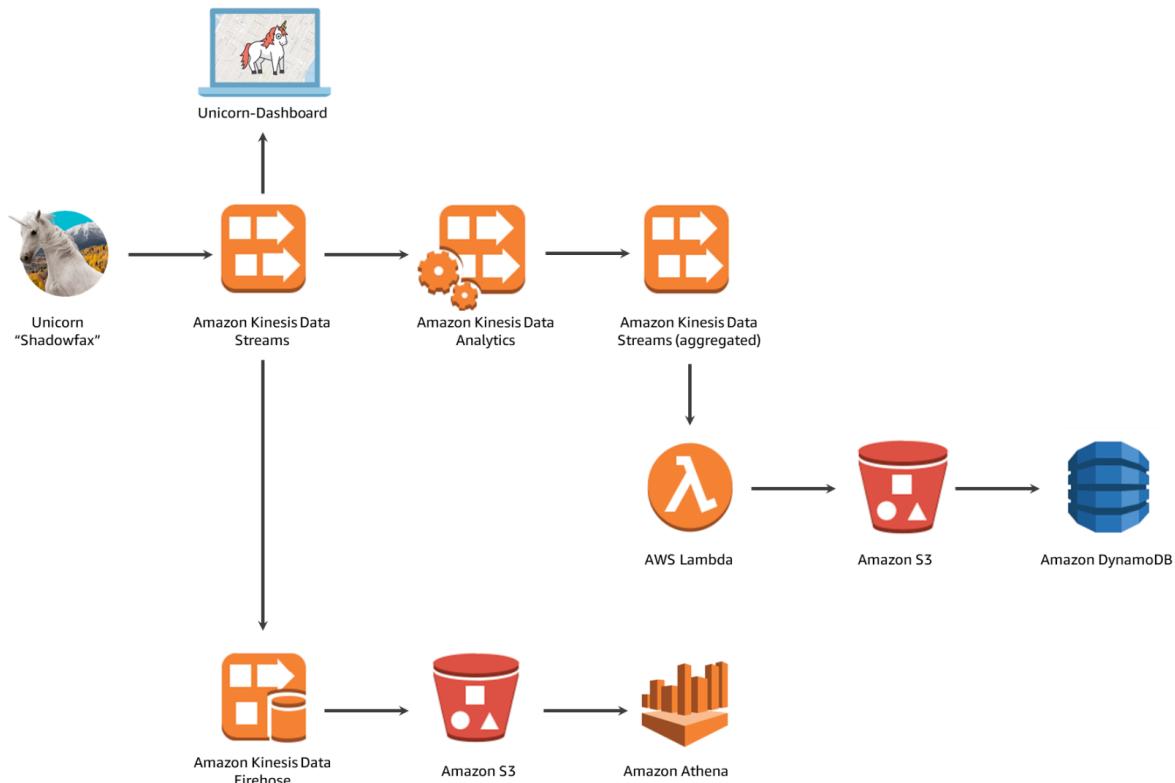


# Build a Serverless Real-Time Data Processing App

Time spent: 150min

## Application Overview



## Related components:

- AWS Lambda
- Amazon Kinesis
- Amazon S3
- Amazon DynamoDB
- Amazon Cognito
- Amazon Athena

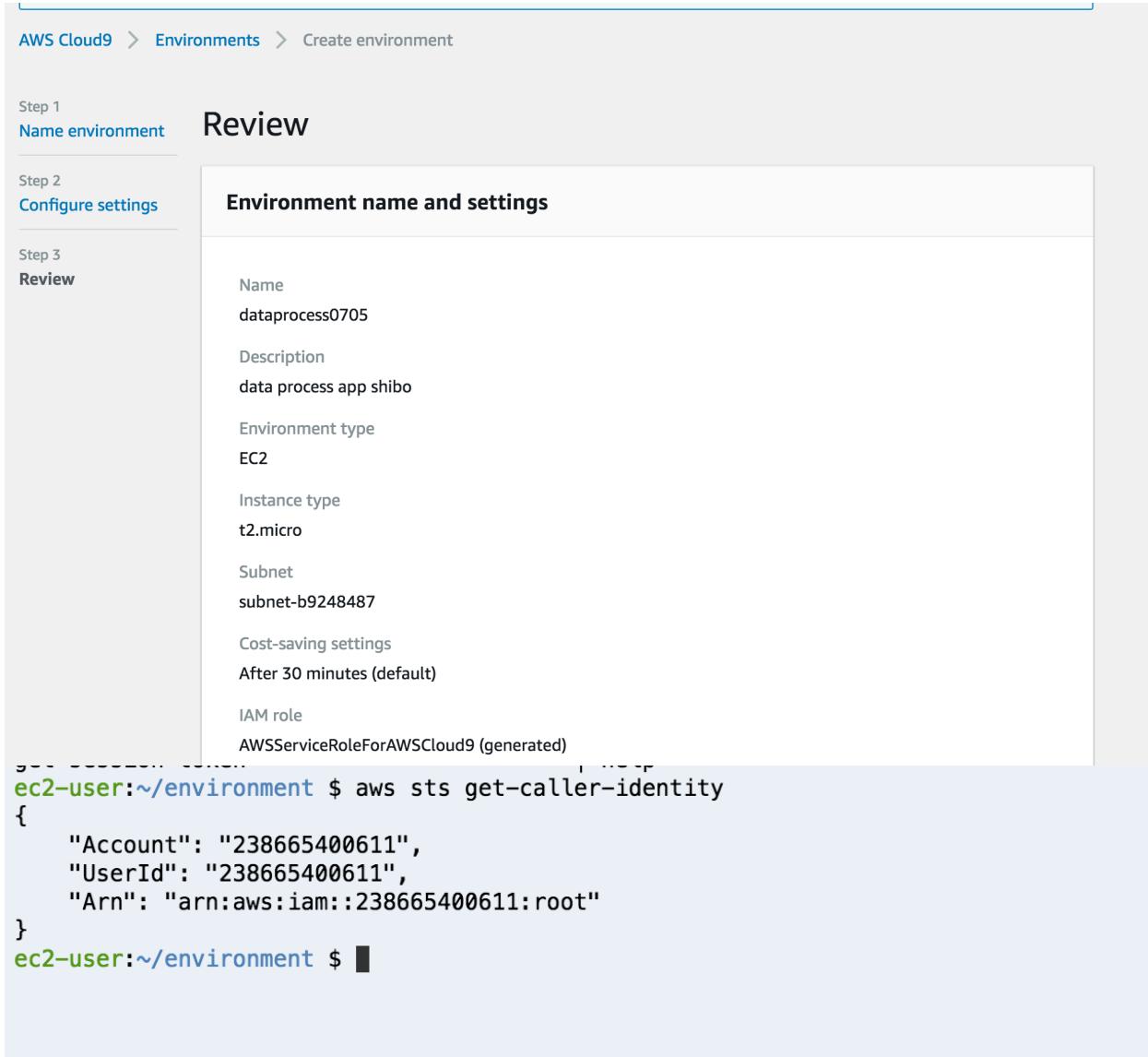
## What I learned:

- How to build a real-time streaming data by Kinesis
- How to use Amazon Cloud9 IDE
- How to create new role in Amazon IAM and new policy
- How to build data analytics application on Kinesis

- How to read the message by application on Kinesis
- How to build the Lambda function
- Use Amazon Athena table to query raw data

## What I built:

- Initiate the Cloud9 environment



AWS Cloud9 > Environments > Create environment

Step 1  
Name environment

Step 2  
Configure settings

Step 3  
Review

**Environment name and settings**

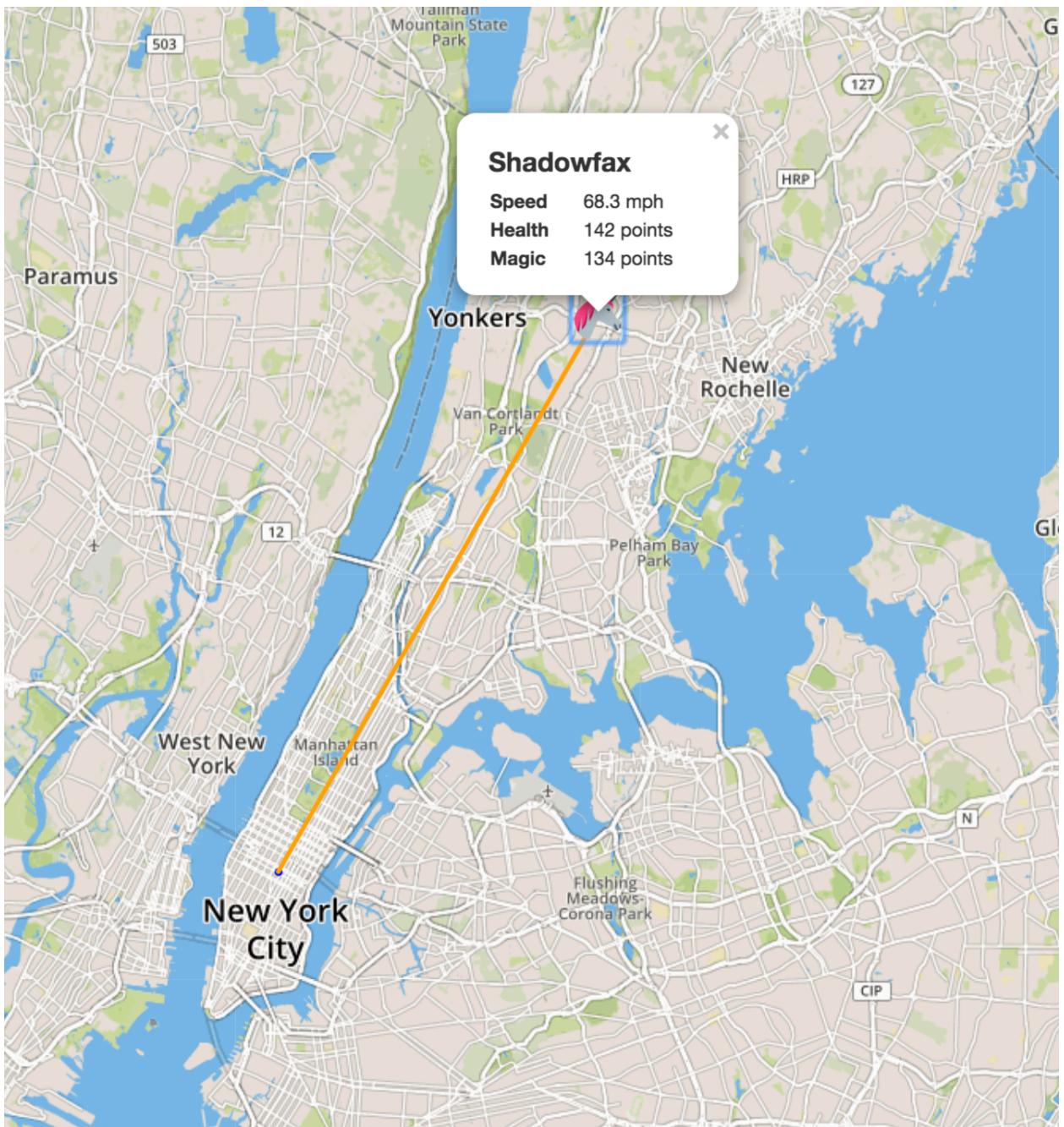
Name	dataprocess0705
Description	data process app shibo
Environment type	EC2
Instance type	t2.micro
Subnet	subnet-b9248487
Cost-saving settings	After 30 minutes (default)
IAM role	AWSServiceRoleForAWSCloud9 (generated)

```
ec2-user:~/environment $ aws sts get-caller-identity
{
    "Account": "238665400611",
    "UserId": "238665400611",
    "Arn": "arn:aws:iam::238665400611:root"
}
ec2-user:~/environment $ █
```

- Build the data stream by Kinesis, consumer can get the data

./producer - "ip-1" × /consumer - "ip- × Immediate (Java) × +

```
ec2-user:~/environment $ ./consumer
{
    "Distance": 29.62814292388923,
    "HealthPoints": 140,
    "Latitude": 40.77360506248107,
    "Longitude": -73.96665330080114,
    "MagicPoints": 134,
    "Name": "Shadowfax",
    "StatusTime": "2018-12-09 21:41:09.462"
}
{
    "Distance": 30.25815004835137,
    "HealthPoints": 140,
    "Latitude": 40.77384201256054,
    "Longitude": -73.96647359482967,
    "MagicPoints": 134,
    "Name": "Shadowfax",
    "StatusTime": "2018-12-09 21:41:10.462"
}
{
    "Distance": 29.355050868836315,
    "HealthPoints": 139,
    "Latitude": 40.77407189051492,
    "Longitude": -73.96629925182656,
    "MagicPoints": 135,
    "Name": "Shadowfax",
    "StatusTime": "2018-12-09 21:41:11.462"
}
{
    "Distance": 29.117967584558254,
    "HealthPoints": 140,
    "Latitude": 40.7742999118818,
    "Longitude": -73.96612631628982,
    "MagicPoints": 136,
```



- Create a new authentication role

Permissions   Trust relationships   Tags   Access Advisor   Revoke sessions

▼ Permissions policies (2 policies applied)

Attach policies   + Add inline policy

Policy name ▾	Policy type ▾
oneClick_Cognito_wildrydesUnauth_Role_1544391845346	Inline policy <span style="float: right;">✖</span>
wildrydesDashboardPolicy	Inline policy <span style="float: right;">✖</span>

▶ Permissions boundary (not set)

- Create two data stream to get multiple unicorns on the map



- Read message from the stream

```

./producer - "ip-1 × ./consumer - "ip-× Immediate (Java × ./consumer - "ip-× ./producer - "ip-1 ×
ec2-user:~/environment $ ./producer -name Bucephalus
-----
ec2-user:~/environment $ ./consumer -stream wildrydes-summary
{
    "Name": "Shadowfax",
    "StatusTime": "2018-12-09 22:12:00.000",
    "Distance": 1801,
    "MinMagicPoints": 147,
    "MaxMagicPoints": 154,
    "MinHealthPoints": 150,
    "MaxHealthPoints": 157
}[
    "Name": "Shadowfax",
    "StatusTime": "2018-12-09 22:13:00.000",
    "Distance": 1797,
    "MinMagicPoints": 153,
    "MaxMagicPoints": 165,
    "MinHealthPoints": 153,
    "MaxHealthPoints": 160
]{
    "Name": "Bucephalus",
    "StatusTime": "2018-12-09 22:13:00.000",
    "Distance": 809,
    "MinMagicPoints": 146,
    "MaxMagicPoints": 152,
    "MinHealthPoints": 149,
    "MaxHealthPoints": 152
}

```

- Build the Lambda function

**WildRydesStreamProcessor**

Throttle Qualifiers Actions Select a test event... Test Save

**Configuration** Monitoring

**Designer**

CloudFront  
CloudWatch Events  
CloudWatch Logs  
CodeCommit  
Cognito Sync Trigger  
DynamoDB  
Kinesis  
S3  
SNS  
SQS

Kinesis Saved

WildRydesStreamProcessor Saved

Layers (0)

Amazon CloudWatch Logs  
Amazon DynamoDB  
Amazon Kinesis

Add triggers from the list on the left

Resources that the function's role has access to appear here

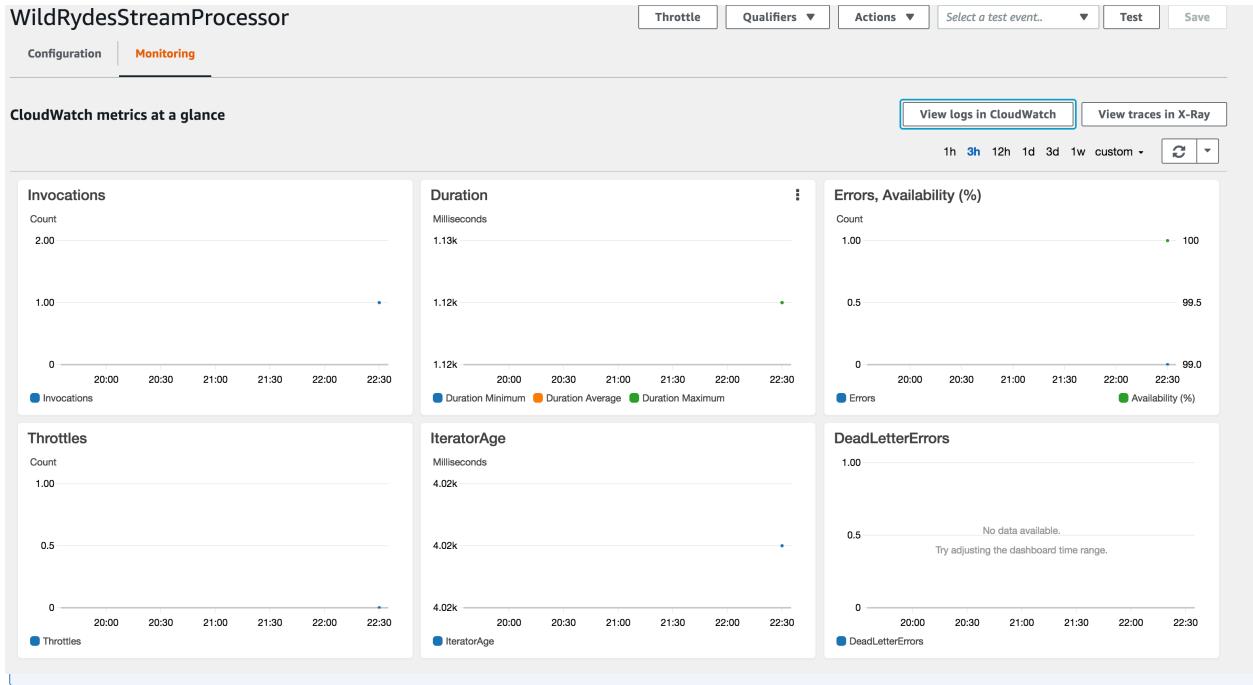
**Kinesis**

wildrydes-summary  
arn:aws:kinesis:us-east-1:238665400611:stream/wildrydes-summary

Batch size: 100 Consumer: NO\_CONSUMER Kinesis stream: kinesis/wildrydes-summary

Creating (~ 1 minute) Delete

- Lambda function result, logs and items in the DB table



CloudWatch > Log Groups > /aws/lambda/WildRydesStreamProcessor > 2018/12/09[\$LATEST]d9bfae61f38e4b5aa03e79841320a8f4

Expand all Row Text

Filter events all 2018-12-08 (22:34:02)

Time (UTC +00:00)	Message
2018-12-09	No older events found at the moment. <a href="#">Retry</a> .
▶ 22:34:02	START RequestId: 9985d824-bc73-4793-bcf5-226e10ec227f Version: \$LATEST
▶ 22:34:04	END RequestId: 9985d824-bc73-4793-bcf5-226e10ec227f
▶ 22:34:04	REPORT RequestId: 9985d824-bc73-4793-bcf5-226e10ec227f Duration: 1124.80 ms Billed Duration: 1200 ms Memory Size: 128 MB Max Memory Used: 38 MB
▶ 22:35:00	START RequestId: faf0f9ba-b572-4963-bf46-04dc1ba4f139 Version: \$LATEST
▶ 22:35:01	END RequestId: faf0f9ba-b572-4963-bf46-04dc1ba4f139
▶ 22:35:01	REPORT RequestId: faf0f9ba-b572-4963-bf46-04dc1ba4f139 Duration: 283.28 ms Billed Duration: 300 ms Memory Size: 128 MB Max Memory Used: 38 MB
	No newer events found at the moment. <a href="#">Retry</a> .

UnicornSensorData Close

Overview	Items	Metrics	Alarms	Capacity	Indexes	Global Tables	Backups	Triggers	Access control	Tags																																								
<a href="#">Create item</a>	<a href="#">Actions</a>																																																	
<p>Scan: [Table] UnicornSensorData: Name, StatusTime ^ Viewing 1 to 4 items</p> <p>Scan [Table] UnicornSensorData: Name, StatusTime ^ Add filter Start search</p> <table border="1"> <thead> <tr> <th></th> <th>Name</th> <th>StatusTime</th> <th>Distance</th> <th>MaxHealthPoint</th> <th>MaxMagicPoints</th> <th>MinHealthPoints</th> <th>MinMagicPoints</th> </tr> </thead> <tbody> <tr> <td><input type="checkbox"/></td> <td>Rocinante</td> <td>2018-12-09 22:34:00.000</td> <td>1468</td> <td>152</td> <td>158</td> <td>147</td> <td>150</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Rocinante</td> <td>2018-12-09 22:35:00.000</td> <td>1807</td> <td>155</td> <td>158</td> <td>146</td> <td>147</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Rocinante</td> <td>2018-12-09 22:36:00.000</td> <td>1801</td> <td>161</td> <td>153</td> <td>156</td> <td>146</td> </tr> <tr> <td><input type="checkbox"/></td> <td>Rocinante</td> <td>2018-12-09 22:37:00.000</td> <td>1807</td> <td>162</td> <td>152</td> <td>155</td> <td>142</td> </tr> </tbody> </table>												Name	StatusTime	Distance	MaxHealthPoint	MaxMagicPoints	MinHealthPoints	MinMagicPoints	<input type="checkbox"/>	Rocinante	2018-12-09 22:34:00.000	1468	152	158	147	150	<input type="checkbox"/>	Rocinante	2018-12-09 22:35:00.000	1807	155	158	146	147	<input type="checkbox"/>	Rocinante	2018-12-09 22:36:00.000	1801	161	153	156	146	<input type="checkbox"/>	Rocinante	2018-12-09 22:37:00.000	1807	162	152	155	142
	Name	StatusTime	Distance	MaxHealthPoint	MaxMagicPoints	MinHealthPoints	MinMagicPoints																																											
<input type="checkbox"/>	Rocinante	2018-12-09 22:34:00.000	1468	152	158	147	150																																											
<input type="checkbox"/>	Rocinante	2018-12-09 22:35:00.000	1807	155	158	146	147																																											
<input type="checkbox"/>	Rocinante	2018-12-09 22:36:00.000	1801	161	153	156	146																																											
<input type="checkbox"/>	Rocinante	2018-12-09 22:37:00.000	1807	162	152	155	142																																											

- Query the data in the bucket

Athena Query Editor Saved Queries History AWS Glue Data Catalog

Database sampledb Tables (2) Views (0)

New query 1 New query 2 +  
1 SELECT \* FROM wildrydes

Run query Save as Create (Run time: 1.54 seconds, Data scanned: 113.76 KB) Format query Clear

Results

	name	statustime	latitude	longitude	distance	healthpoints	magicpoints
1	Rocinante	2018-12-09 22:52:47.882	40.4296	-73.96376	29.652407	162	171
2	Rocinante	2018-12-09 22:52:48.882	40.429333	-73.963745	29.566196	162	171
3	Rocinante	2018-12-09 22:52:49.882	40.429058	-73.96373	30.341145	163	172
4	Rocinante	2018-12-09 22:52:50.882	40.428795	-73.96371	29.16852	162	171
5	Rocinante	2018-12-09 22:52:51.882	40.42852	-73.96369	30.441355	161	172
6	Rocinante	2018-12-09 22:52:52.882	40.428257	-73.96368	29.009014	162	173
7	Rocinante	2018-12-09 22:52:53.882	40.427994	-73.96365	29.315641	162	172

## Summary

In this project, I learned the three main functions of Amazon Kinesis: Create data stream, analytics application and data firehose. Besides, I learned use Amazon IAM to create different roles and how to add different policies to it, which is important for cloud services. After the combination of them, I finished this serverless real-time data processing application.