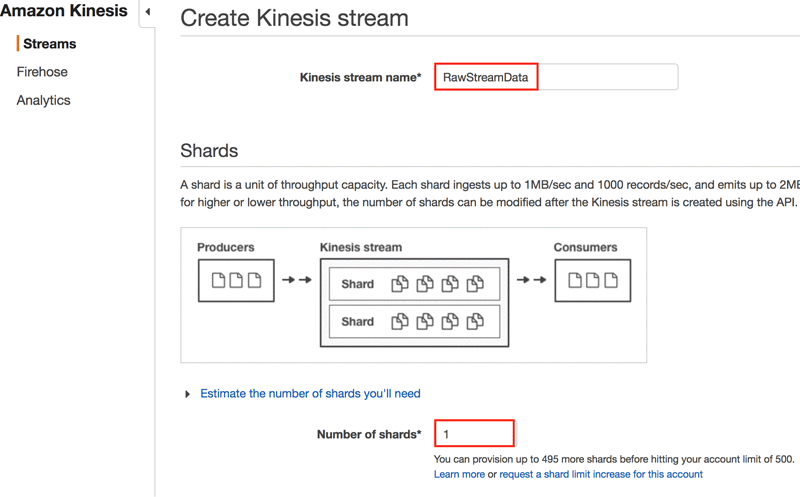
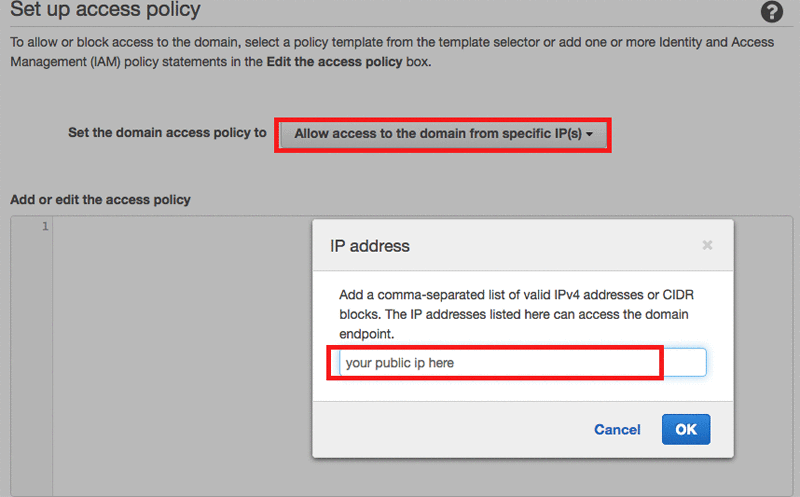
**Steps to perform:**

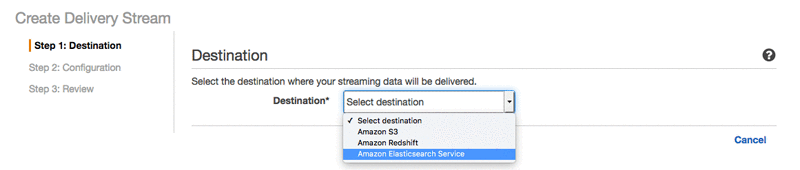
1. **Create Kinesis delivery stream**
2. Open the Amazon Kinesis Streams console
3. Create a new Kinesis stream. Give it a name that indicates it’s for raw incoming stream data—for example, RawStreamData. For Number of shards, type 1



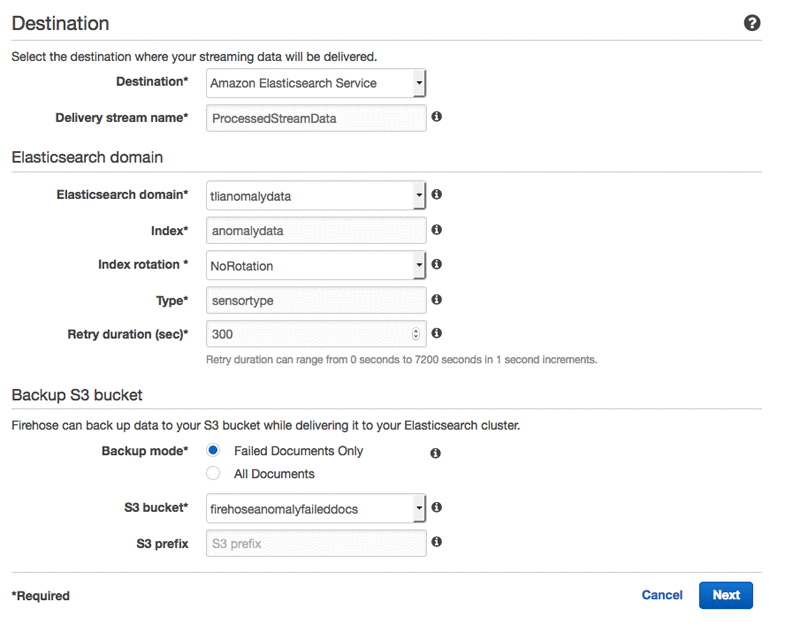
1. **Stream the data to Kinesis Data Stream**
2. Open Command Prompt and Login with AWS credentials as shown in capture.png
3. Run the New.py
4. **Open the Amazon Elasticsearch Service console and create a new domain**.
5. Give a unique name to the domain.
6. In the Configure cluster screen, use the default settings
7. In the Set up access policy screen, choose Allow access to the domain from specific IP(s)
8. Enter the public IPv4 address of your computer and paste into space and click Allow



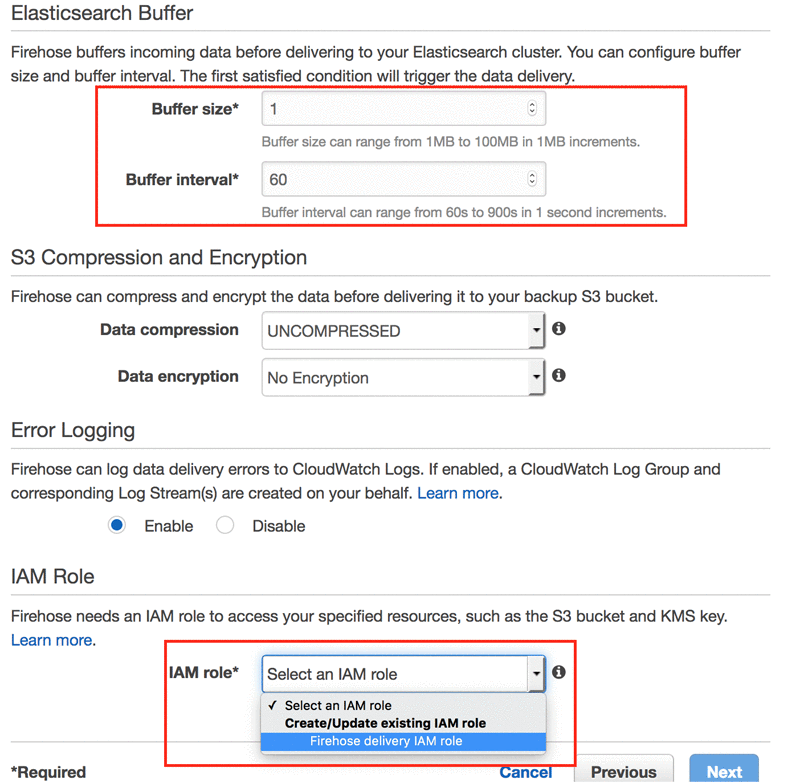
1. **Configure Kinesis Firehose to export the results to Amazon ES**
2. Open the Amazon Kinesis Firehose console and choose Create Delivery Stream
3. In the Destination dropdown list, choose Amazon Elasticsearch Service



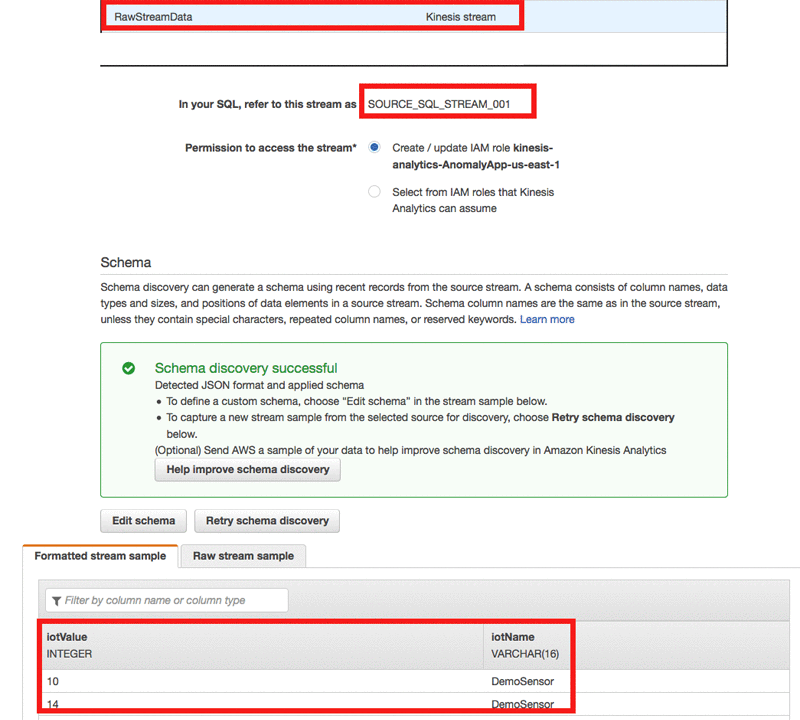
1. Type a stream name, and choose the Amazon ES domain
2. Provide an index name and ES type. In the S3 bucket dropdown list, choose Create New S3 bucket. Choose Next.



1. Under IAM Role, choose Create/Update existing IAM role.
2. Choose Next to move to the Review page.



1. Review the configuration, and then choose Create Delivery Stream.
2. **Open the Amazon Kinesis Analytics**
3. Open console and create a new application. Give the application a name, and then choose Create Application.
4. On the next screen, choose Connect to a source. Choose the raw incoming data stream that you created earlier. (Note the stream name Source\_SQL\_STREAM\_001 because you will need it later.)
5. Use the default settings for everything else. When the schema discovery process is complete, it displays a success message with the formatted stream sample in a table as shown in the following screenshot. Review the data, and then choose Save and continue.



1. Next, choose Go to SQL editor. When prompted, choose Yes, start application.

Copy the following SQL code and paste it into the SQL editor window.

CREATE OR REPLACE STREAM "TEMP\_STREAM" (

"iotName" varchar (40),

"iotValue" integer,

"ANOMALY\_SCORE" DOUBLE);

-- Creates an output stream and defines a schema

CREATE OR REPLACE STREAM "DESTINATION\_SQL\_STREAM" (

"iotName" varchar(40),

"iotValue" integer,

"ANOMALY\_SCORE" DOUBLE,

"created" TimeStamp);

-- Compute an anomaly score for each record in the source stream

-- using Random Cut Forest

CREATE OR REPLACE PUMP "STREAM\_PUMP\_1" AS INSERT INTO "TEMP\_STREAM"

SELECT STREAM "iotName", "iotValue", ANOMALY\_SCORE FROM

TABLE(RANDOM\_CUT\_FOREST(

CURSOR(SELECT STREAM \* FROM "SOURCE\_SQL\_STREAM\_001")

)

);

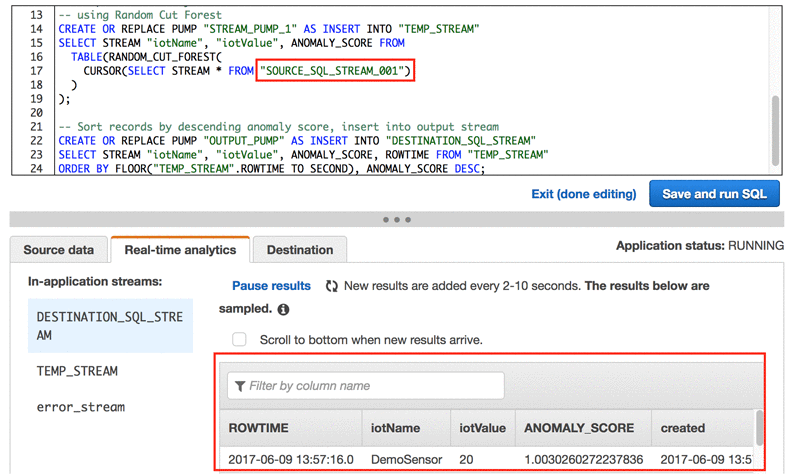
-- Sort records by descending anomaly score, insert into output stream

CREATE OR REPLACE PUMP "OUTPUT\_PUMP" AS INSERT INTO "DESTINATION\_SQL\_STREAM"

SELECT STREAM "iotName", "iotValue", ANOMALY\_SCORE, ROWTIME FROM "TEMP\_STREAM"

ORDER BY FLOOR("TEMP\_STREAM".ROWTIME TO SECOND), ANOMALY\_SCORE DESC;

1. Choose Save and run SQL  
   As the application is running, it displays the results as stream data arrives. If you don’t see any data coming in, run the Python script again to generate some fresh data. When there is data, it appears in a grid as shown in the following screenshot.



Note that you are selecting data from the source stream name Source\_SQL\_STREAM\_001 that you created previously. Also note the ANOMALY\_SCORE column. This is the value that the [Random\_Cut\_Forest](http://docs.aws.amazon.com/kinesisanalytics/latest/sqlref/random-cut-forest.html) function calculates based on the temperature ranges provided by the Python script. Higher (anomaly) temperature ranges have a higher score.Looking at the SQL code, note that the first two blocks of code create two new streams to store temporary data and the final result. The third block of code analyzes the raw source data (Stream\_Pump\_1) using the Random\_Cut\_Forest function. It calculates an anomaly score (ANOMALY\_SCORE) and inserts it into the TEMP\_STREAM stream. The final code block loads the result stored in the TEMP\_STREAM into DESTINATION\_SQL\_STREAM.

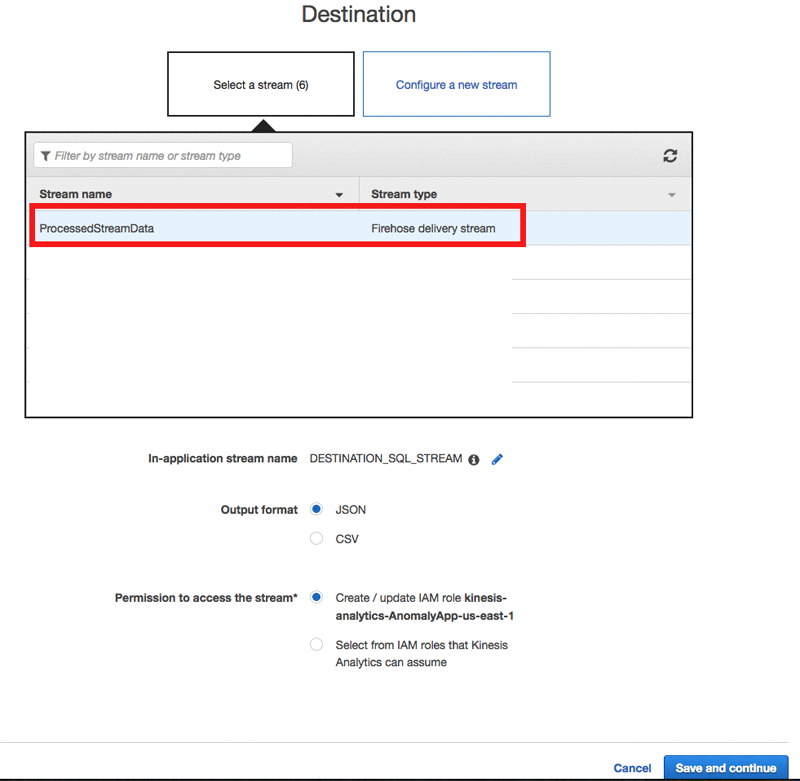
Choose Exit (done editing) next to the Save and run SQL button to return to the application configuration page.

1. Load processed data into the Kinesis Firehose delivery stream

Now, you can export the result from DESTINATION\_SQL\_STREAM into the Amazon Kinesis Firehose stream that you created previously.

On the application configuration page, choose Connect to a destination.

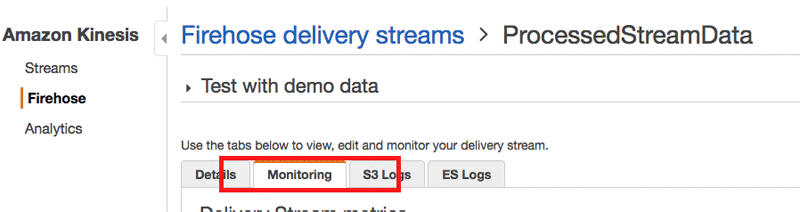
Choose the stream name that you created earlier, and use the default settings foreverything else. Then choose Save and Continue.



On the application configuration page, choose Exit to Kinesis Analytics applications to return to the Amazon Kinesis Analytics console.

Run the Python script again for 4–5 minutes to generate enough data to flow through Amazon Kinesis Streams, Kinesis Analytics, Kinesis Firehose, and finally into the Amazon ES domain.

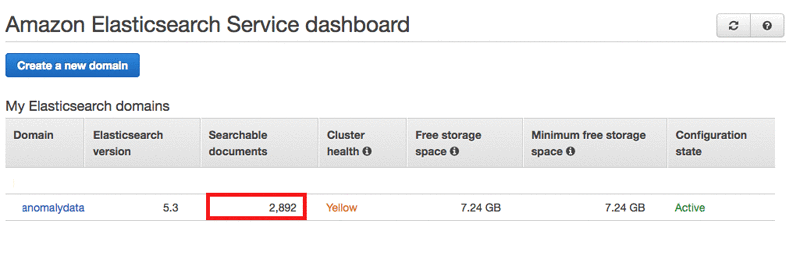
Open the Kinesis Firehose console, choose the stream, and then choose the Monitoring



As the processed data flows into Kinesis Firehose and Amazon ES, the metrics appear on the Delivery Stream metrics page. Keep in mind that the metrics page takes a few minutes to refresh with the latest data.

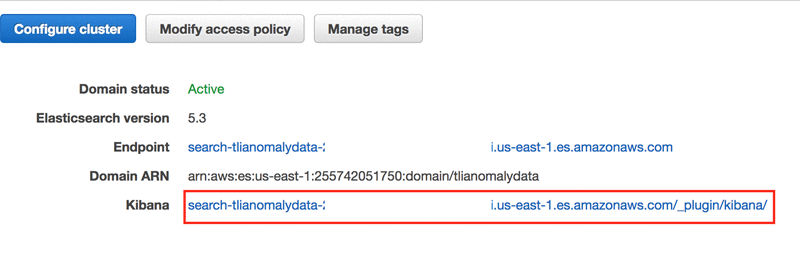


Open the Amazon Elasticsearch Service dashboard in the AWS Management Console. The count in the Searchable documents column increases as shown in the following screenshot. In addition, the domain shows a cluster health of Yellow. This is because, by default, it needs two instances to deploy redundant copies of the index. To fix this, you can deploy two instances instead of one.

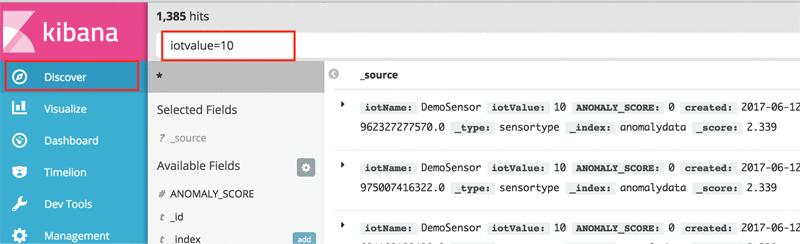


1. **Visualize the data using Kibana**

Use the ES domain link to go to the cluster detail page, and then choose the Kibana link as shown in the following screenshot.



In the Kibana dashboard, choose the Discover tab to perform a query.



You can also visualize the data using the different types of charts offered by Kibana. For example, by going to the Visualize tab, you can quickly create a split bar chart that aggregates by ANOMALY\_SCORE per minute.

