CLOUD NATIVE
COMPUTING FOUNDATION

Fluent Bit v1.5
Webinar

July 17, 2020

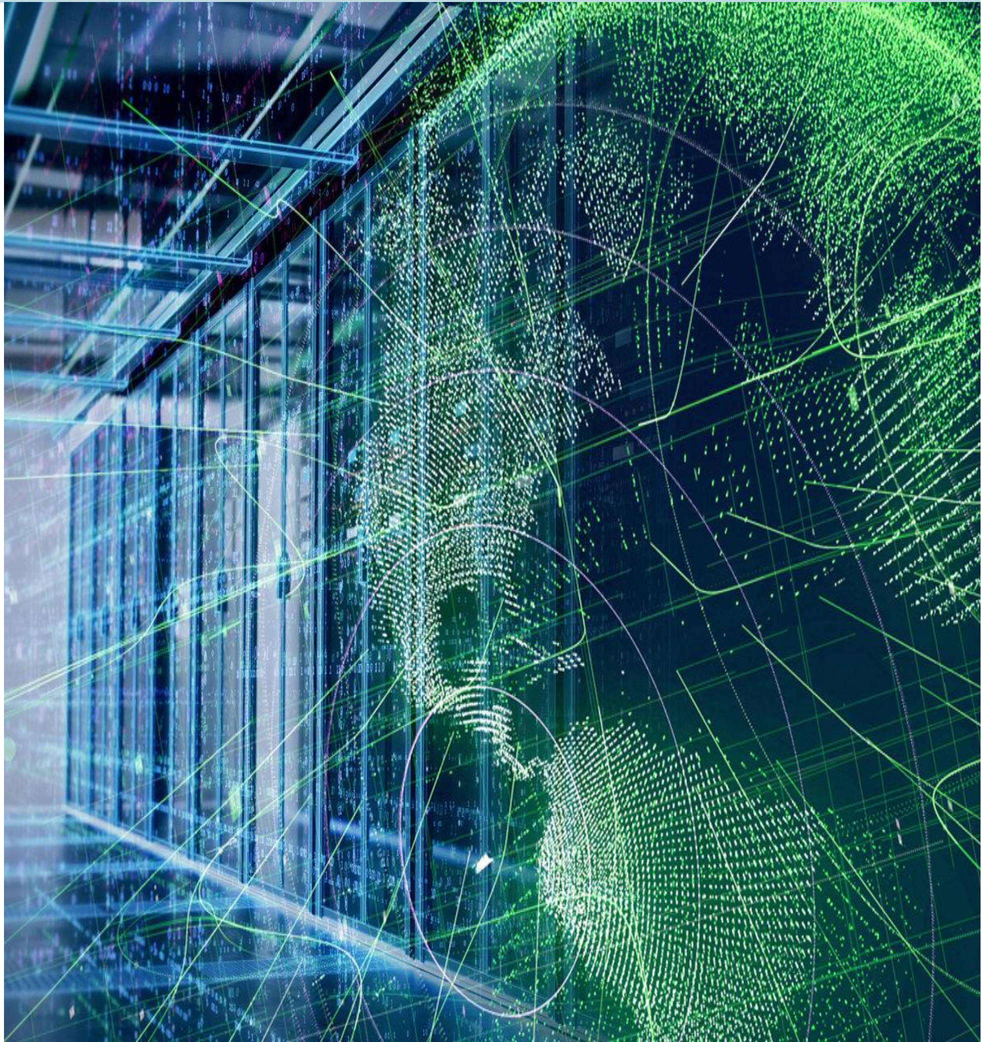
Eduardo Silva
eduardo@treasure-data.com
@edsiper

Wesley Pettit
wppttt@amazon.com
@PettitWesley

Masoud Koleini
masoud.koleini@arm.com
@koleini

Agenda

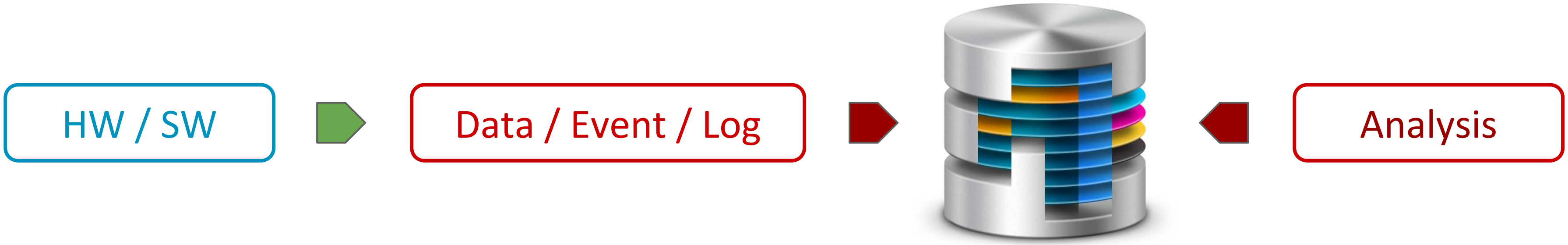
- Introduction to Fluent Bit
- Fluent Bit v1.5
- Migrating AWS plugins from Go to **C**
- Stream Processing
- Q&A



Introduction to **Fluent Bit**

End to End

Communication Workflow



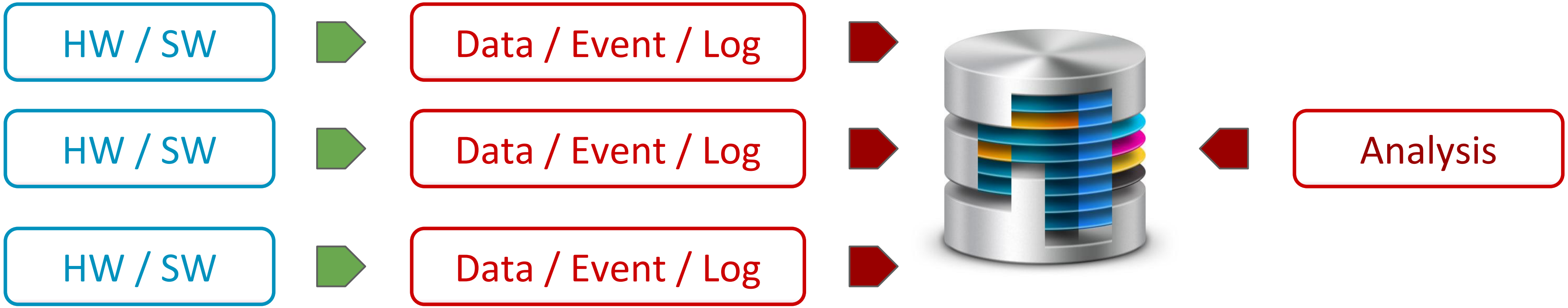
HW / SW

Data / Event / Log

Analysis

Data Ingestion

Performance Penalties



HW / SW

HW / SW

HW / SW

Data / Event / Log

Data / Event / Log

Data / Event / Log

Analysis

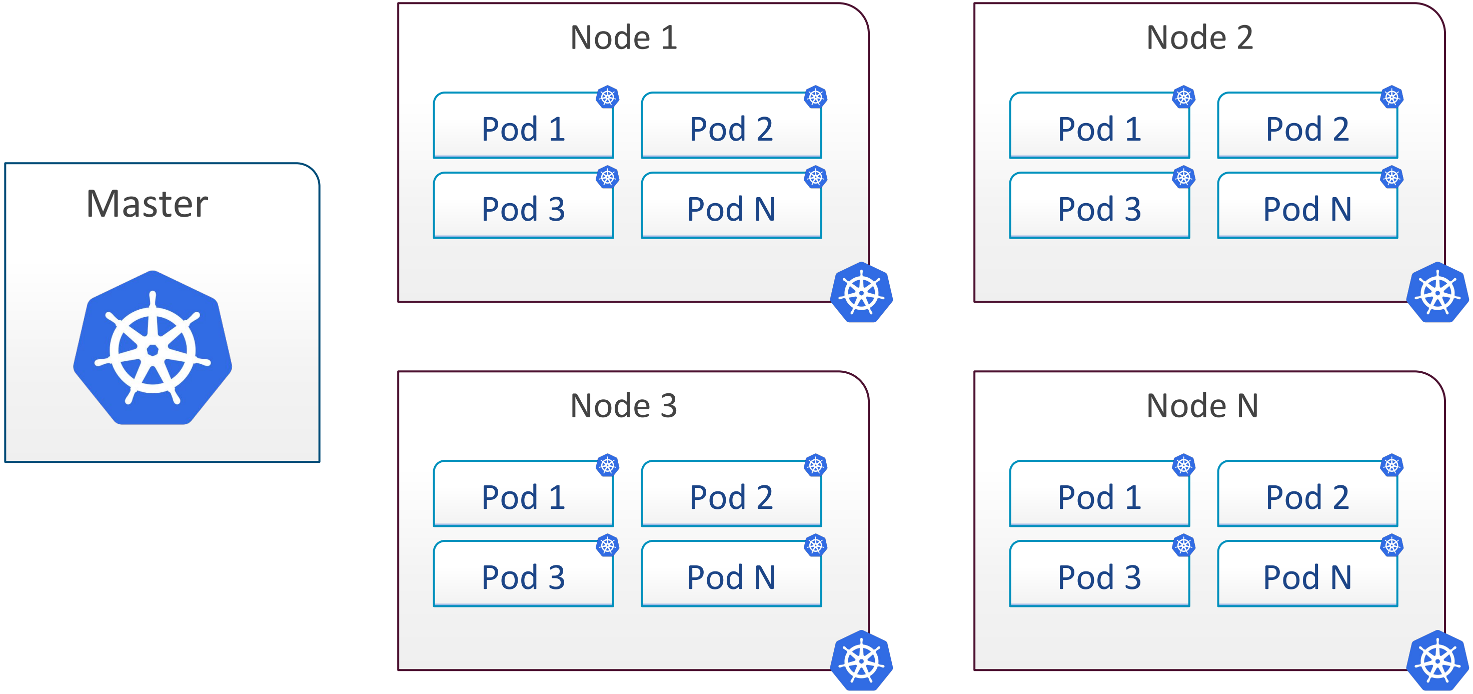
Data Challenges

Multiple Sources of Information

- Network protocols: TCP / UDP
- File system, common log files
- Systemd / Journald
- Others

Data Challenges

Distributed Environments: e.g: Kubernetes



Data & Logging Challenges

.. and each one with different data formats, structure ?

- Apache Logs

  ```
  [14/Mar/2019:23:43:52 +0000] GET /Frasera HTTP/1.0 500 2216
  ```

- MySQL

  ```
  2019-04-30T21:32:39.095880Z 0 [Note] InnoDB: Mutexes use GCC atomic builtins
  ```

- JSON Maps

  ```
  {“log”: “Hey GEC!”, “stream”: “stdout”, “time”: “2019-05-07T10:03:11.33507113Z”}
  ```

- Many others...!

Data & Logging Challenges

.. and each one with different data formats

- Apache Logs

```
[14/Mar/2019:23:43:52 +0000] GET /Frasera HTTP/1.1 500 2216
```

- MySQL

```
2019-04-30T21:32:39.095880Z 0 [Note] InnoDB: Mutexes use GCC atomic builtins
```

- JSON Maps

```
{ “log”: “Hey GEC!” , “stream”: “stdout” , “time”: “2019-05-07T10:03:11.33507113Z” }
```

- Many others...!

Before Data Analysis we need:

Ideal tool

- Collect data from **different sources**
- Convert from **unstructured** to **structured** messages
- Data **enrichment** & filtering
- Delivery: **multiple destinations** like databases or cloud services

Apache License v2.0

CNCF Ecosystem

Fluent Bit is a **CNCF** sub-project under the umbrella of **Fluentd**

fluentbit

CLOUD NATIVE
COMPUTING FOUNDATION

fluentd

About

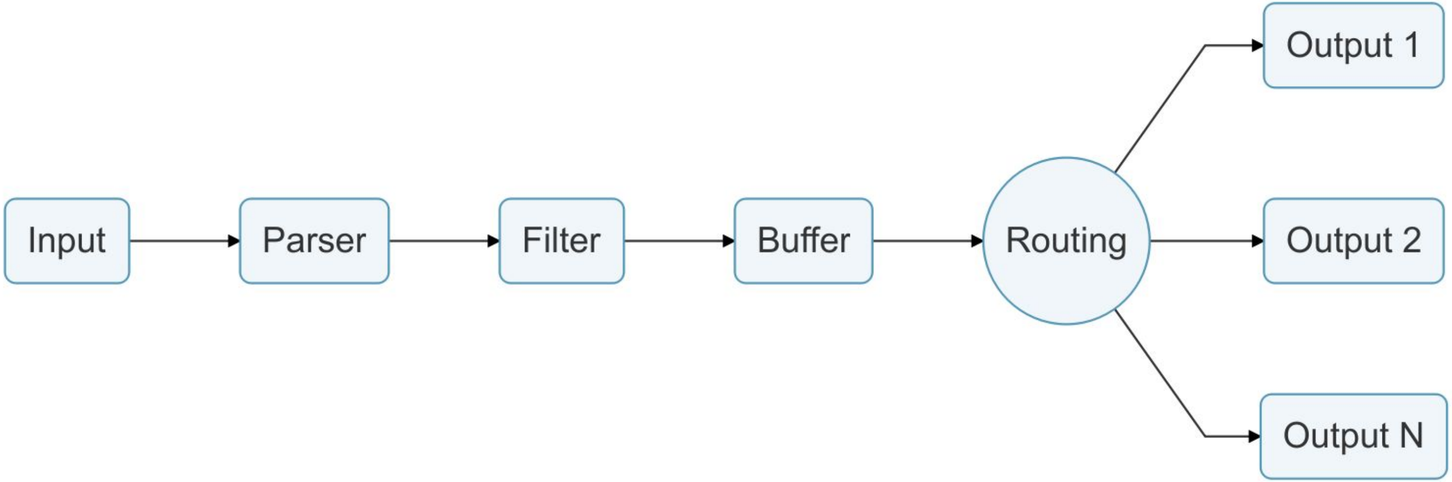
Fluent Bit

- Started in 2015
- Origins: Lightweight log processor for Embedded Linux
- Quickly evolved as a solution for the Cloud space
- Apache License v2.0

Fluent Bit

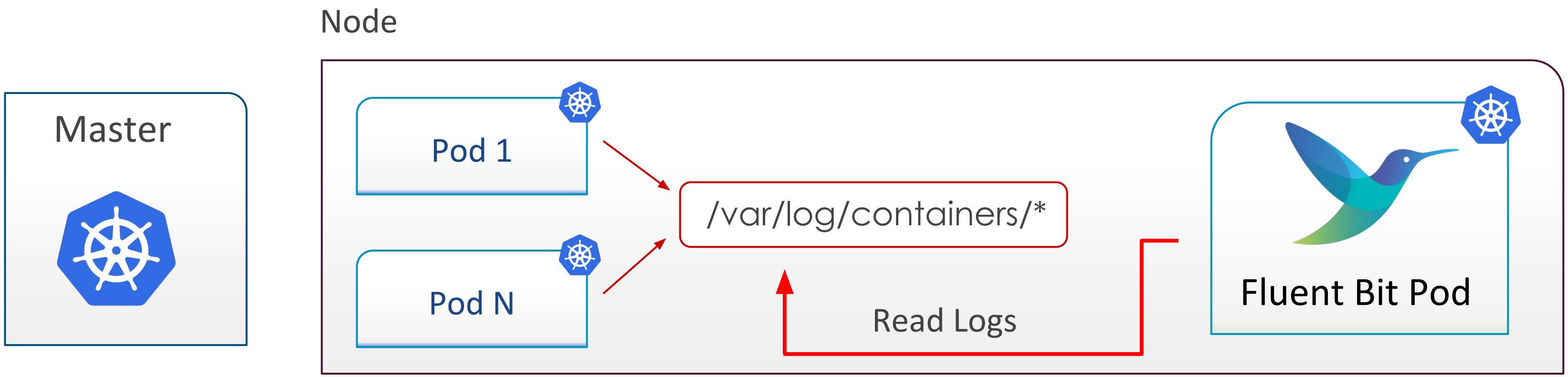
Design & Internals

- Written in **C** language
- **Low** memory and CPU footprint (memory around **600KB**)
- Pluggable Architecture (**> 60** plugins available)
- Built-in security: TLS on Network I/O


Input
Parser
Filter
Buffer
Routing
Output 1
Output 2
Output N

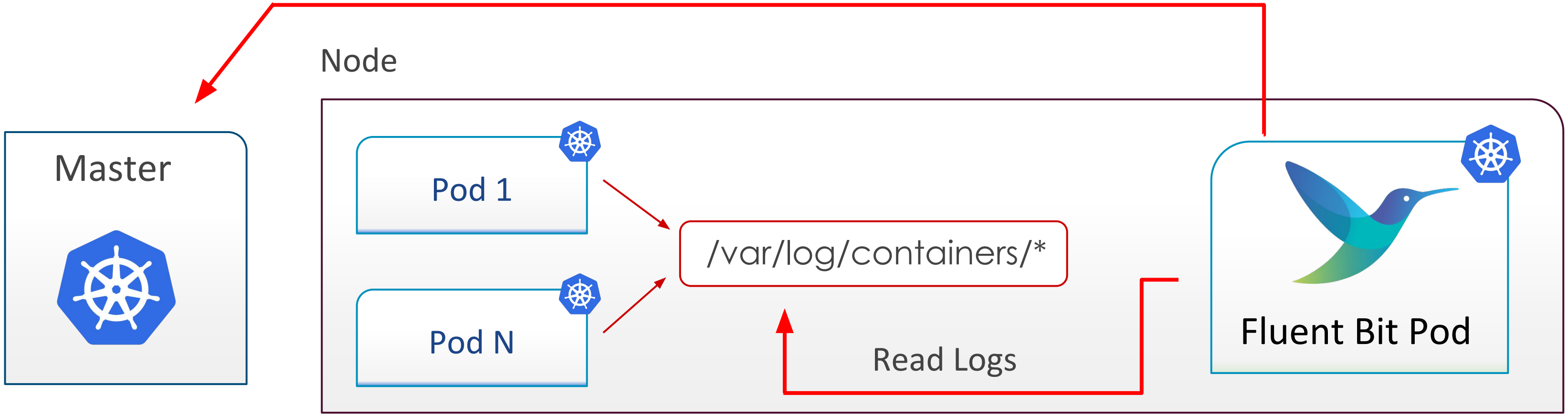
Logging Processing in Kubernetes

Read Logs from the Filesystem or Journald



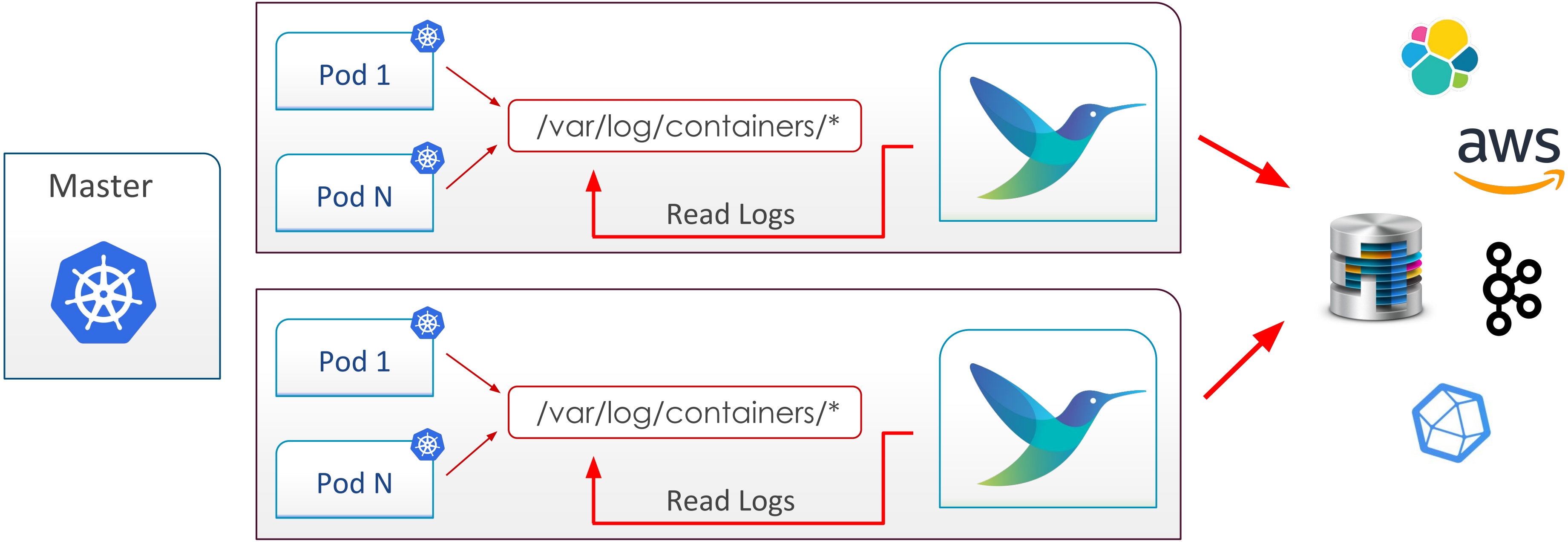
Logging Processing in Kubernetes

Read Logs from the Filesystem or Journald



Logging Processing in Kubernetes

Read Logs from the Filesystem or Journald



Fluent Bit v1.5

by Eduardo Silva

Fluent Bit v1.5

Core: Networking and KeepAlive

- Connect Timeouts
- Custom Source Address / network interface
- Keep Alive for TCP and TLS Sessions
- Keep Alive Idle Timeouts

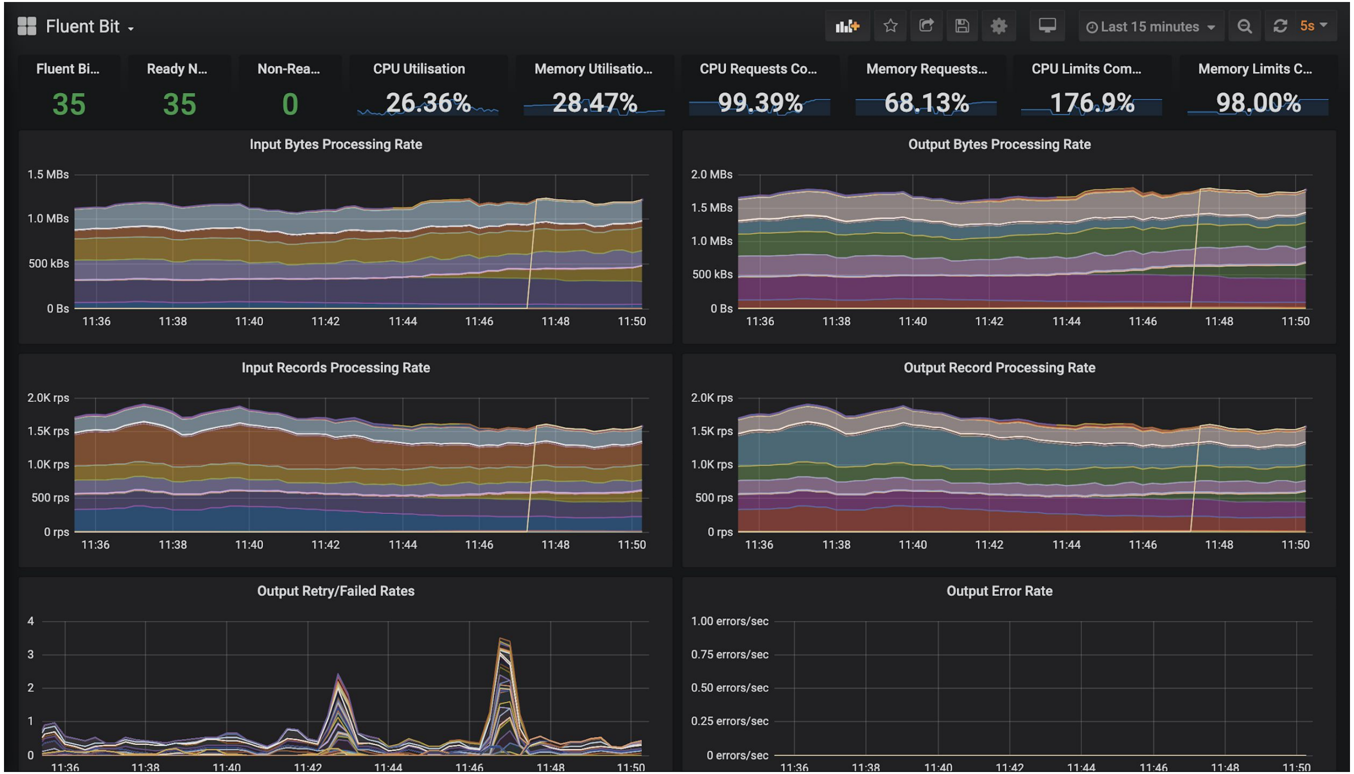
Fluent Bit v1.5

Windows Support Improvements

- Windows Service Support
- Windows Event Log Input Plugin: full UTF-8 encoding
- Kubernetes Support

Fluent Bit v1.5

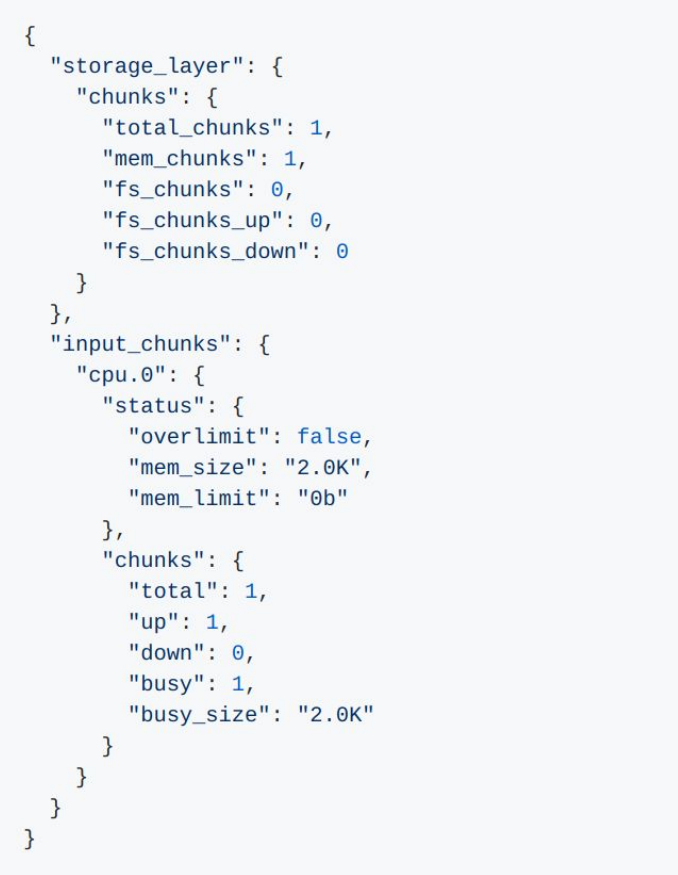
Monitoring: Grafana Dashboards



Fluent Bit v1.5

Monitoring: Storage Metrics

- Storage Layer Chunks
  - Memory
  - File System
- Input plugin granular stats



```
{
  "storage_layer": {
    "chunks": {
      "total_chunks": 1,
      "mem_chunks": 1,
      "fs_chunks": 0,
      "fs_chunks_up": 0,
      "fs_chunks_down": 0
    }
  },
  "input_chunks": {
    "cpu.0": {
      "status": {
        "overlimit": false,
        "mem_size": "2.0K",
        "mem_limit": "0b"
      },
      "chunks": {
        "total": 1,
        "up": 1,
        "down": 0,
        "busy": 1,
        "busy_size": "2.0K"
      }
    }
  }
}
```

Fluent Bit v1.5

New Enterprise Connectors

Amazon Elasticsearch Service

Amazon CloudWatch

logdna

New Relic®

Fluent Bit v1.5

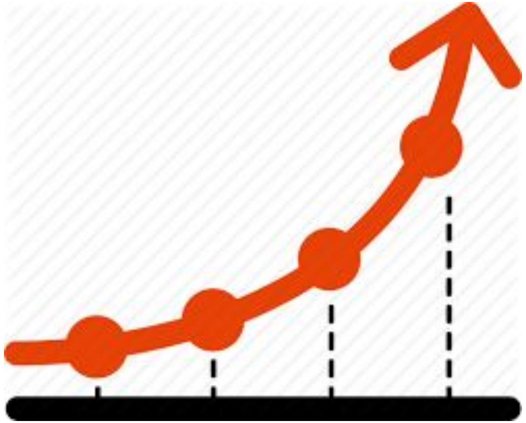
Highly Improved: Google Stackdriver

- Kubernetes resources types: containers, pods and nodes
- Labels as special fields
- Add associated operation as a special field

Google Cloud

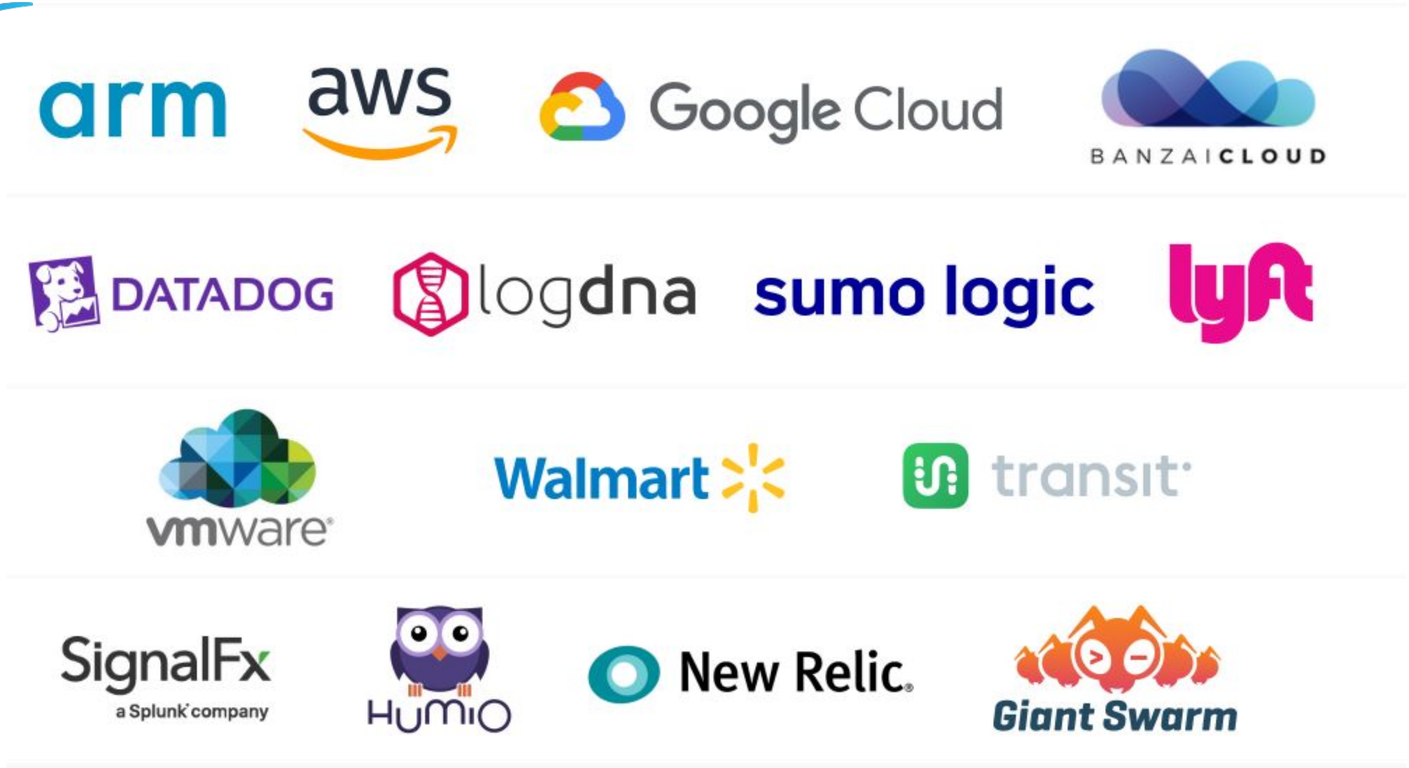
Project Status

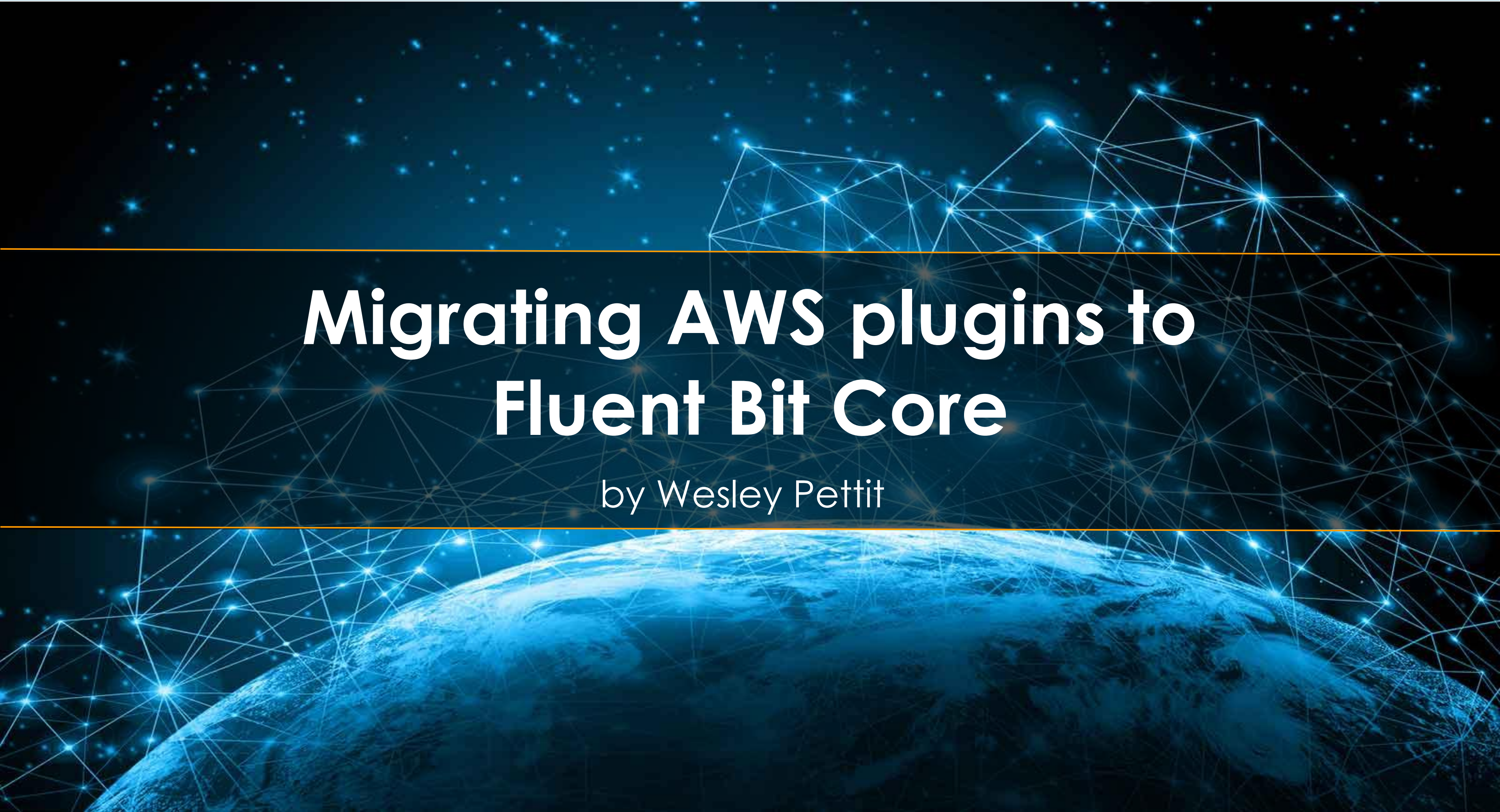
Adoption as of **July 2020**



	Deployments
2020	**106 Million**
2019	**62 Million**
2018	**18 Million**

Enterprise Adoption





Migrating AWS plugins to Fluent Bit Core

by Wesley Pettit

Last Year: Go plugins

Launched AWS for Fluent Bit with Go plugins

- Amazon CloudWatch Logs
- Amazon Kinesis Data Firehose
- Amazon Kinesis Data Streams

Go plugins: Why ?

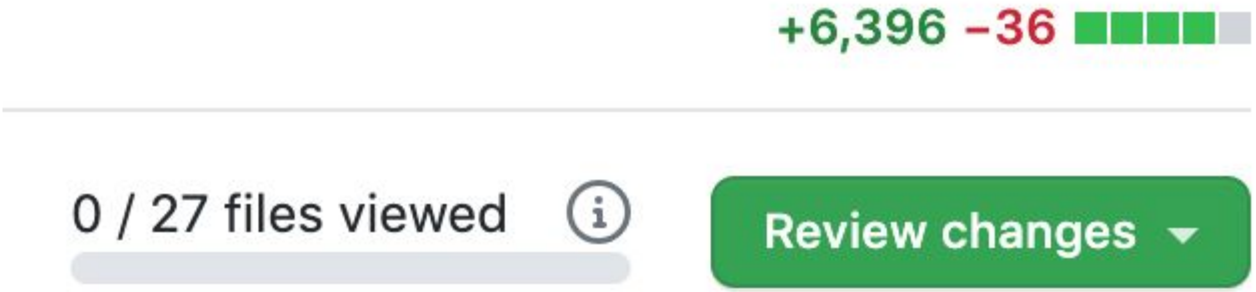
- Primary reason: AWS SDK for Go
- Secondary reason: Speed of Development

AWS Authentication

- Custom Auth Algorithm: Sigv4 Signing
- Many sources for Credentials
    - ECS IAM Roles for Tasks
    - EKS IAM Roles for Service Accounts
    - EC2 Instance Role
    - Local AWS Profile in shared credential file
    - Environment Variables
    - STS Assume Role

New in Fluent Bit 1.5: Core C Library for AWS Auth

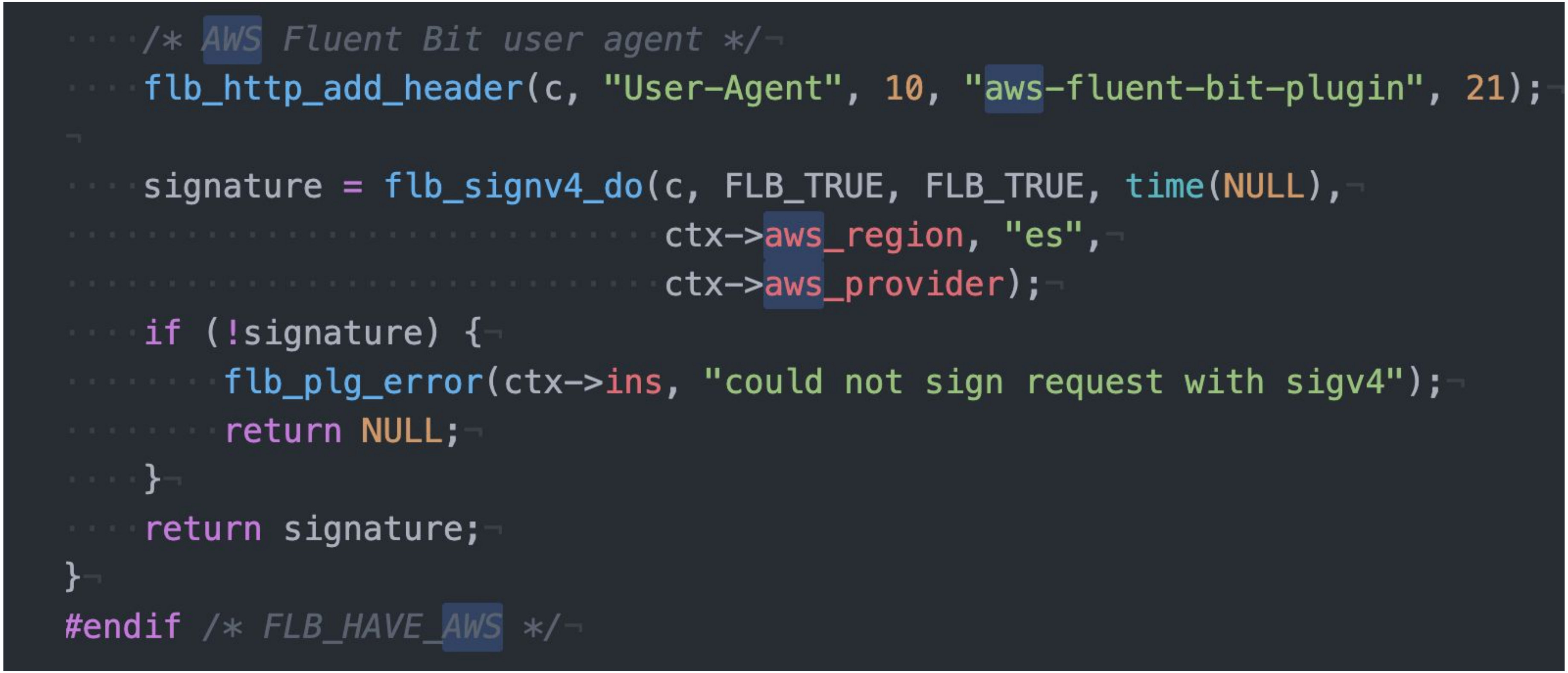
Custom Library that uses Fluent Bit’s built in HTTP Client and concurrency features



+6,396 −36

0 / 27 files viewed

New in Fluent Bit 1.5: Core C Library for AWS Auth



```
    /* AWS Fluent Bit user agent */
    flb_http_add_header(c, "User-Agent", 10, "aws-fluent-bit-plugin", 21);

    signature = flb_signv4_do(c, FLB_TRUE, FLB_TRUE, time(NULL),
                              ctx->aws_region, "es",
                              ctx->aws_provider);
    if (!signature) {
        flb_plg_error(ctx->ins, "could not sign request with sigv4");
        return NULL;
    }
    return signature;
}
#endif /* FLB_HAVE_AWS */
```

Amazon ElasticSearch Service Support

Fluent Bit Configuration for AWS Elasticsearch

```
[OUTPUT]
    Name        es
    Match       *
    Host        vpc-test-domain-ke7thhzo7ite7y.us-west-2.es.amazonaws.com
    Port        443
    Index       my_index
    Type        my_type
    AWS_Auth    On
    AWS_Region us-west-2
    TLS         On
```

Amazon ElasticSearch Service Support

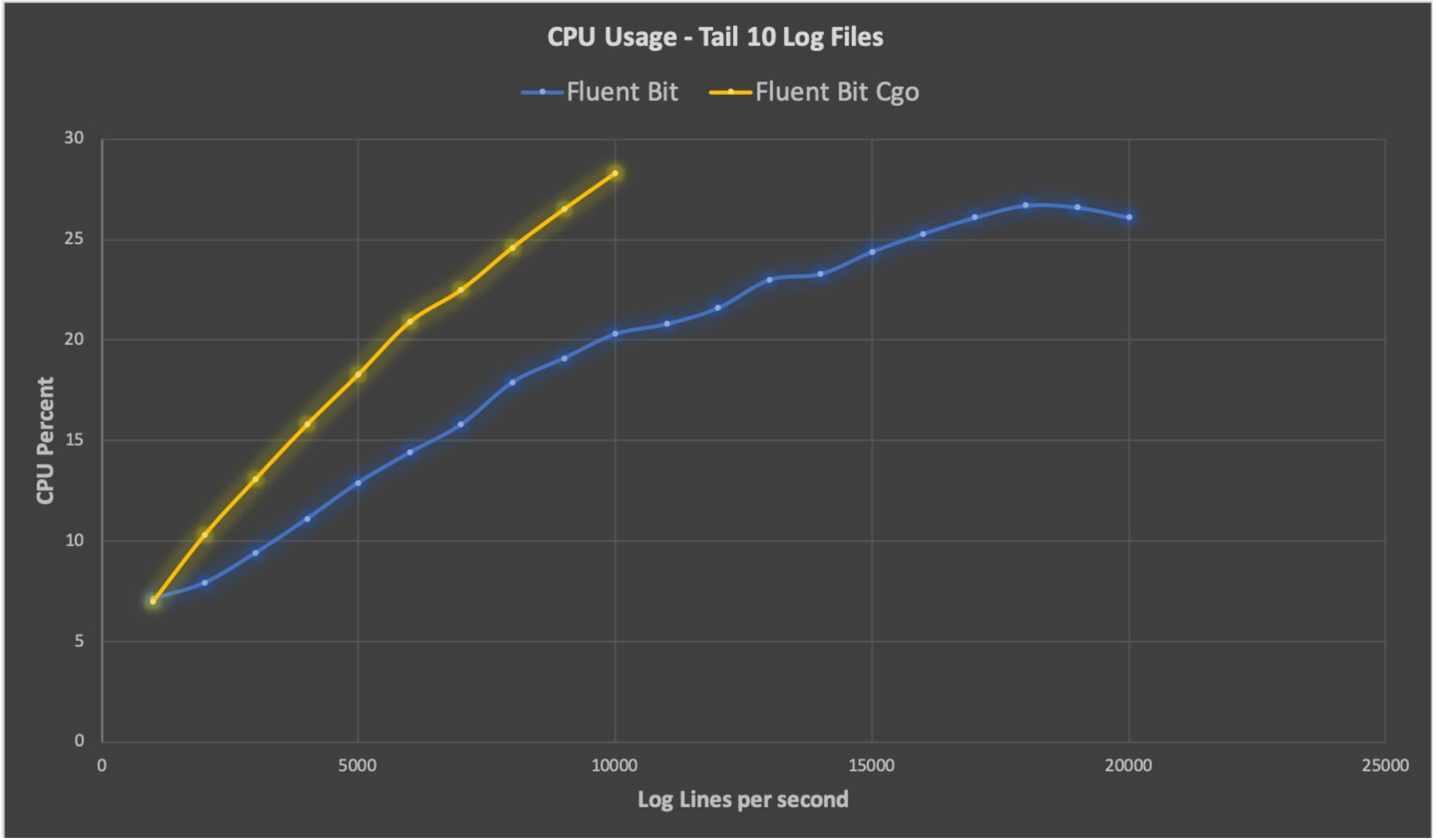
Fluent Bit Configuration for AWS Elasticsearch with Role

```
[OUTPUT]
    Name            es
    Match           *
    Host            vpc-test-domain-ke7thhzo7ite7y.us-west-2.es.amazonaws.com
    Port            443
    Index           my_index
    Type            my_type
    AWS_Auth        On
    AWS_Region      us-west-2
    AWS_Role_ARN arn:aws:iam::11111111111:role/provider-testing
    TLS             On
```

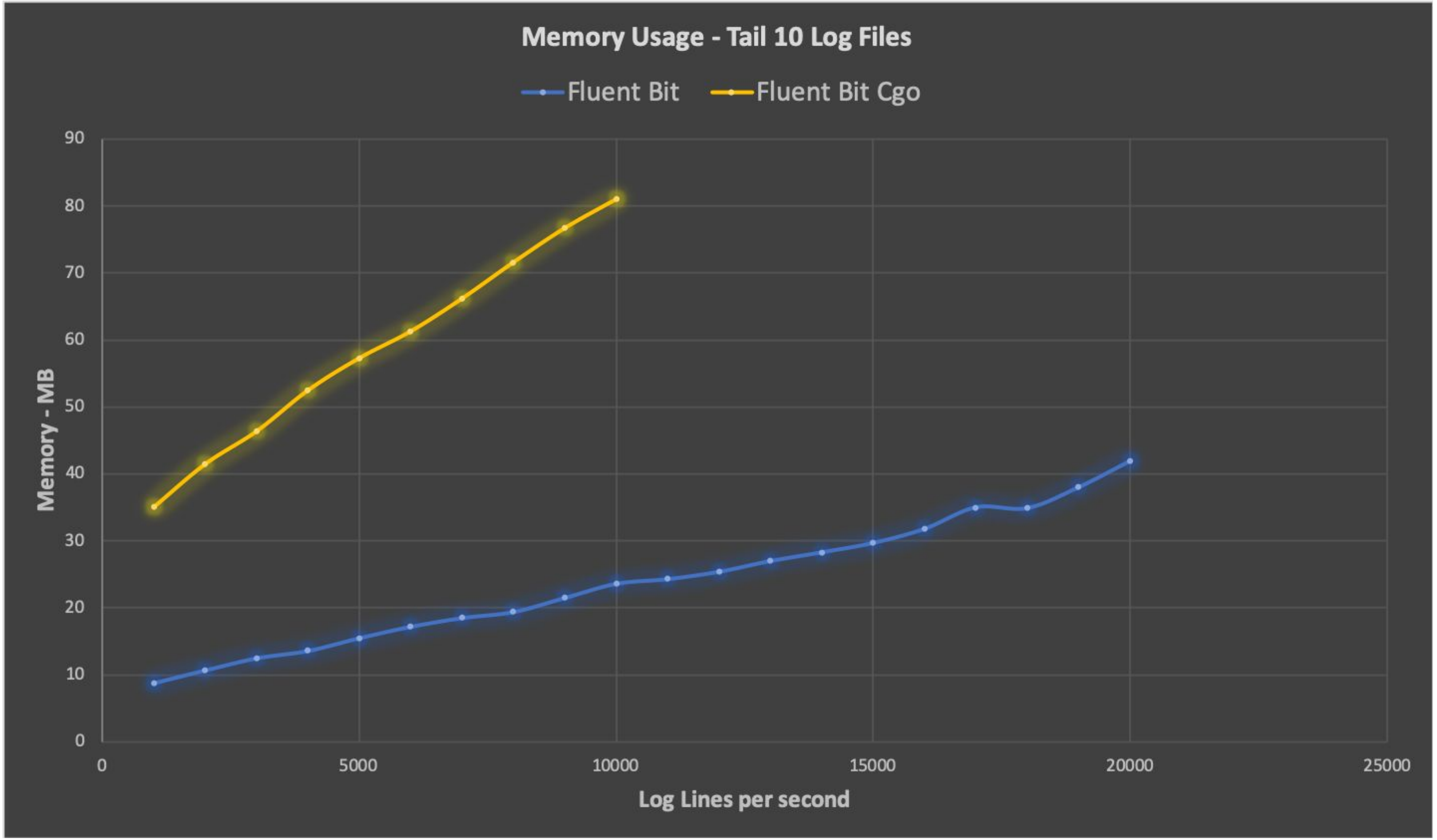
New CloudWatch Logs Plugin in C

```
[OUTPUT]
    Name              cloudwatch_logs
    Match             *
    region            us-east-1
    log_group_name    fluent-bit-cloudwatch
    log_stream_prefix from-fluent-bit-
    auto_create_group On
```

New CloudWatch Plugin: Performance



New CloudWatch Plugin: Performance



Long term plan

- Rewrite all 3 Go plugins in C in core of Fluent Bit
- Deprecate Go Plugins
- Alias Go plugin names to C plugins
- Timeline uncertain

What am I working on now?

- Amazon S3 output support
- If you have thoughts or ideas, post on GitHub

S3 Support

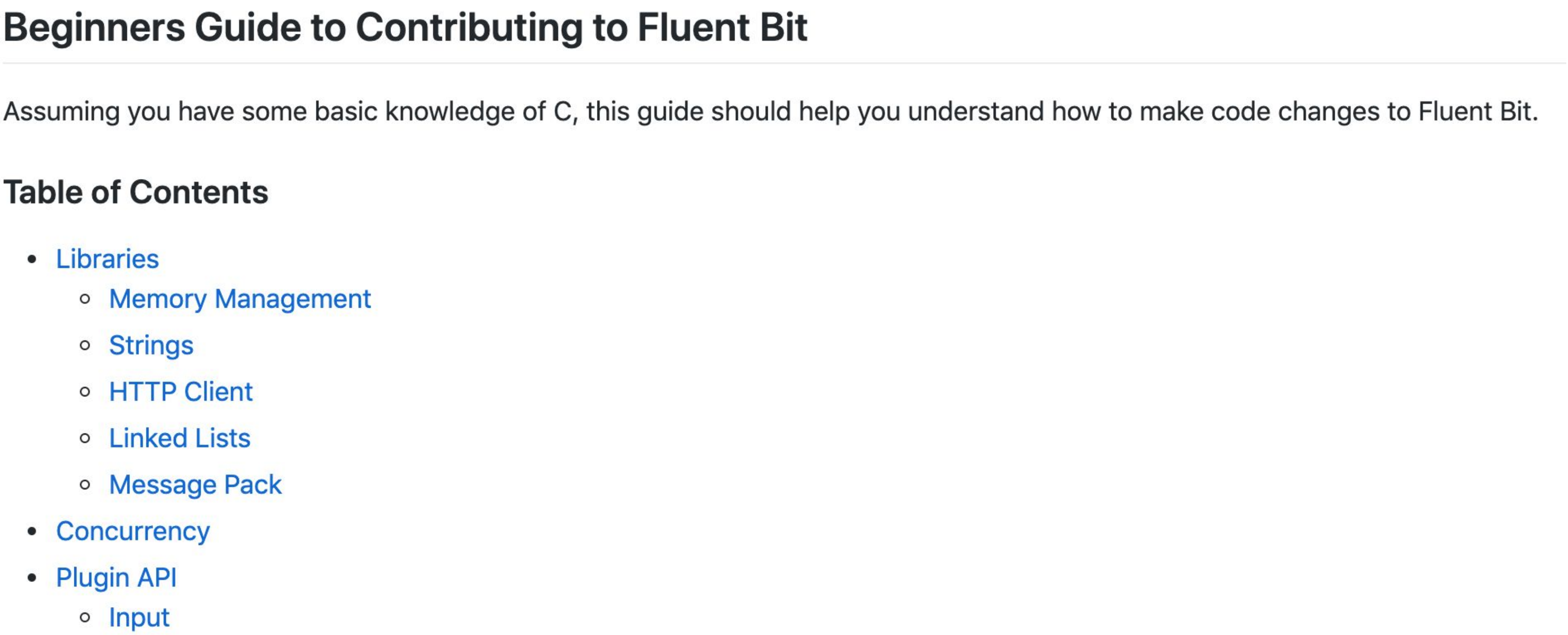
- Multipart Uploads
  - Send data in small chunks frequently
  - Minimal local buffering

```
[OUTPUT]
      Name      s3
      Match     *
      bucket    my-bucket
      region    us-west-2
      file_size 250M
```

Fluent Bit + AWS: How to get help

- Open issue on fluent/fluent-bit and mention @PettitWesley
- **Preferred**: Open issue on aws/aws-for-fluent-bit repo

Contributing: Learning Fluent Bit Code

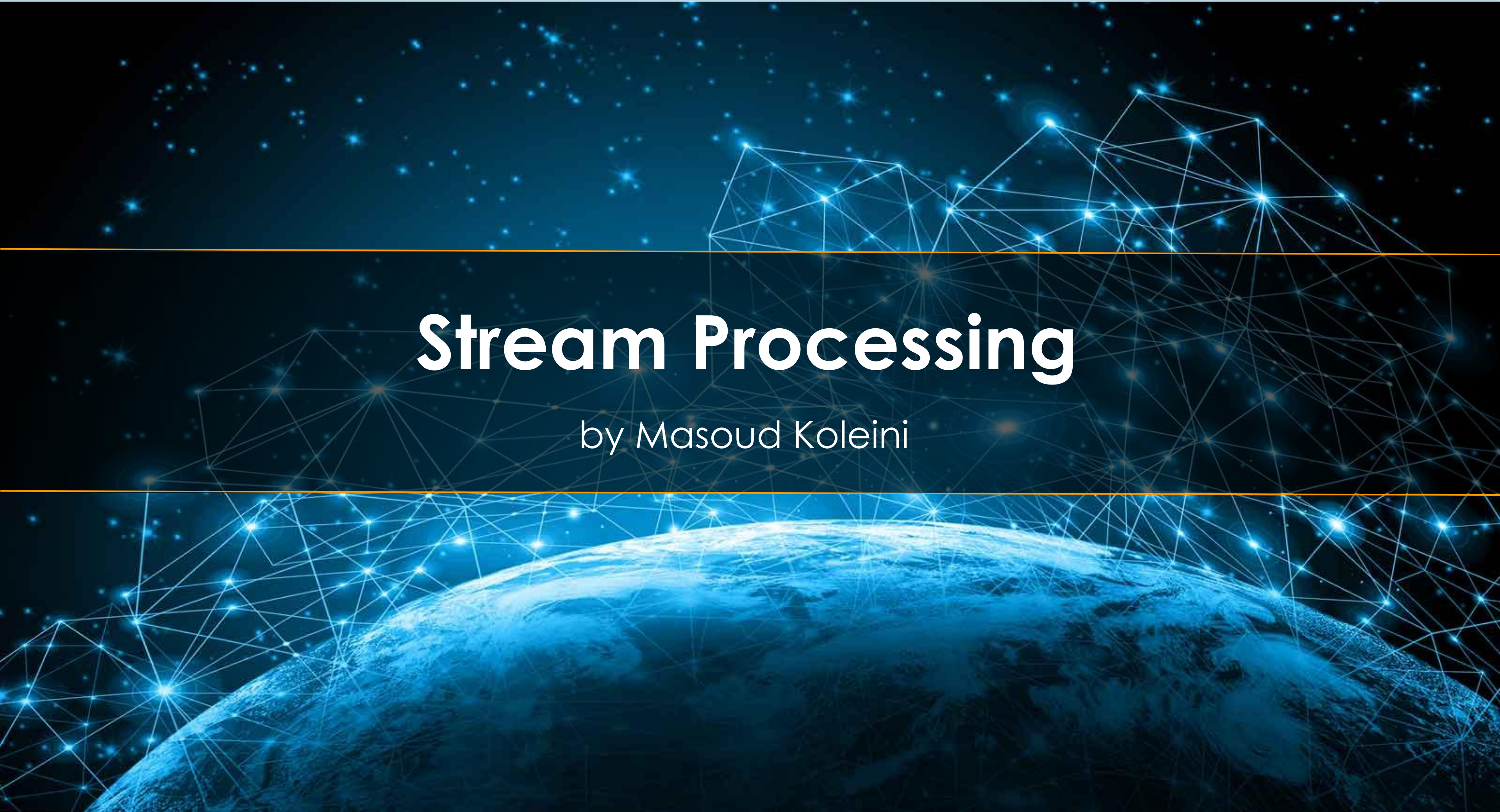


Beginners Guide to Contributing to Fluent Bit

Assuming you have some basic knowledge of C, this guide should help you understand how to make code changes to Fluent Bit.

Table of Contents

- Libraries
  - Memory Management
  - Strings
  - HTTP Client
  - Linked Lists
  - Message Pack
- Concurrency
- Plugin API
  - Input



Stream Processing

by Masoud Koleini

Stream Processing

> “ It’s the ability to perform
>
> Data Processing while **it** Still in Motion ”

Stream Processing

Events

- **Records** emitted by applications, services or hardware
- Events are **structured** messages
- Composition
    - Timestamp: specify when the event was created
    - Message: the event informational data

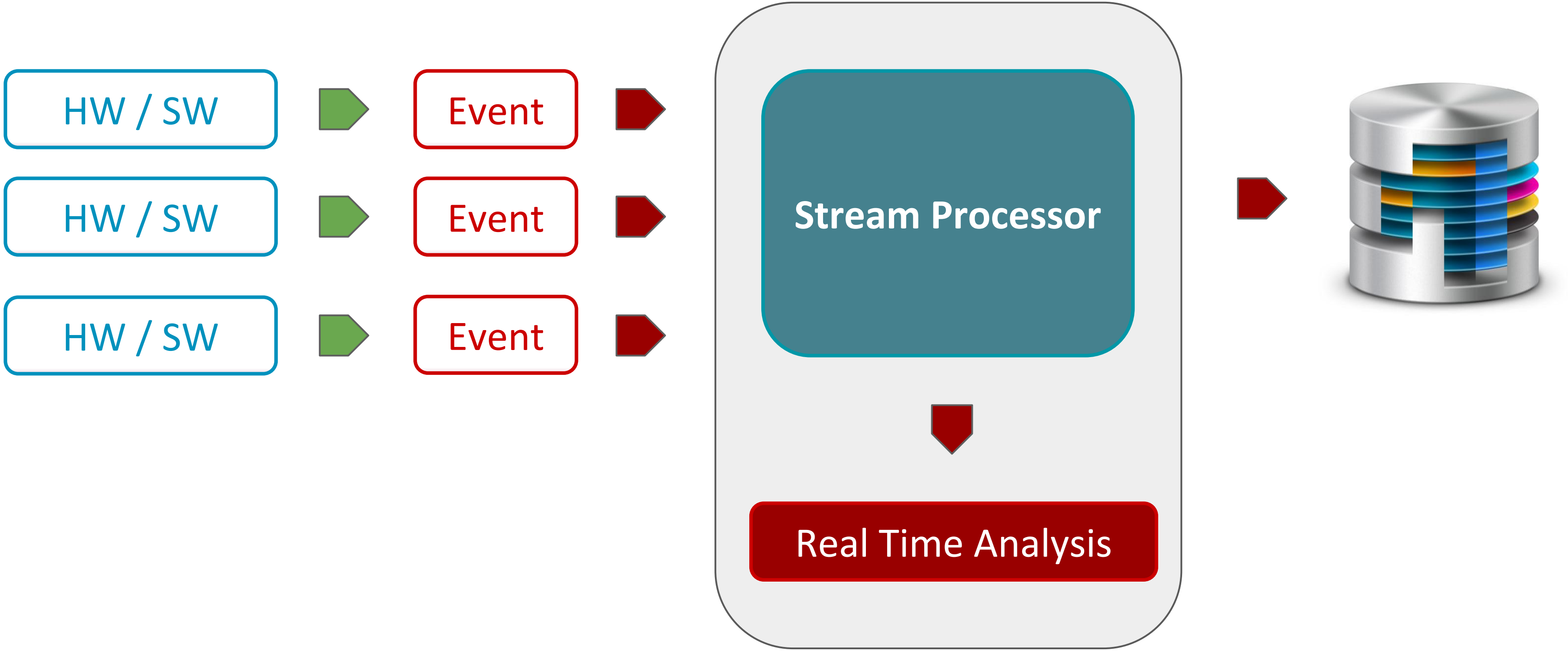
Stream Processing

General Goals

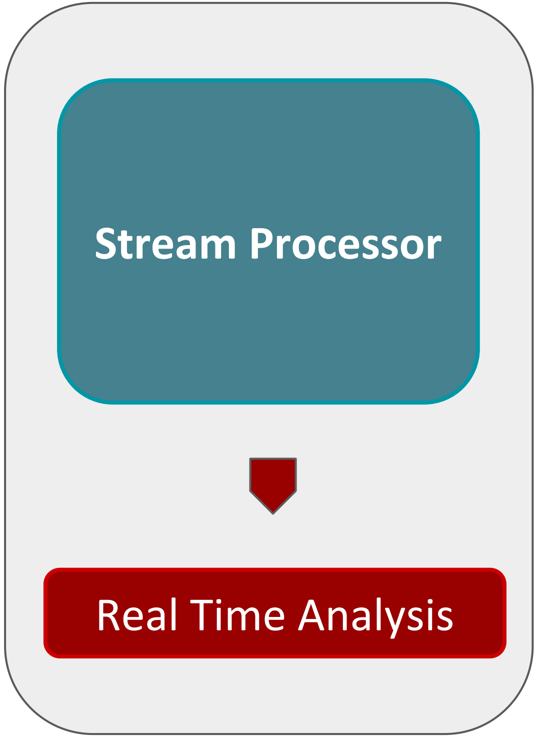
- Fast and Lightweight Data Processing
- No Tables
- No Indexing / Index-Free
- Easy to use programming model

How ?

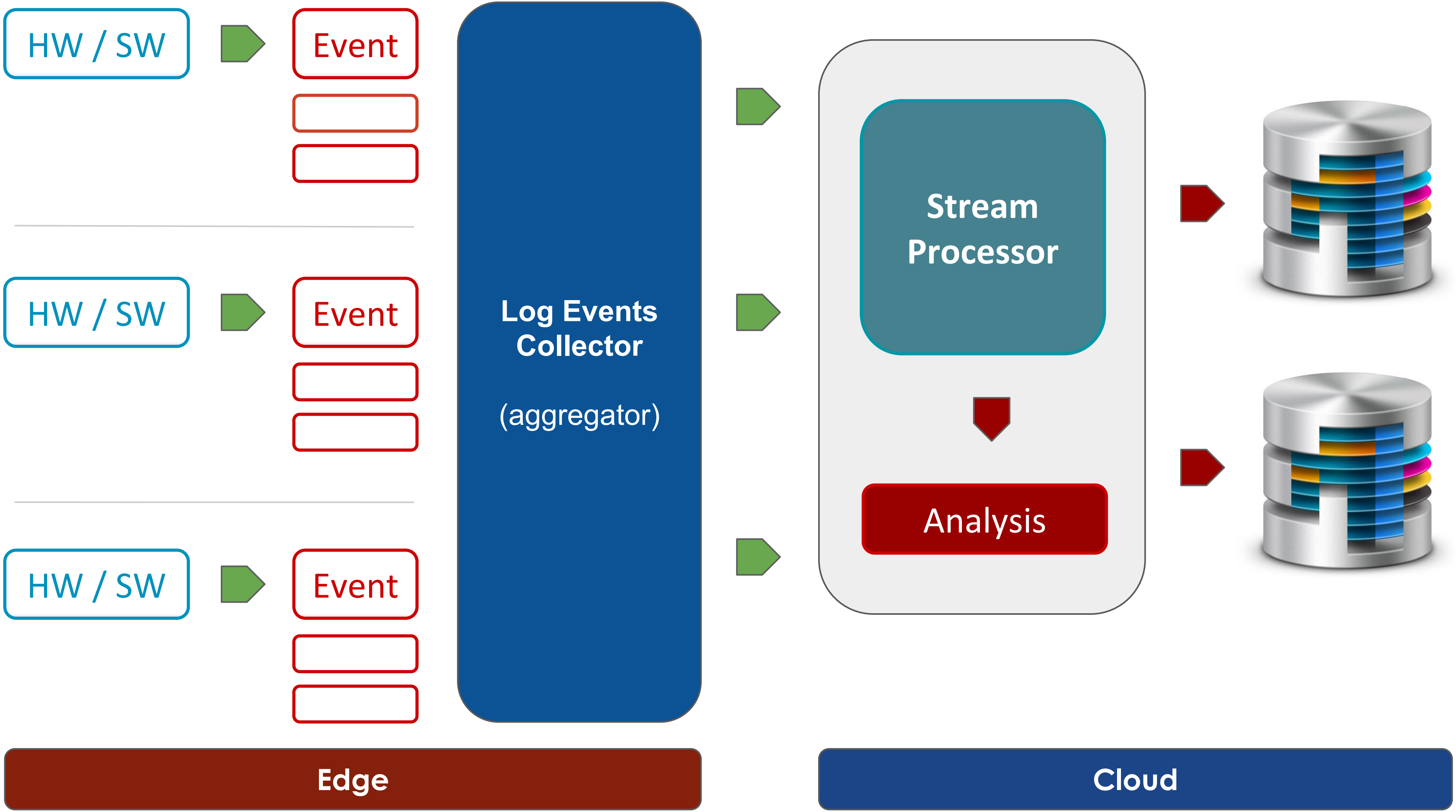
Stream Processor

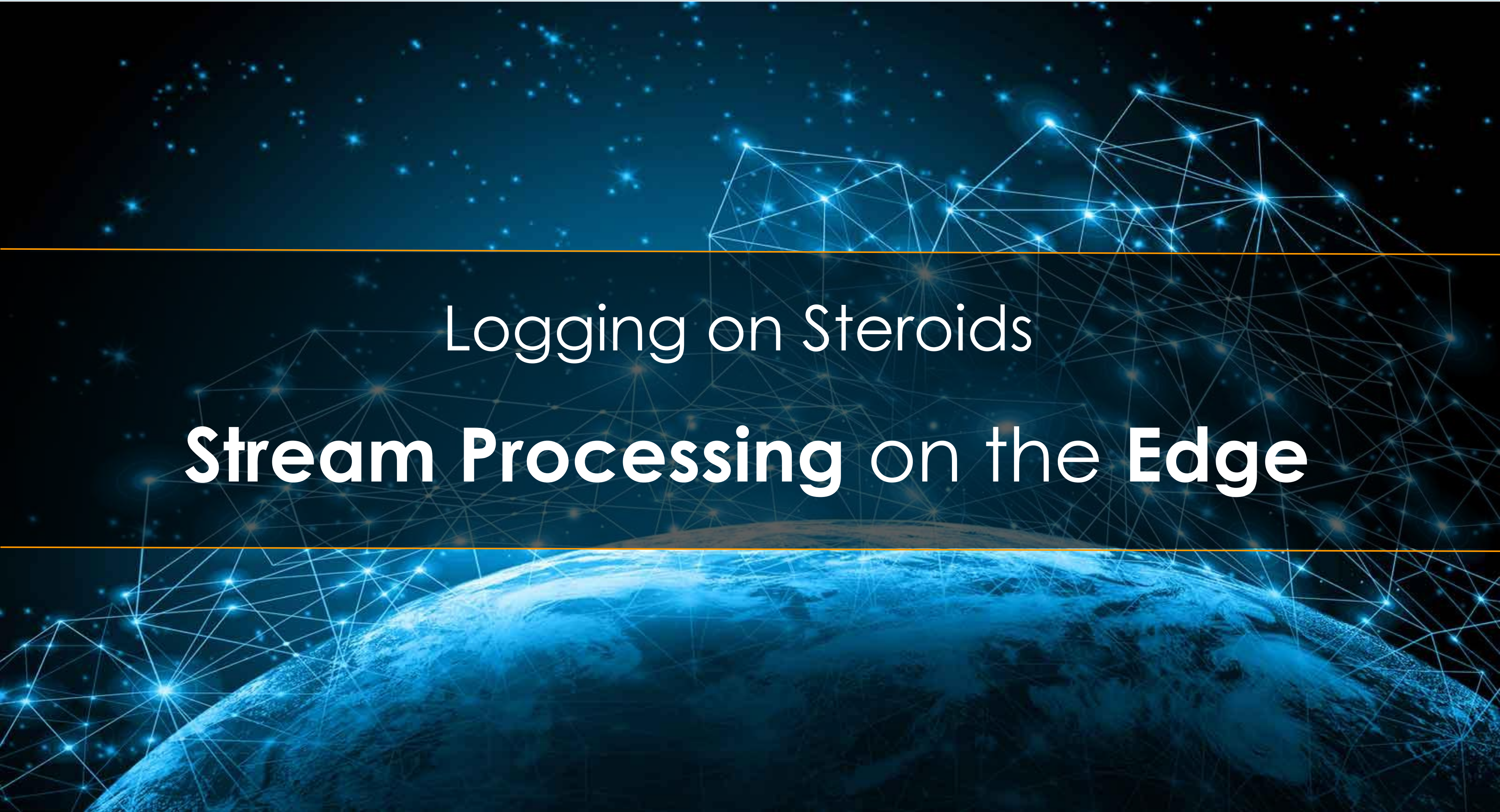


Stream Processor



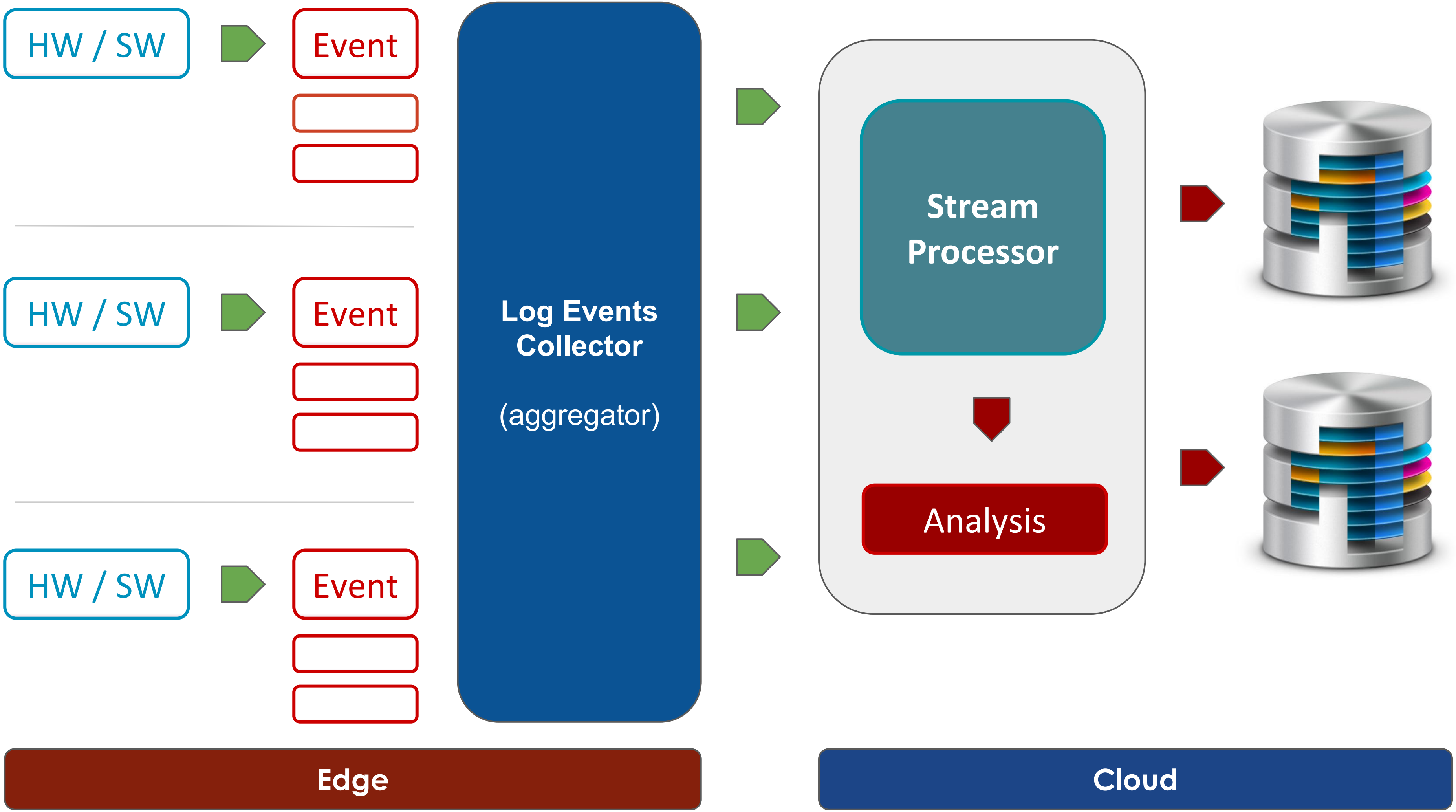
- Receive **structured** events (records)
- Expose a Query Language
  - Keys selection
  - Filtering
  - Aggregation Functions
  - Events Routing
- Do processing **in-memory**

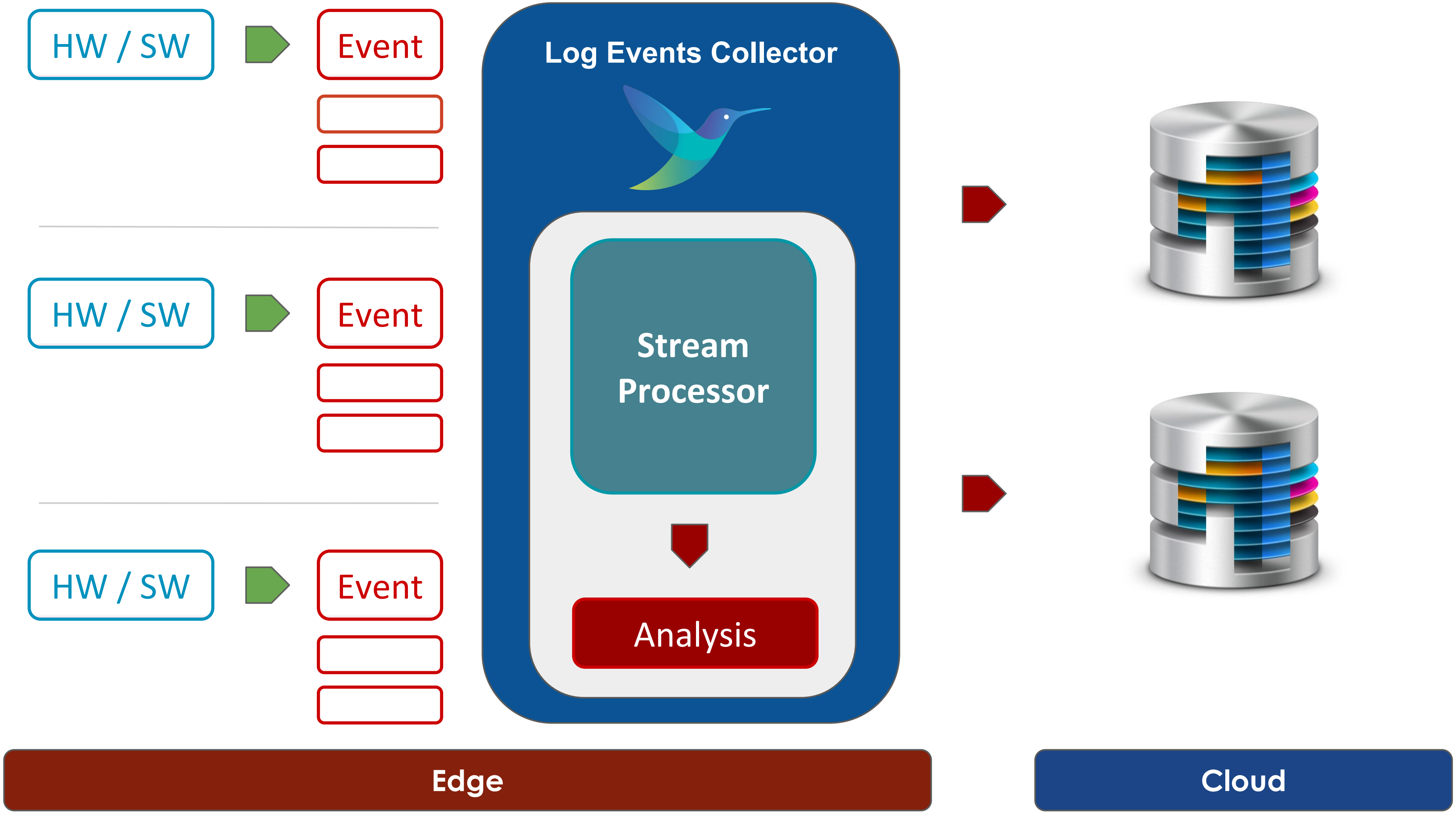

HW / SW
Event
HW / SW
Event
HW / SW
Event
Log Events Collector
(aggregator)
Stream Processor
Analysis
Edge
Cloud



Logging on Steroids

Stream Processing on the **Edge**


HW / SW
Event
HW / SW
Event
HW / SW
Event
Log Events Collector
(aggregator)
Stream Processor
Analysis
Edge
Cloud

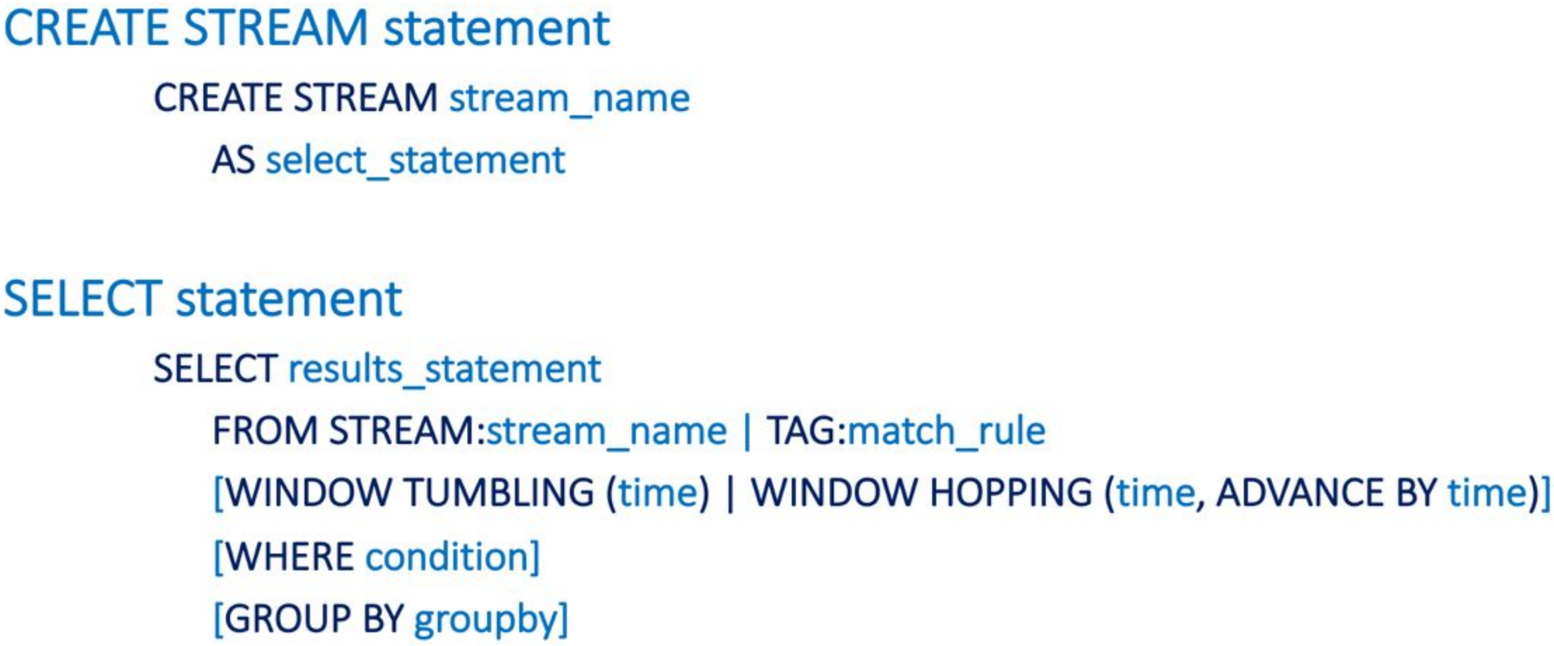

HW / SW
Event
HW / SW
Event
HW / SW
Event
Log Events Collector
Stream Processor
Analysis
Edge
Cloud

Goals and Features

Analysis of streaming data (logs, metrics, etc.) in real time

- Stream Processing features
  - Offloads computations from servers to data collectors
  - Only sends required data to cloud
  - Uses declarative SQL-like language to express the computations
  - Integrated in Fluent Bit core

CREATE STREAM Syntax



CREATE STREAM statement

```
CREATE STREAM stream_name
    AS select_statement
```

SELECT statement

```
SELECT results_statement
    FROM STREAM:stream_name | TAG:match_rule
    [WINDOW TUMBLING (time) | WINDOW HOPPING (time, ADVANCE BY time)]
    [WHERE condition]
    [GROUP BY groupby]
```

Stream Processor Functions

- Aggregation Functions:

  AVG(key), **COUNT**(key), **COUNT**(*), **MIN**(key), **MAX**(key), **SUM**(key)

- Time Functions:

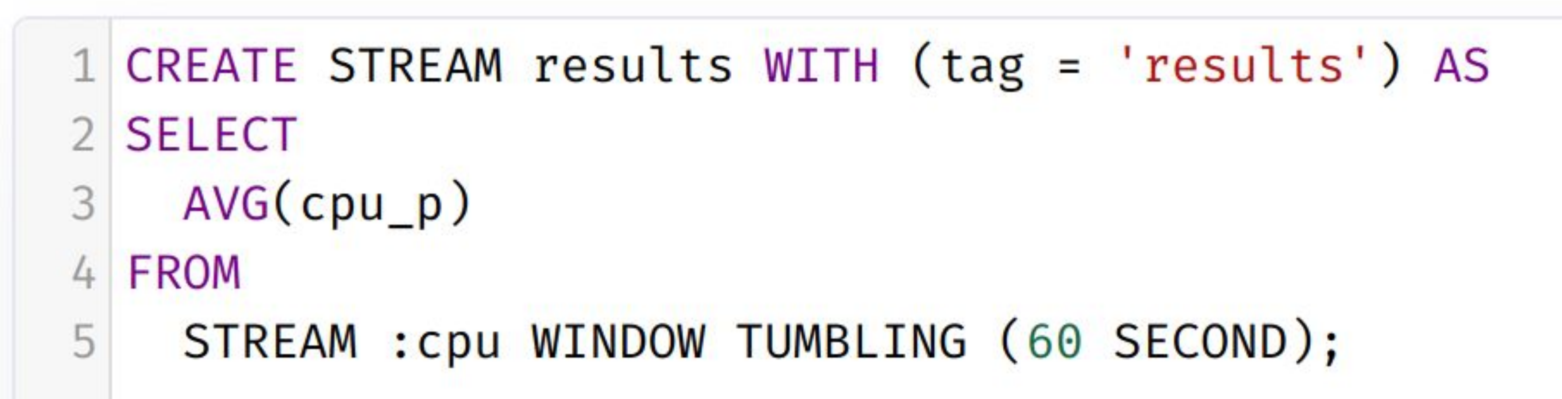
  NOW(), **UNIX_TIMESTAMP**()

- Timeseries Function:

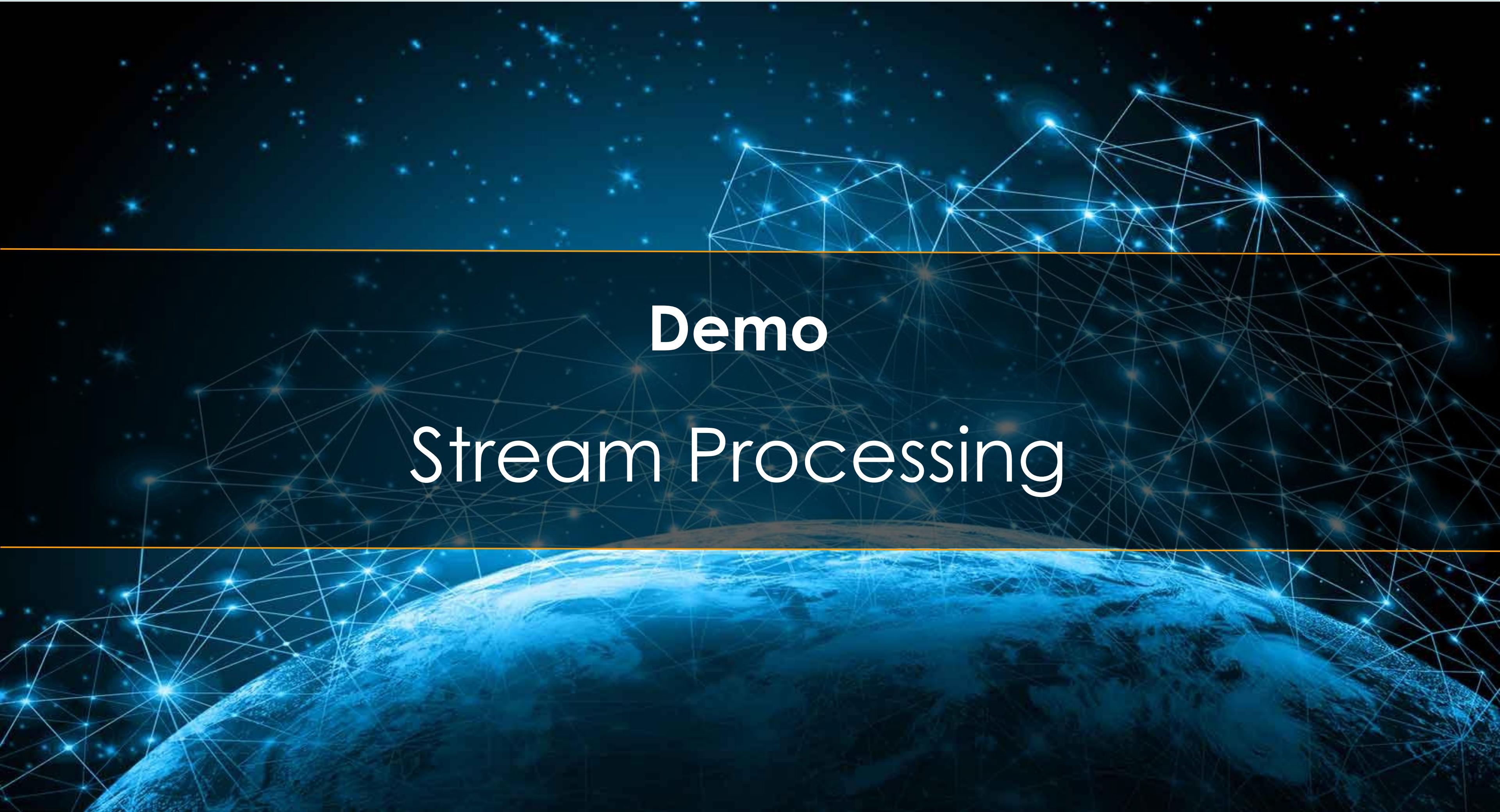
  TIMESERIES_FORECAST(key1, key2, value)
  TIMESERIES_FORECAST_R(key1, key2, value, max)

Fluent Bit Stream Processing syntax support subkeys, for instance: key[sub1][sub2]

Example: Stream Creation



```
CREATE STREAM results WITH (tag = 'results') AS
SELECT
  AVG(cpu_p)
FROM
  STREAM :cpu WINDOW TUMBLING (60 SECOND);
```



Demo

Stream Processing

Q & A

fluentbit.io

fluent/**fluent-bit**