-14 23:27:00.000",
    "Distance": 1794,
    "MinMagicPoints": 118,
    "MaxMagicPoints": 125,
    "MinHealthPoints": 123,
    "MaxHealthPoints": 129
}{
    "Name": "Bucephalus",
    "StatusTime": "2018-11-14 23:27:00.000",
    "Distance": 1803,
    "MinMagicPoints": 134,
    "MaxMagicPoints": 142,
    "MinHealthPoints": 157,
    "MaxHealthPoints": 166
}
```

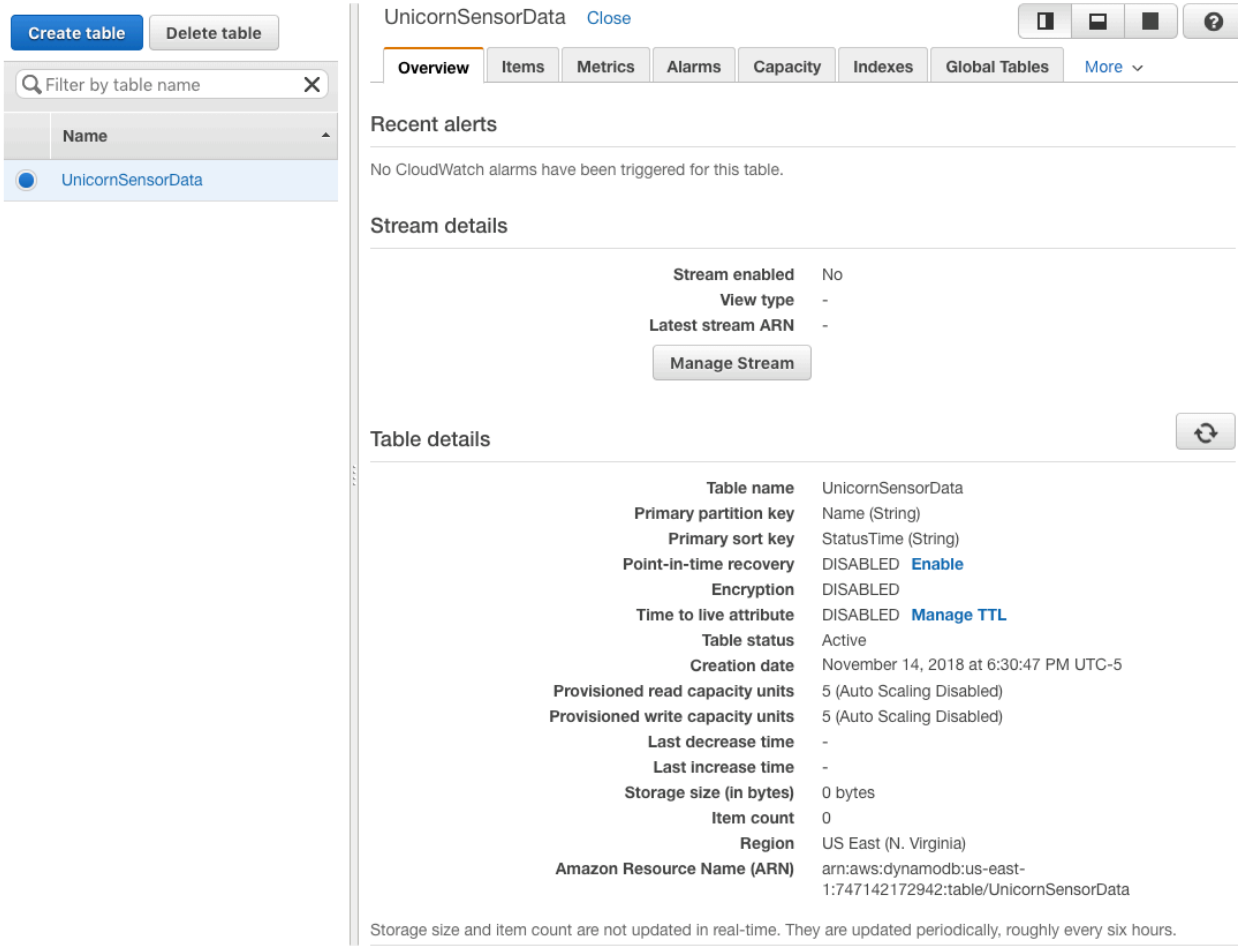
Step 4: Experiment with the producer

Multiple unicorn



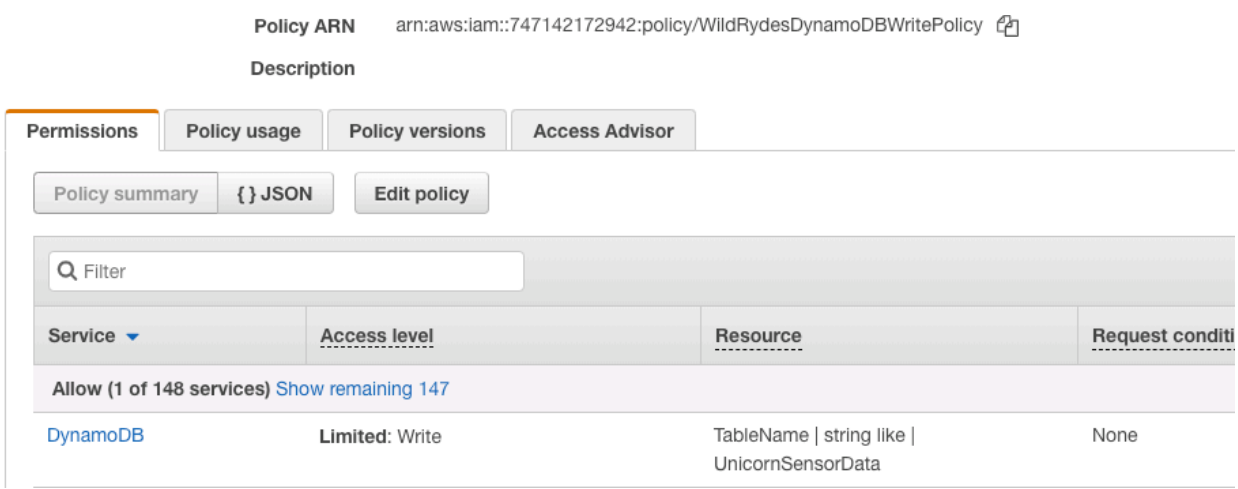
Process streaming data

Step 1: Create an Amazon DynamoDB tables

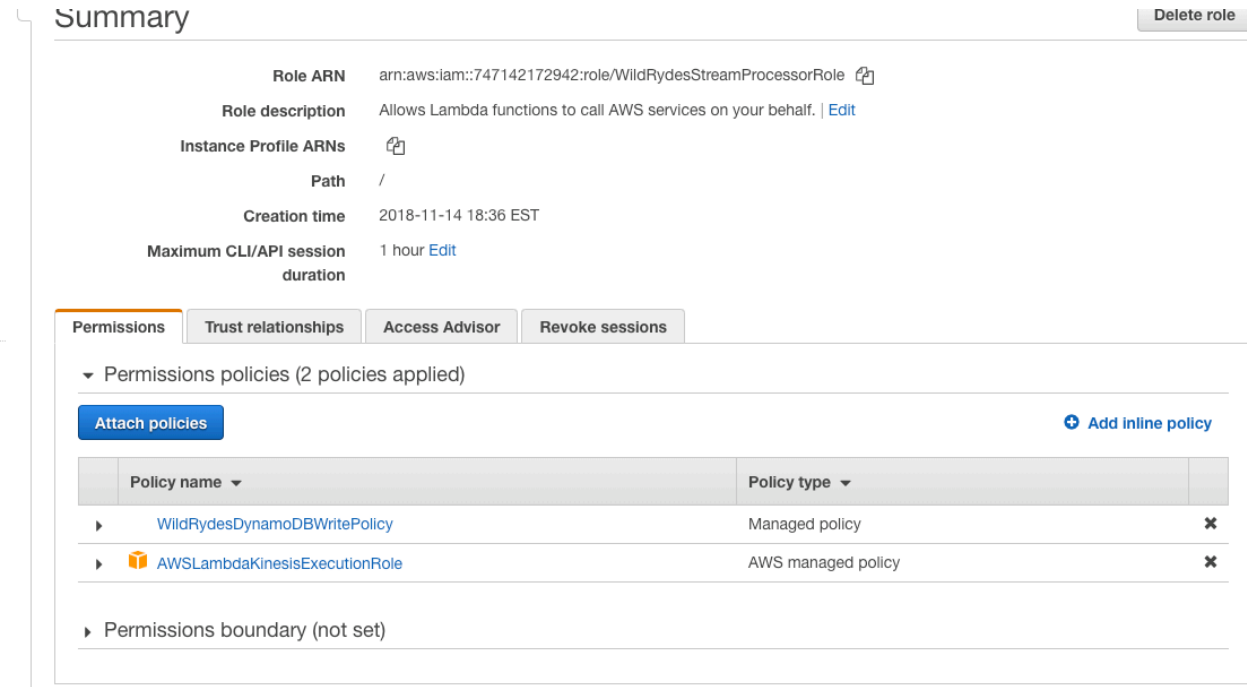


Step 2: Create an IAM role for your Lambda function

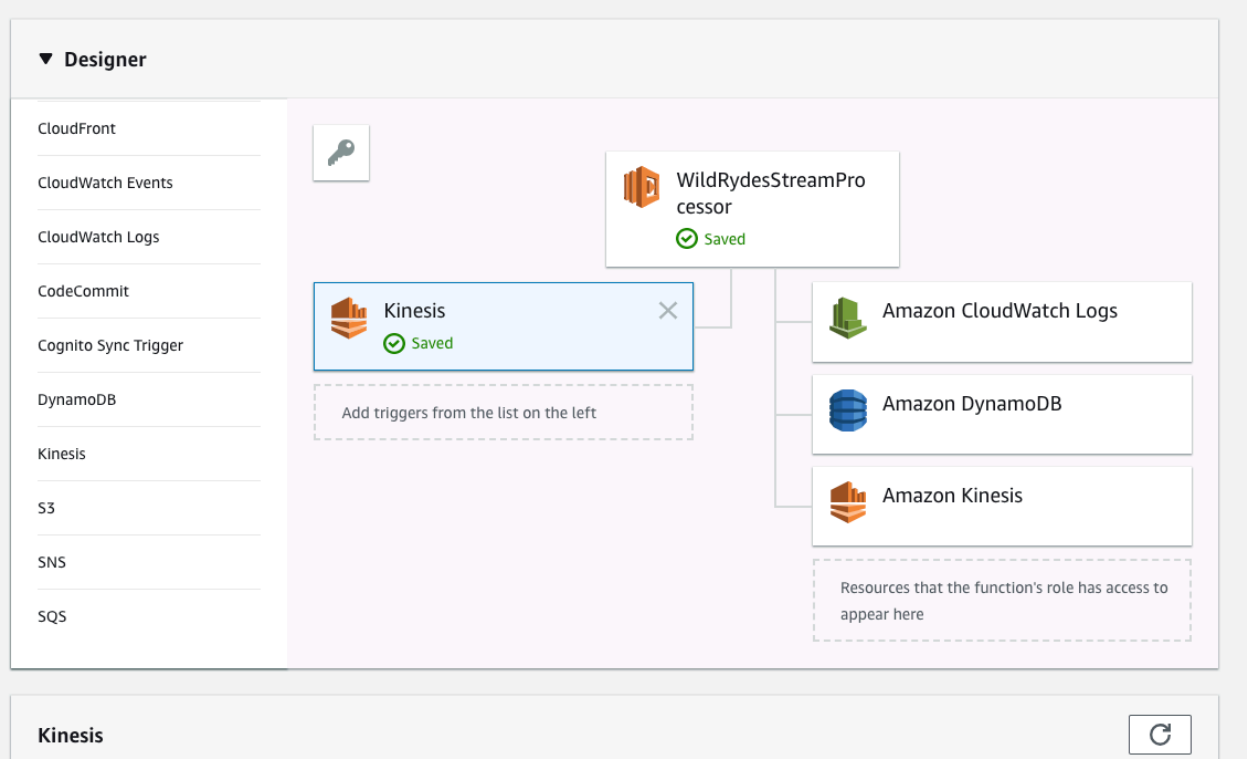
Create policy



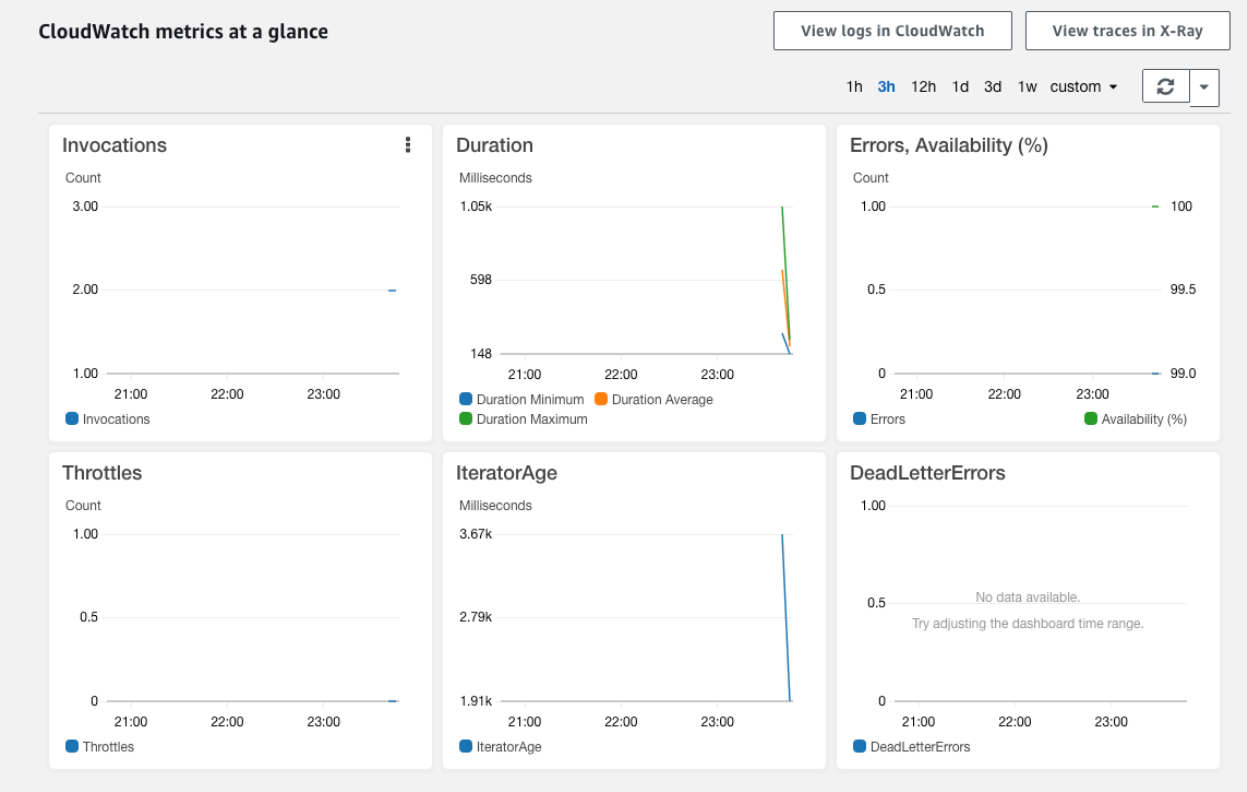
Create role



Step 3: Create a Lambda function to process the stream

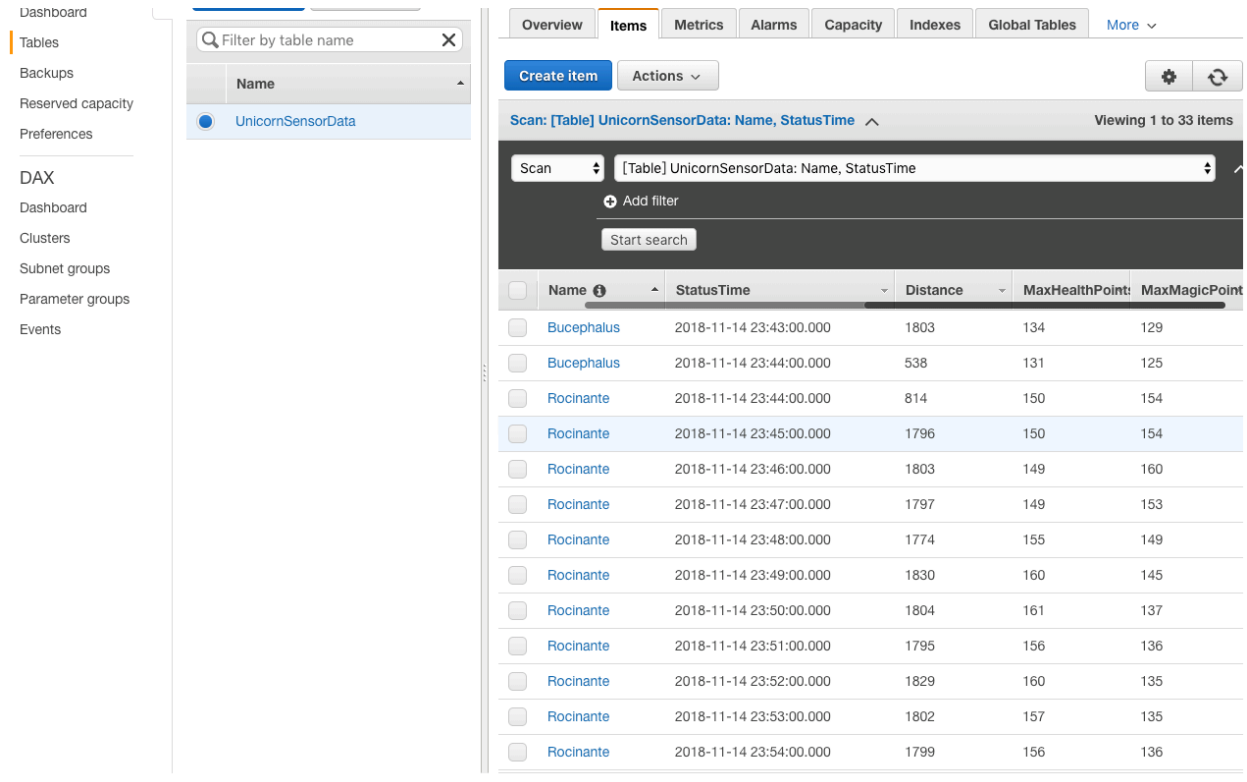


Step 4: Monitor the Lambda function

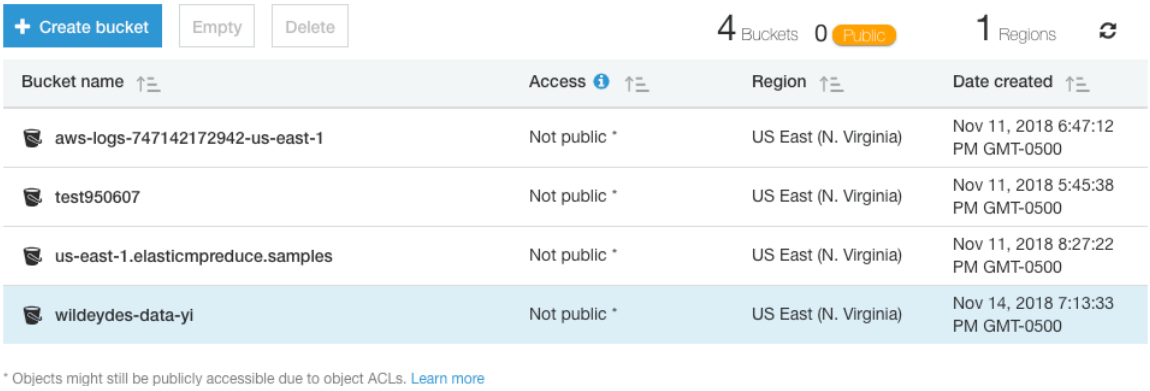


Step 5: Query the DynamoDB table

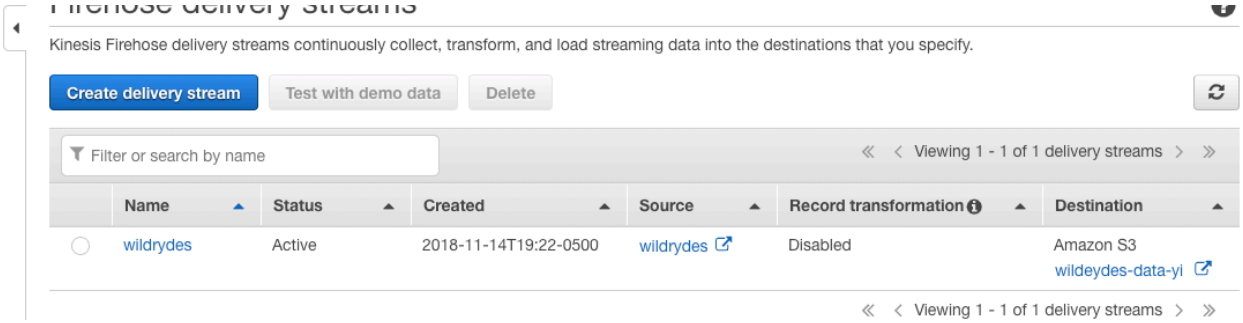
Store & query data



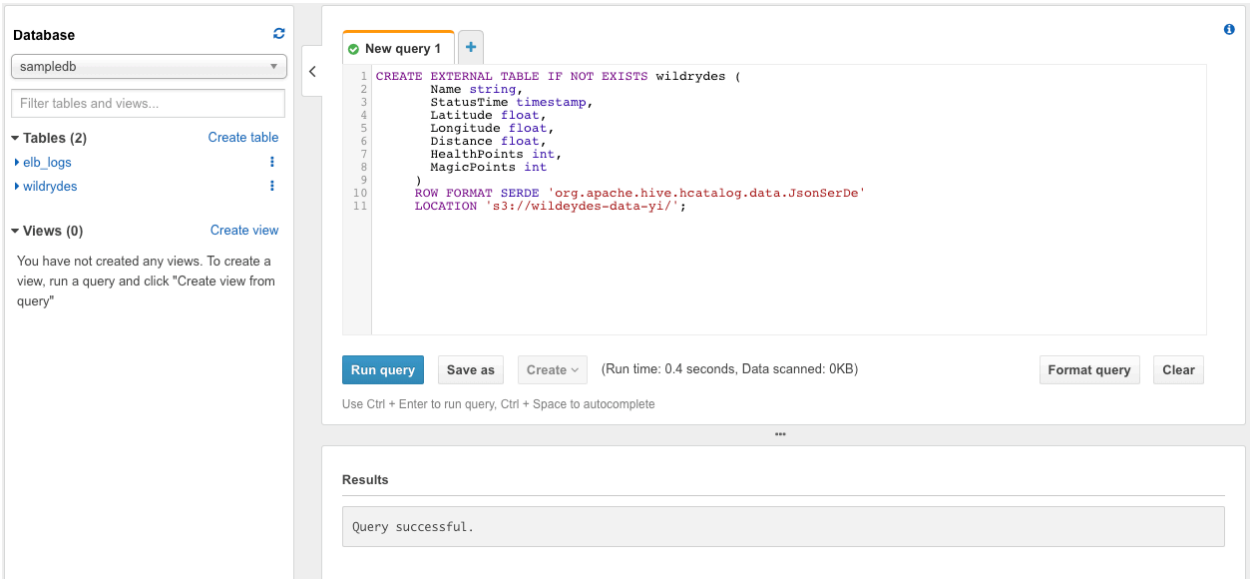
Step 1: Create an Amazon S3 bucket



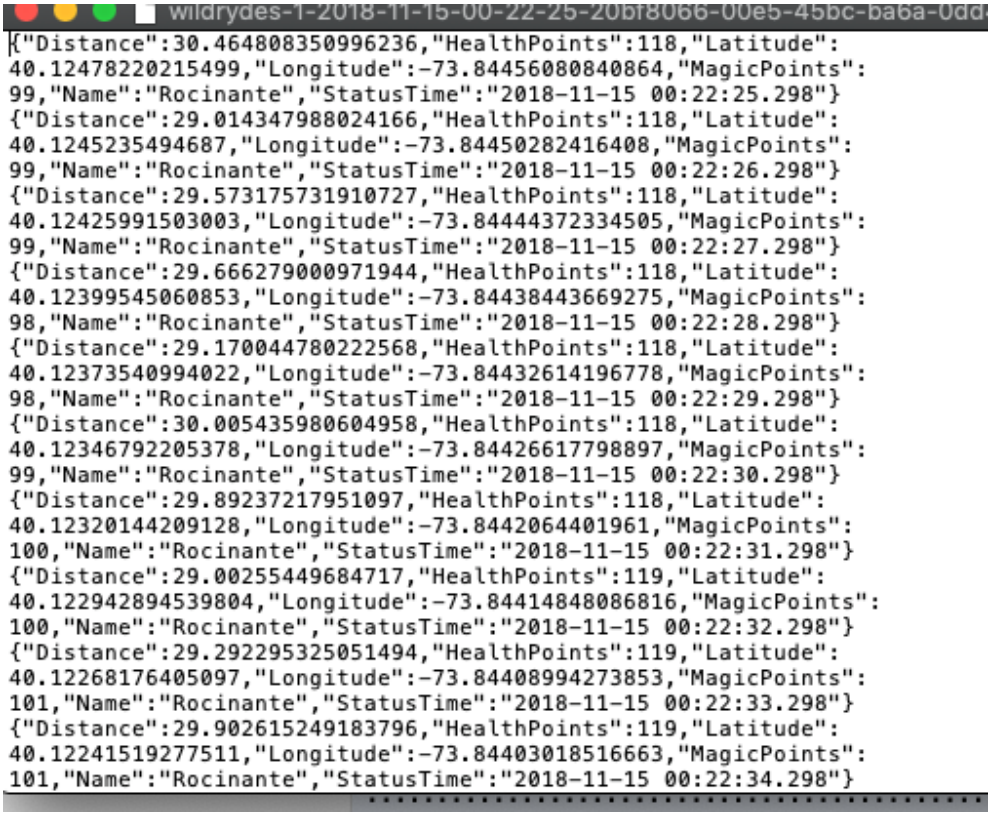
Step 2: Create an Amazon Kinesis Data Firehose delivery stream



Step 3: Create an Amazon Athena table



Step 4: Explore the batched data files



Step 5: Query the data file