Let's Chat About Splunk and ELK...

Kate Lawrence-Gupta

Platform Architect Splunk |
klawrencegupta@splunk.com

Forward-Looking Statements

During the course of this presentation, we may make forward-looking statements regarding future events or plans 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 may differ materially. The forward-looking statements made in the this presentation are being made as of the time and date of its live presentation. If reviewed after its live presentation, it may not contain current or accurate information. We do not assume any obligation to update any forward-looking statements made herein.

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 functionalities described or to include any such feature or functionality in a future release.

Splunk, Splunk>, Turn Data Into Doing, 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. © 2019 Splunk Inc. All rights reserved.

Kate

Almost 20 years experience in infrastructure management, systems operations, security & big data architecture

Spent the last 6 years with Comcast

- Principal Engineer (Splunk)
- Senior Manager of Engineering & Software Development
  - Focus on open source integrations with existing data platforms

Inaugural SplunkTrust member & 2013 Revolution Award Winner (Innovation)

Joined Splunk ~18 months ago as Platform Architect in the Global Engineering team

Data is Critical

Extracting Value

Which path to take?

Splunk or ELK...?

What is ELK?

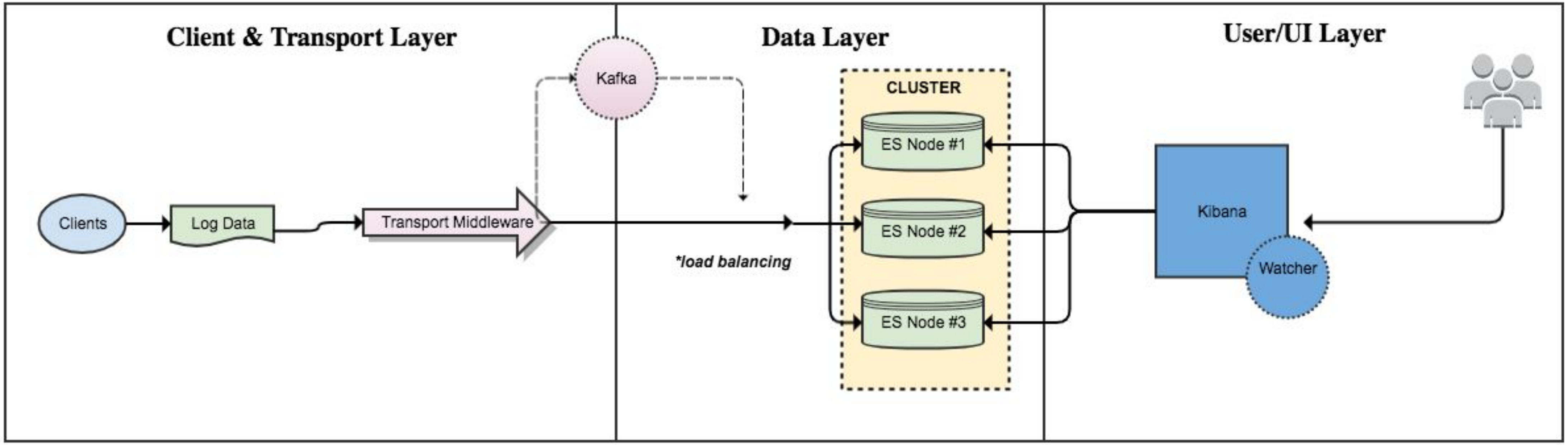
ELK represents a suite of open source tools that work together in a stack to provide a complete experience to the end user for managing log data.

- **ElasticSearch**
  - The data layer where the log data is physically stored on disk in indices.
- **Logstash**
  - The transport or middleware layer that allows the log data to be sent from clients to ElasticSearch
  - This is also where the schema or format of your data is defined that allows for analysis

This is commonly referred to as schema-on-write methodology

- **Kibana**
  - The user interface (UI) that allows the user to investigate, analyze & visualize the data stored in ElasticSearch

ELK Logical Overview



ELK

PROS

- Open Source
- Active Development Community
- Container Based Ecosystem
- Lucene is a robust Query Language
- Later versions have addressed storage requirements & data compression
- Hosted Solutions are available (AWS, Elastic Cloud, GCP)
- Additional support is available from vendors for on-premise or cloud based deployments
- Beats Central Config/Logstash Pipeline UI available for client/middleware management
- Flexible integration models available
- Learning curve is relatively low

CONS

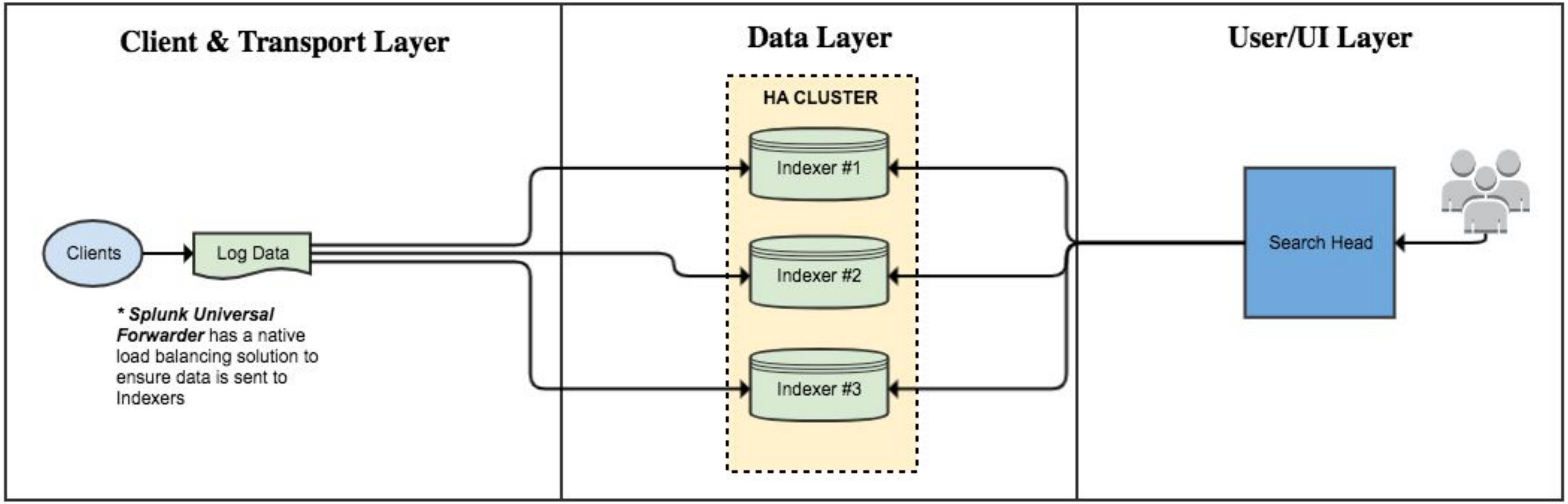
- Schema-on-write methodology can be difficult to manage and does not work well for unstructured data sets
- Very basic capability to extract fields at search time
- Other Domain Specific Languages will be used depending on the product. Lucene Query Syntax, Elastic Query Syntax, Kibana Query Syntax, SQL, etc.
- Data enrichment generally take place at ingest time which requires the users to know the questions of data ahead of time. Joins (Canvas) & Lookups (Kibana) can be used to mitigate some of these factors.
- Very large scale architecture can also be challenging in terms of design
- Kibana performance issues with large datasets & proximity searches (*Elasticsearch aggregations can assist here)
- Managing large deployments can be complex due to sharding strategies

What is Splunk?

Splunk is a an enterprise ready commercial solution designed for machine data search and analysis. It has 3 major components:

- Universal Forwarder
  - This is the client layer that with the Splunk forwarding agent deployed will tail logs, monitor TCP ports, or run custom scripts and is designed to send data to Splunk indexers.
- Indexers & Cluster Master
  - The is the data layer where log data is stored & aggregated for search & other analysis.
- Search Head
  - The user interface (UI) that allows the user to investigate, aggregate & visualize the data stored in Splunk

Splunk Logical Overview



***Splunk Universal Forwarder** has a native load balancing solution to ensure data is sent to Indexers*

Splunk

PROS

- Hosted solutions are available with Splunk Cloud (AWS)
- Schema-on-demand design allows for greater flexibility for data ingest
- Data Model Acceleration & Summary Indexing features implement schema-on-write operations for better search performance.
- Indexed fields for JSON/CSV data with known structure can be implemented with schema-on-write & schema-on-demand for increased search performance.
- SmartStore feature allows use of cheaper S3-object storage to further reduce costs
- Workload Manager available to allocate search/ingest capacity (using cgroups)
- Built-in user management, LDAP & SAML integrations
- Distributed map reduction capabilities will process 100’s of millions of data points
- Data compression of 50%+ allows for more data in a smaller storage footprint
- Minimal logical limits on per cluster data storage.
- 1000’s of 3rd party apps and plugins
- Strong user-driven support community

CONS

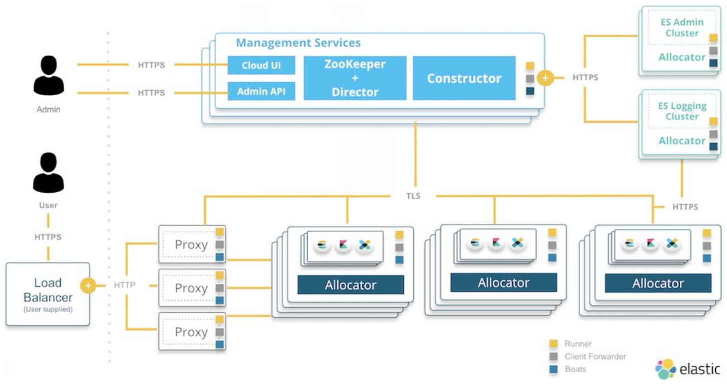
- Cost can be perceived as high
- Moderate learning curve to enable advanced analysis and features
- Logical limits on clustering/bucket replication
- Needs more nuanced documentation & reference implementation guides.
- Very large scale architecture can also be challenging in terms of design

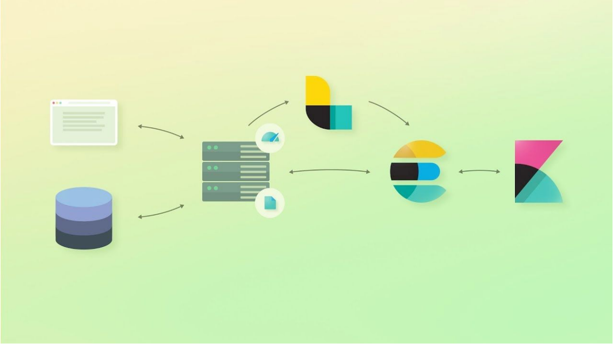
ELK - SSH Monitoring Use Case

In this ELK scenario the company *Stark Industries* would need to deploy the following high-level components (* we will assume this is all net-new deployment)

1. Deploy a Beats agent to each of the 5000 hosts
2. Build out a filebeat.yml configuration for input of SSH related logs & output to Logstash
3. Optional: Build out a Grok parsing config that matches the data to be ingested
4. Optional: Build a transport layer (logstash-server) & deploy the logstash.yml
5. Provision an ElasticSearch cluster & Kibana that matches your retention and ingest needs.
6. If Beats is the only agent used then index patterns will be set by default.
7. Using the built-in Watcher UI define the alert condition

ELK - SSH Monitoring Use Case





1. Filebeat.yml – stores the client's configuration of what files to monitor and where to output the log data. If using Beats can now be managed via the Beats Config Manager

2. Optional: Logstash.yml - Using the pipeline editor this can be managed via the UI *not shown

3. Updated Management Services component allows for UI administration of many Elastic functions that are also available via the API

4. Alert Condition – this is defined via the UI through the Watcher plugin

ELK - SSH Monitoring Use Case – Additional Considerations

Scaling

- Logstash & ElasticSearch scale horizontally
- As throughput of the deployment goes up:
  - Additional (optional) Logstash nodes & storage will be needed to process the data **before** it's indexed.

Logging also tends to be spiky meaning that capacity will need to be provisioned for peak ingest times.

This layer also requires a back-pressure memory configuration to handle persistent queuing. The default is 4Gb but may need to be increased to 8GB as deployment throughput goes up.

- 6.x version now include Lucene 7 and the _all field is now disabled as default.
  - This provides a 50% reduction in size for indexes with sparse fields
  - Removal of the_all field can mean a 40% reduction in all index sizes
- Frozen indices are also available for data that requires searching but at a throttled rate

Operating

- In an ELK deployment you are looking at multiple tools;
  - Unified management console allows for easier management of the deployment
  - Containerized architecture is run on Docker & will use the Conductor to manage the efficiency and scaling of the nodes used.
  - Life Cycle Management was introduced to allow easier migration of older data to lower cost hardware

This may require managed multiple tiers of different resources

- Large time commitment to creating in-house knowledge, documentation and training to use and maintain a customized system
- Additional configuration management & monitoring tools are needed.

Support

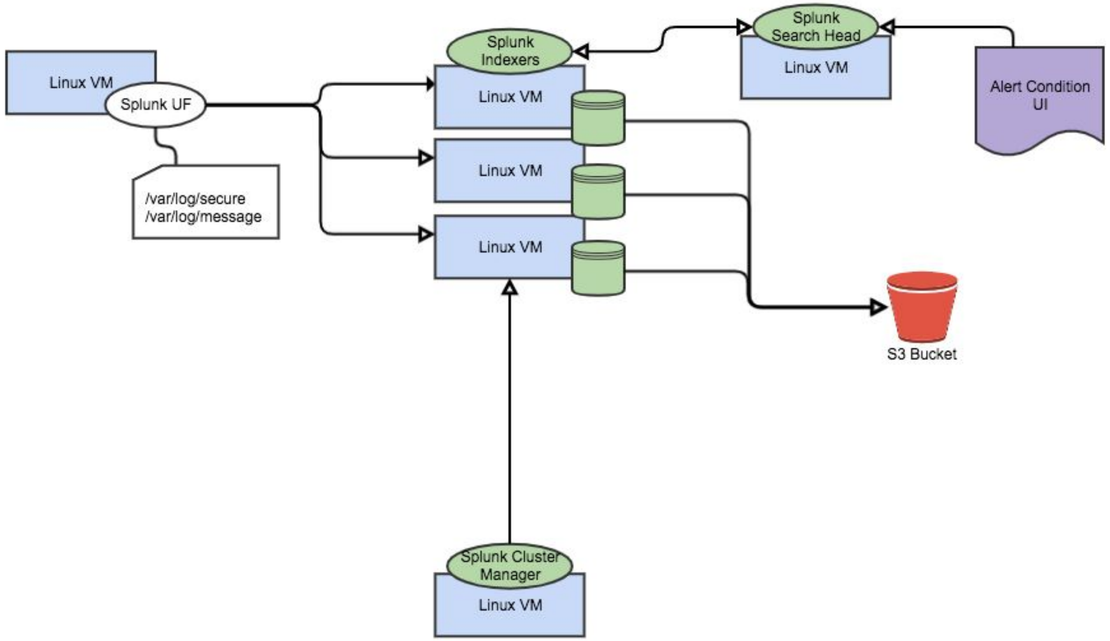
- Vendor support & consulting is available at additional cost

Splunk - SSH Monitoring Use Case

In this Splunk scenario the company *Stark Industries* would need to deploy the following high-level components (* we will assume this is all net-new deployment)

1. Deploy the Universal Splunk forwarder to each of the 5000 hosts
2. Build out an inputs configuration to capture SSH related data
3. Build an outputs configuration to send data to Splunk indexers
4. Provision a cluster of Splunk indexers & object storage for SmartStore
5. Provision a cluster manager to manage indices on Splunk indexers
6. Stand up a dedicated Splunk search head & peer to the provisioned Splunk indexers
7. Verify the data & setup the alert with the Splunk UI

Splunk - SSH Monitoring Use Case



1. Inputs.conf– stores the client’s configuration of what files to monitor and metadata (index, sourcetype, etc.)
2. Server.conf – stores the cluster manager specifications for replication & search factor
3. Outputs.conf- stores the clients configuration of where to send the monitored data
4. SavedSearch.conf – where the alert configuration is actively stored. However the alert itself is defined through the UI
5. Indexes.conf – this is where the index/object-store configuration is defined
6. Using the Splunk UI alerts can be easily generated from any search

Splunk - SSH Monitoring Use Case: Additional Considerations

Scaling

- Splunk will also scale horizontally
  - Indexing & Search layers will scale independently based on search need and indexing need
- As throughput of the deployment goes up:
  - Additional indexers will be needed to accommodate more ingest but with the compression ratio of 50% you can keep more log data in a smaller footprint.
  - Using newer NVME or Local SSD based storage + SmartStore we can increase the efficiency of Splunk nodes to a greater scaling factor – driving down instance sprawl and costs.
- Deployments have been able to scale to several PBs per day
- Splunk has a built-in queueing mechanism that allows for more flexibility over peak ingest times.
- Splunk integration with S3 objects stores can assist in reducing TCO for storage & assist in retention mandates (SmartStore)
  - All data is searched using the same resources in a SmartStore configuration as all data is only searched from the local cache.
  - Buckets are made smaller in a SmartStore configuration to reduce transfer time from the S3 object-store

Operating

- Splunk is a single eco-system that has a consistent framework throughout helping make overall administration more manageable by fewer staff
- Splunk has a built-in configuration management system (Deployment Server & Deployers) to help with consistent configuration, but also easily integrates with Ansible, Chef, Jenkins and other frameworks.

Support

- Professional Training & Certification paths available
- Community support through Splunk Answers & SplunkTrust MVP Program available.
- Vendor support & consulting is available at with an Enterprise License purchase

Q&A

References

https://www.elastic.co/guide/en/logstash/current/deploying-and-scaling.html

https://thoughts.t37.net/designing-the-perfect-elasticsearch-cluster-the-almost-definitive-guide-e614eabc1a87

https://calculator.s3.amazonaws.com/index.html

https://www.elastic.co/guide/en/logstash/current/tuning-logstash.html

http://splunk-sizing.appspot.com/

https://logz.io/blog/elastic-stack-6-new/

https://www.elastic.co/guide/en/elasticsearch/hadoop/current/mapreduce.html

https://www.elastic.co/blog/how-to-enable-saml-authentication-in-kibana-and-elasticsearch

https://www.datadoghq.com/blog/elasticsearch-performance-scaling-problems/

https://www.elastic.co/guide/en/x-pack/current/xpack-introduction.html

https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/petabyte-scale.html

https://www.elastic.co/guide/en/elasticsearch/reference/current/index-lifecycle-management.html

https://www.elastic.co/guide/en/cloud-enterprise/current/index.html

https://www.elastic.co/guide/en/cloud-enterprise/current/ece-architecture.html#ece-overview-admin

.conf19

splunk>

Thank You!

Go to the .conf19 mobile app to

RATE THIS SESSION