
# Old Meets New: Syslog and Kafka

## Aggregated Data Collection with Splunk Connect for Kafka

Mark Bonsack, CISSP | Staff Sales Engineer
Scott Haskell | Principal SE Architect

October 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

# We Will Discuss:

1. Syslog/Splunk Best Practice Review
2. Rudiments of Splunk Connect for Kafka
3. New! Connect for Kafka and Syslog
4. Connect for Kafka Configuration
5. Wrap-up/Resources

# Syslog and Splunk: A Review

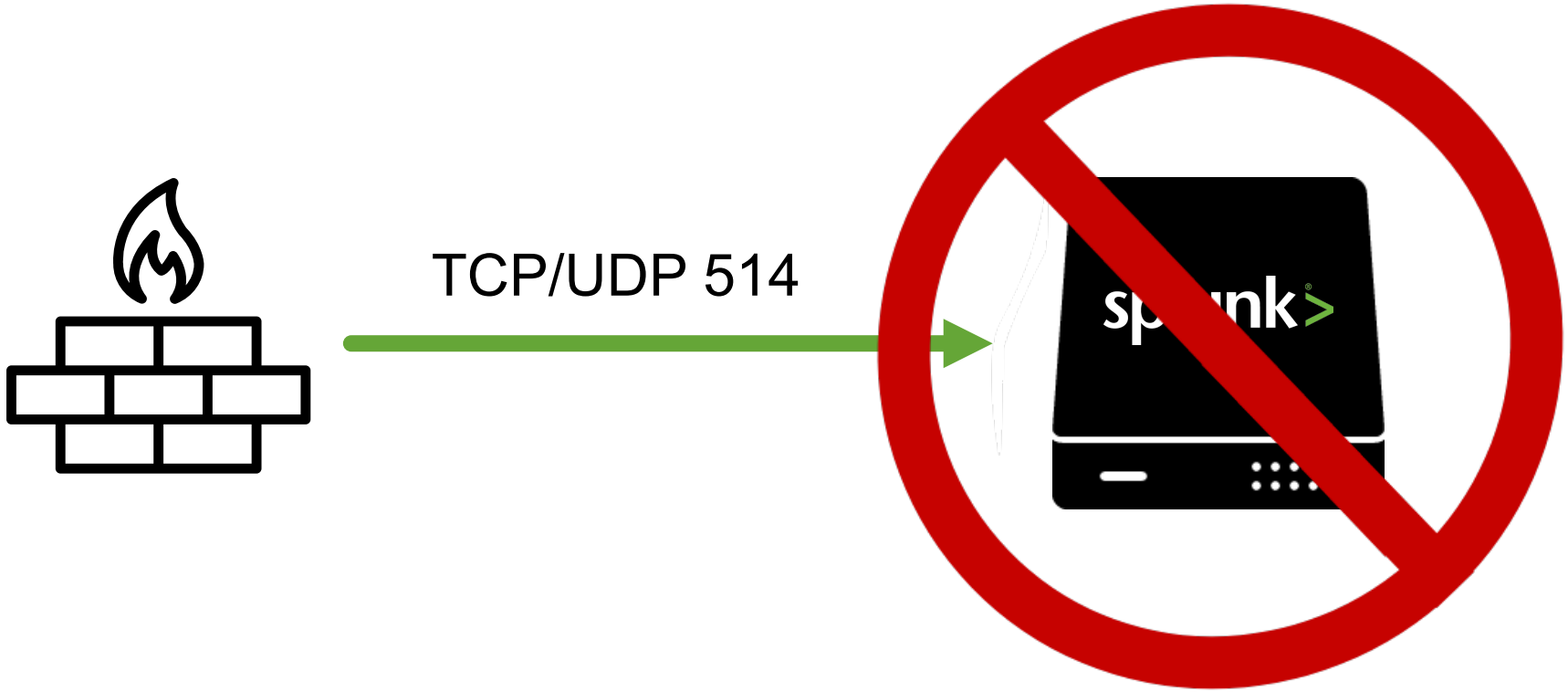
**Syslog lives on!**

splunk> .conf18

# Remember this from last year...

## Do *not* send syslog traffic (on any port) directly to Splunk indexers

(Except in the smallest of installations.  Or other corner cases.  There are *always* corner cases.)



# A Lot has Changed in a Year...

- Message buses have become more prevalent, particularly in Cloud installations
- Kafka has emerged as the de facto “on-prem” message bus
  - cf. Kinesis (AWS), EventHub (Azure), etc.
- There are now plenty of options for ingesting syslog-based data into Splunk
  - With and without message buses
  - With and without a traditional syslog server
- Other sessions will explore these options

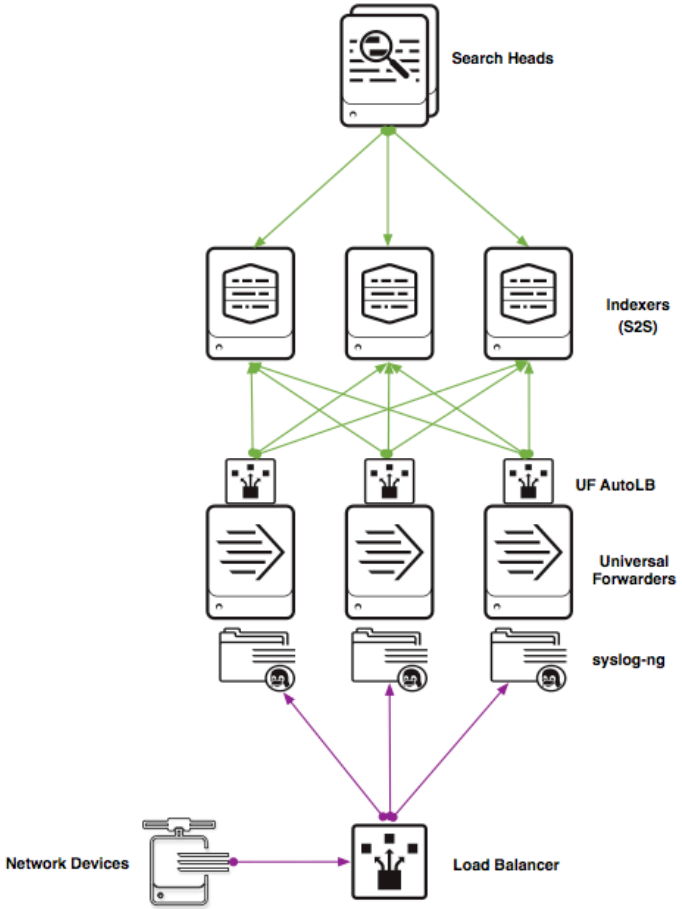
# ...And a Lot has Not

Syslog as popular as ever

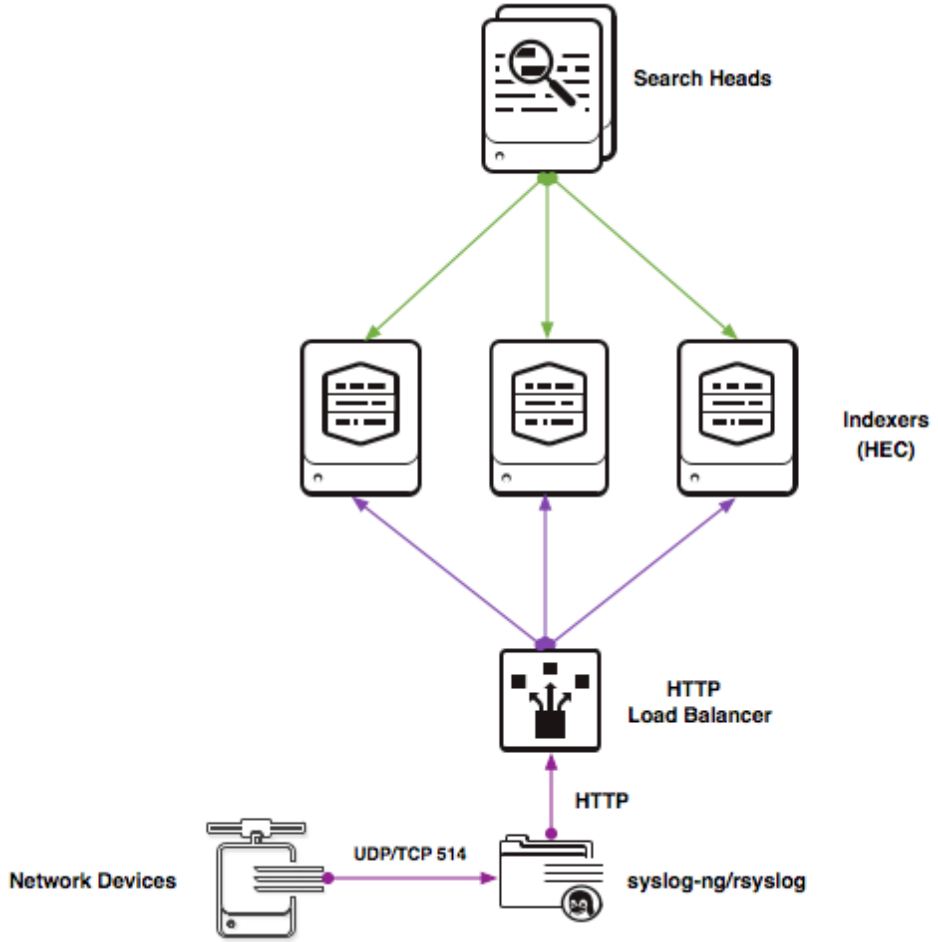
- Efficient and easy-to-implement protocol still popular with device vendors
- Syslog still the most common “sourcetype” ingested by our customers
- Most customers still ingesting syslog incorrectly
- Both syslog server flavors (syslog-ng and rsyslog) have stewardship issues
  - Recent versions may not be officially supported by distro vendors
  - Compiled binaries at times can be difficult to track down (recent links provided in wrap-up)

# syslog/HEC Introduced Last Year...

## UF




## HEC



# ...With Some Caveats

Along with some very successful deployments

- For highest scale, messages sent to HEC should be batched
- For this reason, a script (`omsplunkhec.py`) was developed to enable batching events from syslog servers to the HEC /raw endpoint
- Script only supports /raw endpoint in HEC
  - Cannot automatically create indexed fields
  - Originating host and other critical fields must be parsed out of the general payload
- Script cannot persist data to disk in case of infrastructure loss
- Though simple (less than one page of NCSS), and in production at many customer installations, it is not officially supported by Splunk
- Even so, still a viable architecture and remains a recommended option

# Scalable Data Collection from Kafka Introduced

## Introducing Splunk Connect for Kafka!

- Replaces a number of early TAs
- Significantly higher scale
- Sources (devices/syslog servers) and sink (Splunk) can be widely distributed
- Allows use of *both* /raw and /event HEC endpoints
  - And with some tricks, both at the same time
- Allows the option of omitting the syslog server (using syslog source connector)
  - More later on this...
- Supported by Splunk!

# Are syslog servers still necessary with Kafka?

Syslog protocol is (still) very rich

- Traditional servers decode traffic, at scale, using a very well-defined protocol
  - One of the earliest protocols developed in the heyday of BSD Unix
  - RFC 3164 (BSD) – latest revision documents behavior of earlier syslog implementations
  - RFC 5424 (IETF) – significant revision written in 2009; supersedes 3164
  - Many devices still (and especially by default) emit only BSD-style (3164) syslog
- Many customers have syslog servers in production today, and would like to continue to use them with Splunk.
- Syslog servers are very well documented; even the “NG” versions are now pushing 20 years old
- Stewardship still an issue

# Or Not?

Direct syslog ingest via kafka-connect-syslog or generic UDP/TCP Kafka connectors

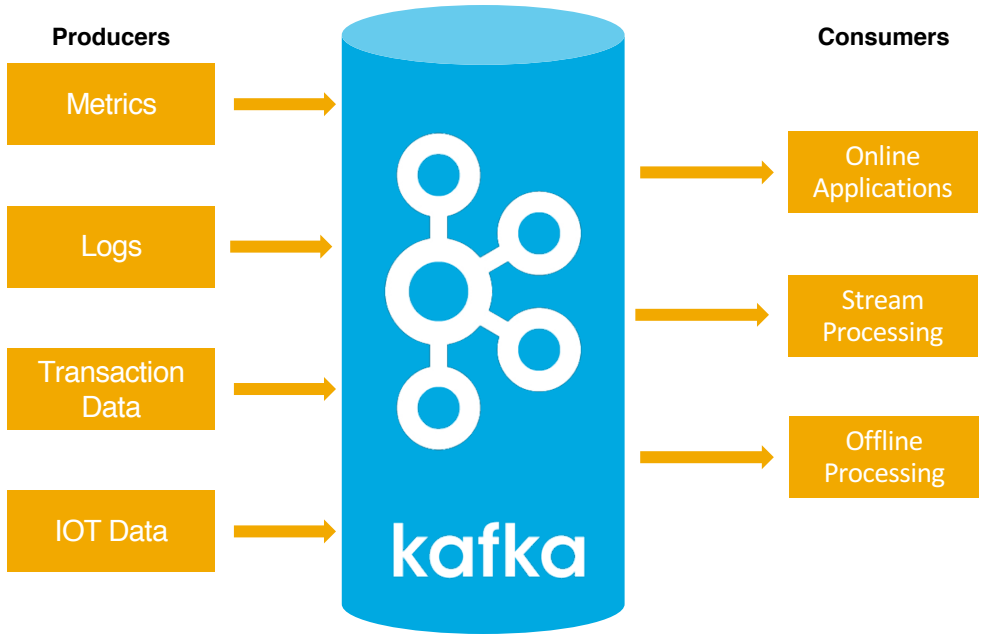
- The new “hotness” is stream processing in Kafka/Kinesis/etc.
  - Significant work needed to replicate true syslog protocol decode
  - Significant props/transforms work may be necessary
- Omitting syslog servers can simplify architecture
- Can be adequate if separate ports can be used for each sourcetype
  - Very difficult to do in practice
  - See session (SEC1905, McKesson) for an example of this approach

# Rudiments of Splunk Connect for Kafka

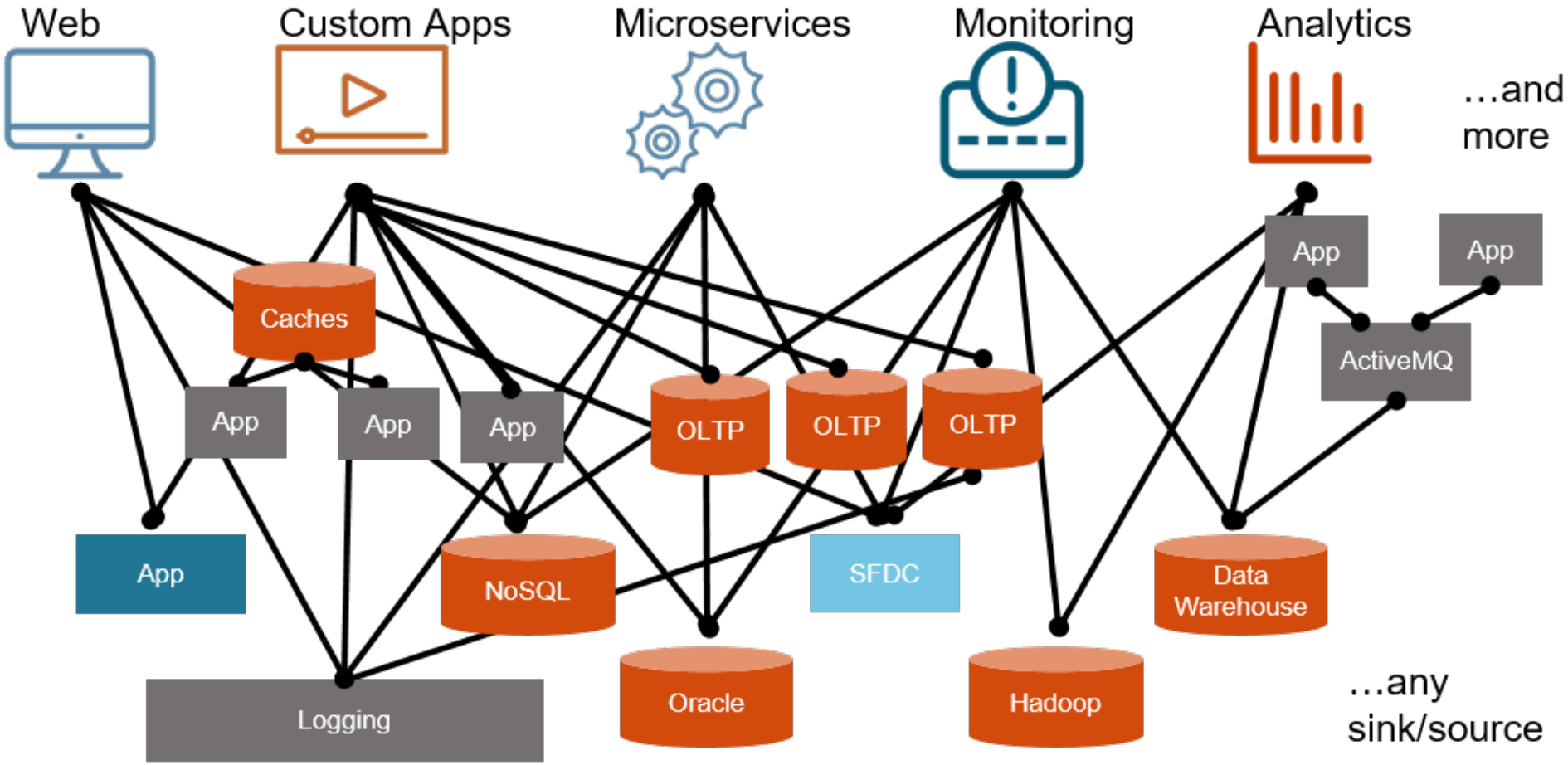
## Scalable Splunk Data Collection from Kafka

splunk> .conf18

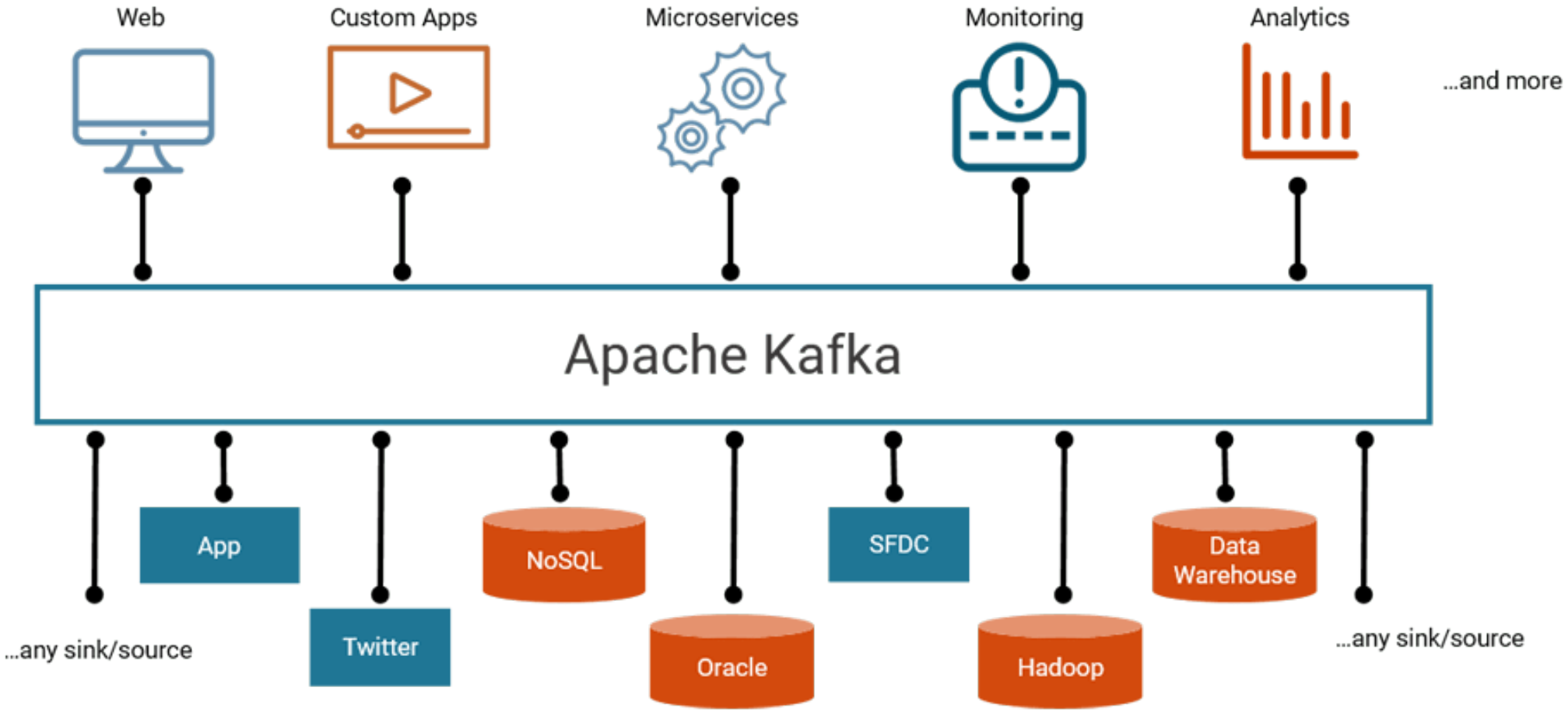
# What Is Kafka?




- Distributed
- Fault Tolerant
- Persistent
- Scalable
- Low Latency
- Secure
- Streaming
- Common API's
- It's a log!!!




Source: Confluent.io



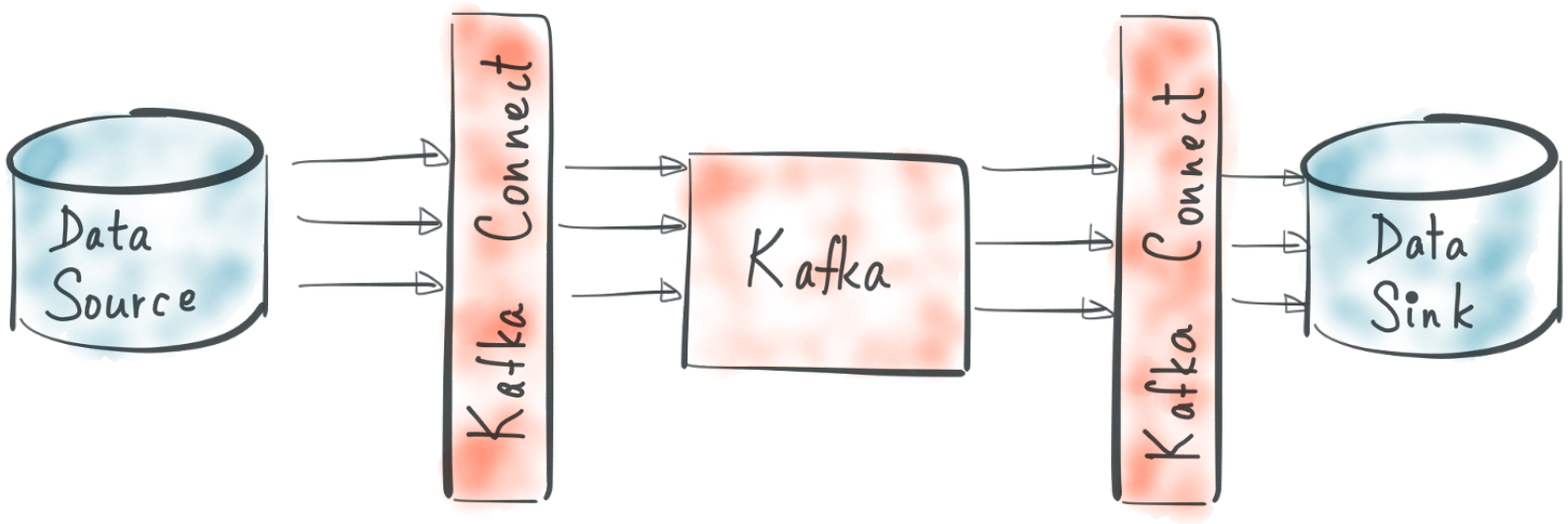

Source: Confluent.io

# Splunk Add-On for Kafka

**https://www.splunk.com/blog/2016/10/31/splunking-kafka-at-scale.html**

- Frequency based polling (modular input)
- Didn’t scale (simple consumer)
- Painful configuration
- Lacking fault tolerance
- Pre-built panels to operationally monitor Kafka (JMX)

# Kafka Connect



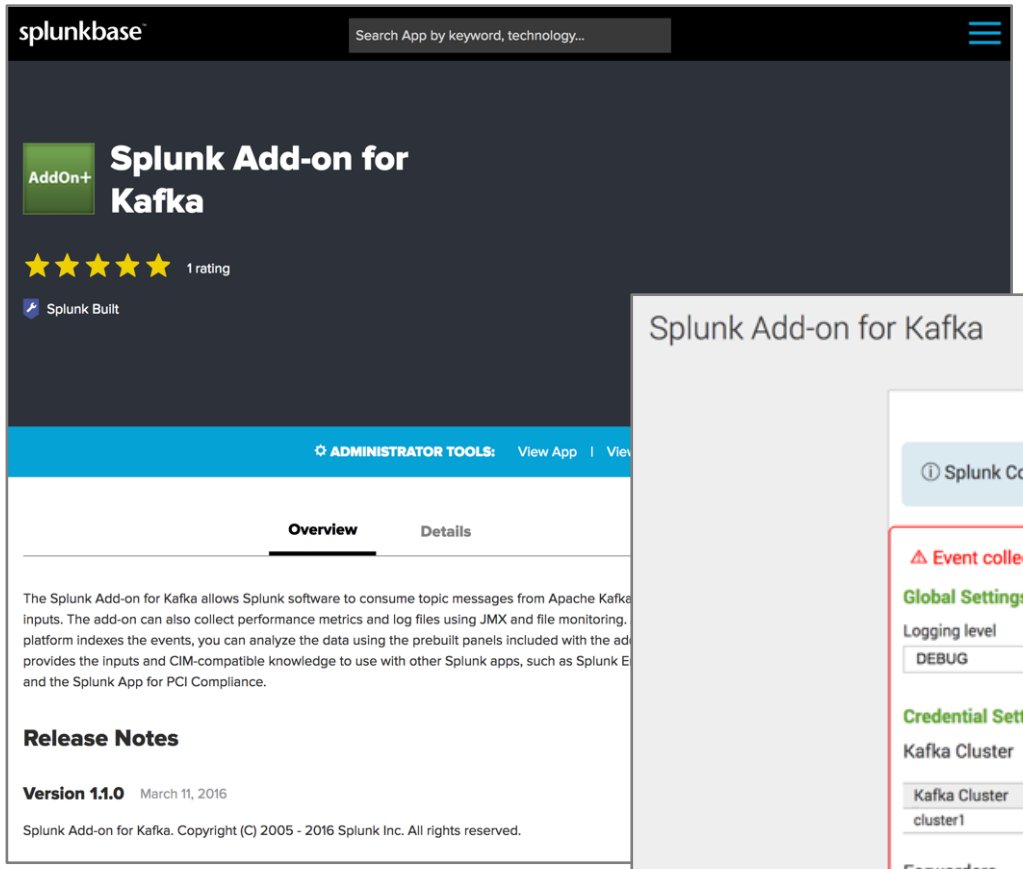

- Scalable & Reliable Data Streaming
- Tightly Coupled API
- Flexible Deployment
- Source & Sink Connectors
- Streaming & Batch
- Connectors, Workers, Tasks
- Management REST API

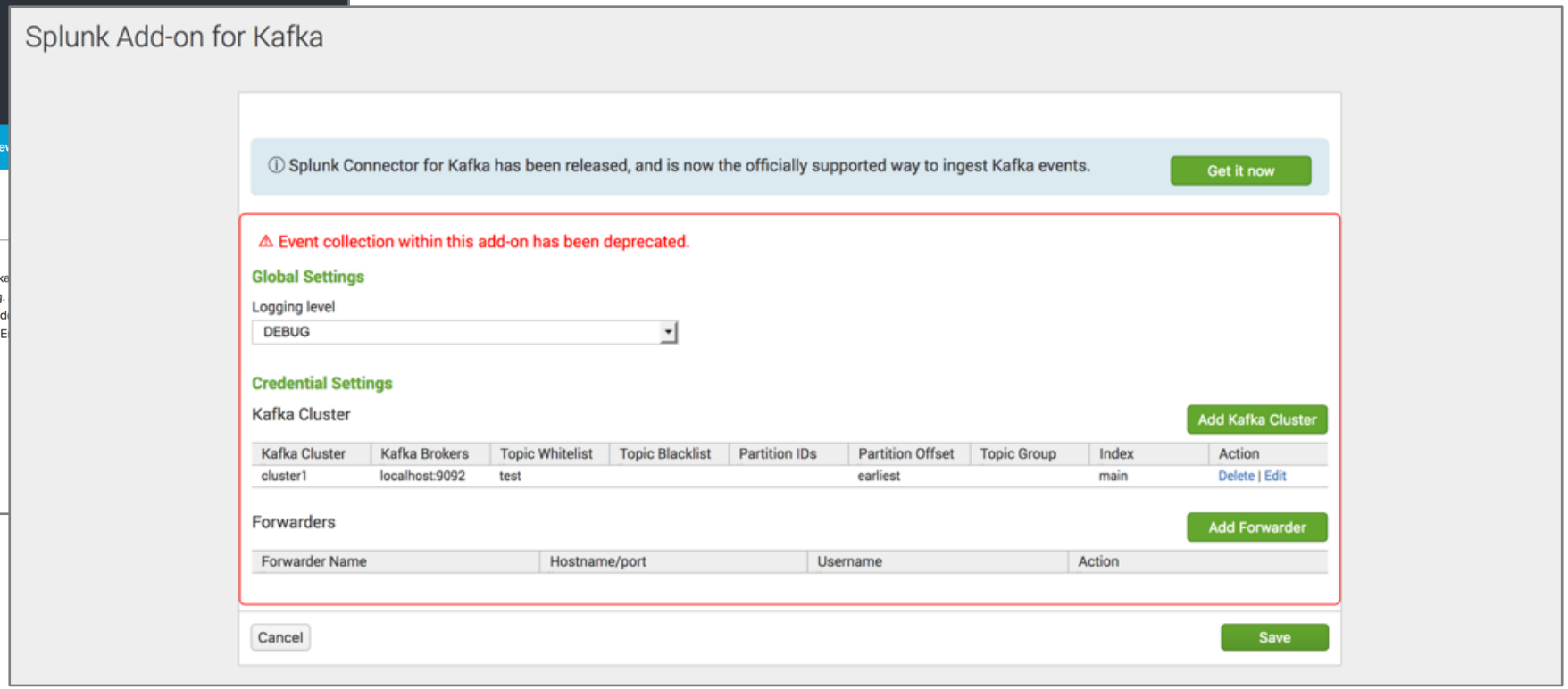
Source: Confluent.io

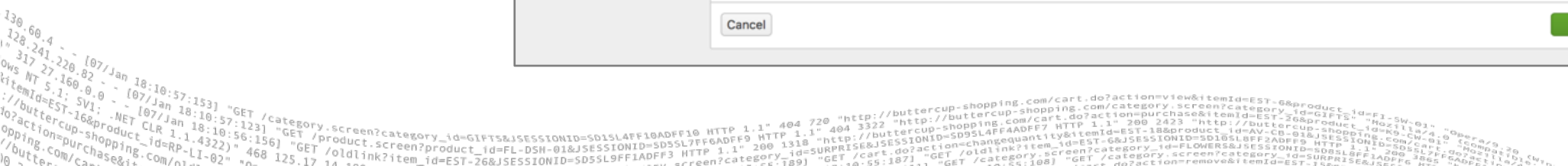
# Splunk Connect for Kafka

- Kafka Connect Sink Connector (distributed, fault tolerant, scalable)
- Java HEC client
- Load balancing – internal round robin or hardware/software LB
- Per topic/event routing
- Metrics Store – collectd ➡ /services/collector/raw [collectd_http]
- Indexer acknowledgement
- Splunk supported

# Legacy Splunk Add-on for Kafka



splunk> .conf18

# New! Splunk Connect for Kafka and Syslog

**Scalable, Performant, and Supported!**

splunk> .conf18

# What Drives the Need?

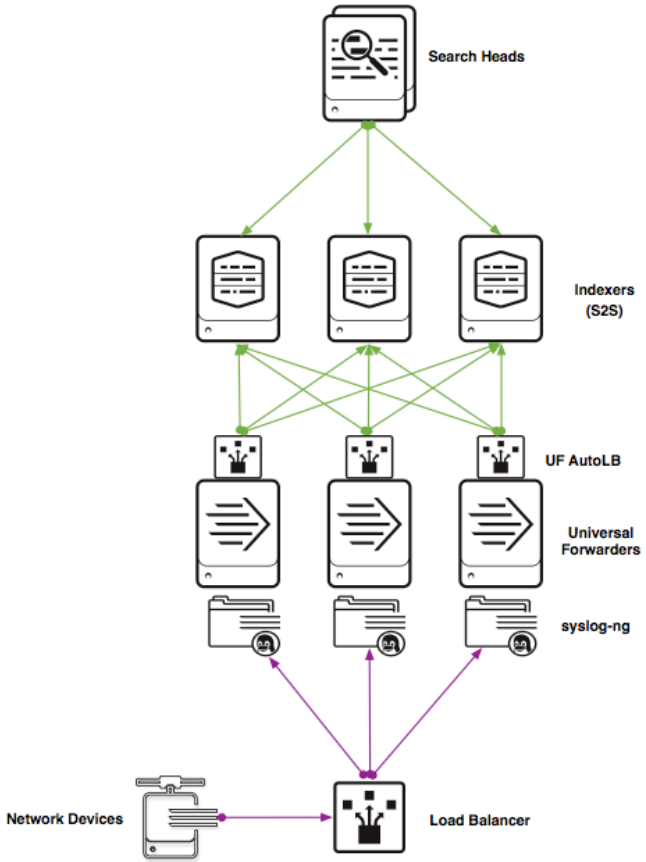
**This is where the subtitle goes**

- Scale requirements stretch UF limits
- Distributed Data Collection/Resiliency
- Improved HEC Compatibility
- Supportability
- Alignment with evolving customer needs

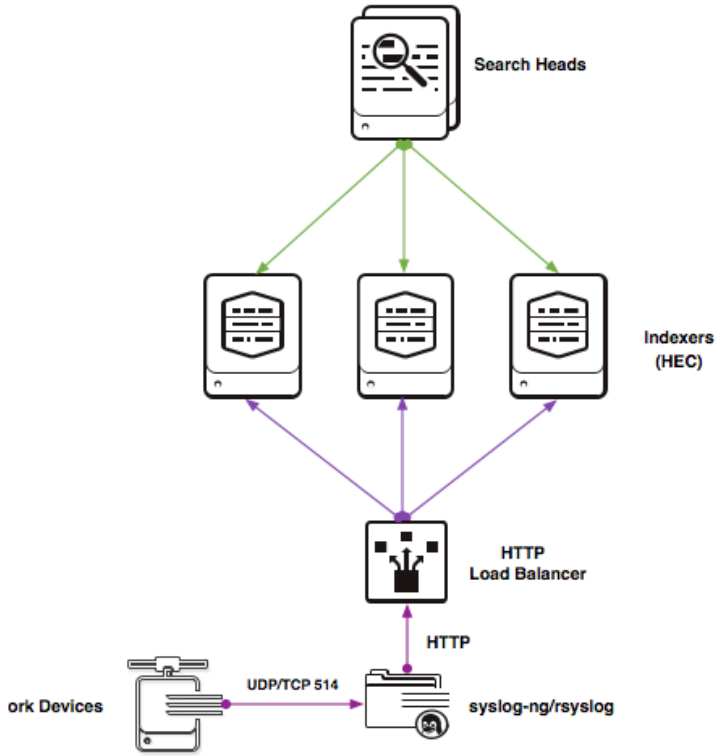
# New for 2018: Syslog with Splunk Connect for Kafka!

## An Extension of the HEC/Syslog Architecture

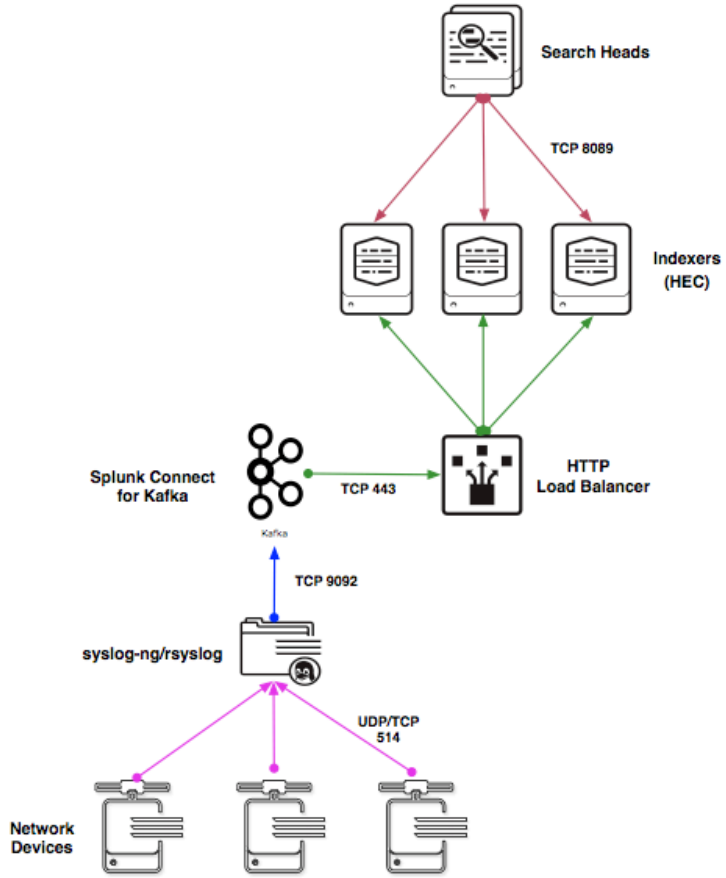
### UF




### HEC




### New! Kafka/HEC

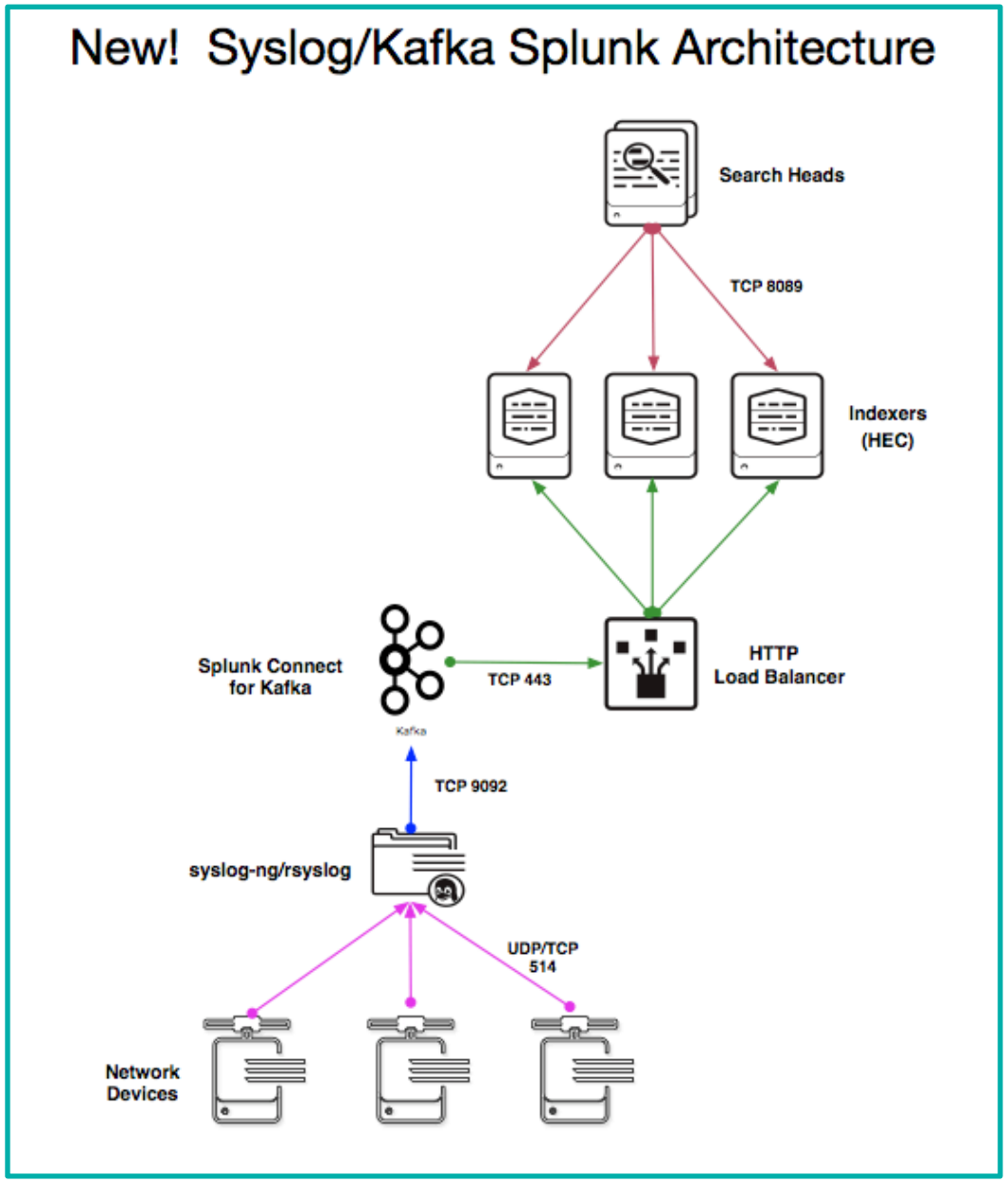



splunk> .conf18

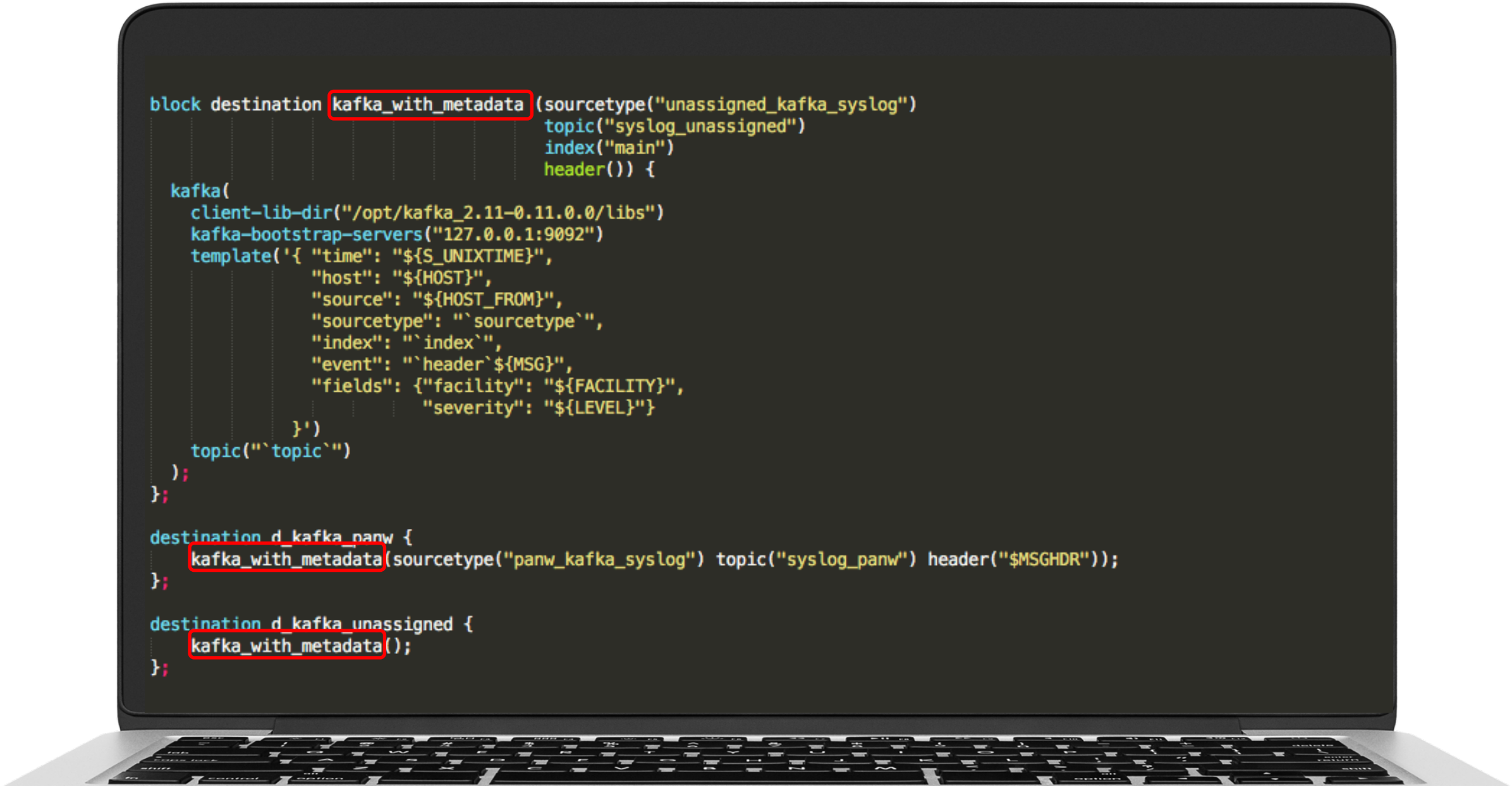
# Syslog/Kafka Architecture

The evolution of syslog/HEC

- Takes full advantage of Kafka
- Allows use of /event, /raw, and even a “hybrid” mode
- Full persistence to disk and distributed data availability
- Very simple configuration changes from original HEC syslog-ng/rsyslog configurations from last year
- Can be used without a syslog server in certain cases

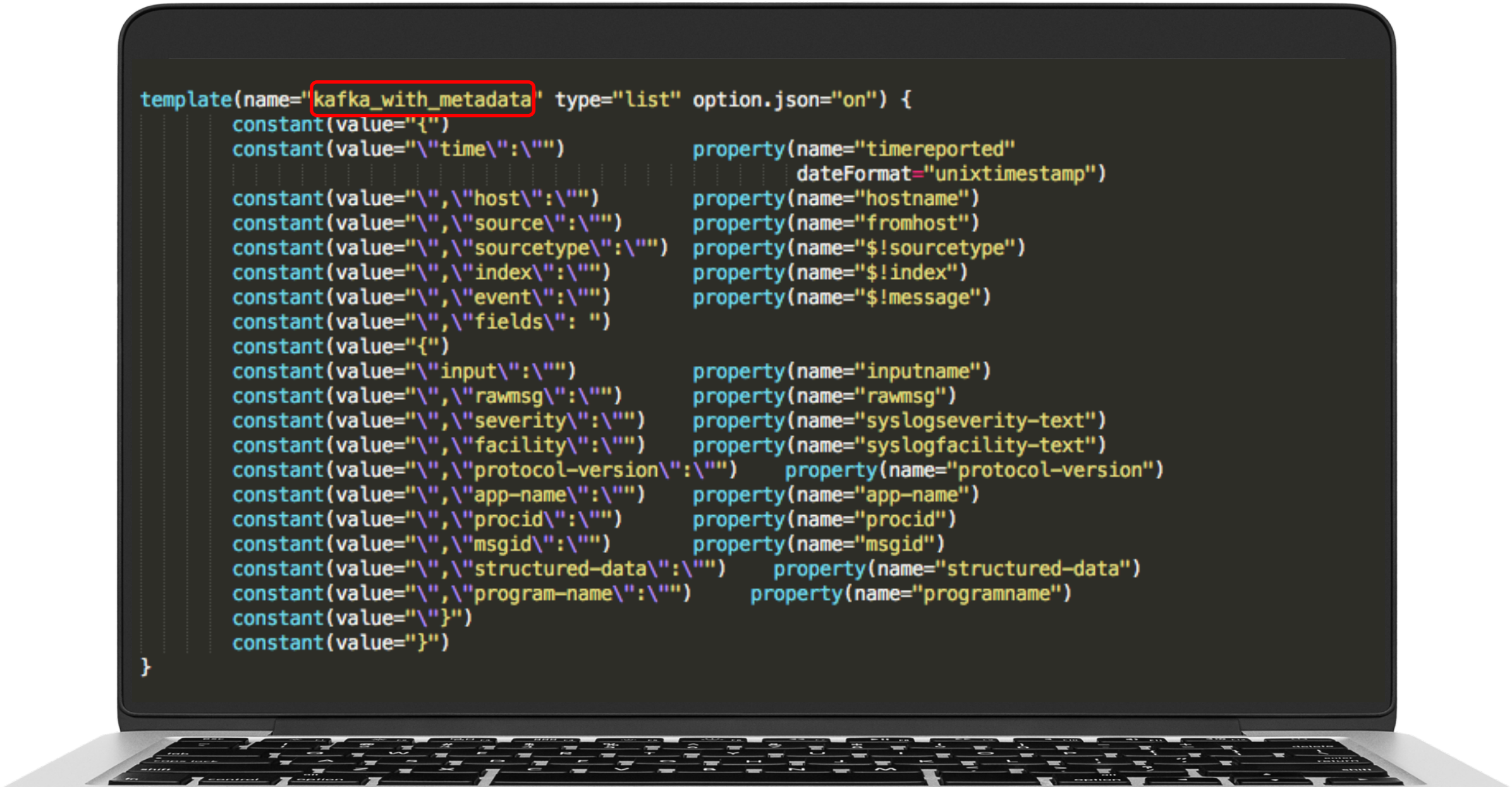


# Syslog-ng Destination Configuration for Kafka



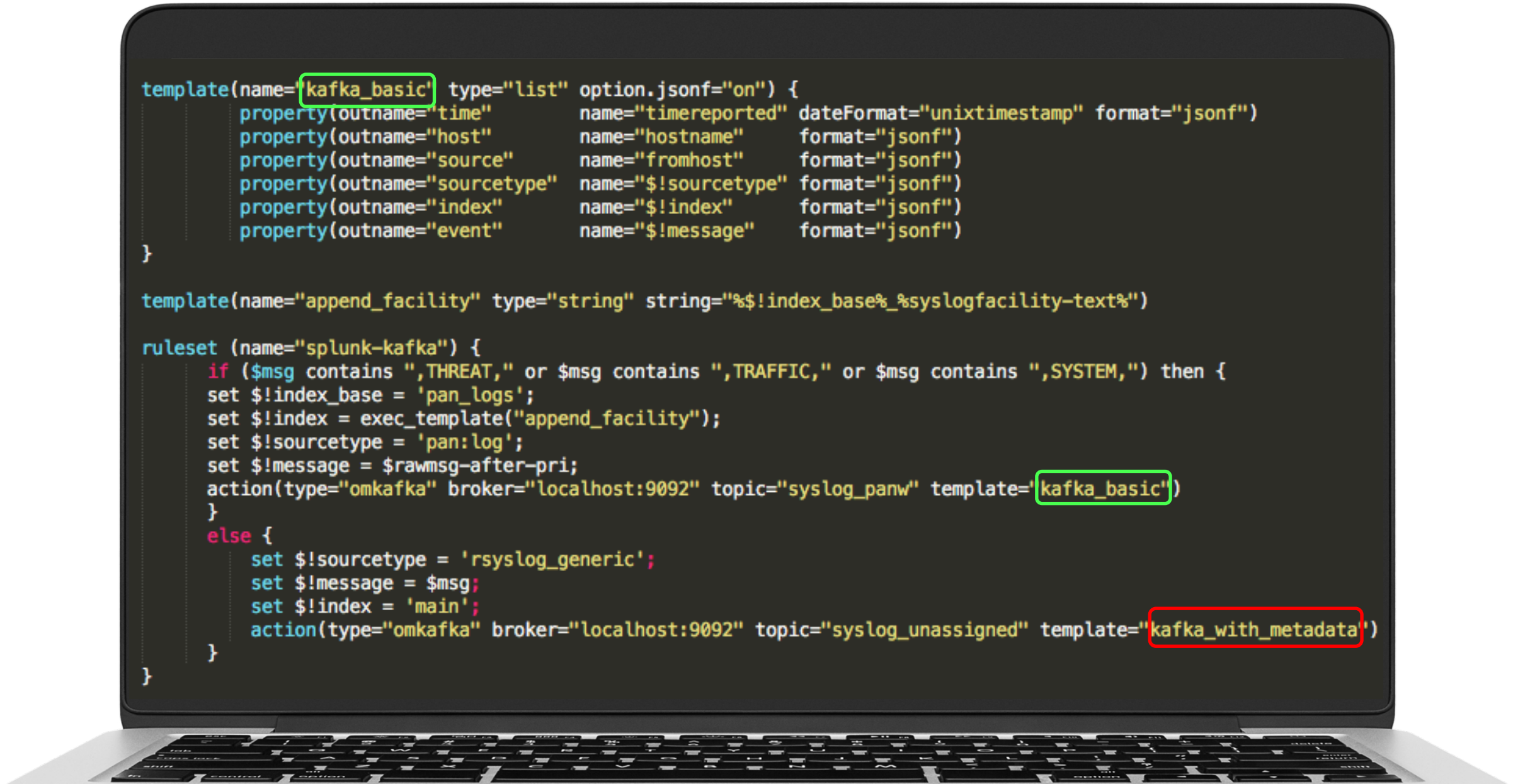
```
block destination kafka_with_metadata (sourcetype("unassigned_kafka_syslog")
                                       topic("syslog_unassigned")
                                       index("main")
                                       header()) {
  kafka(
    client-lib-dir("/opt/kafka_2.11-0.11.0.0/libs")
    kafka-bootstrap-servers("127.0.0.1:9092")
    template('{ "time": "${S_UNIXTIME}",
                "host": "${HOST}",
                "source": "${HOST_FROM}",
                "sourcetype": "`sourcetype`",
                "index": "`index`",
                "event": "`header`${MSG}",
                "fields": {"facility": "${FACILITY}",
                           "severity": "${LEVEL}"}
              }')
    topic("`topic`")
  );
};

destination d_kafka_panw {
    kafka_with_metadata(sourcetype("panw_kafka_syslog") topic("syslog_panw") header("$MSGHDR"));
};

destination d_kafka_unassigned {
    kafka_with_metadata();
};
```

# rsyslog Template Configuration for Kafka



```
template(name="kafka_with_metadata" type="list" option.json="on") {
        constant(value="{")
        constant(value="\"time\":\"")           property(name="timereported"
                                                         dateFormat="unixtimestamp")
        constant(value="\",\"host\":\"")        property(name="hostname")
        constant(value="\",\"source\":\"")      property(name="fromhost")
        constant(value="\",\"sourcetype\":\"")  property(name="$!sourcetype")
        constant(value="\",\"index\":\"")       property(name="$!index")
        constant(value="\",\"event\":\"")       property(name="$!message")
        constant(value="\",\"fields\": ")
        constant(value="{")
        constant(value="\"input\":\"")          property(name="inputname")
        constant(value="\",\"rawmsg\":\"")      property(name="rawmsg")
        constant(value="\",\"severity\":\"")    property(name="syslogseverity-text")
        constant(value="\",\"facility\":\"")    property(name="syslogfacility-text")
        constant(value="\",\"protocol-version\":\"")    property(name="protocol-version")
        constant(value="\",\"app-name\":\"")    property(name="app-name")
        constant(value="\",\"procid\":\"")      property(name="procid")
        constant(value="\",\"msgid\":\"")       property(name="msgid")
        constant(value="\",\"structured-data\":\"")    property(name="structured-data")
        constant(value="\",\"program-name\":\"")     property(name="programname")
        constant(value="\"}")
        constant(value="}")
}
```

# rsyslog Ruleset Configuration for Kafka



```
template(name="kafka_basic" type="list" option.jsonf="on") {
          property(outname="time"        name="timereported" dateFormat="unixtimestamp" format="jsonf")
          property(outname="host"        name="hostname"     format="jsonf")
          property(outname="source"      name="fromhost"     format="jsonf")
          property(outname="sourcetype"  name="$!sourcetype" format="jsonf")
          property(outname="index"       name="$!index"      format="jsonf")
          property(outname="event"       name="$!message"    format="jsonf")
}

template(name="append_facility" type="string" string="%$!index_base%_%syslogfacility-text%")

ruleset (name="splunk-kafka") {
      if ($msg contains ",THREAT," or $msg contains ",TRAFFIC," or $msg contains ",SYSTEM,") then {
      set $!index_base = 'pan_logs';
      set $!index = exec_template("append_facility");
      set $!sourcetype = 'pan:log';
      set $!message = $rawmsg-after-pri;
      action(type="omkafka" broker="localhost:9092" topic="syslog_panw" template="kafka_basic")
      }
      else {
          set $!sourcetype = 'rsyslog_generic';
          set $!message = $msg;
          set $!index = 'main';
          action(type="omkafka" broker="localhost:9092" topic="syslog_unassigned" template="kafka_with_metadata")
      }
}
```

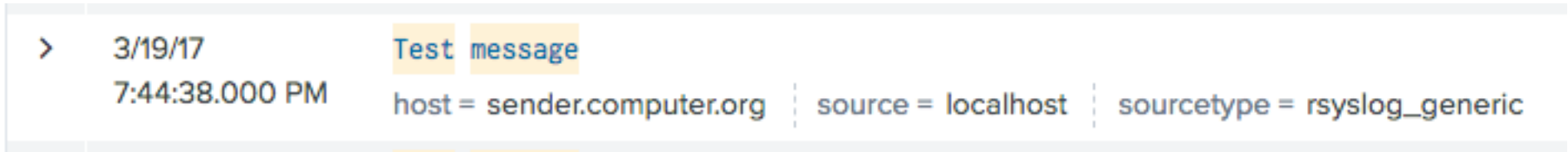
# What does all this look like in Splunk?

Using the prior configuration example

- A raw RFC 5424-compliant message:

```
<165>1 2017-03-19T23:44:38+00:00 sender.computer.org evententry - ID47 [example
iut="3" eventSource="Application" eventID="1011"] Test message
```

- Looks like this using the `splunk-kafka` rsyslog ruleset (/event HEC endpoint):




- This is great, but where's the rest of my stuff?

# What does all this look like in Splunk?

## The beauty of indexed fields in HEC!

| i | Time | Event |
|---|---|---|
| ⌄ | 3/19/17 7:44:38.000 PM | Test message |

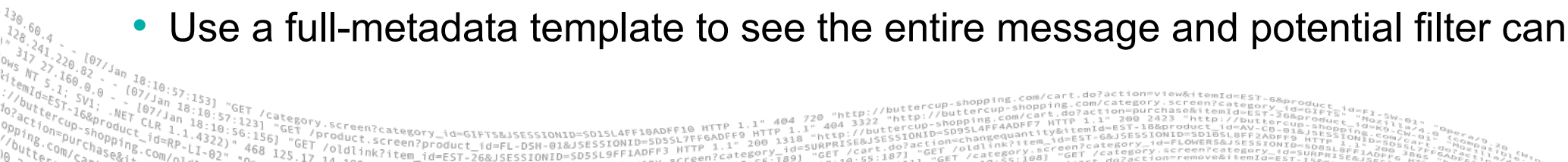
Event Actions ▾

| Type | Field | Value | Actions |
|---|---|---|---|
| Selected | host ▾ | sender.computer.org | ⌄ |
| | source ▾ | localhost | ⌄ |
| | sourcetype ▾ | rsyslog_generic | ⌄ |
| Event | app-name ▾ | evententry | ⌄ |
| | facility ▾ | local4 | ⌄ |
| | input ▾ | imudp | ⌄ |
| | msgid ▾ | ID47 | ⌄ |
| | procid ▾ | - | ⌄ |
| | program-name ▾ | evententry | ⌄ |
| | protocol-version ▾ | 1 | ⌄ |
| | rawmsg ▾ | <165>1 2017-03-19T23:44:38+00:00 sender.computer.org evententry - ID47 [example iut="3" eventSource="Application" eventID="1011"] Test message | ⌄ |
| | severity ▾ | notice | ⌄ |
| | structured-data ▾ | [example iut="3" eventSource="Application" eventID="1011"] | ⌄ |
| Time | _time ▾ | 2017-03-19T19:44:38.000-04:00 | |

# Tips for Managing Config Files

The ultimate goal is automation

- Use a good text editor! (e.g. Sublime with bash syntax)
  - Use good formatting technique; align/separate JSON grammar for readability
  - Color highlighting will make spotting formatting errors much easier!
- Build generic templates (i.e. functions) that can be called by the rulesets (rsyslog) or destinations (syslog-ng) specific to each technology.
  - This will significantly aid automation
- Change as little as possible in the syslog config files when adding a new data source.
  - The actual regex/property filter and the sourcetype/index should be all that needs to be changed for most installations.
- Use a “catch-all” filter to find rogue data sources.
  - Use a full-metadata template to see the entire message and potential filter candidates.

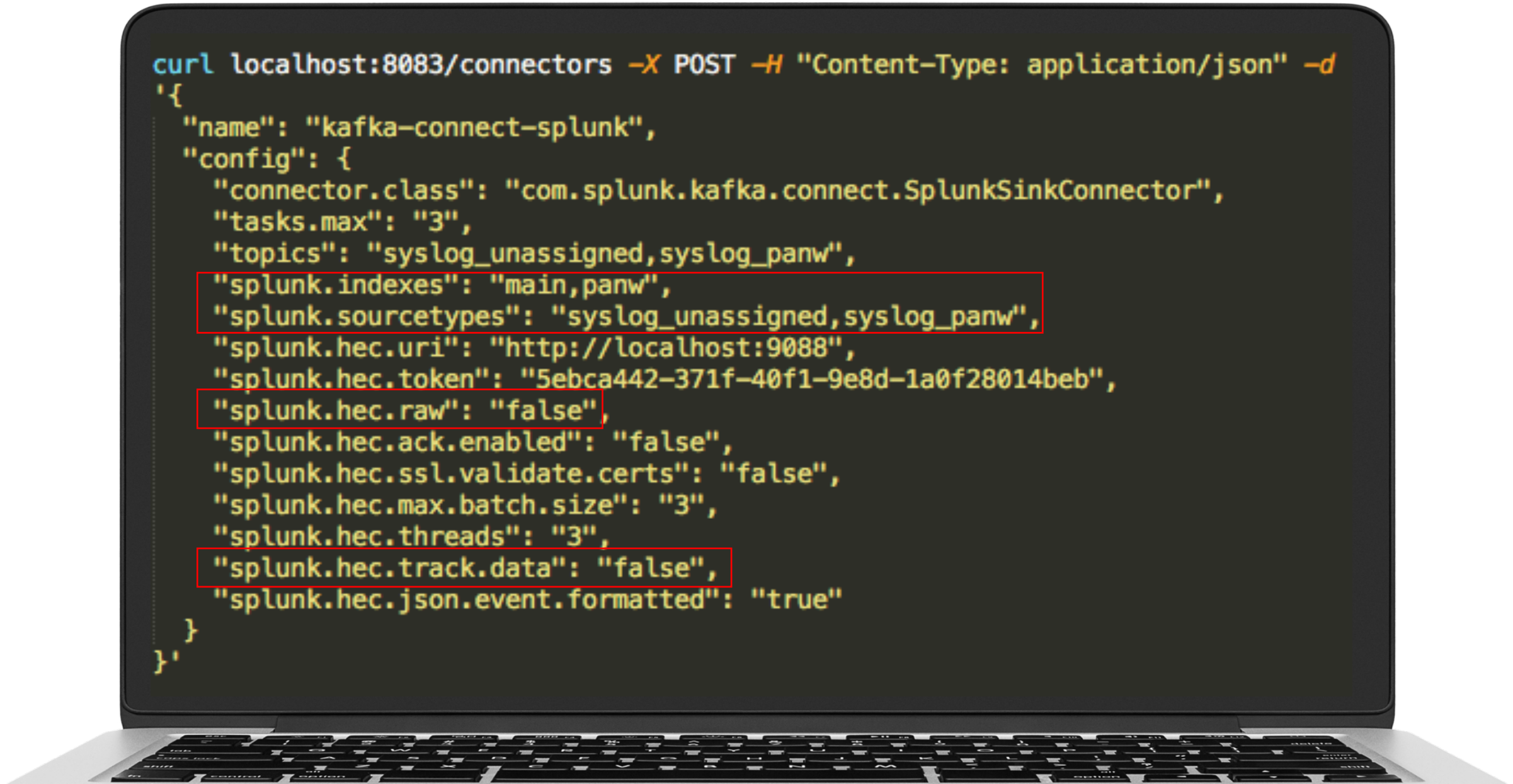


# Splunk Connect for Kafka Configuration

## With a look at Scale

splunk> .conf18

# Splunk Connect for Kafka Configuration



# A Look at Scale

- Syslog-ng:  Up to 78K EPS to Kafka
  - https://www.syslog-ng.com/community/b/blog/posts/testing-the-performance-of-log-streaming-to-kafka-with-syslog-ng
  - Key factors for syslog-ng scale to Kafka:
    - Kafka receipt acknowledgement (destroys scale, as expected)
    - Template complexity (use basic templates when syslog metadata from vendor is sparse)
- Rsyslog:
  - TrueCar Tests: https://www.drivenbycode.com/how-truecar-uses-kafka-for-high-volume-logging-part-2/
  - 1 VM rsyslog; 5 VM kafka cluster; 5 partitions; **100,000 EPS**
  - Librdkafka tests: https://github.com/edenhill/librdkafka/blob/master/INTRODUCTION.md
  - 2 brokers, 2 partitions, required.acks=2, 100 byte messages: **850,000 EPS**, **85 MB/s**
  - Several “nerd knobs” can be configured for queue sizes, buffering to disk, etc.
- Splunk Connect for Kafka
  - Single Connect Worker (16GB JVM, 16CPU) = 45.3 MB/s
  - 600 datagens on 12 servers, 20 Server Kafka Cluster, 20 Connect Workers, 600 tasks, 50 indexers
    - **1,350,000 EPS; 730 MB/s**

splunk> .conf18

# Wrap-up

## Additional Resources

splunk> .conf18

# Key Takeaways

**This is where the subtitle goes**

1. Use a syslog server! Do not send syslog traffic directly to Splunk.
2. Kafka provides data persistence, resiliency, and scale
3. Splunk Connect for Kafka provides a *supported* extension to the syslog/HEC architecture introduced last year
4. There are many helpful resources, both Splunk and open source

# Install/Prepare syslog servers for Kafka

Both require additional steps for Kafka integration

## Syslog-ng

- https://www.syslog-ng.com/products/open-source-log-management/3rd-party-binaries.aspx (All distros)
- Prepare Java (needed for Kafka destination support)
  - https://www.syslog-ng.com/technical-documents/doc/syslog-ng-open-source-edition/3.16/administration-guide/35#TOPIC-956517
  - https://www.syslog-ng.com/community/b/blog/posts/troubleshooting-java-support-syslog-ng
- There is a librdkafka implementation (C/C++ instead of Java)
  - Again, theoretically faster
  - Suffers from lack of formatting flexibility due to mod-python (syslog-ng-python) requirement
  - For more info: https://syslogng-kafka.readthedocs.io/en/latest/index.html

splunk> .conf18

# Install/Prepare syslog servers for Kafka

Both require additional steps for Kafka integration

## Rsyslog (CentOS/RHEL)

- https://www.rsyslog.com/ubuntu-repository (Ubuntu)
- https://www.rsyslog.com/rhelcentos-rpms (CentOS/RHEL)
- Don’t forget to also install rsyslog-kafka for either one
  - Required module for sending data to Kafka
  - Insert the the directive `$ModLoad omkafka` near top of rsyslog.conf file
  - Uses librdkafka; again could scale higher than syslog-ng Java implementation
    - https://github.com/edenhill/librdkafka/blob/master/INTRODUCTION.md

# Helpful Additional Resources

- Foundational syslog/HEC integration:
  - https://www.splunk.com/blog/2017/03/30/syslog-ng-and-hec-scalable-aggregated-data-collection-in-splunk.html (foundational blog introducing syslog/HEC)
- Additional Resources
  - https://www.splunk.com/blog/2016/05/05/high-performance-syslogging-for-splunk-using-syslog-ng-part-2.html (good overview of syslog-ng server configuration and optimization)
  - https://www.syslog-ng.com/technical-documents/doc/syslog-ng-open-source-edition/3.16/administration-guide (syslog-ng documentation)
  - http://www.rsyslog.com/rsyslog-configuration-builder/ (rsyslog configuration tool (beta))
  - http://www.rsyslog.com/doc/v8-stable/ (rsyslog documentation)

# Additional relevant sessions this week

**Don’t miss these!**

- SEC1905 - 159 Security Use Cases in Record Time With Splunk and Kafka
  - Wednesday, Oct 03, 3:15 p.m. - 4:00 p.m.
- FN1211 - Don't Miss the Bus — Splunking Kafka at Scale
  - Wednesday, Oct 03, 4:30 p.m. - 5:15 p.m.
- FN1616 - The Critical Tricks to Getting Syslog Into Splunk the Right Way
  - Thursday, Oct 04, 11:00 a.m. - 11:45 a.m.

# Q&A

**Mark Bonsack | Staff Sales Engineer**

**Scott Haskell | Principal Sales Engineer**

splunk> .conf18

# Thank You

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