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, Splunk>, Listen to Your Data, The Engine for Machine Data, Splunk Cloud, Splunk Light and SPL are trademarks and registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2017 Splunk Inc. All rights reserved.
Where We Started
Searchable
Query logs in an easy fashion

Secure
Control who can see the contents

Stored
Retained indefinitely
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

Events Created

Search Platform

Event Pipeline

Cold Storage
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

Ingress stream(s) → Processing service → Many stream(s) → Stream service

Custom metadata service
Recap

Event Pipeline

Cold Storage

Events Created

Search Platform

© 2017 SPLUNK INC.
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

Event Pipeline

Cold Storage

Events Created

Security

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

Event Pipeline

Cold Storage

Events Created

All Atlassian
Splunk Ingestion

Splunk stream(s) → “Ingestion” service → ELB

“Ingestion” service

Custom metadata service

Indexers running HEC
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
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
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

<table>
<thead>
<tr>
<th>ServiceID</th>
<th>Name</th>
<th>Organisation</th>
<th>Owner</th>
<th>Daily Capacity MB</th>
<th>Index</th>
<th>Filter</th>
<th>Cluster</th>
<th>ACL</th>
</tr>
</thead>
<tbody>
<tr>
<td>rJ44X1ds</td>
<td>TestService</td>
<td>SRE</td>
<td>acitizen</td>
<td>1500</td>
<td>teamA</td>
<td>env:prod</td>
<td>first</td>
<td>all-staff</td>
</tr>
<tr>
<td>f500GjX1</td>
<td>McCloudface</td>
<td>Cloudy</td>
<td>jdoe</td>
<td>4500</td>
<td>teamB</td>
<td>env:prod,tag:weblogs env:prod,tag:*</td>
<td>first</td>
<td>cloud-team,security all-staff</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td>second</td>
<td>all-staff</td>
</tr>
</tbody>
</table>
## What Metadata Service Looks Like

### Luigi

#### McCloudface

<table>
<thead>
<tr>
<th>General</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last updated</td>
<td>Filter</td>
</tr>
<tr>
<td>Logging ID</td>
<td>env:prod, tag:weblogs</td>
</tr>
<tr>
<td>Organization</td>
<td>LDAP group</td>
</tr>
<tr>
<td>Owner</td>
<td>all-staff</td>
</tr>
<tr>
<td>Index</td>
<td>Cluster</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Capacity</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Current daily capacity</td>
<td>first</td>
</tr>
<tr>
<td>Comments</td>
<td>second</td>
</tr>
</tbody>
</table>

Last updated 3 minutes ago by jdoe

Logging ID: VyghDekHjl

Organization: cloudy

Owner: acitizen

Index: teamA

Current daily capacity: 1 GB

Comments: Default
How Did We Go
2015 Capacity
“Splunk makes me feel smart"
“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
90,000 Searches Every Day
Over 4x Our Planned Capacity
What Worked And What Didn’t
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

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