splunk> .conf2017

Atlassian's Journey Into Splunk

The Building Of Our Logging Pipeline On AWS

Tim Clancy | Engineering Manager, Observability
James Mackie | Infrastructure Engineer, Observability

September 2017 | Washington, DC

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.

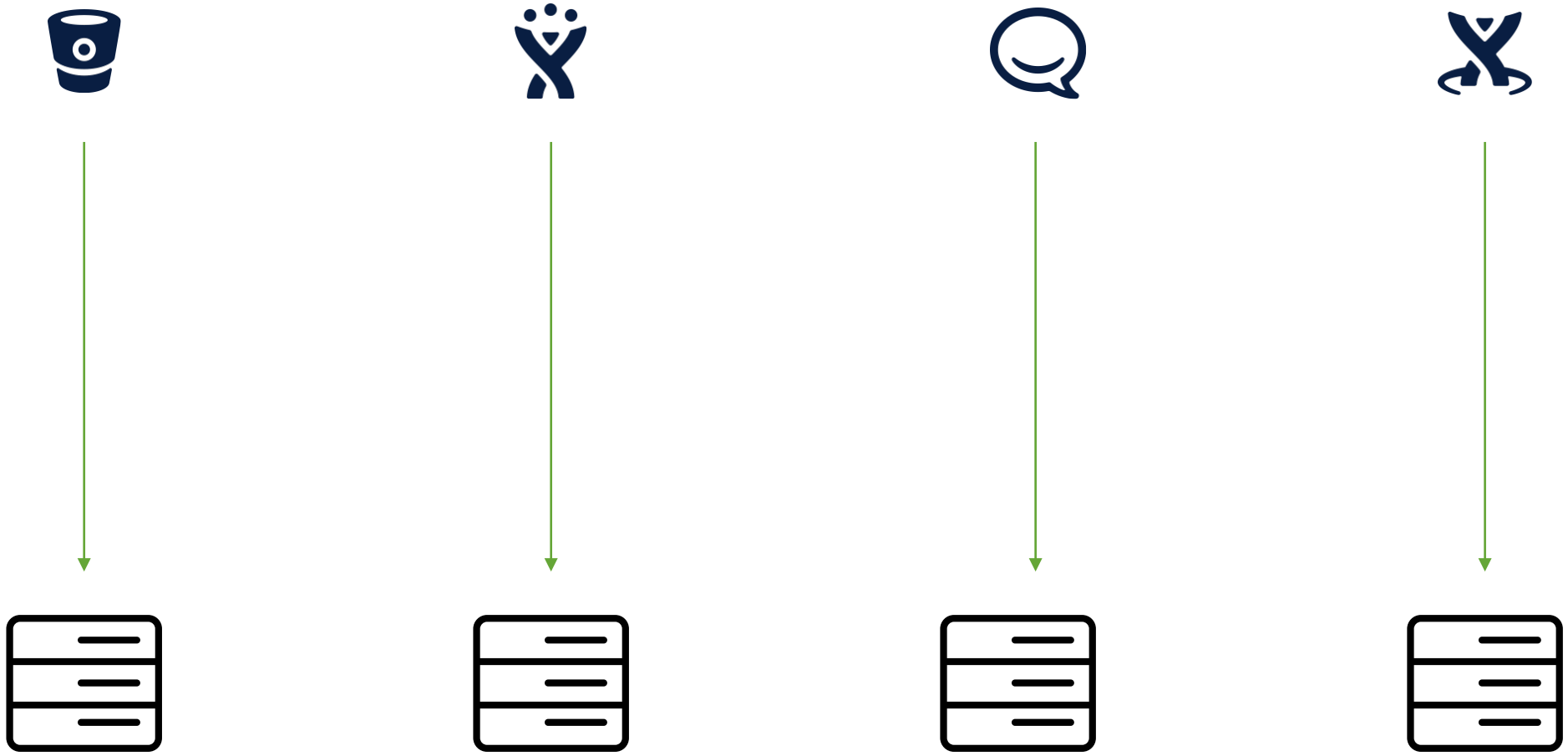
The forward-looking statements made in this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, this presentation may not contain current or accurate information. We do not assume any obligation to update any forward looking statements 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.

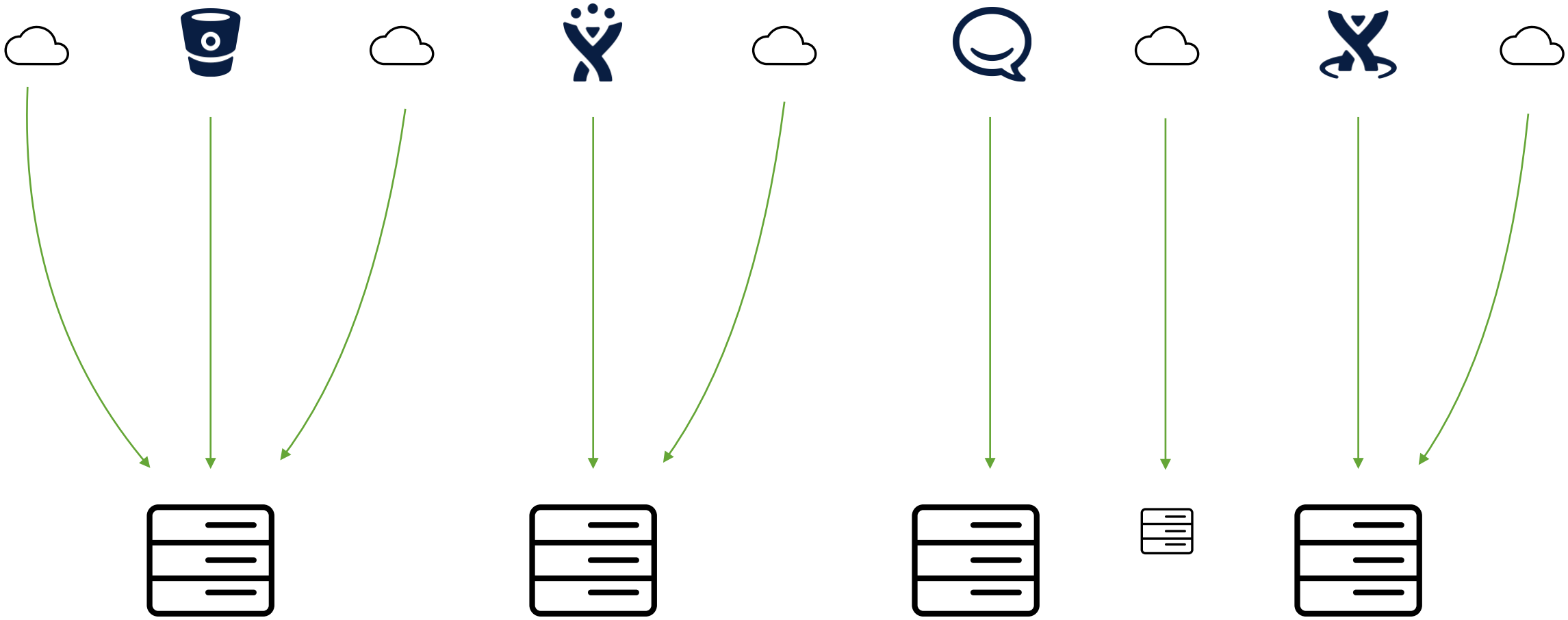
splunk> .conf2017

Where We Started

2015

splunk> .conf2017





splunk> .conf2017

Searchable

Query logs in an easy fashion

Secure

Control who can see the contents

Stored

Retained indefinitely

splunk> .conf2017

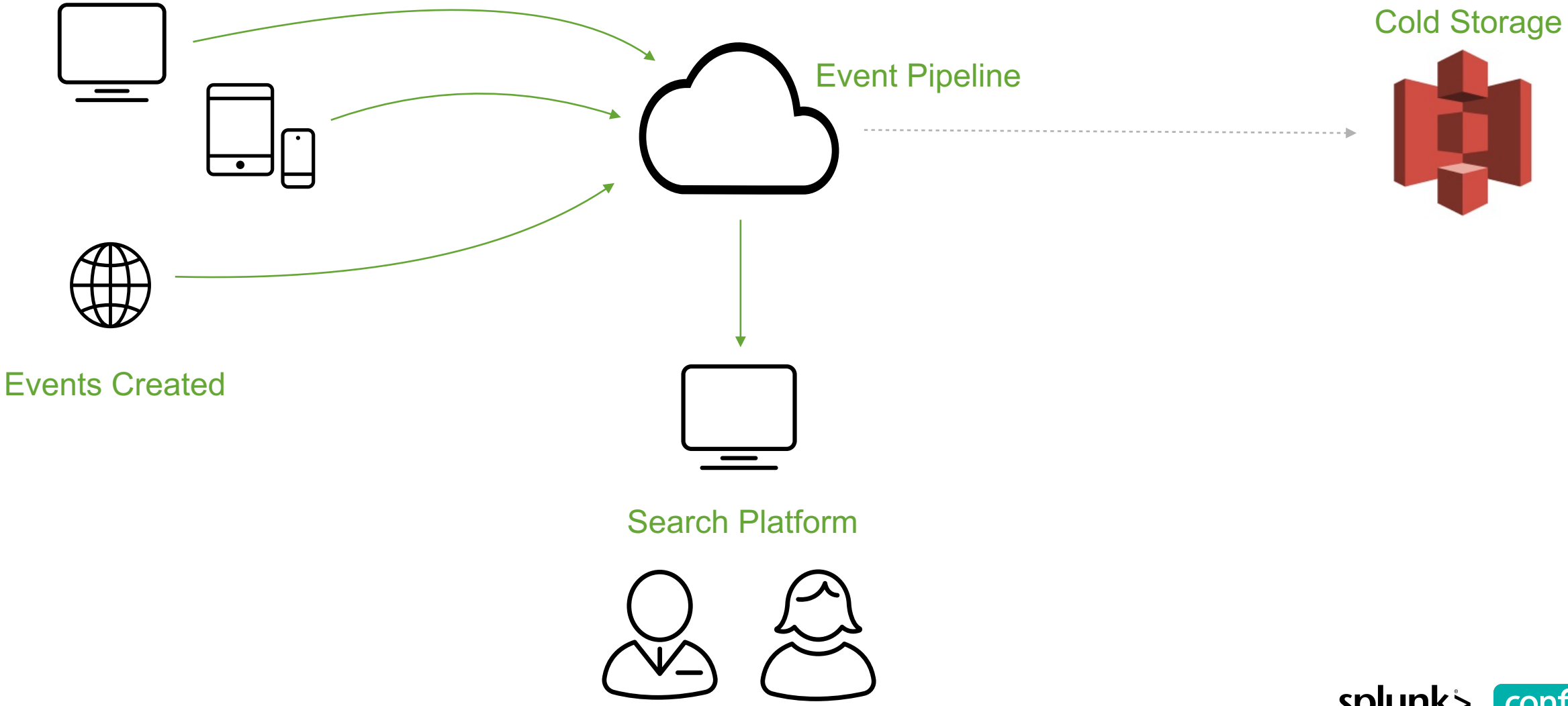
Centralized Logging

- **Core team**
  With knowledge of how to run a logging platform

Centralized Logging

- **Core team**
  With knowledge of how to run a logging platform
- **One common way**
  To send, receive and store logs

Architecture



Event Structure

- **JSON**
  With knowledge of how to run a logging platform

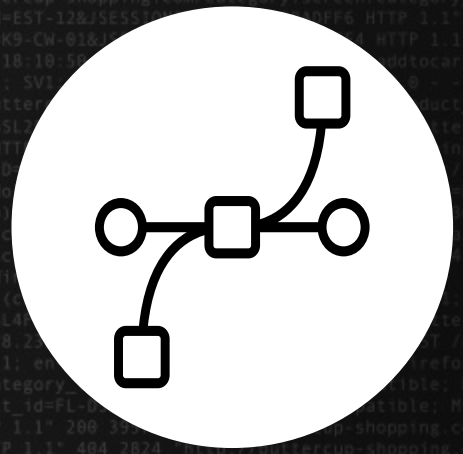
Event Structure

- **JSON**
  With knowledge of how to run a logging platform
- **Service ID**
  Unique way to identify the generating service

Event Structure

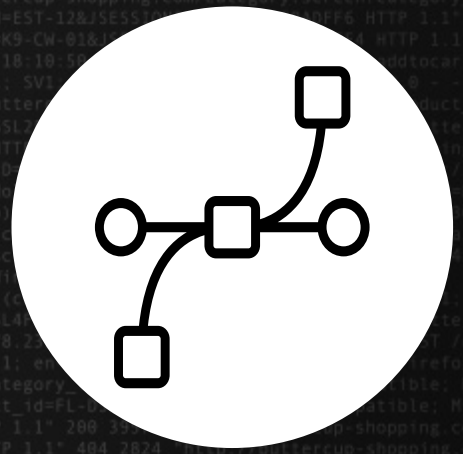
- **JSON**
  With knowledge of how to run a logging platform
- **Service ID**
  Unique way to identify the generating service
- **Time**
  Time of creation in UTC

Pipeline Requirements



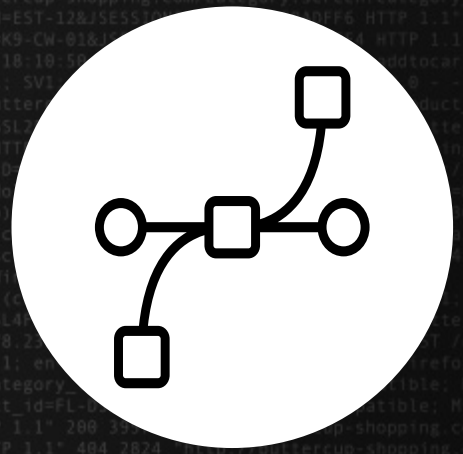
- **Scalable**
  Easy and quickly add capacity

Pipeline Requirements

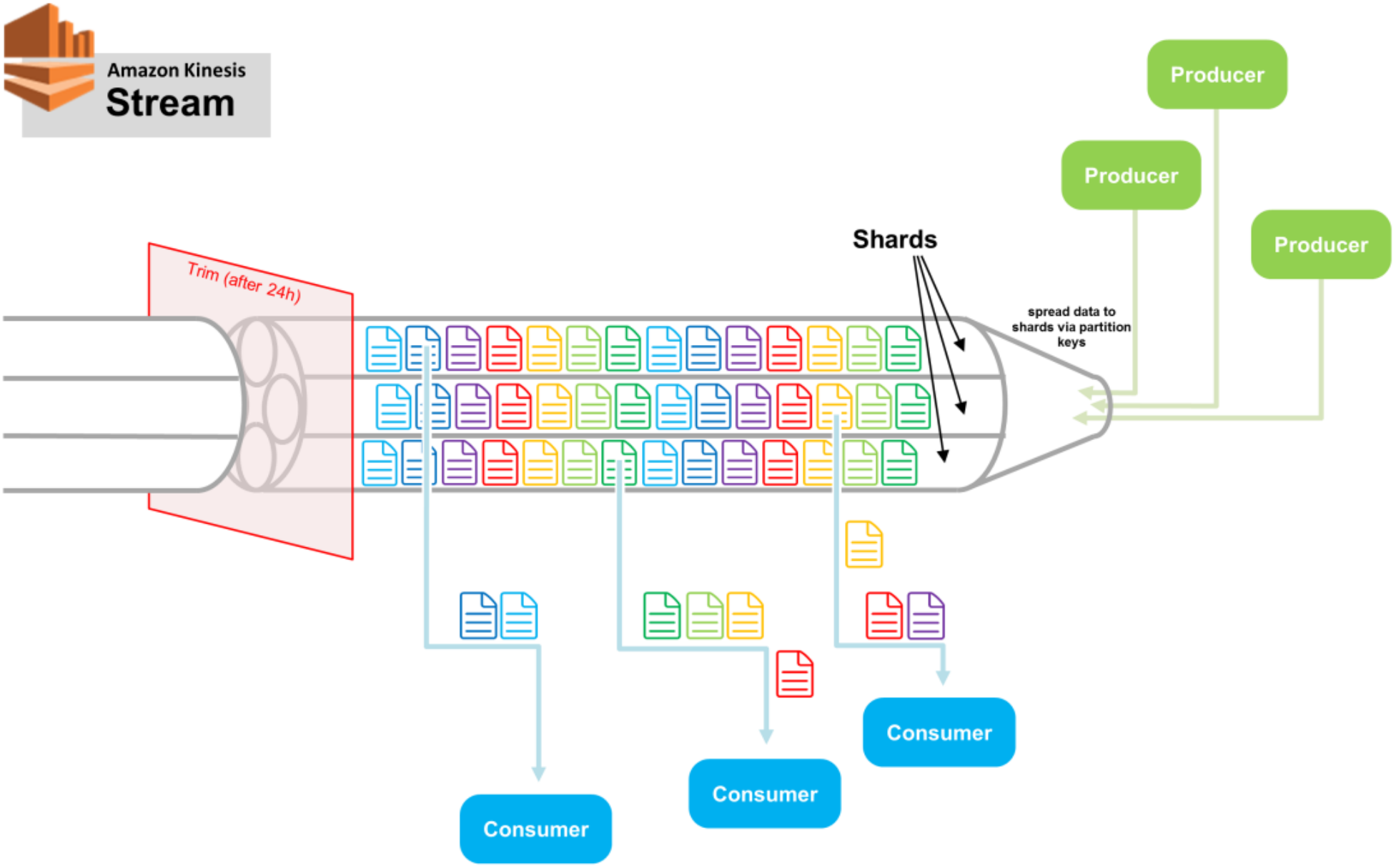


- **Scalable**
  Easy and quickly add capacity
- **Queued**
  Store events even if our consumers stop processing

Pipeline Requirements

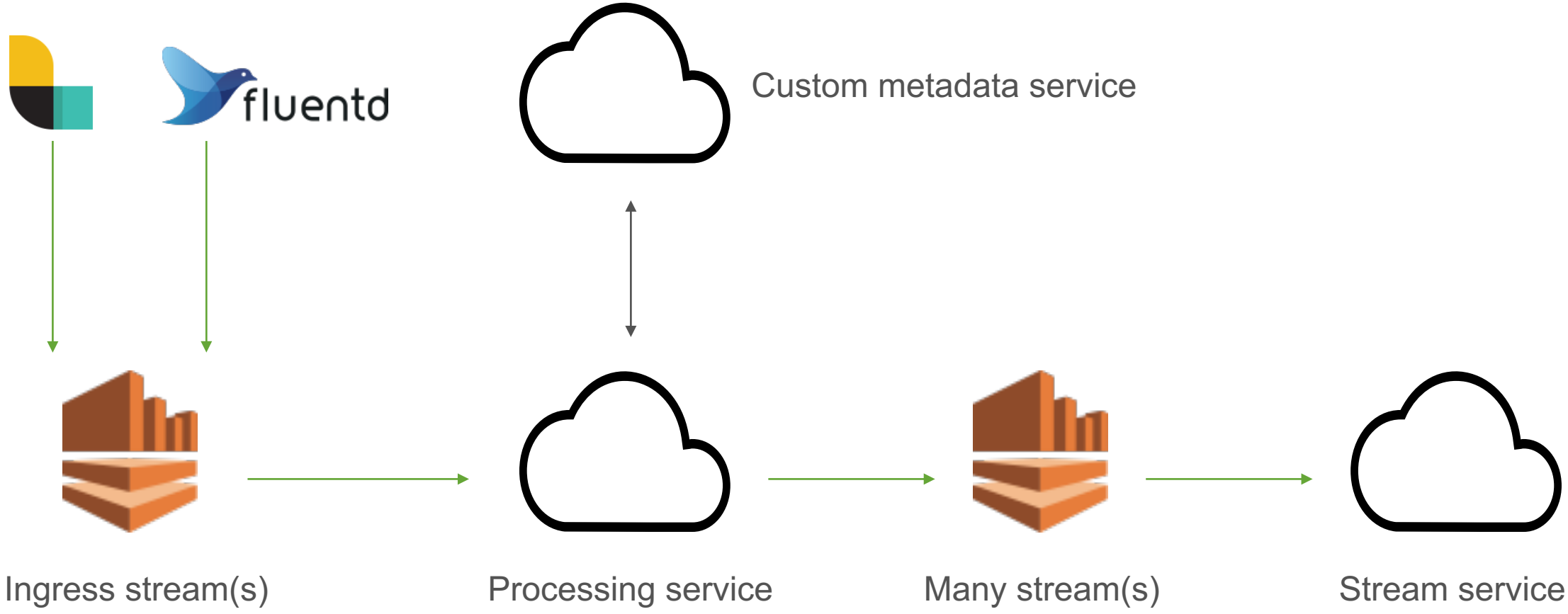


- **Scalable**
  Easy and quickly add capacity
- **Queued**
  Store events even if our consumers stop processing
- **Large Volume**
  Thousands of producers, many consumers

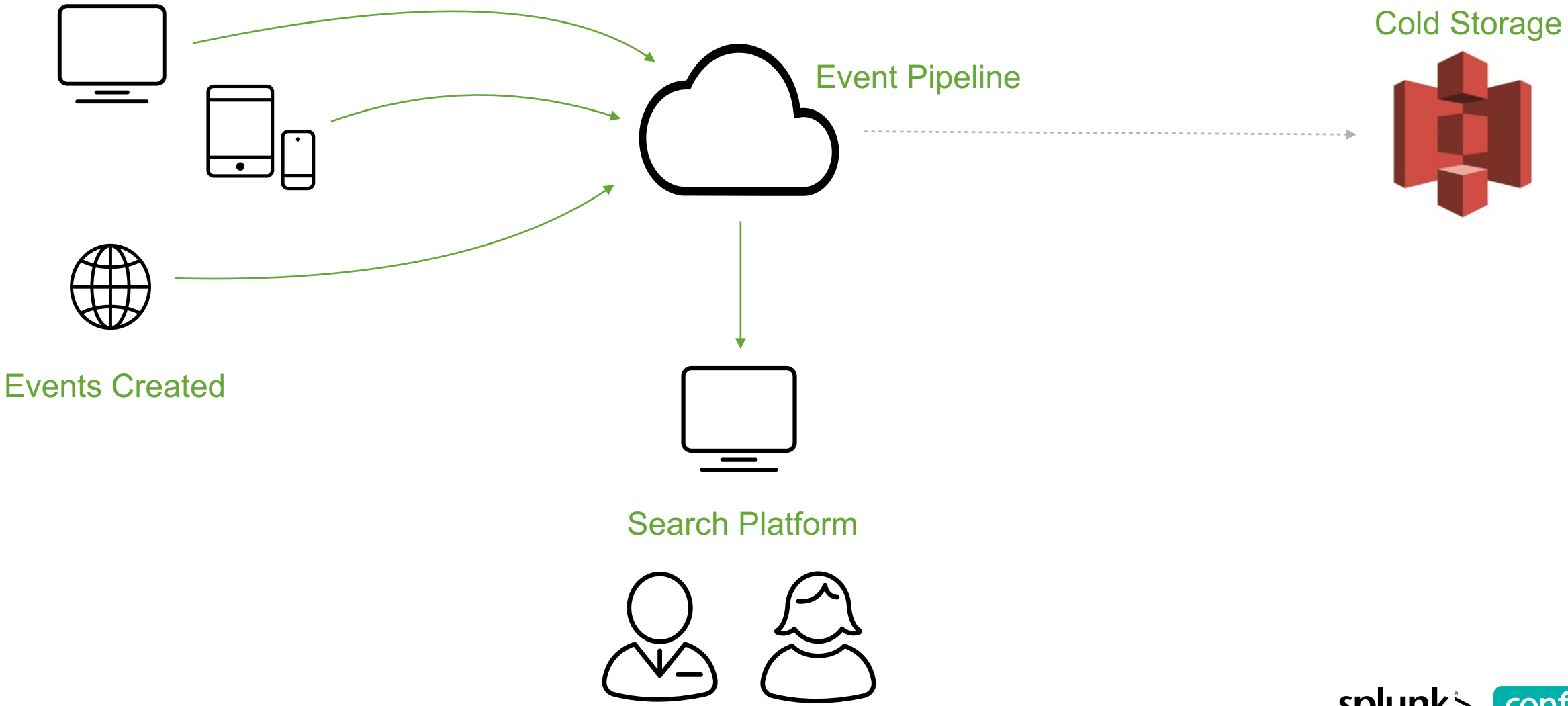


http://fbrnc.net/blog/2016/03/messaging-on-aws

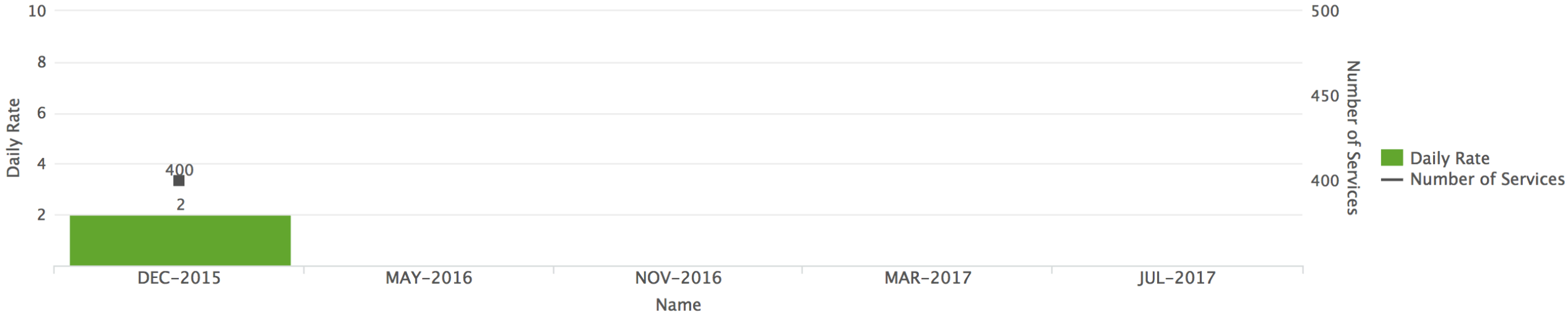
Event Pipeline



Recap



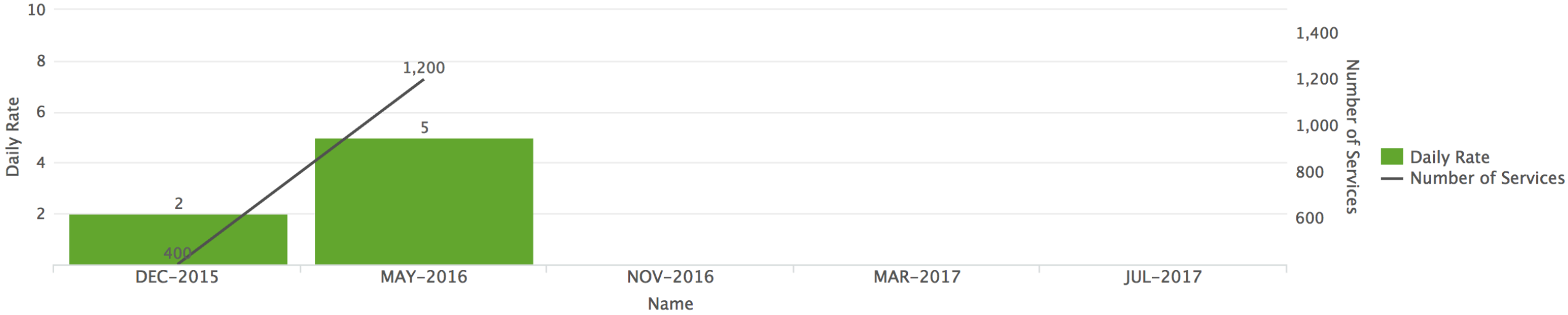
2015 Capacity



2016

splunk> .conf2017

2015 Capacity



Problems

- **Stability**
  Ingestion not keeping up resulting in long delays

Problems

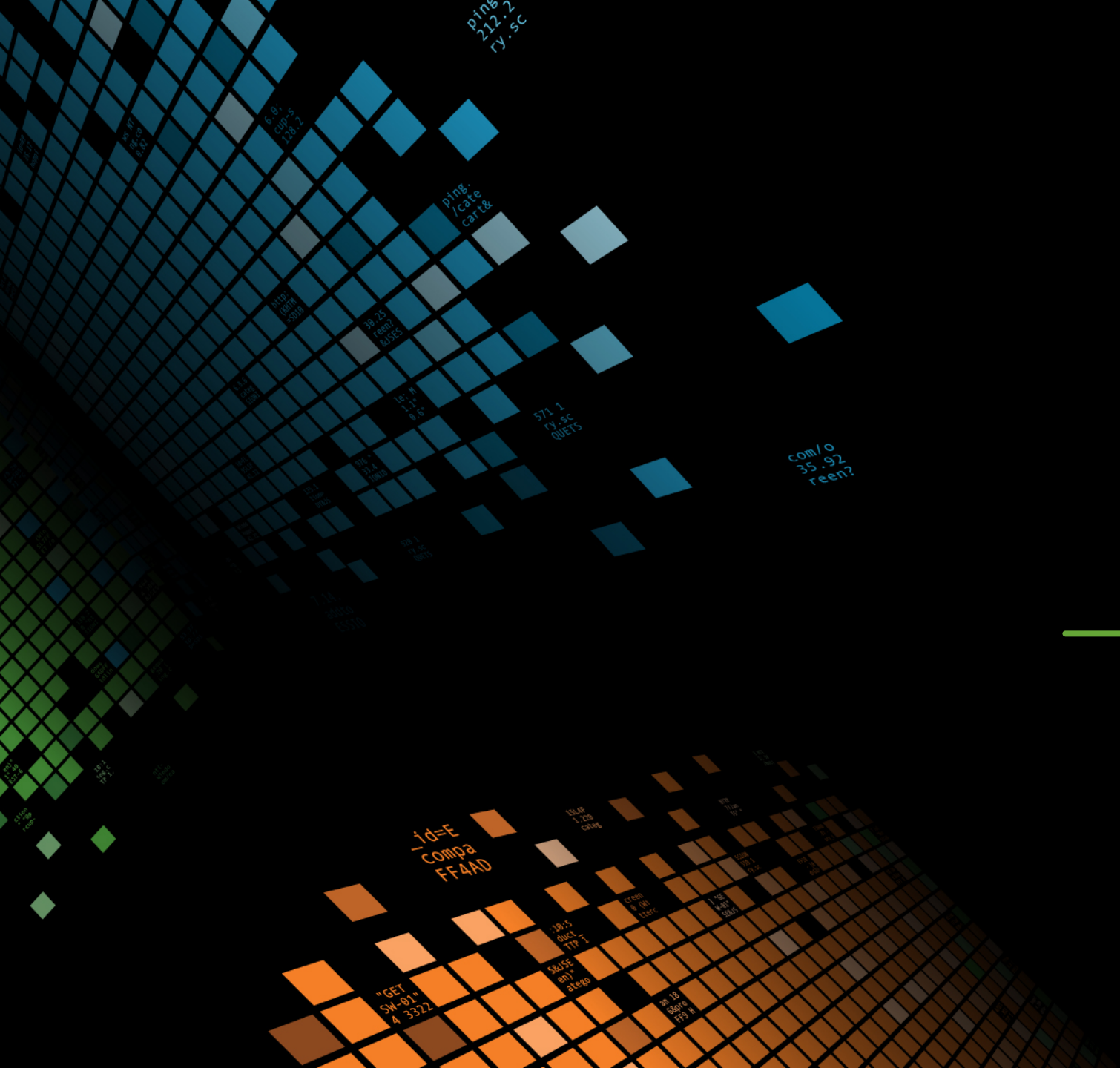
- **Stability**
  Ingestion not keeping up resulting in long delays
- **Scale**
  Indexing model didn’t keep pace as more services were brought onboard

Problems

- **Stability**
  Ingestion not keeping up resulting in long delays
- **Scale**
  Indexing model didn't keep pace as more services were brought onboard
- **User Experience**
  One of the top complaints on our shared development platform

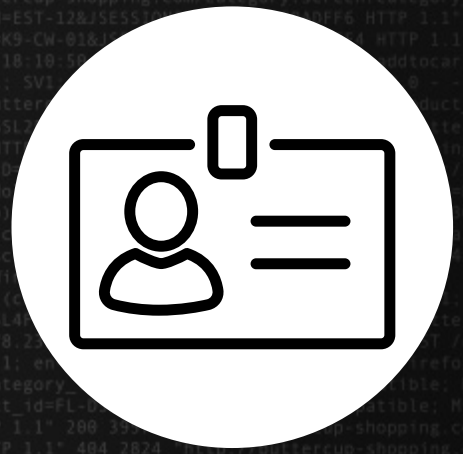
Let's Move To Splunk!

(oh, and do it in 3-4 months)



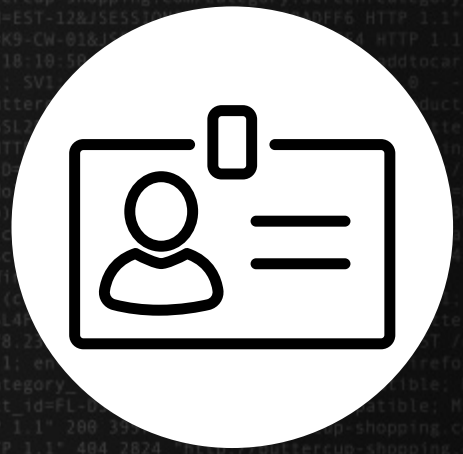
How We Did It

Splunk Had A Place In Security



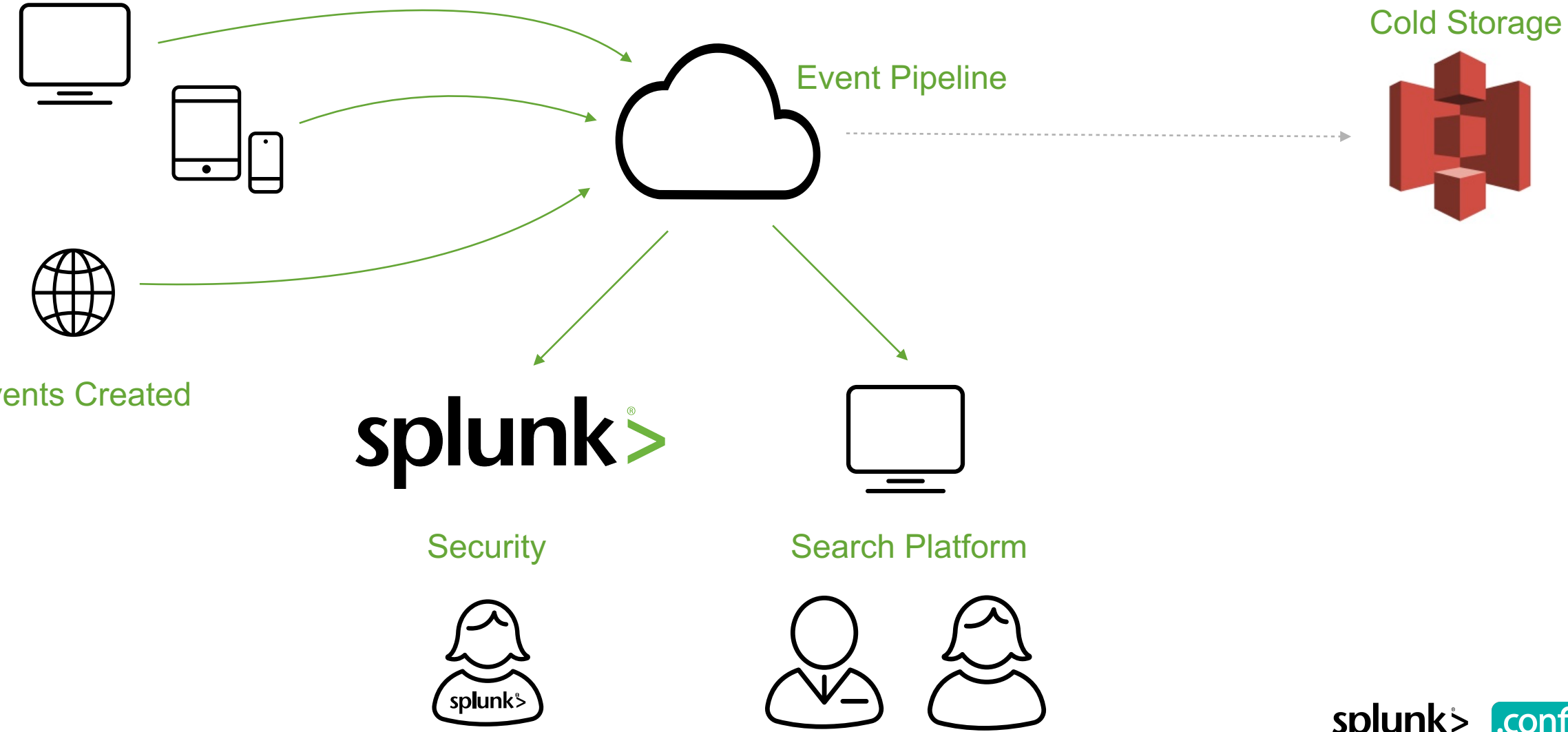
- Security incident detection

Splunk Had A Place In Security

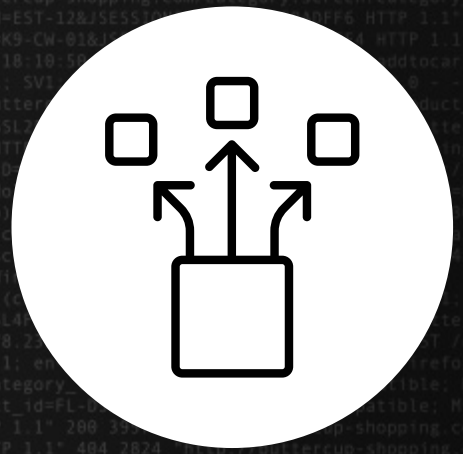


- Security incident detection
- **Limited users & data**
  Very restricted user base and subset of logs

Architecture

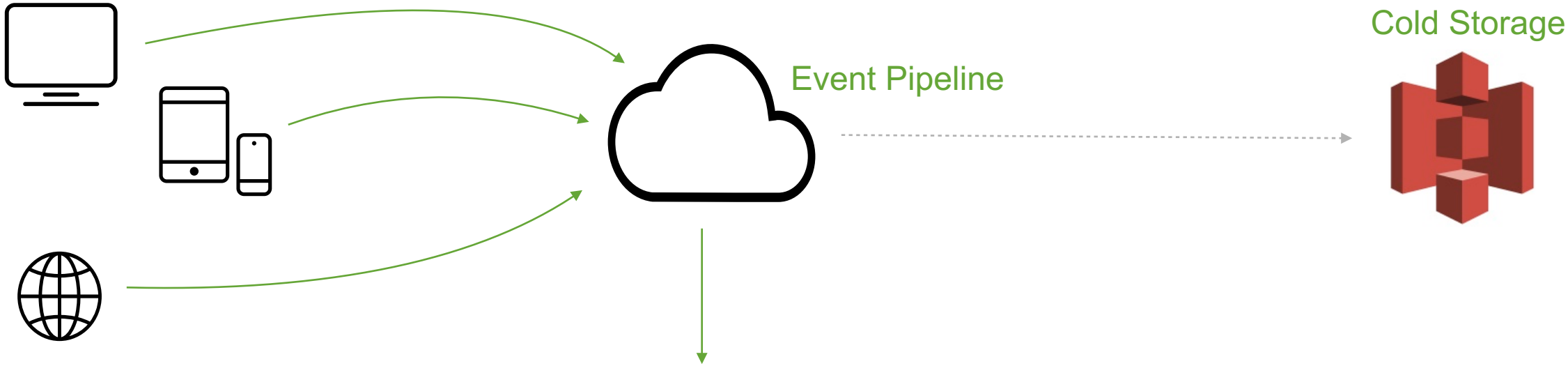


HTTP Event Collector



- Our log producers and pipeline already support JSON structured logs. Input to Splunk what is already being produced.

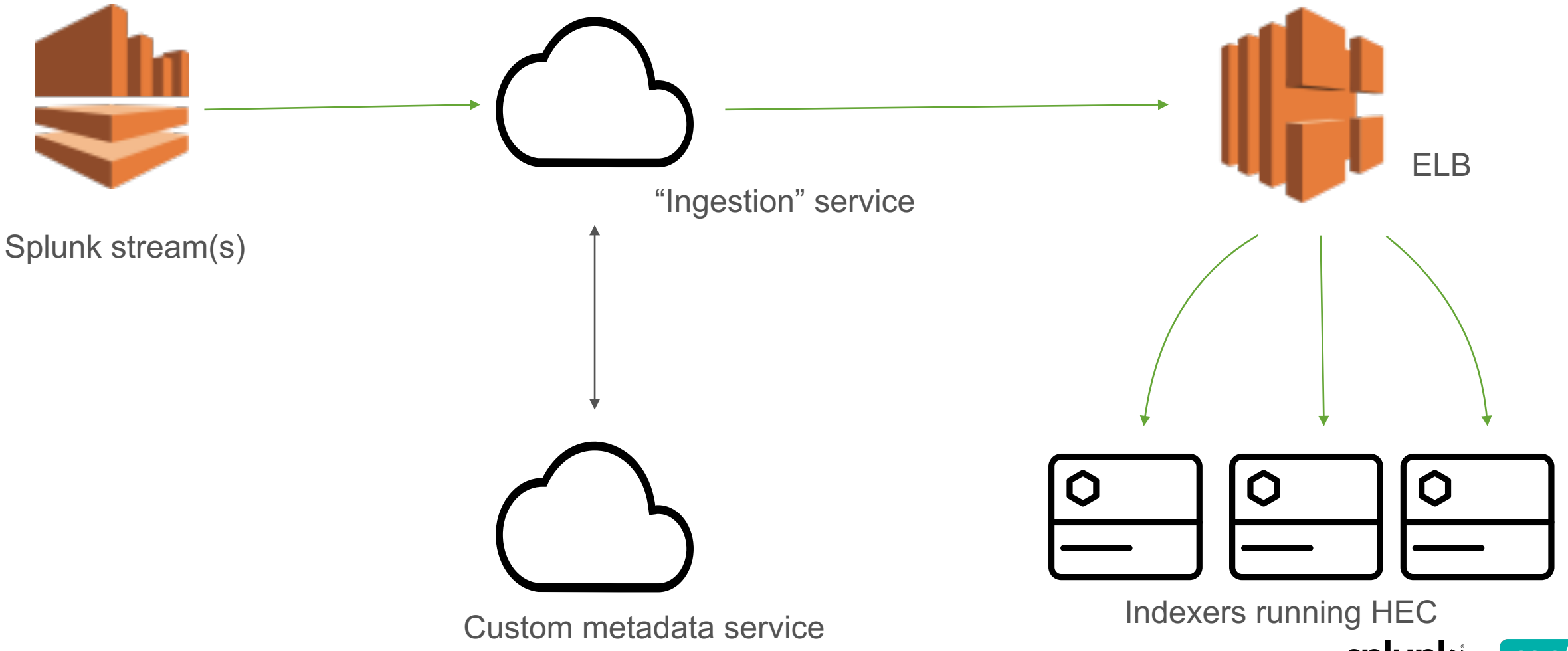
Goal



splunk>

All Atlassian

Splunk Ingestion



Indexing Strategy

Thousands Of Services

Handle having many thousands of services - too many for index per service

Make Them Easy To Find

Map our serviceID to ‘source’
Auto generate eventtype

No Custom Retention

Clustering Strategy

AZ is Site

AWS AZ maps nicely onto Splunk site, allowed multi-site clusters

Multiple Indexing Clusters

Different indexer clusters for different types of logs

Initially just ‘production’ or ‘non production’

Single SHC

No special treatment for teams

Ansible

Infrastructure

Provision our AWS infrastructure/roles. Heavy use of dynamic cloud formation lookups

Configure

Make changes to .conf files - join clusters

Business As Usual

Create indexes, deploy apps

Metadata Service

Control Access

Provide LDAP restrictions on access to particular sets of logs - mapped to splunk groups

Manage Capacity

Top services provide their throughput and retention periods - assists in provisioning

Routing

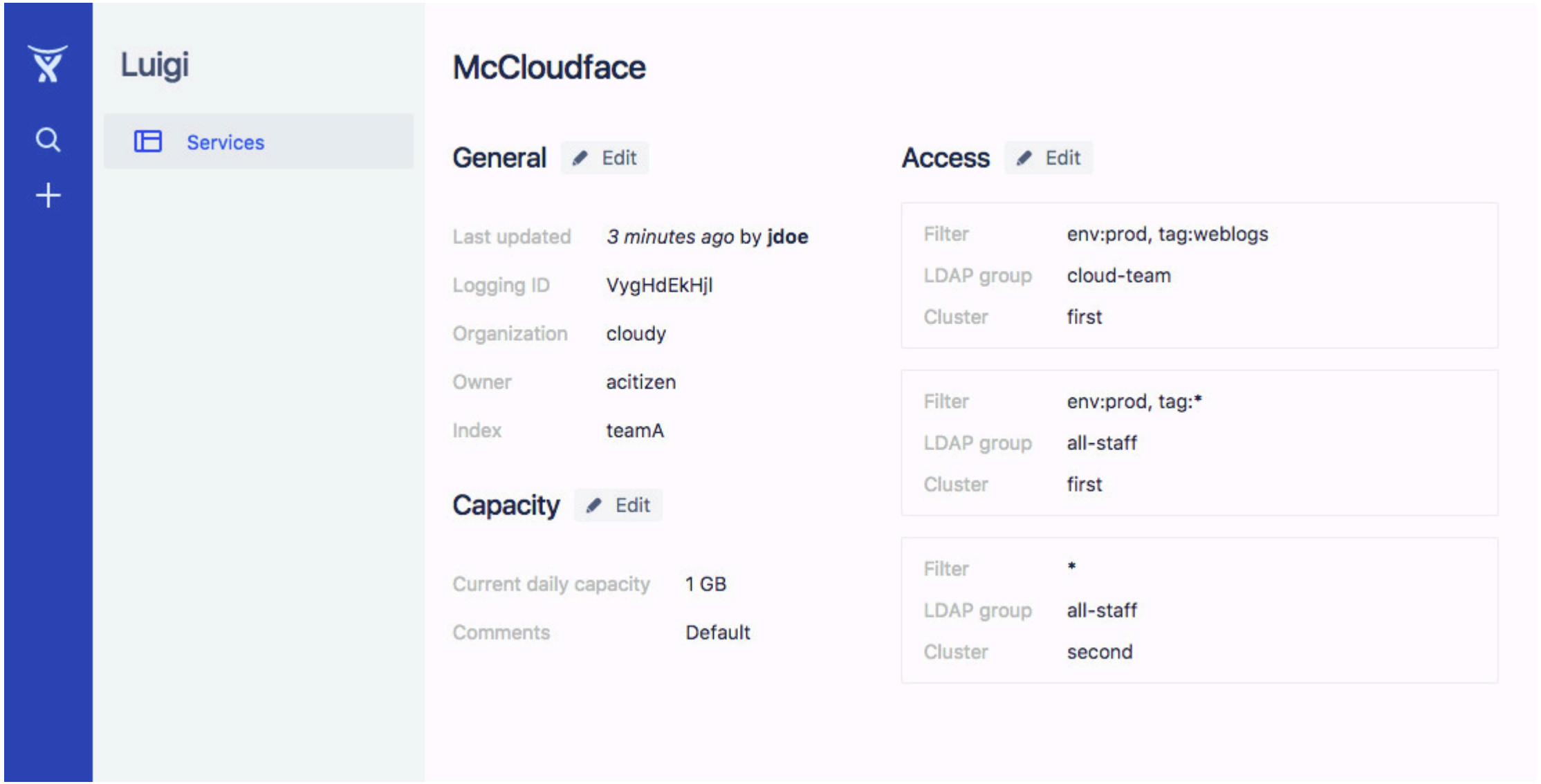
Split logs to particular clusters and indexes

What Metadata Service Looks Like

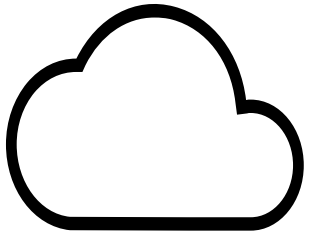
If it was backed by Google Docs

A	B	C	D	E	F	G	H	I
ServiceID	**Name**	**Organisation**	**Owner**	**Daily Capacity MB**	**Index**	**Filter**	**Cluster**	**ACL**
rJ44X1ds	TestService	SRE	acitizen	1500	teamA	env:prod	first	all-staff
						*	second	all-staff
f500GjX1	McCloudface	Cloudy	jdoe	4500	teamB	env:prod,tag:weblogs	first	cloud-team,security
						env:prod,tag:*	first	all-staff
						*	second	all-staff

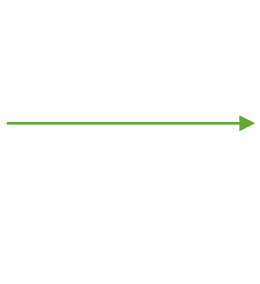
What Metadata Service Looks Like



Splunk Changes



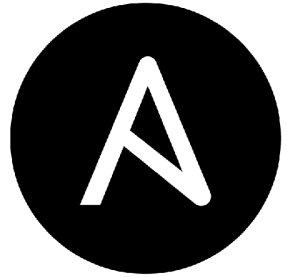
Custom metadata service



crontab

splunk>

Infrastructure
ASG, CF

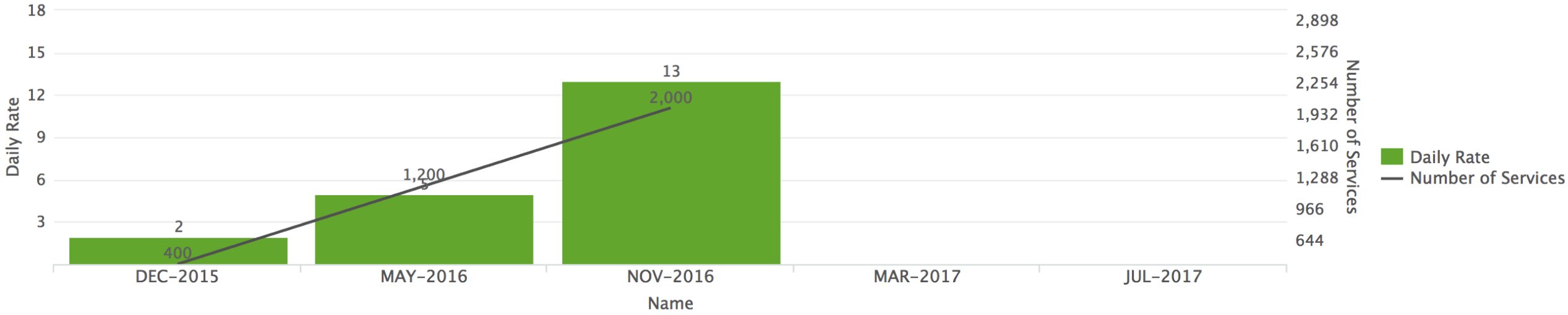


Configuration
indexes, apps

How Did We Go

splunk> .conf2017

2015 Capacity

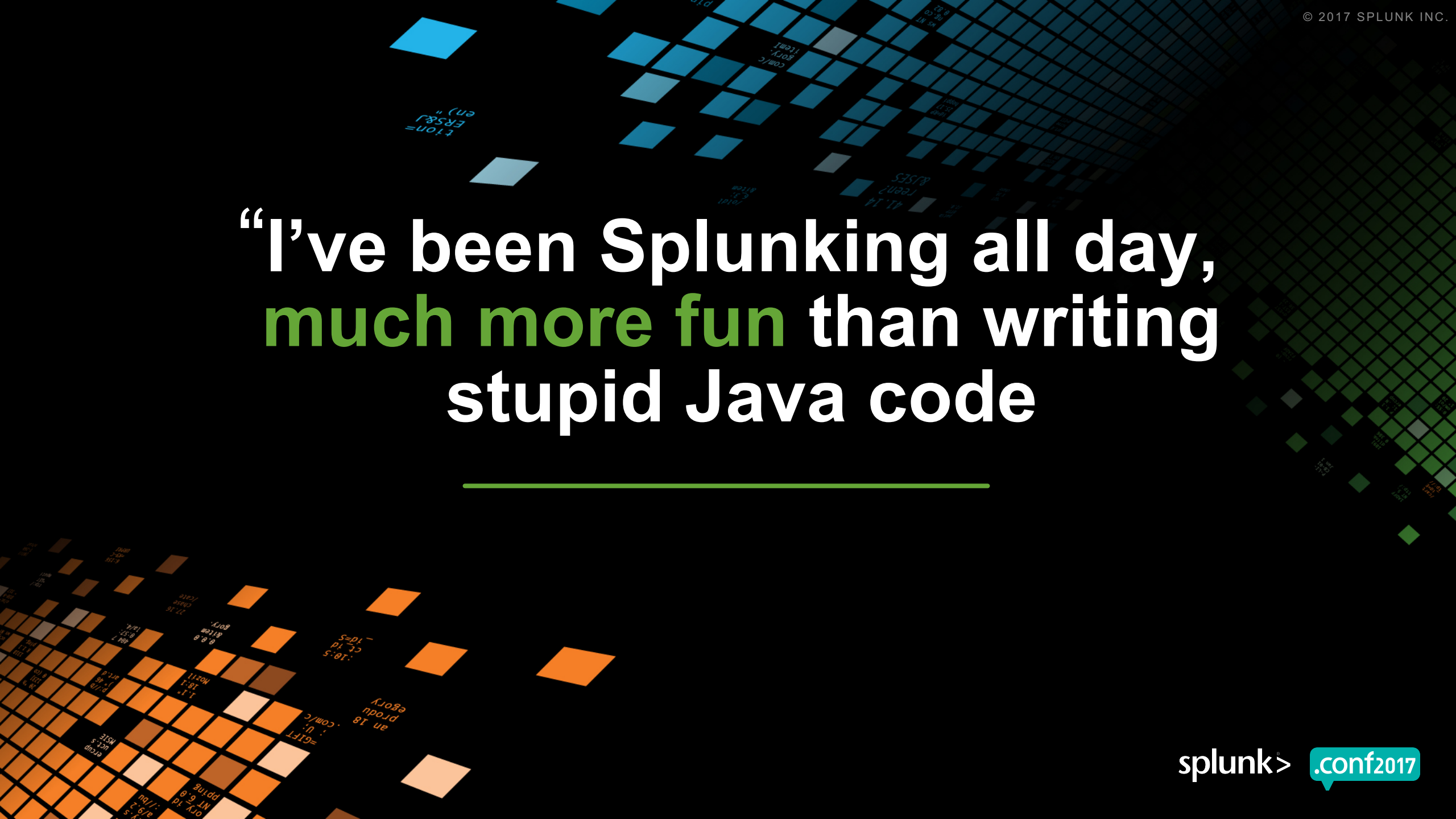


“Splunk makes me feel smart

splunk> .conf2017

"There are so many things we can do on Splunk that were never possible in the old system!

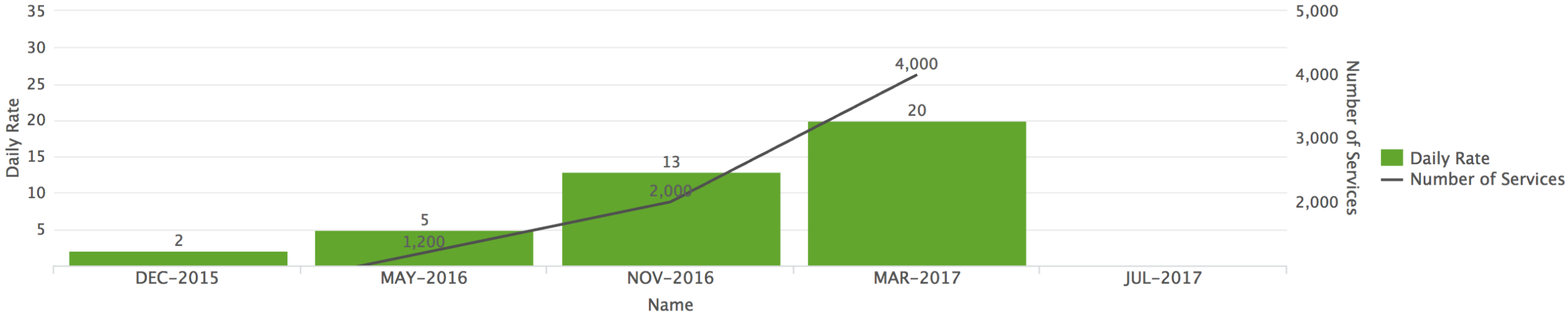
"The Splunk team helped me get my feet wet, and showed me that there *is* life after the old system.



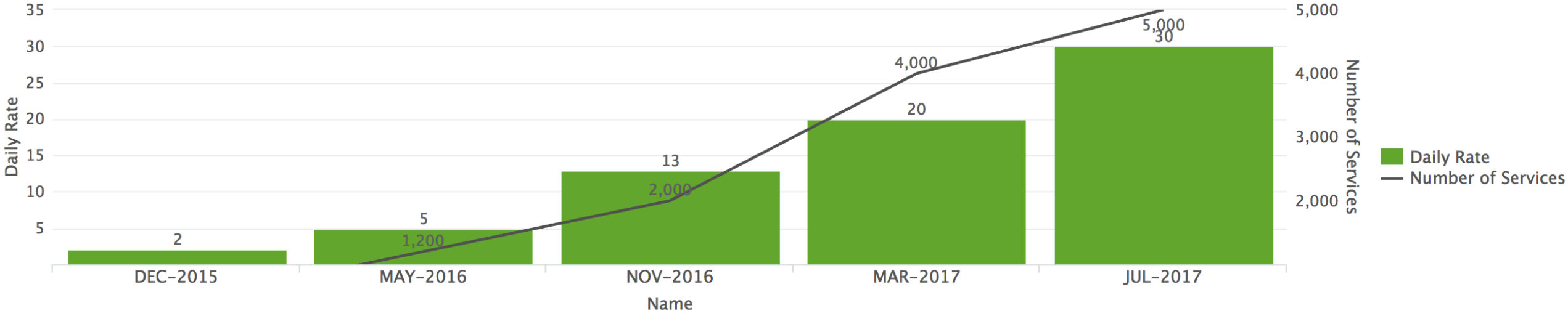
“I’ve been Splunking all day, much more fun than writing stupid Java code

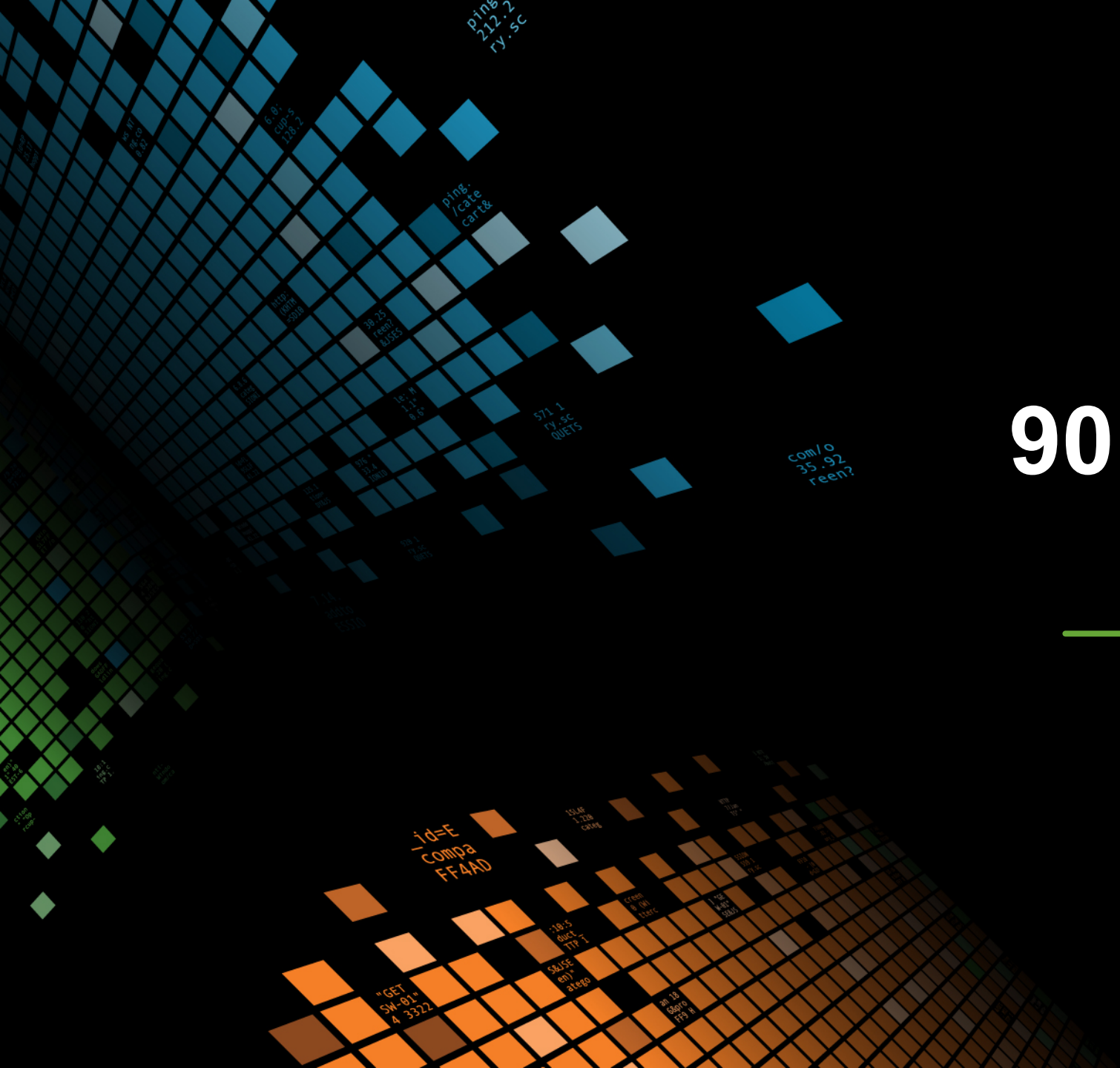
Logging went from a constant pain point to a feature of the platform.

2015 Capacity



2015 Capacity





90,000 Searches Every Day

splunk> .conf2017

Over 4x Our Planned Capacity

splunk> .conf2017

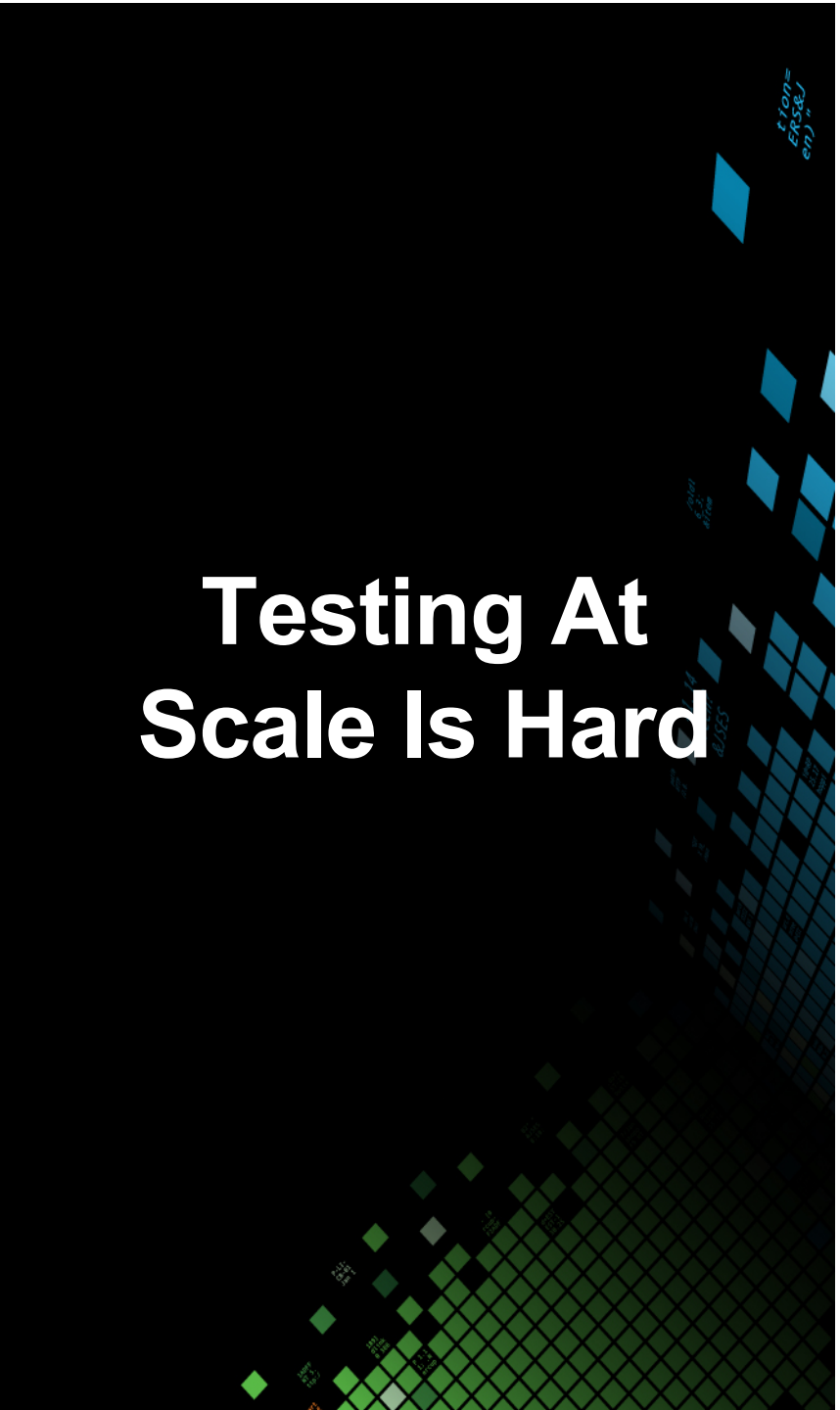
What Worked And What Didn't

splunk> .conf2017

Using Kinesis & the HEC at scale

- Read off Kinesis with KCL workers, not Lambda
- HEC on each indexer, all behind an ELB
- Load balancing pools need an accurate healthcheck

Testing At Scale Is Hard



- Difficult to replicate production load for tests
- Run into problems only in prod
- Game Week!

Game Week

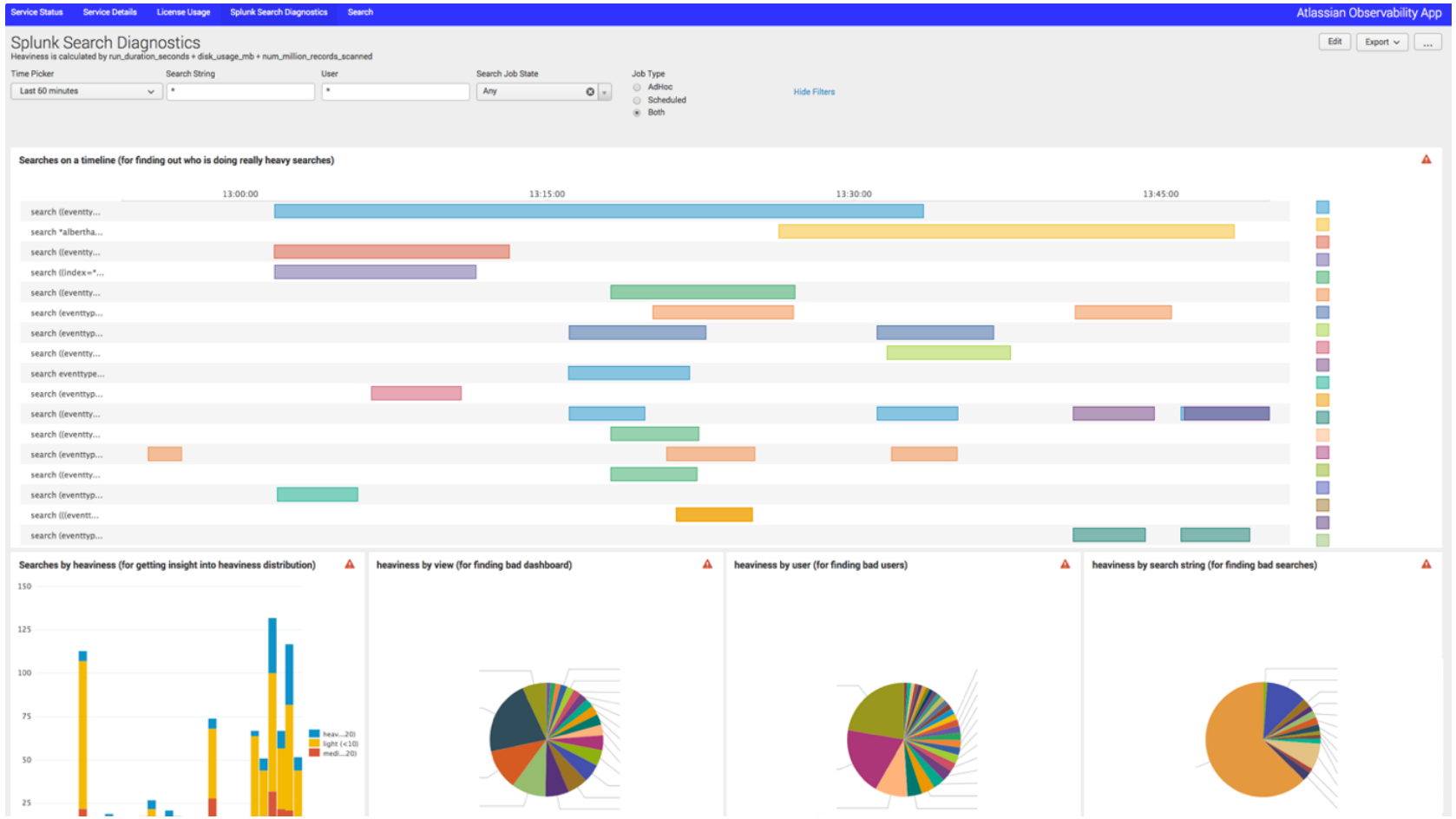
Confluence Spaces People Questions Calendars Create ••• Feedback

Disaster	Actions	Expected Alerts	Expected Monitoring (graphs, trends, dashboards)	Expected Loss of Functionality	Expected Recovery Process	✅ Alerts	✅ Monitoring	✅ Functionality	✅ Recovery
Lose an indexer node	• Terminate an indexer instance in AWS console	• host alert - HipChat • no PD	• DMC shows host down • node count reduced by 1	• OE: None • Users: None	Autoscaling group brings up a new node. Team runs playbook to configure it.	✅ HipChat alert when node goes down ❌ Get PD because missing data in datadog ❌ Unexpected ingestion delay alert	✅ DMC shows node was down ✅ Splunkd process count went down ✅ Replication traffic went up	❌ Unexpected ingestion delay	✅
Lose an AZ where master is not in	• Isolate the AZ where indexer master is. • Use network ACL to isolate the site • Detail step see step 3 on: Runbook - Amazon AZ Failure - Staging DR Test - 29th November 2016	• host alert - HipChat	• DMC shows site is down, part of the searchhead cluster is down	• OE: None • Users: None	Autoscaling group brings back up the site in the other AZ. Team runs playbook to configure it.	❌ Unexpected ingestion delay alert ❌ No host alerts ❌ Unexpected splunk process count alert	✅ DMC monitoring ✅ Ingestion monitoring	❌ We didn't expect ingestion delay ❌ We didn't expect indexer throttling	✅
Lose an AZ	• Use network	• ingestion	• DMC	• OE: None	Autoscaling	✅	✅ DMC	✅ Ingestion	✅ Self Heal

An Open Platform Has Challenges

- Openness and flexibility are important to us
- Splunk per-user and global limits will save you
- Sledge hammer capabilities (can_delete, admin_all_objects)

Search Diagnostics



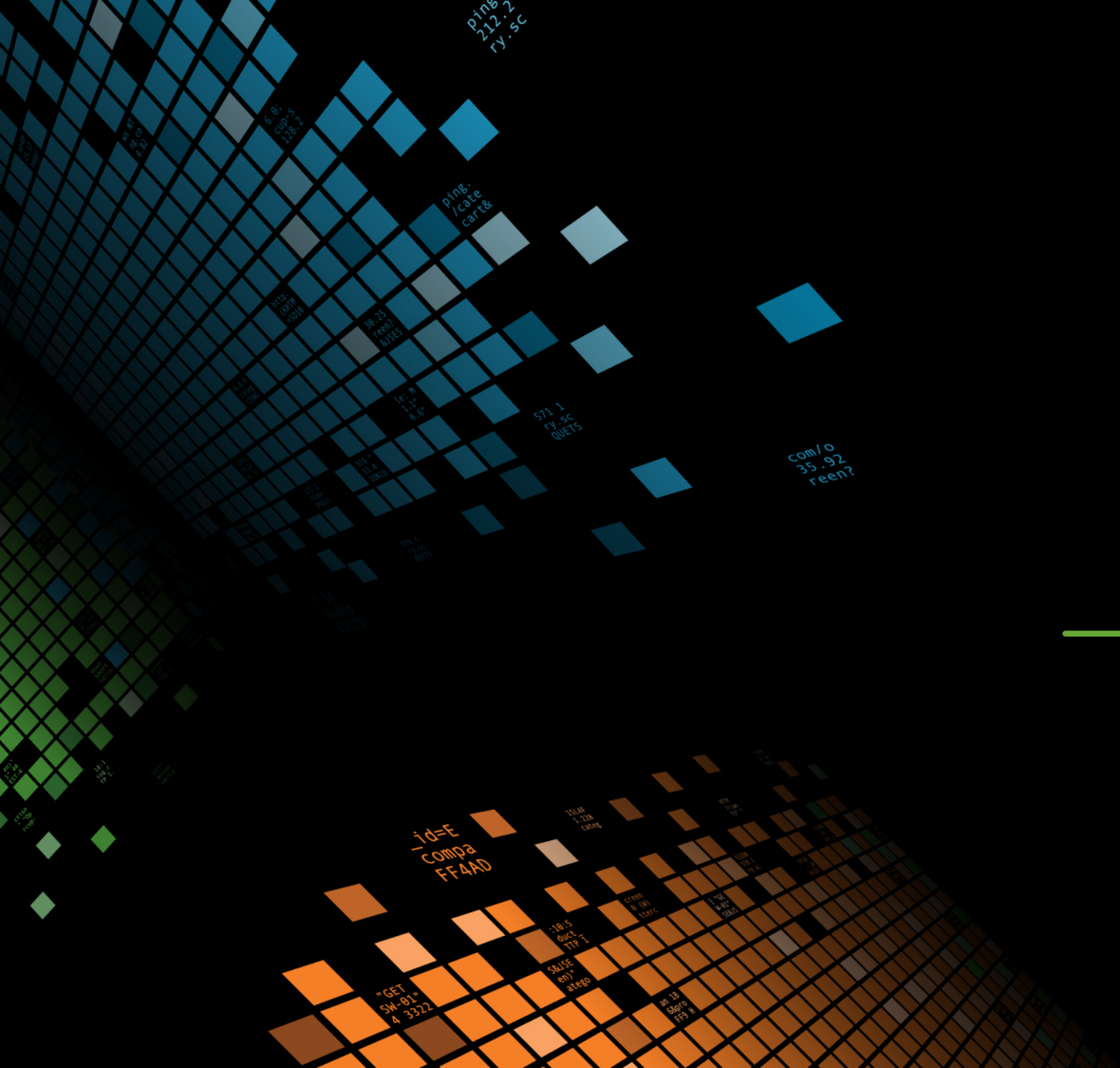
splunk> .conf2017

Know Your Resource Constraints

- Splunk likes disk IOPs, a lot
- Fixup tasks, bucket copying, heavy search load
- Limits help, but eventually you will run into resource constraints

Review Your Clustering Strategies

- Indexing and clustering choices affect performance
- Making the right choice is worth the time



The Future

Partitioned Services

Make better use of the resources we have

More Clusters

Limit blast radius of bad actors

Containers

Docker/Kube

Thank You!

Don't forget to rate this session in the .conf2017 mobile app