Splunk can Drink from a Firehose

An AWS Kinesis Firehose

Gary Mikula | FINRA | Senior Director, Cyber & Information Security
Kuljeet Singh | FINRA | Lead Security Engineer

October 3, 2018

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or the expected performance of the company. We caution you that such statements reflect our current expectations and estimates based on factors currently known to us and that actual events or results could differ materially. For important factors that may cause actual results to differ from those contained in our forward-looking statements, please review our filings with the SEC.

The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements we may make. In addition, any information about our roadmap outlines our general product direction and is subject to change at any time without notice. It is for informational purposes only and shall not be incorporated into any contract or other commitment. Splunk undertakes no obligation either to develop the features or functionality described or to include any such feature or functionality in a future release.

splunk> .conf18

What's In It For You

Deep Dive into AWS Firehose/Splunk Integration

- High Level Understanding of AWS Firehose
- How it Integrates with Splunk
- How to Get an Initial Service Up and Running
- How to Splunk-ify Your Data
- How to Design for Errors
- How to Cost Estimate the Service

- **All the Tools Necessary to Build Production Ready Services**

Who We Are and How We Got Here

FINRA's Roadmap with Splunk and AWS

splunk> .conf18

We Are FINRA

Financial Industry Regulatory Authority

- An independent, non-governmental regulator for all securities firms doing business with the public in the United States
- Regulates 630,000 brokers and 3,800 brokerage firms and by monitoring trading on U.S. stock markets
- FINRA monitors up to 100 billion transactions analyzed per day. Take the number of:
  - VISA charges (29M) + tweets (0.5B) + likes and updates (2.7B)
  - Add them together, multiply by 20, and you almost get to 100B

Our Journey with Splunk

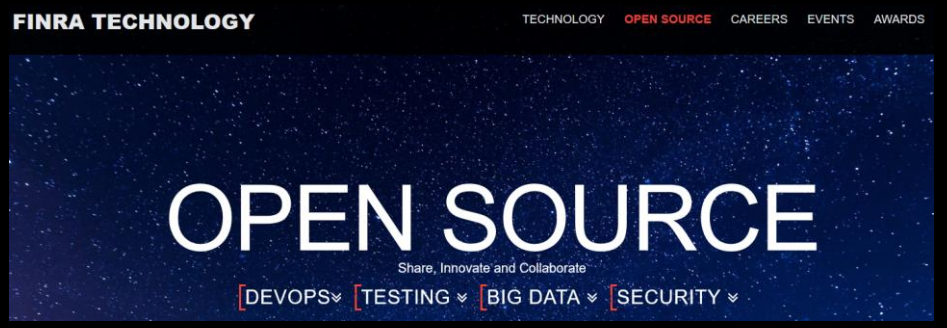
- Became one of the First Large SplunkCloud Customers in 2013
- Over 1/3 of Technology Staff uses Splunk every week
- Became Enterprise Security Customer in 2018
- Key Splunk/AWS Integrations:
  - Orchestrated Splunk into all Elastic Compute IaaS
  - IAM Compliance
  - Bootstrapped Splunk agent into Elastic Map Reduce (EMR)
  - Splunk Side-Car Container Service for ECS
  - Financial System of Record for all AWS charges
  - Security Group Analysis/Maintenance/Alerting
  - Event Driven Surveillance
  - And…..

Shameless Self Promotion

If You Are Here, You Might Enjoy…..

- .conf2017 – Integrating Splunk and AWS Lambda
  - A Better way to Get AWS Cloud Trail Logs into Splunk
  - Open Sourced as “CT Grazer”
  - https://github.com/FINRAOS/CTGrazer

- FINRA is Active in the Open-Source Community
  - DevOps
  - Security
  - Big Data
  - http://technology.finra.org/opensource.html



splunk> .conf18

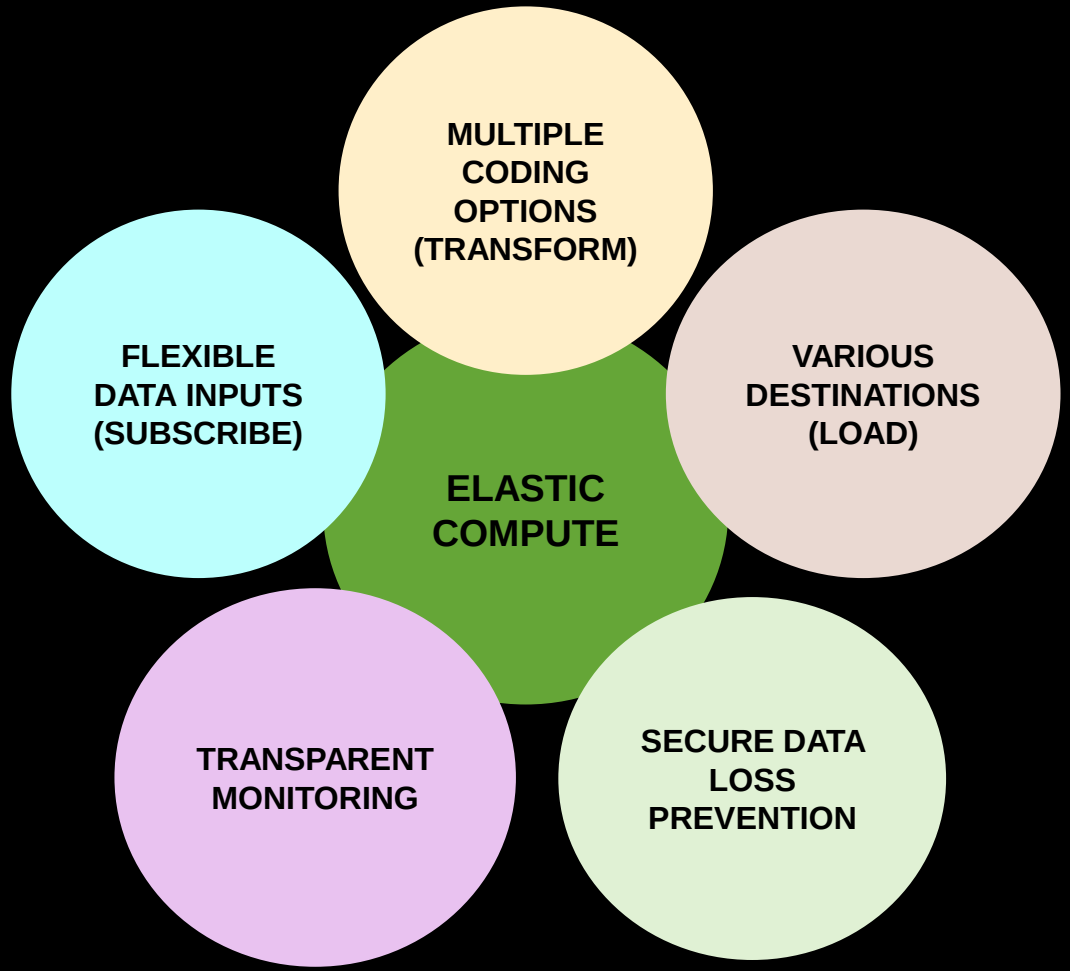
What is the AWS Firehose Service

Let's Deconstruct into Sub-Services

splunk> .conf18

Subscribe/Send – Transform - Load

Financial Industry Regulatory Authority

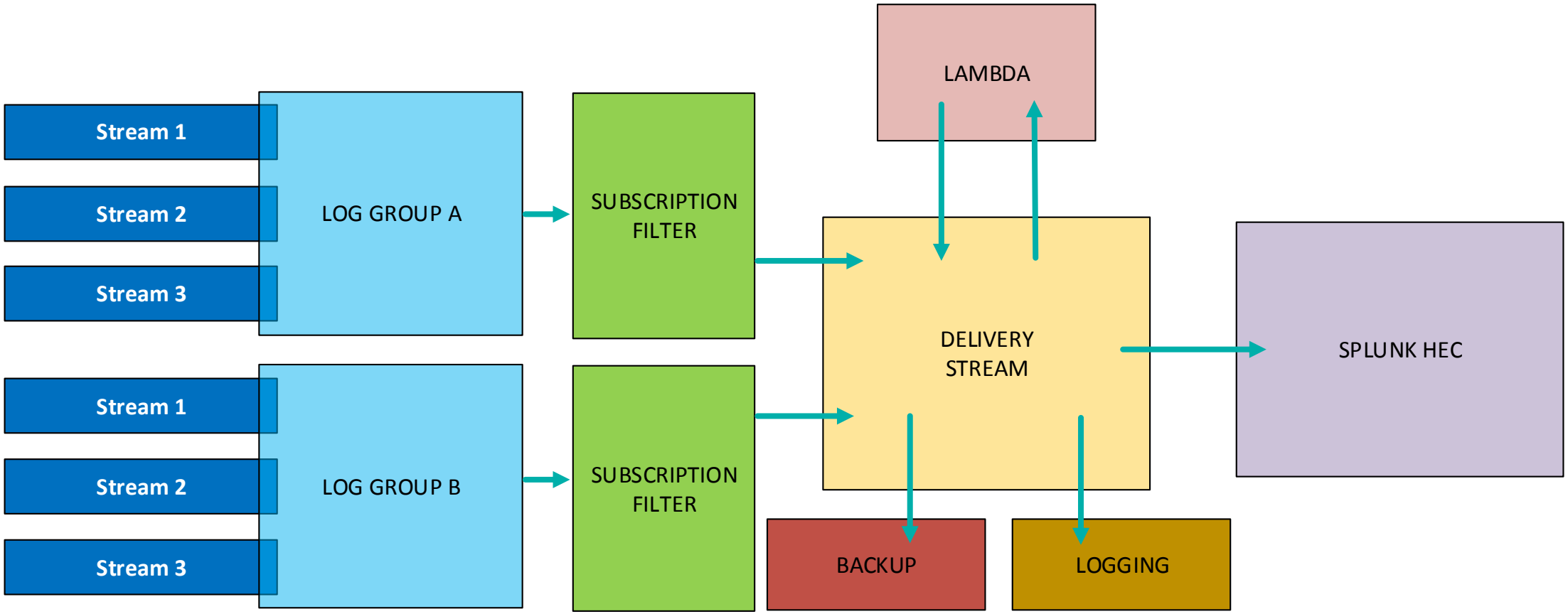


Configuring Firehose to Send Data to Splunk

Everything You Need to Get Your First Integration Working

splunk> .conf18

Overview of CWL into Firehose

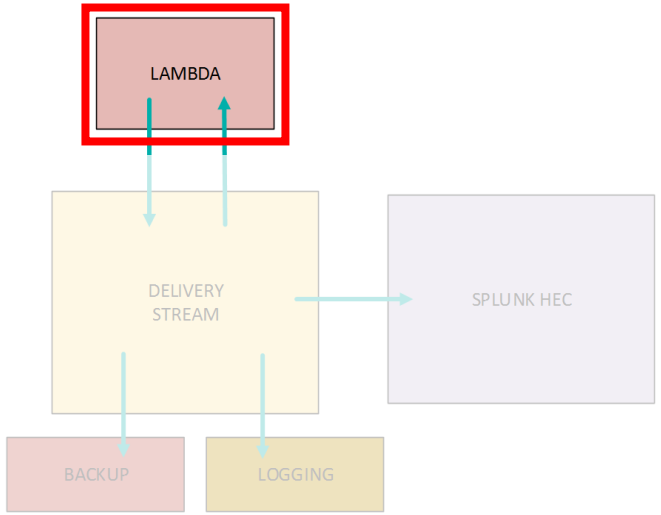


Data Processing

Lambda configuration for Data Transformation

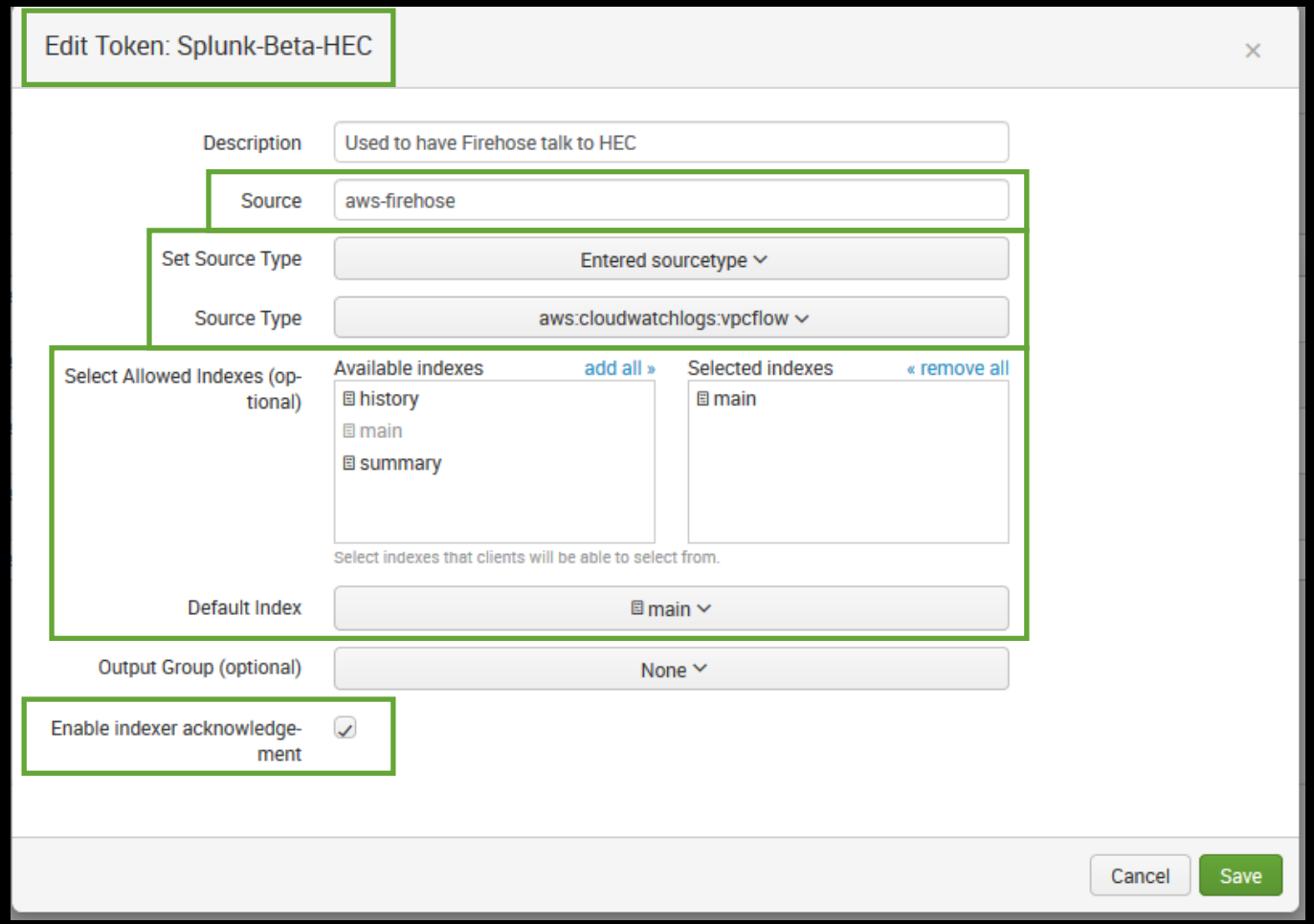
```
"ProcessingConfiguration": {
  "Enabled": true,
  "Processors": [{
    "Type": "Lambda",
    "Parameters": [{
      "ParameterName": "LambdaArn",
      "ParameterValue": "arn:aws:lambda:us-east-1:123456789123:function:SPLUNK-VPC-TRANSFORM:$LATEST“ },
      { "ParameterName": "RoleArn",
        "ParameterValue": "arn:aws:iam::123456789123:role/SVC_FIREHOSE_SPLUNK_SR“ },
      { "ParameterName": "NumberOfRetries", "ParameterValue": "3“ },
      { "ParameterName": "BufferSizeInMBs", "ParameterValue": “1“ },
      { "ParameterName": "BufferIntervalInSeconds", "ParameterValue": "60“ }]
    }]
},
```

- Lambda Function is required for processing CWLs
- FH CWL Blueprints for Lambda
  - Python & Node.js
  - kinesis-firehose-cloudwatch-logs-processor
- IAM Role with access to required resources
- Buffer size



Data Destination Token

Let's look at SPLUNK HEC Token options



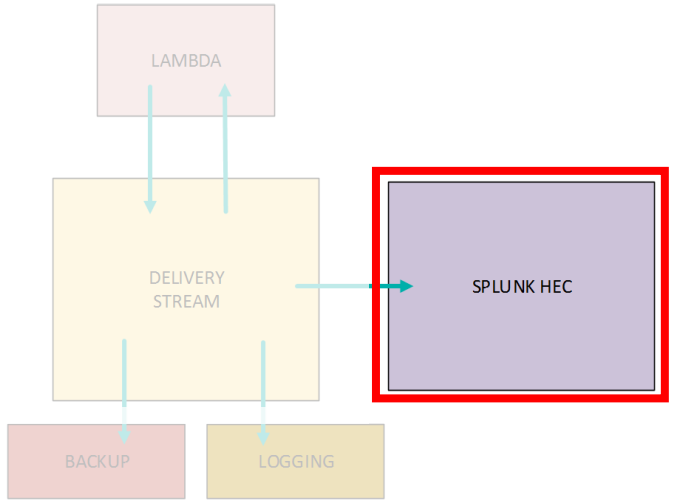
- Name of HEC Token
- Source if not defined here
  - http:<tokenName>
- Set sourcetype, ensure
  - "splunk-add-on-for-amazon-kinesis-firehose" installed
  - pulldown_type is enabled
- Index configuration
- Indexer acknowledgement

splunk> .conf18

Data Destination

SPLUNK HEC configuration

```
"HECEndpoint": "https://yourhec.your.org:8088",
"HECEndpointType": "Raw",
"HECToken": "2d626201-8bab-45f4-b280-24499c56f1f4",
"HECAcknowledgmentTimeoutInSeconds": 180,
"RetryOptions": {
        "DurationInSeconds": 300
},
```



- Before you begin, ensure:
  - DS and HEC endpoints can communicate
  - Enable sticky session for: proxy/LB
- Endpoint Type
  - Raw
    - All Events Strung Together
    - New Line delimiter
  - Event
    - JSON structured
- CA-signed certificate
- Indexer Acknowledgement timeout

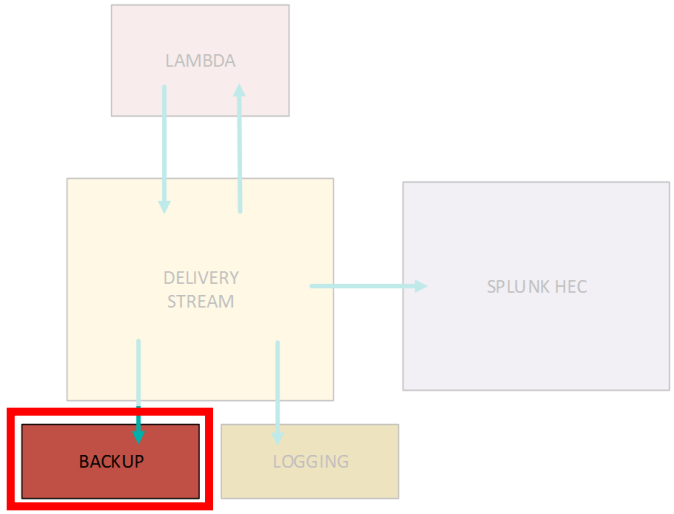
splunk> .conf18

Data Backup

S3 configuration in case of delivery failure

```
"S3BackupMode": "FailedEventsOnly",
"S3Configuration": {
  "RoleARN": "arn:aws:iam::123456789123:role/SVC_FIREHOSE_SPLUNK_SR",
  "BucketARN": "arn:aws:s3:::1234-5678-9123-logs",
  "Prefix": "SPLUNK/firehose",
  "BufferingHints": { "IntervalInSeconds": 300, "SizeInMBs": 5 },
  "CompressionFormat": "UNCOMPRESSED",
  "EncryptionConfiguration": {
    "KMSEncryptionConfig": {
      "AWSKMSKeyARN": "arn:aws:kms:us-east-1:123456789123:key/63852895-684e-47b5-9740-78a1da0a31f0“
} },
```

- Backup Failed Events Only or All Events
- S3 bucket
- IAM Role for writing to S3
- Compress events
  - $$ vs Ease of Processing
- Encryption



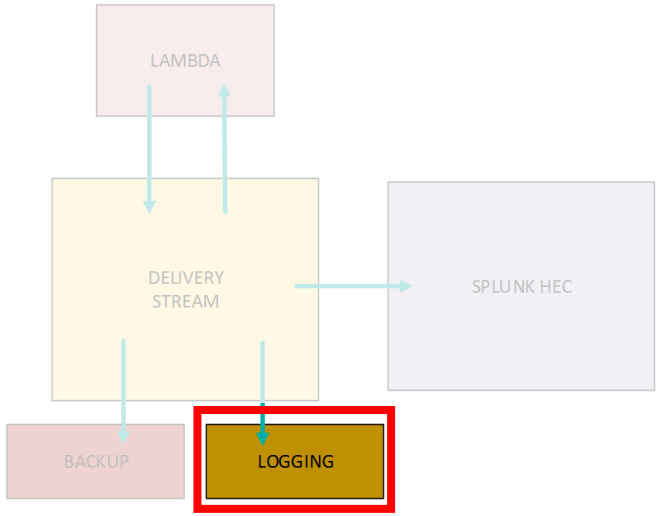
splunk> .conf18

CWL Logging

DS self error logging

```
"CloudWatchLoggingOptions": {
        "Enabled": true,
        "LogGroupName": "/aws/firehose/<LogGroupName>",
        "LogStreamName": "S3Delivery"
        }
},
"CloudWatchLoggingOptions": {
        "Enabled": true,
        "LogGroupName": "/aws/firehose/<LogGroupName>",
        "LogStreamName": "SplunkDelivery"
        }
}
```

- Enable logging for
  - S3 delivery
  - Splunk delivery
- LogStream – a place you write events
- LogGroup – can contain multiple log streams



Delivery Stream Configuration

Configuration via CLI Command & JSON

aws firehose create-delivery-stream

--delivery-stream-name “<NameOfDeliveryStream>”

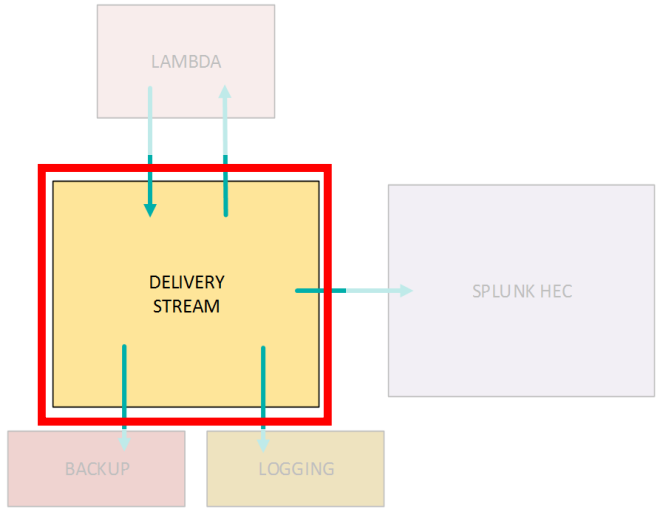
- Name must be unique in AWS region per account

--splunk-destination-configuration file:///<PATH-TO-JSON-FILE>

- JSON Formatted
- All Configuration Settings for Splunk Destination

[--delivery-stream-type “DirectPut”]

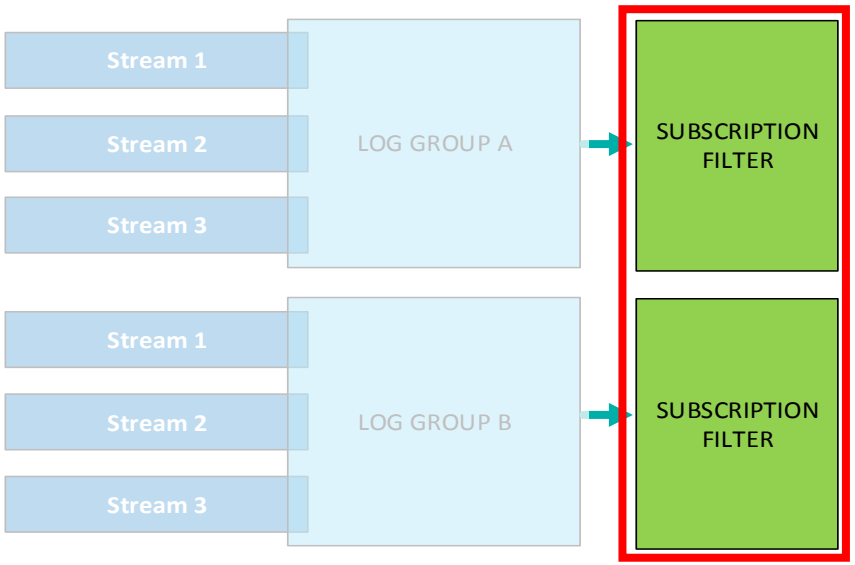
- Type of stream. Default type is “DirectPut”

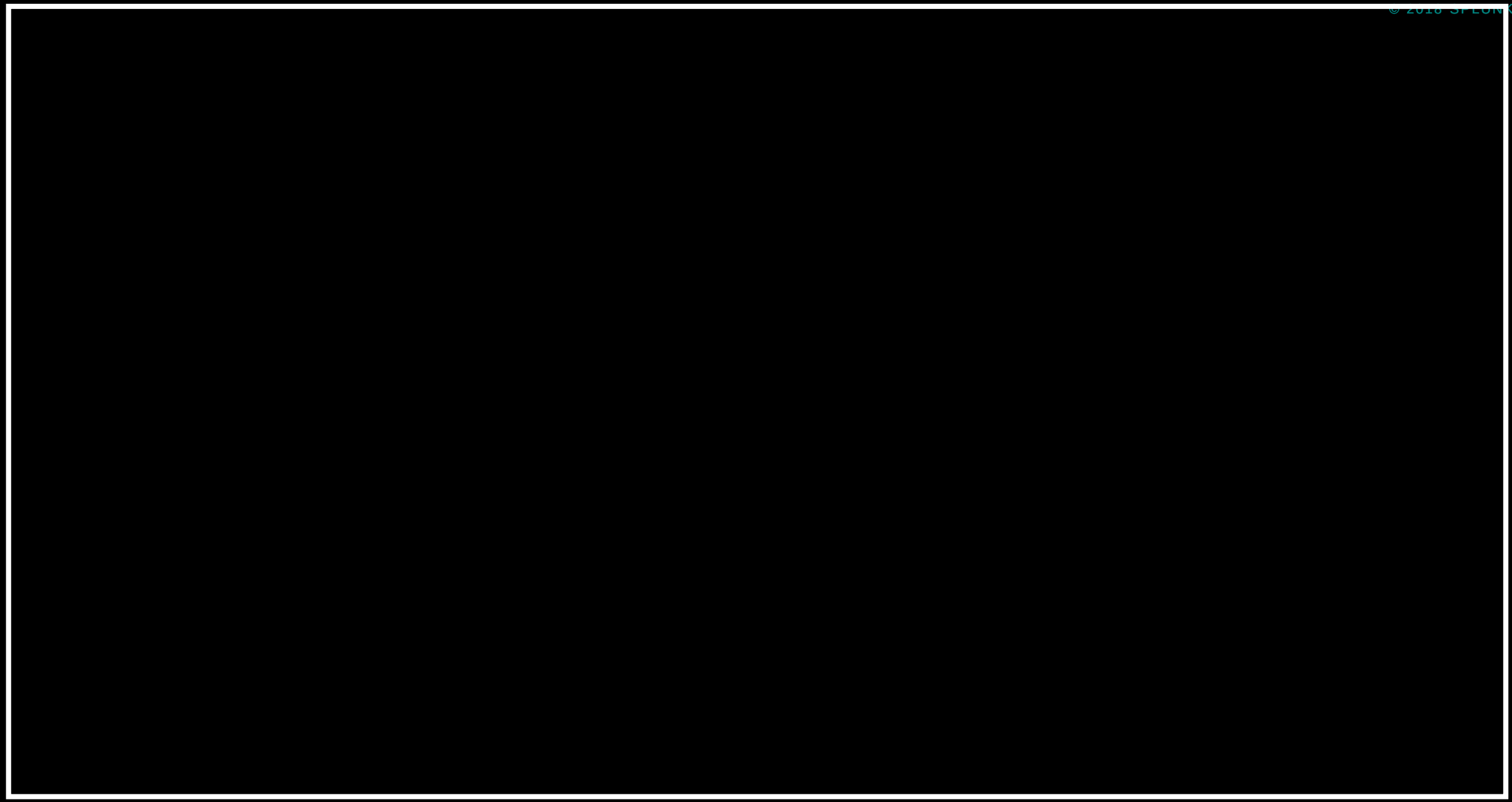


Delivery Stream

Associate subscription filter to log group

- put-subscription-filter
  - --log-group-name <value>
    - Log Group can only have ONE subscription filter
  - --filter-name
    - Unique Name of your subscription filter
  - --filter-pattern
    - Method to filter Streams from a LogGroup
  - --destination-arn
    - The Delivery Stream we just built
  - --role-arn
    - Must be able to read from CloudWatch and write to the Delivery Stream





Digging Deep into the Transform AWS Lambda Function

kinesis-firehose-cloudwatch-logs-processor-python

splunk> .conf18

Why Would I *EVER* Want To Do This?

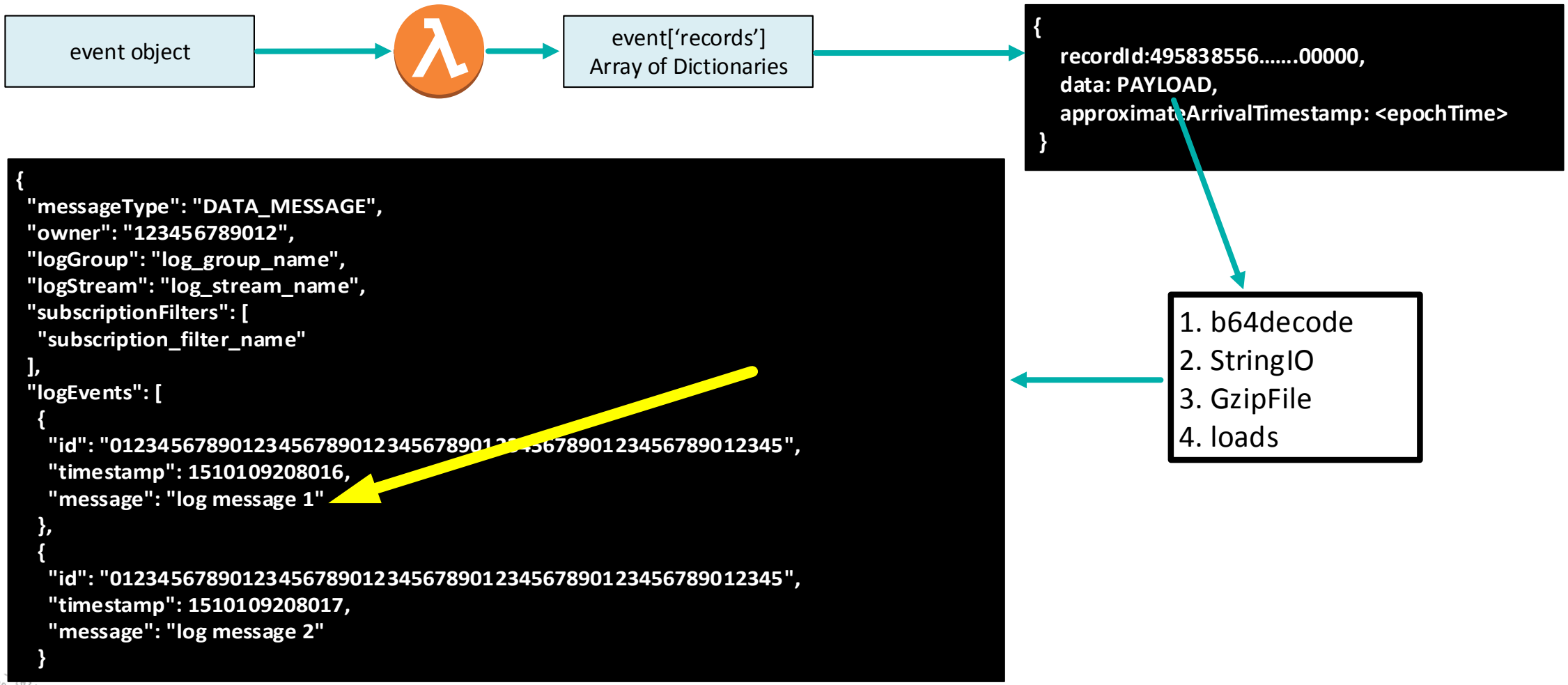
Saving Money & Better User Experience

- Filter Events
- Event Enrichment
  - Send to JSON HEC Key
- Put Your Events on a Diet
- Enhanced Logging
  - Give me more information about what's going on!!!
  - Optimize your configuration settings
- Lambda Efficiencies
  - Small changes are multiplied
- It is *only* a Blueprint

Configure
Customize
Optimize

Getting to the Log Messages

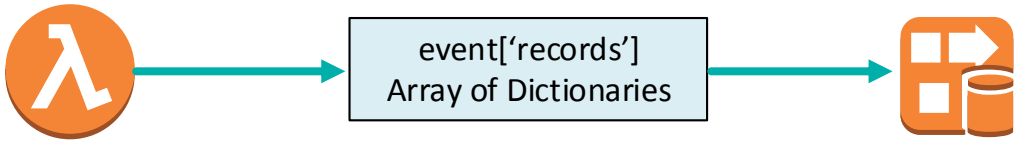
kinesis-firehose-cloudwatch-logs-processor-python



splunk> .conf18

What You Send Back

- data
  - base-64 encoded
  - FORMATTED_PAYLOAD
    - EVENT – new line delimited
    - JSON – Serialized Array of Dictionaries
  - NOT GZIPPED!!!
- result
  - Ok – Send this record's data to the HEC
  - Dropped – All is well, nothing to send to HEC
  - ProcessingFailed – Raises an error in Monitoring
- recordid
  - Accounted for by the Delivery Stream




```
{
  'data': FORMATTED_PAYLOAD,
  'recordId': 495838556.....00000,
  'result': RESULT_VALUE
}
```

Filtering Data

Knowing Your Data Saves $$$

- Omit Chatty, Low Value Events
  - Successful queries to valid service address
    - Lines 1, 3-5, transformLogEvent
- Ensure you test for empty sets
  - Line 7 processRecords
- Efficiency equals cost savings
  - Regex is ~10x slower than ‘IN’
  - Find is ~2x slower than ‘IN’
  - Lines 3-5, transformLogEvent
- This is HEC sourcetype=
  - aws:cloudwatchlogs:vpcflow

```
1 for e in data['logEvents']:
 2         event = transformLogEvent(e)
 3         if(event):
 4             tmpData.append(event)
 5
 6 # if we have no data, mark as dropped
 7     if(tmpData):
 8         data = base64.b64encode(''.join(tmpData))
 9         yield {
10                 'data': data,
11                 'result': 'Ok',
12                 'recordId': recId
13             }
14     else:
15         yield {
16                 'result': 'Dropped',
17                 'recordId': recId
18             }
```

processRecords

```
1if 'NODATA' in log_event['message']:
2         return 0
3 elif(( ' 9997 ' in log_event['message']) and
4       ( ' ACCEPT ' in log_event['message'])):
5         return 0
6 else:
7         return log_event['message'] + '\n'
```

transformLogEvent

splunk> .conf18

Enriching Data

- Make Life Easy for Your Users
- Only Send Variables through the Delivery Stream, Add Constants in the Config
- Naming Standards?
- Fields must be entered in fields.conf to be searchable as fields
- This is HEC sourcetype=_json
- Remember to remove the newline from the event
- Convert the dictionary to string before sending back

```
eventDict                                = {}
eventDict['time']                        = log_event["timestamp"]
functionName                             = re.split('\/',logGroup)[-1]
if( '-' in functionName ):
   ags                    = re.split('-',functionName,maxsplit=1)[0]
else:
   ags                    = 'UNK'
eventDict['sourcetype']                            = ags.lower()
eventDict['source']                                = 'aws:lambda'
eventDict['fields']                                = {}
eventDict['fields']['region']                      = region.lower()
eventDict['fields']['account']                     = account
eventDict['fields']['logGroup']                    = logGroup
eventDict['fields']['logStream']                   = logStream
eventDict['fields']['functionName']                = functionName
eventDict['fields']['subscriptionFilter'] = subscriptionFilter

eventDict.update({ "event" : log_event['message'].rstrip('\n')})

return json.dumps(eventDict)
```

splunk> .conf18

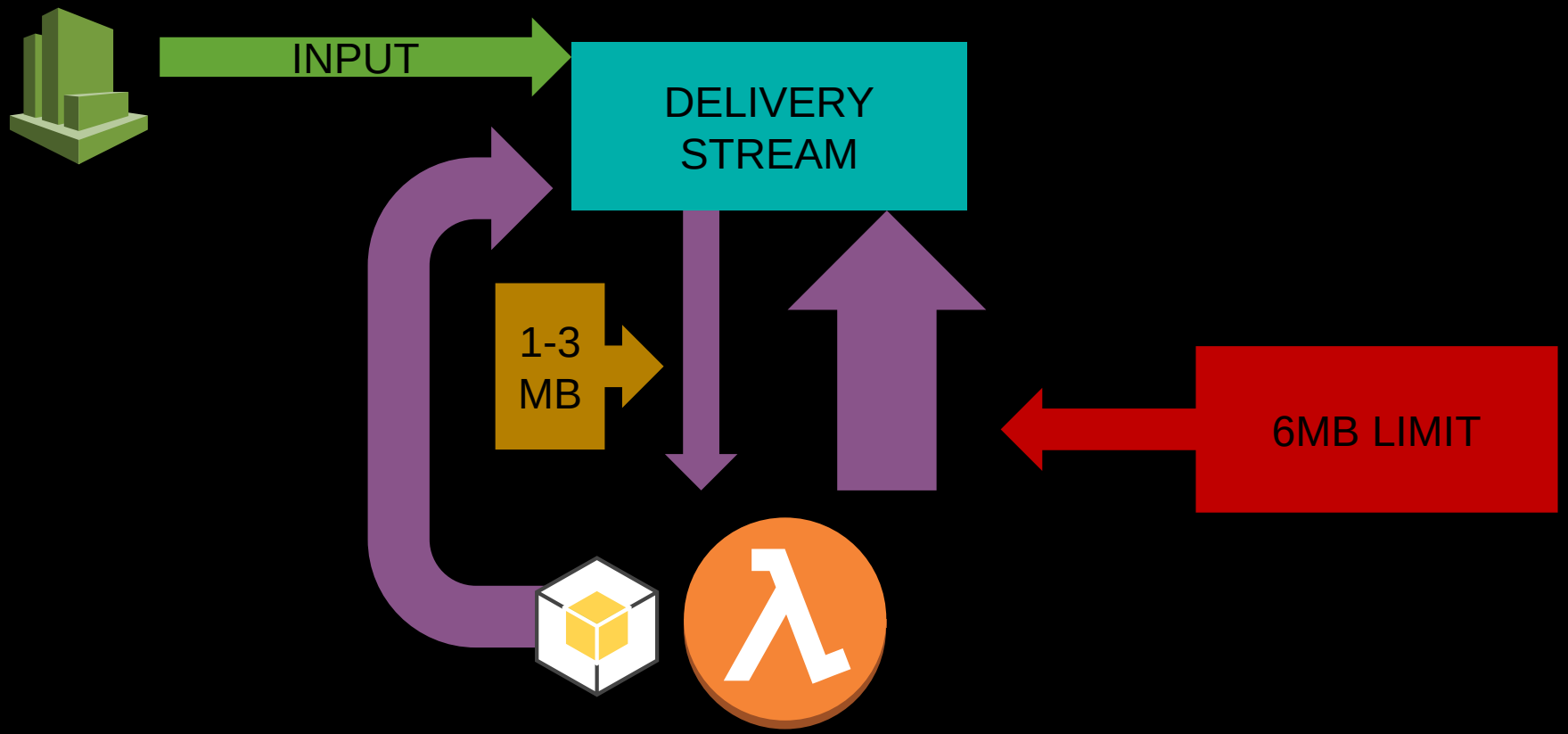
Understanding ReIngestion

Dealing with Extremely High Volume CloudWatch Streams

splunk> .conf18

What Is Re-Ingestion

And Why Do We Need to Care



Blueprint Performance Tweak

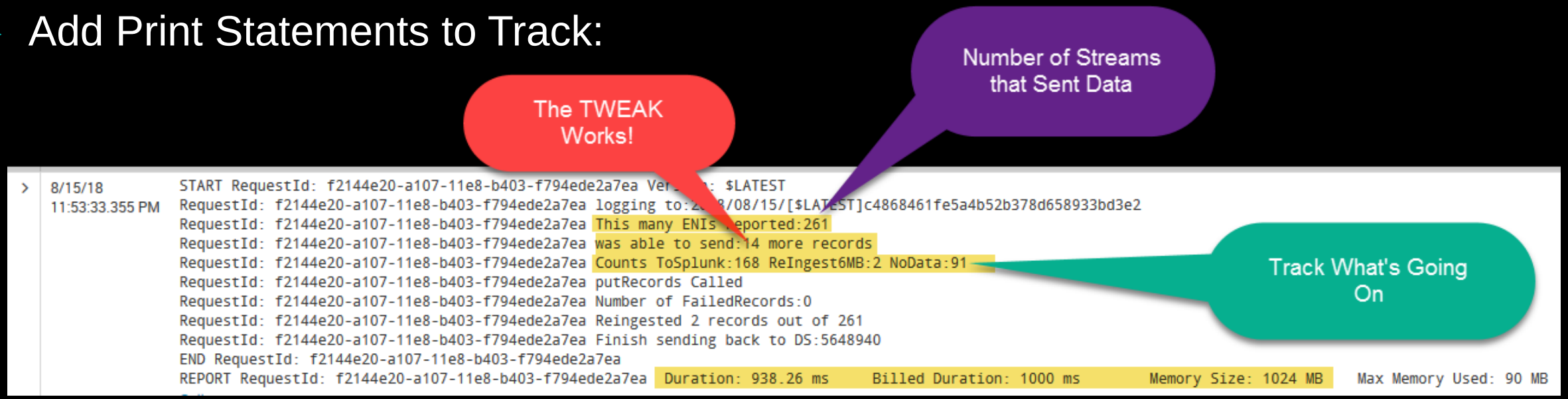
All Records are NOT Created Equal

```
if( projectedSize + len(rec['data']) + len(rec['recordId']) < 6000000 ):
        projectedSize += len(rec['data']) + len(rec['recordId'])
```

splunk> .conf18

Things You Need to Know

- AWS CLI calls put_record() & put_record_batch() are Relatively Expensive
- Some Re-Ingestion Normal, BUT Excessive Re-Ingestion can Drive Lambda Charges and Cause Latency
- Add Print Statements to Track:



- Adjust Your AWS design accordingly
  - Two Delivery Streams can be less expensive than One

splunk> .conf18

Planning for Errors

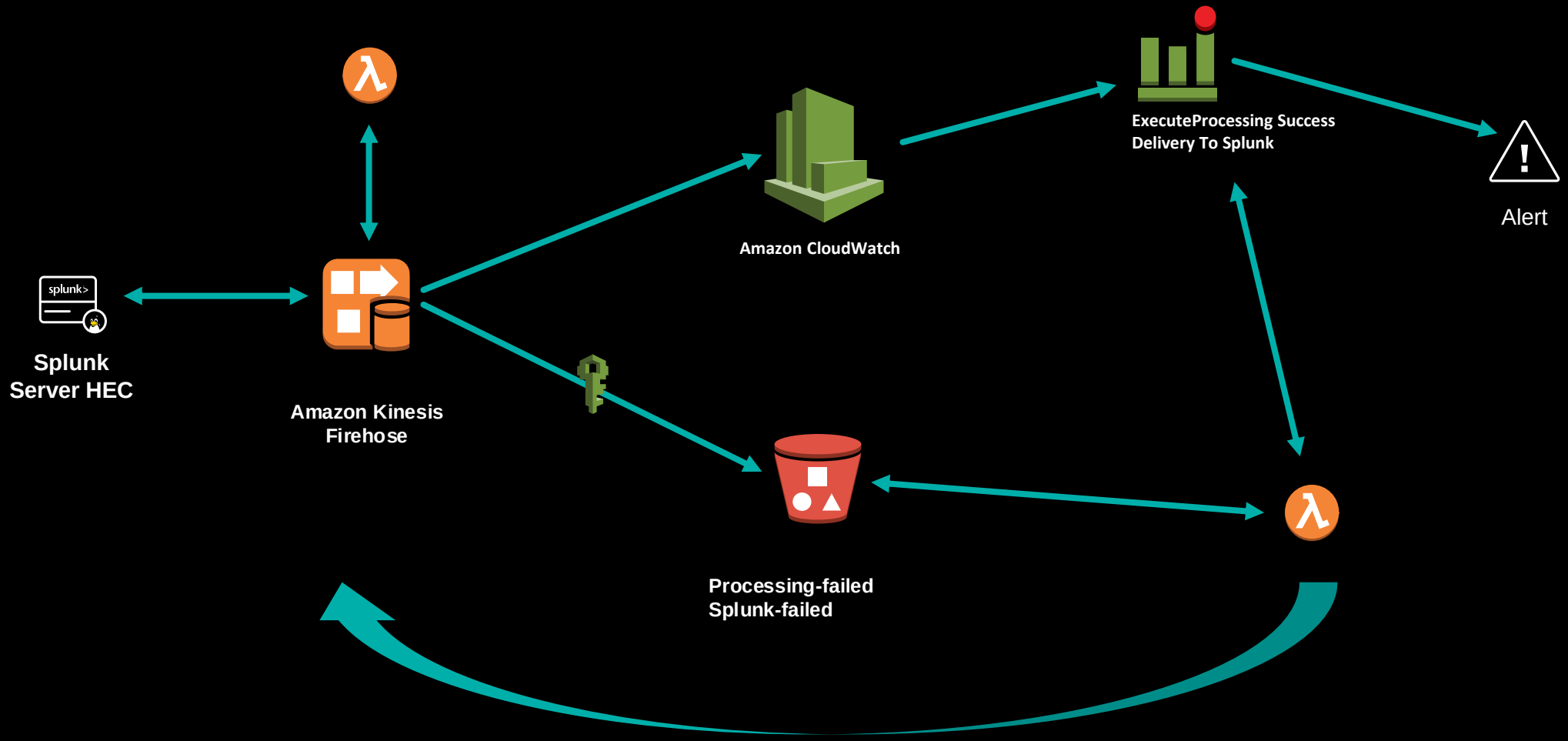
Automating for Bad Times

splunk> .conf18

Error Types

Know Your Enemy (Sun Tzu)

- Transient
  - Delete/Add lag for Lambda function (use aliases)
  - Delivery Stream had some difficulty (extremely rare occurrence)
  - Service is HEALTHY
- Persistent
  - New version of Lambda has logical errors (Processing Error)
  - HEC Key was removed (Splunk Error)
  - SG was incorrectly updated (Splunk Error)
  - Service is NOT HEALTHY
- Both Are Sent to the Configured S3 Bucket

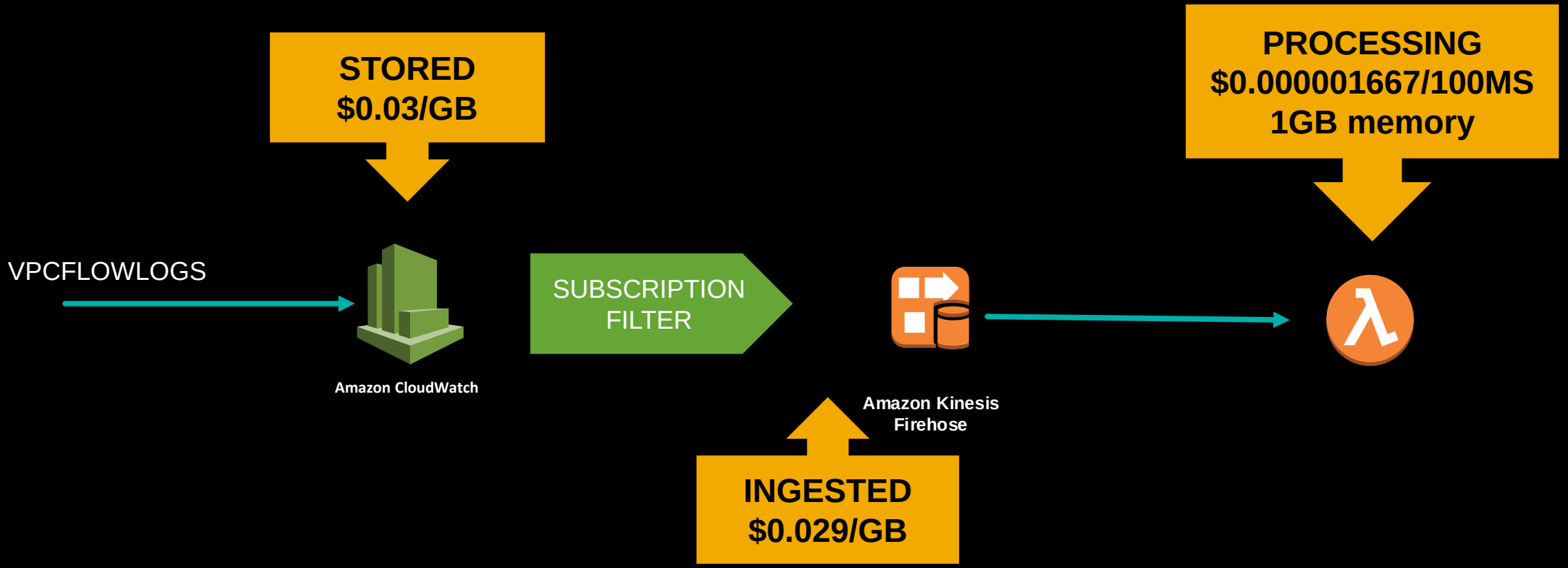


When DS State is Healthy

Pricing Out the Costs of a Solution

“Always the Dollars…” Joe Pesci, Casino

splunk> .conf18



Let's Do the Math

100 GB of VPCFLOWLOGS into Splunk per Day

- Processing
  - $50K \frac{runs}{day} x \frac{365\ day}{year} x \frac{900ms}{run} x \frac{\$0.000001667}{100ms}$ = $274/year
- Ingested

50% Compression Rate

Filtering 20% of Events

100GB to Splunk>

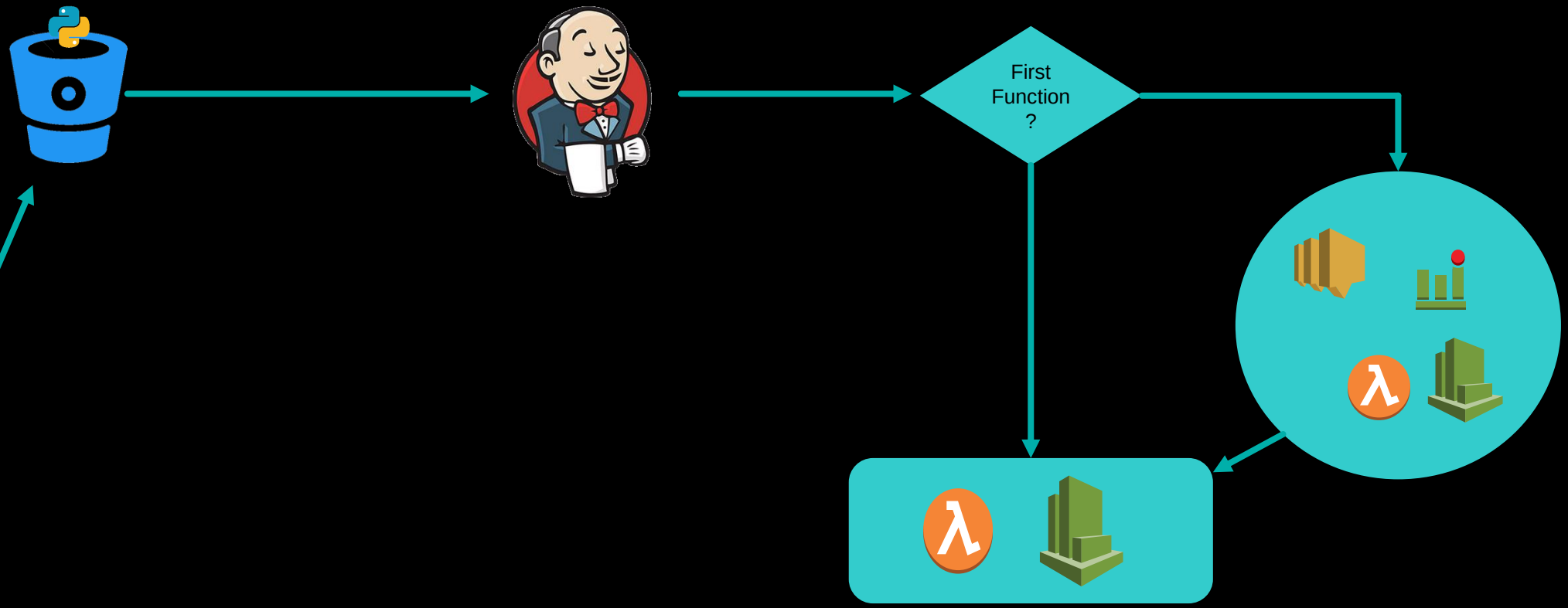
  - $42 \frac{GB}{Day}\ X \frac{\$0.029}{GB}\ X\ 365 \frac{Days}{Year}$ = $445/year
- Stored (savings)
  - $42 \frac{GB}{Day} X \frac{\$0.03}{GB}\ X\ 365 \frac{Days}{Year} X \frac{11}{12}$ = $422
- Total
  - $\$445 + \$274\ - \$422 = \$297$ + error processing

splunk> .conf18

Putting it All Together

What is a Conductor with No Orchestra?

splunk> .conf18



splunk>

3 Key Takeaways

Because there is *always* three

1. Using AWS Firehose with a Splunk Destination is a Flexible, Cost-Effective, Fast, and Reliable method to ingest AWS logs

2. Take the time to Design a Robust, Reusable, and Hands-Free Eco-System

3. Let Splunk and AWS do what they do best individually, will provide greater ROI and User Acceptance

Thank You

Don't forget to rate this session in the .conf18 mobile app