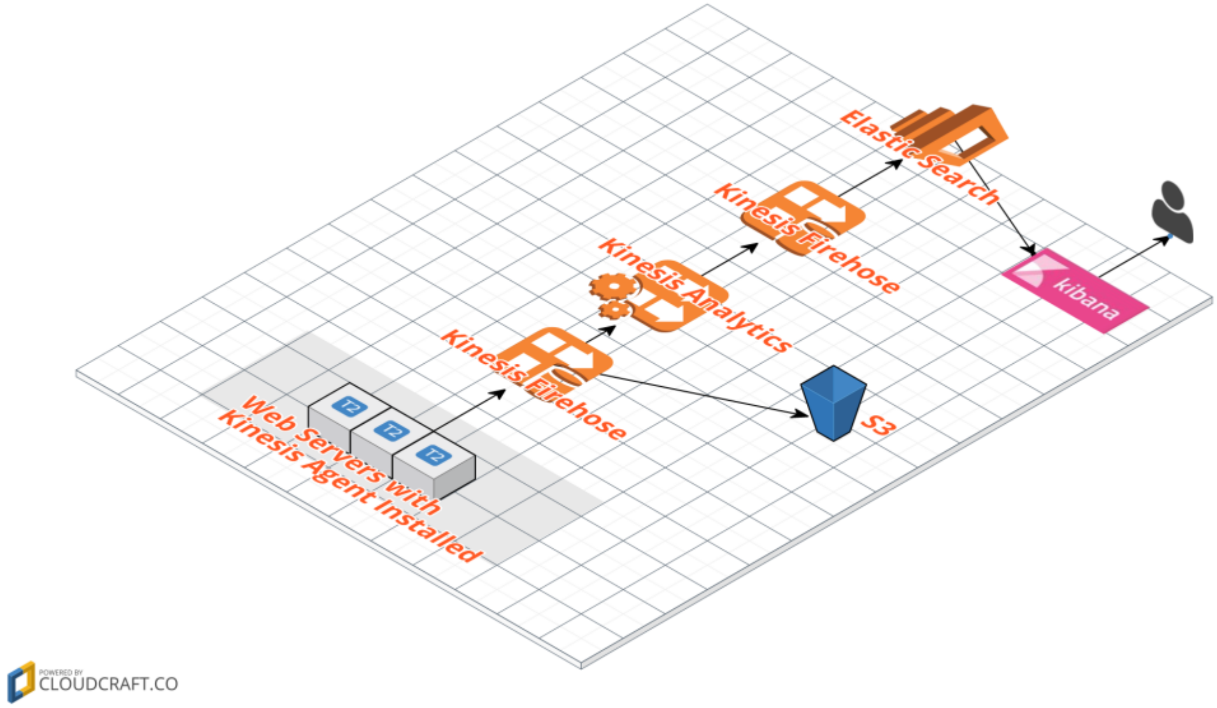
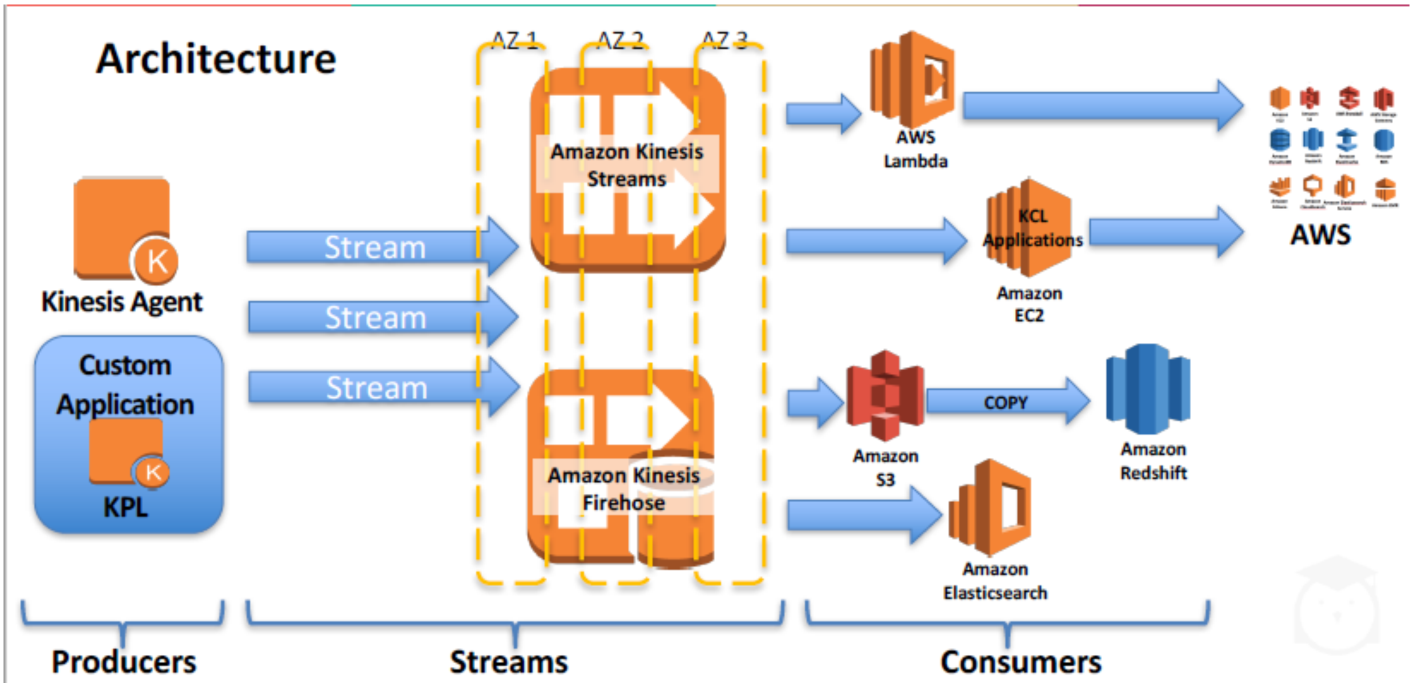
Apache Log Analytics - Kinesis Firehose, Analytics and Elastic Search

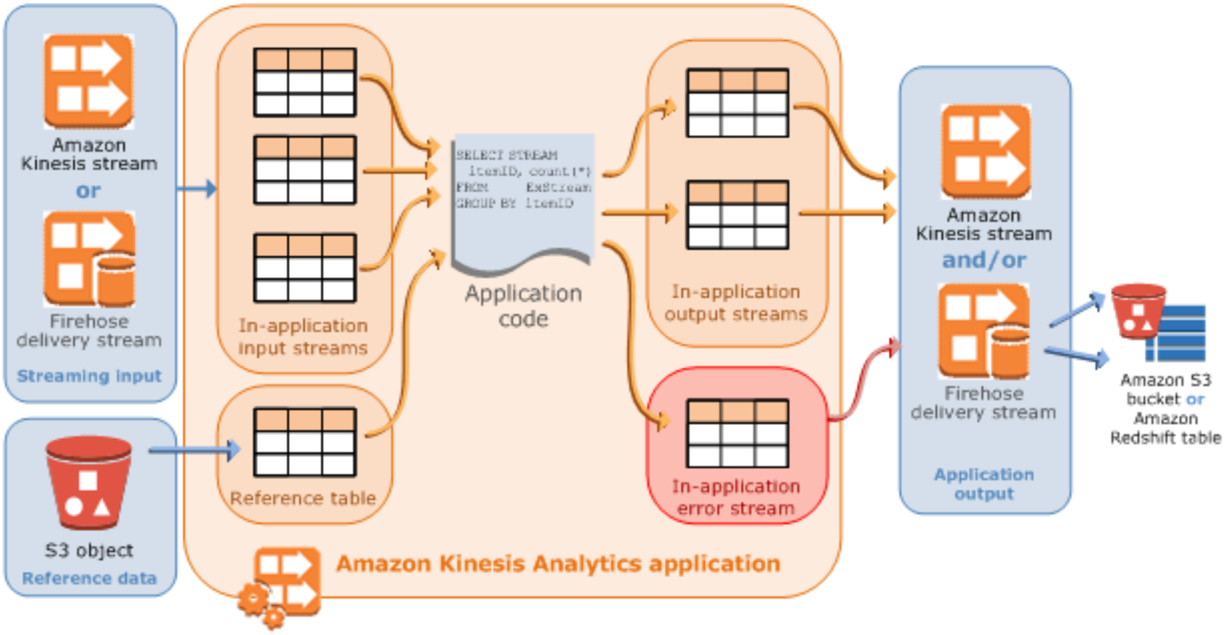


Kinesis Fireshose -



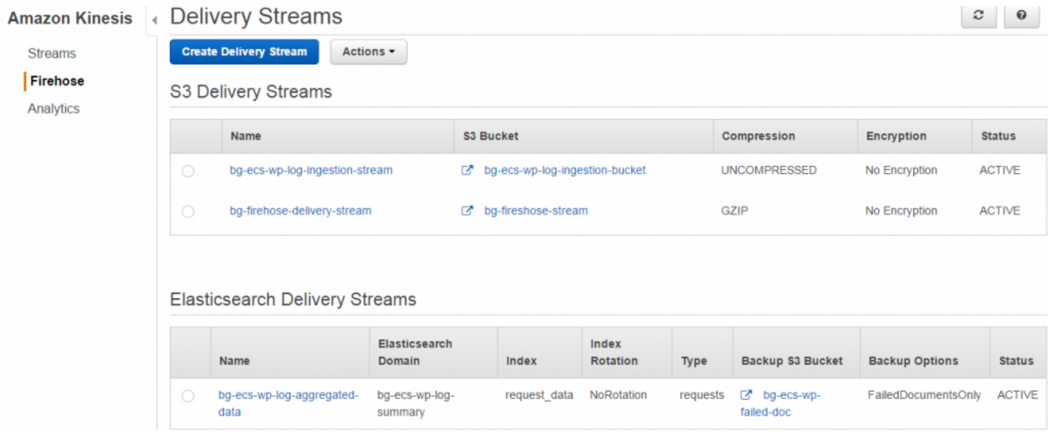
- Firehose can scale to gigabytes of streaming data per second, and allows for batching, encrypting and compressing of data.
- Easiest way to load streaming data into AWS.
- Firehose will automatically scale to meet demand.

Kinesis Analytics -



- Amazon Kinesis Analytics enables users to run standard SQL queries over live streaming data
- Readily query Kinesis Stream/Firehose data and export the output to destination like S3
- In-flight analytics, Standard SQL Interface, Elastic scaling up or down
- data dump to S3, ES, and Redshift
- Auto-schema detection
- Pre-build stream processing templates.

Follow Build a Log Analytics Solution on AWS Tutorial to perform this exercise.



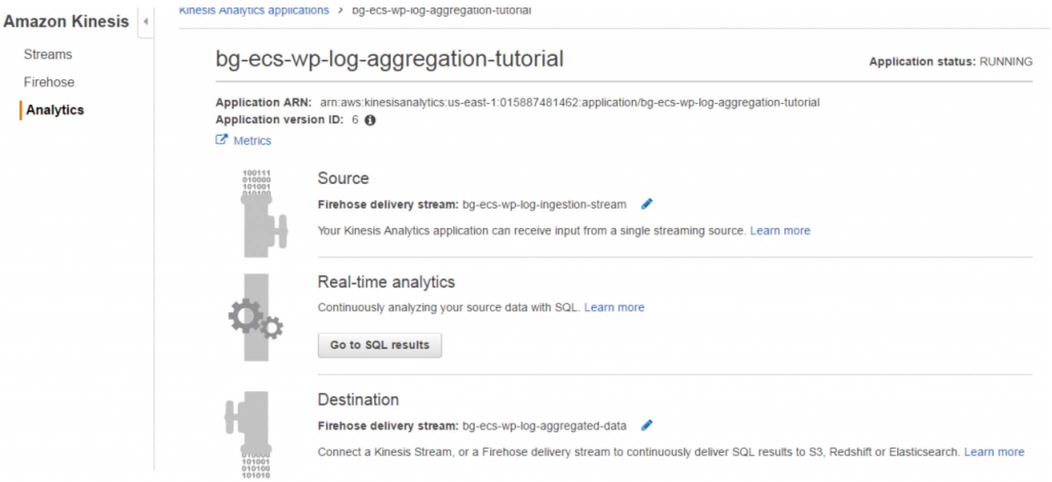
Delivery Streams

S3 Delivery Streams

	Name	S3 Bucket	Compression	Encryption	Status
○	bg-ecs-wp-log-ingestion-stream	bg-ecs-wp-log-ingestion-bucket	UNCOMPRESSED	No Encryption	ACTIVE
○	bg-firehose-delivery-stream	bg-fireshose-stream	GZIP	No Encryption	ACTIVE

Elasticsearch Delivery Streams

	Name	Elasticsearch Domain	Index	Index Rotation	Type	Backup S3 Bucket	Backup Options	Status
○	bg-ecs-wp-log-aggregated-data	bg-ecs-wp-log-summary	request_data	NoRotation	requests	bg-ecs-wp-failed-doc	FailedDocumentsOnly	ACTIVE



Amazon Kinesis
Streams
Firehose
Analytics
Kinesis Analytics applications › bg-ecs-wp-log-aggregation-tutorial

bg-ecs-wp-log-aggregation-tutorial

Application status: RUNNING

Application ARN: arn:aws:kinesisanalytics:us-east-1:015887481462:application/bg-ecs-wp-log-aggregation-tutorial
Application version ID: 6

Source

Firehose delivery stream: bg-ecs-wp-log-ingestion-stream

Your Kinesis Analytics application can receive input from a single streaming source. Learn more

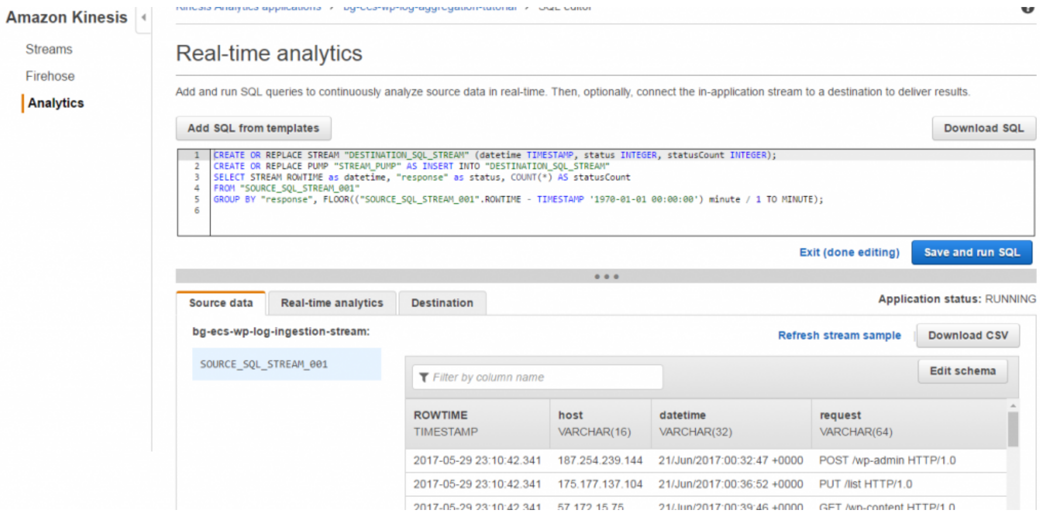
Real-time analytics

Continuously analyzing your source data with SQL. Learn more

Destination

Firehose delivery stream: bg-ecs-wp-log-aggregated-data

Connect a Kinesis Stream, or a Firehose delivery stream to continuously deliver SQL results to S3, Redshift or Elasticsearch. Learn more



Amazon Kinesis
Streams
Firehose
Analytics
Kinesis Analytics applications › bg-ecs-wp-log-aggregation-tutorial › SQL editor

Real-time analytics

Add and run SQL queries to continuously analyze source data in real-time. Then, optionally, connect the in-application stream to a destination to deliver results.

Add SQL from templates
Download SQL

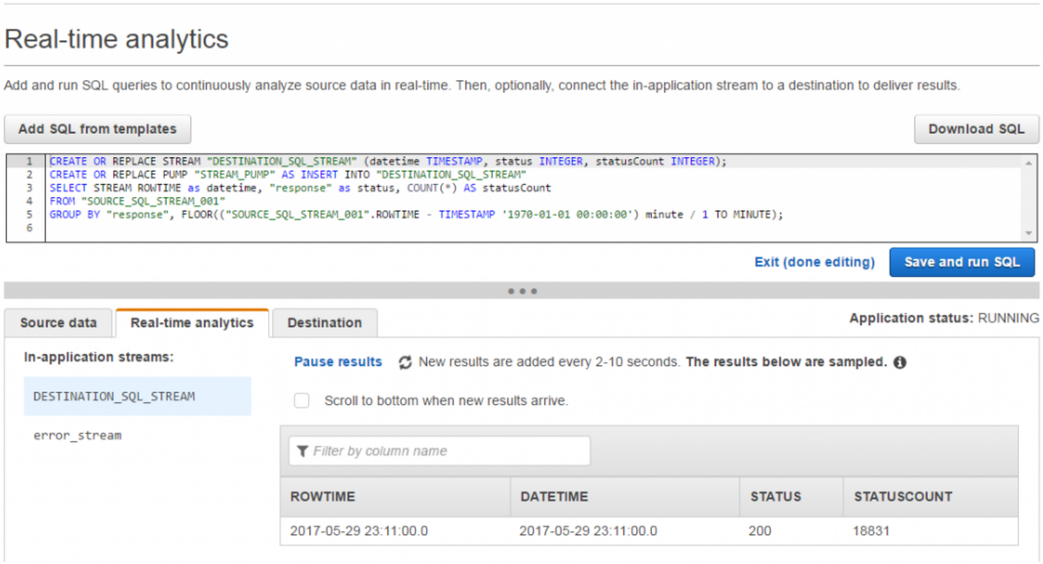
```
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (datetime TIMESTAMP, status INTEGER, statusCount INTEGER);
CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
SELECT STREAM ROWTIME as datetime, "response" as status, COUNT(*) AS statusCount
FROM "SOURCE_SQL_STREAM_001"
GROUP BY "response", FLOOR(("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-01 00:00:00') minute / 1 TO MINUTE);
```

Application status: RUNNING

bg-ecs-wp-log-ingestion-stream:

SOURCE_SQL_STREAM_001

ROWTIME TIMESTAMP	host VARCHAR(16)	datetime VARCHAR(32)	request VARCHAR(64)
2017-05-29 23:10:42.341	187.254.239.144	21/Jun/2017:00:32:47 +0000	POST /wp-admin HTTP/1.0
2017-05-29 23:10:42.341	175.177.137.104	21/Jun/2017:00:36:52 +0000	PUT /list HTTP/1.0
2017-05-29 23:10:42.341	57.172.15.75	21/Jun/2017:00:39:46 +0000	GET /wp-content HTTP/1.0



Real-time analytics

Add and run SQL queries to continuously analyze source data in real-time. Then, optionally, connect the in-application stream to a destination to deliver results.

```
CREATE OR REPLACE STREAM "DESTINATION_SQL_STREAM" (datetime TIMESTAMP, status INTEGER, statusCount INTEGER);
CREATE OR REPLACE PUMP "STREAM_PUMP" AS INSERT INTO "DESTINATION_SQL_STREAM"
SELECT STREAM ROWTIME as datetime, "response" as status, COUNT(*) AS statusCount
FROM "SOURCE_SQL_STREAM_001"
GROUP BY "response", FLOOR(("SOURCE_SQL_STREAM_001".ROWTIME - TIMESTAMP '1970-01-01 00:00:00') minute / 1 TO MINUTE);
```

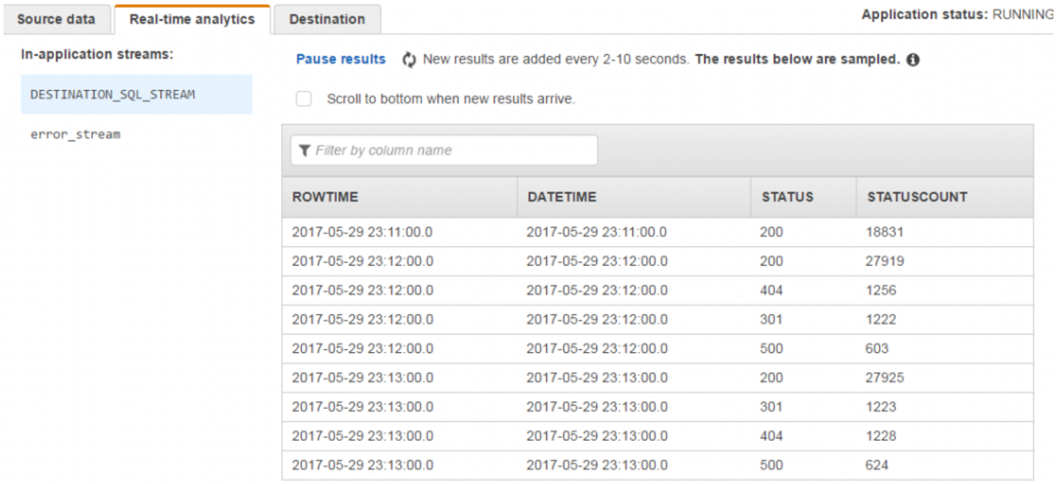
Application status: RUNNING

In-application streams:

New results are added every 2-10 seconds. **The results below are sampled.**

☐ Scroll to bottom when new results arrive.

ROWTIME	DATETIME	STATUS	STATUSCOUNT
2017-05-29 23:11:00.0	2017-05-29 23:11:00.0	200	18831



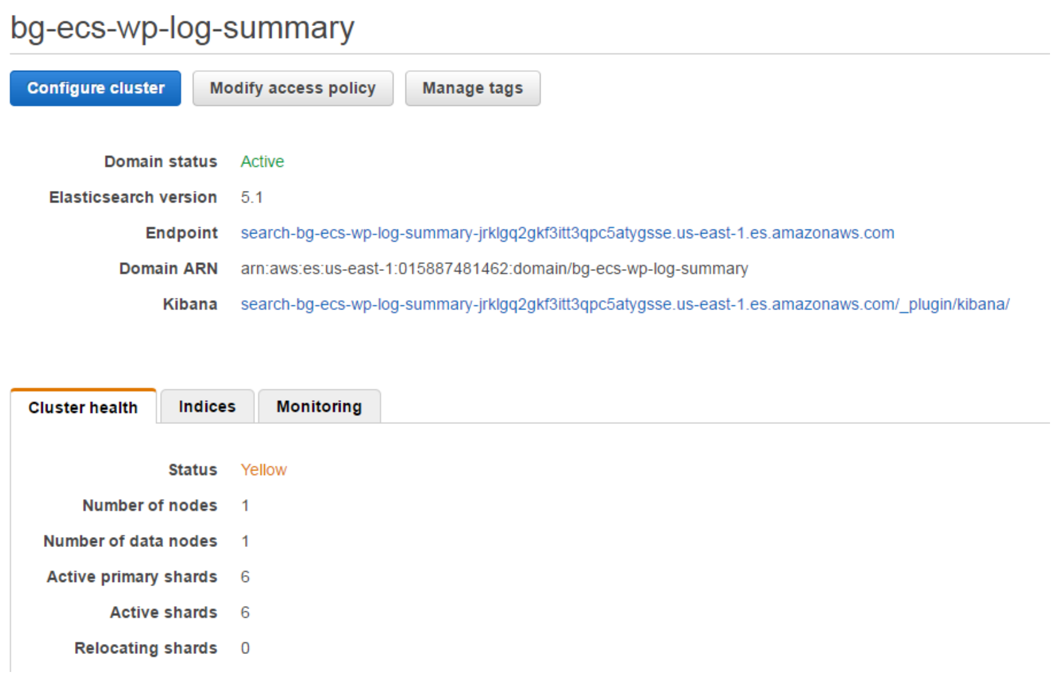
Application status: RUNNING

In-application streams:

New results are added every 2-10 seconds. **The results below are sampled.**

☐ Scroll to bottom when new results arrive.

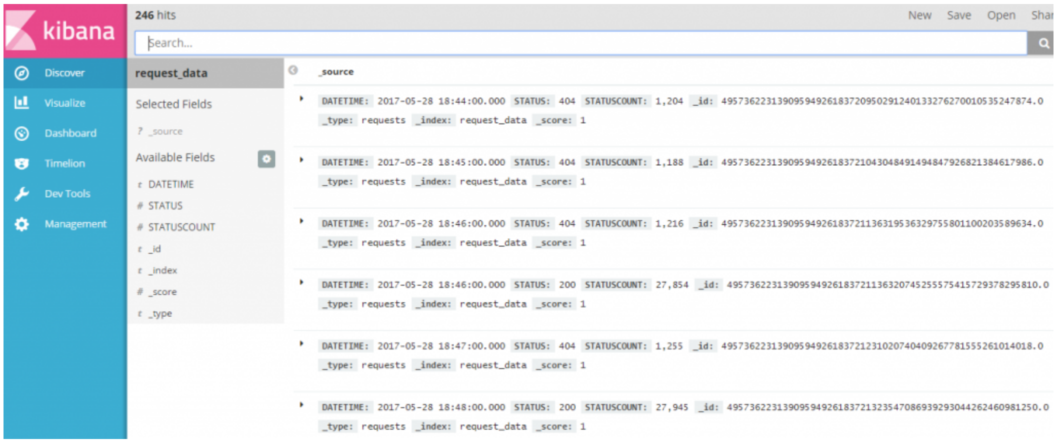
ROWTIME	DATETIME	STATUS	STATUSCOUNT
2017-05-29 23:11:00.0	2017-05-29 23:11:00.0	200	18831
2017-05-29 23:12:00.0	2017-05-29 23:12:00.0	200	27919
2017-05-29 23:12:00.0	2017-05-29 23:12:00.0	404	1256
2017-05-29 23:12:00.0	2017-05-29 23:12:00.0	301	1222
2017-05-29 23:12:00.0	2017-05-29 23:12:00.0	500	603
2017-05-29 23:13:00.0	2017-05-29 23:13:00.0	200	27925
2017-05-29 23:13:00.0	2017-05-29 23:13:00.0	301	1223
2017-05-29 23:13:00.0	2017-05-29 23:13:00.0	404	1228
2017-05-29 23:13:00.0	2017-05-29 23:13:00.0	500	624



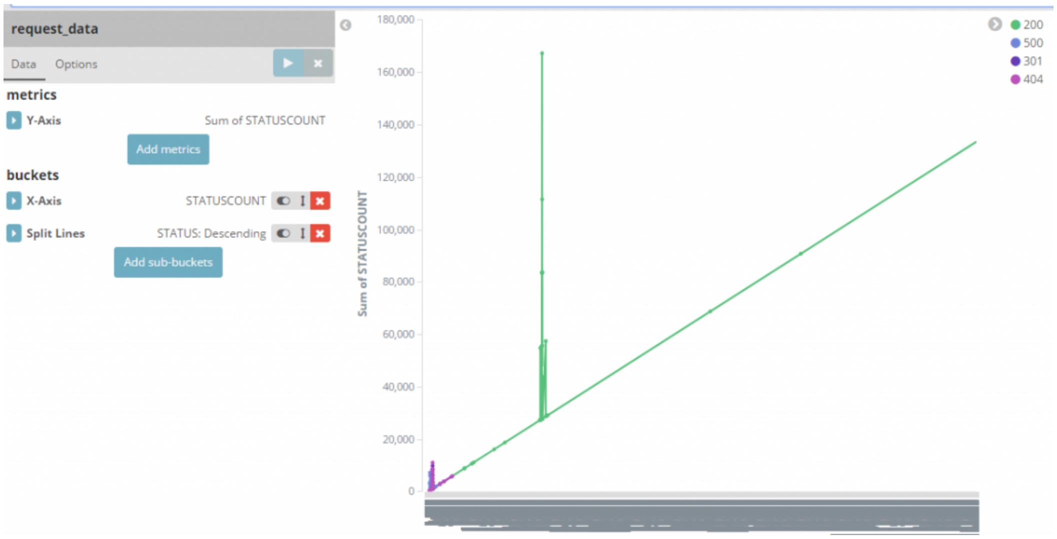
bg-ecs-wp-log-summary

Domain status	Active
Elasticsearch version	5.1
Endpoint	search-bg-ecs-wp-log-summary-jrklgq2gkf3itt3qpc5atygsse.us-east-1.es.amazonaws.com
Domain ARN	arn:aws:es:us-east-1:015887481462:domain/bg-ecs-wp-log-summary
Kibana	search-bg-ecs-wp-log-summary-jrklgq2gkf3itt3qpc5atygsse.us-east-1.es.amazonaws.com/_plugin/kibana/

Status	Yellow
Number of nodes	1
Number of data nodes	1
Active primary shards	6
Active shards	6
Relocating shards	0



```
DATETIME: 2017-05-28 18:44:00.000 STATUS: 404 STATUSCOUNT: 1,204 _id: 49573622313909594926183720950291240133276270010535247874.0
_type: requests _index: request_data _score: 1

DATETIME: 2017-05-28 18:45:00.000 STATUS: 404 STATUSCOUNT: 1,188 _id: 49573622313909594926183721043048491494847926821384617986.0
_type: requests _index: request_data _score: 1

DATETIME: 2017-05-28 18:46:00.000 STATUS: 404 STATUSCOUNT: 1,216 _id: 49573622313909594926183721136319536329755801100203589634.0
_type: requests _index: request_data _score: 1

DATETIME: 2017-05-28 18:46:00.000 STATUS: 200 STATUSCOUNT: 27,854 _id: 49573622313909594926183721136320745255575415729378295810.0
_type: requests _index: request_data _score: 1

DATETIME: 2017-05-28 18:47:00.000 STATUS: 404 STATUSCOUNT: 1,255 _id: 49573622313909594926183721231020740409267781555261014018.0
_type: requests _index: request_data _score: 1

DATETIME: 2017-05-28 18:48:00.000 STATUS: 200 STATUSCOUNT: 27,945 _id: 49573622313909594926183721323547086939293044262460981250.0
_type: requests _index: request_data _score: 1
```



request_data

Data Options

metrics

Y-Axis Sum of STATUSCOUNT

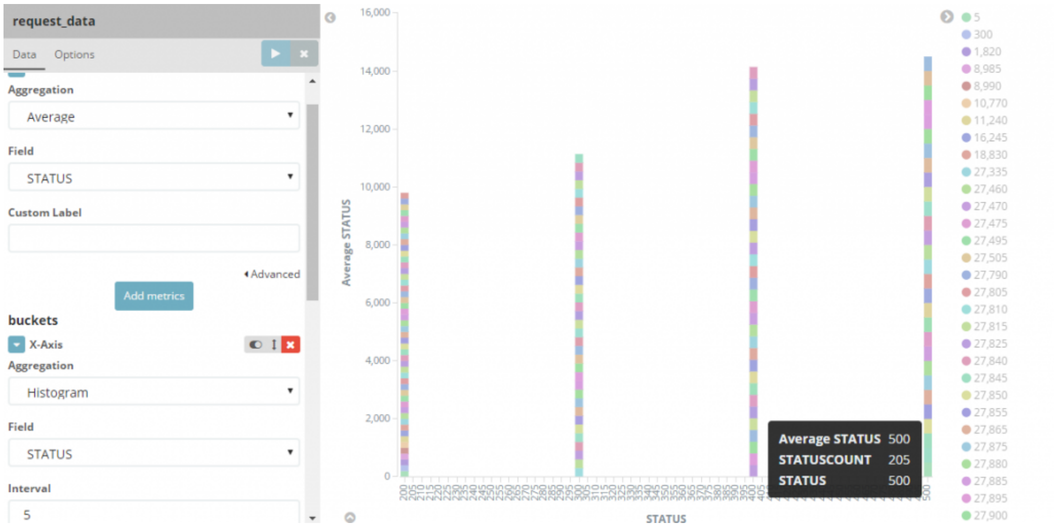
Add metrics

buckets

X-Axis STATUSCOUNT

Split Lines STATUS: Descending

Add sub-buckets



request_data

Data Options

Aggregation: Average

Field: STATUS

Custom Label:

Advanced

Add metrics

buckets

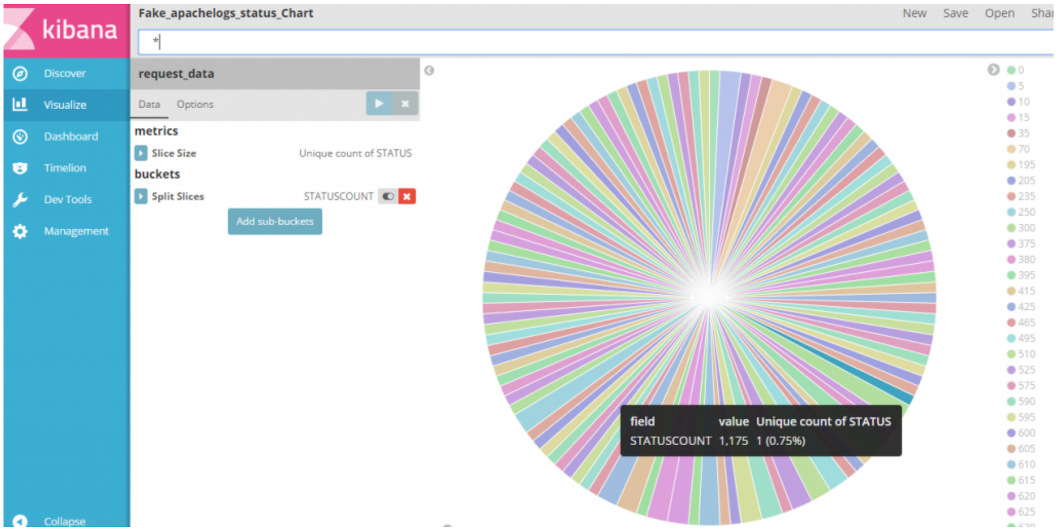
X-Axis

Aggregation: Histogram

Field: STATUS

Interval: 5

Average STATUS 500
STATUSCOUNT 205
STATUS 500



Fake_apachelogs_status_Chart

New Save Open Shai

*

request_data

Data Options

metrics

Slice Size Unique count of STATUS

buckets

Split Slices STATUSCOUNT

Add sub-buckets

field value Unique count of STATUS
STATUSCOUNT 1,175 1 (0.75%)