Go Big or Go Home : Primer on large scale deployments

Sean Delaney – Principal Architect
Mustafa Ahamed – Principal Architect

October 2018 | Version 1.0

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

Agenda

- Basic Guidelines of large scale deployment
- Sample real world deployments
- Key Learnings from large deployments
- Tips and Tricks
- Q & A

About Us

Sean Delaney

- Principal Field Architect
- 7+ years at Splunk
- Large scale deployments
- 9th .conf

About Us

Mustafa Ahamed

- Principal Architect – looking after large scale customers in APAC
- Led Splunk Enterprise Product Management team for 6 years
- Passionate about finding simple solutions to complex problems

Basic Guidelines of Large Deployments

How Good Architecture helps in easier deployments

1. Understand the use cases and the end goals
2. Consider the management aspects of deployments early
3. Embrace relevant product features as soon as you can
4. Spread the search workloads to all indexers evenly

splunk> .conf18

Sample Real World Deployments

Deployment A

- 900 TB ingestion per day
- 1050 indexers
  - 12 core to 24 core CPU, 40 to 60 GB of RAM
- Local Disk
  - No SSD; Hot / warm / cold on same disk
- Total searches executed per day is 25,000

Key Takeaways

- 1 TB / day ingestion per indexer
- Index-heavy, Low-search environment

Deployment B

- 1.5 PETA Byte ingestion per day
- 3540 indexers
  - Xeon E5-2650, 128 GB of RAM
- Local Disk
  - No SSD; Hot / warm / cold on same disk ; RAID 10
- Total searches executed per day is 429,000

Key Takeaways

- Balance of data ingest to search load

Deployment C

- 1.65 PETA Byte ingestion per day
- 2950 indexers
  - Xeon E5-2650, 128 GB of RAM
- Local Disk
  - No SSD; Hot / warm / cold on same disk ; RAID 10
- Total searches executed per day is 325,000

Key Takeaways

- Heavy Search environment
- Lower ingest per Indexer

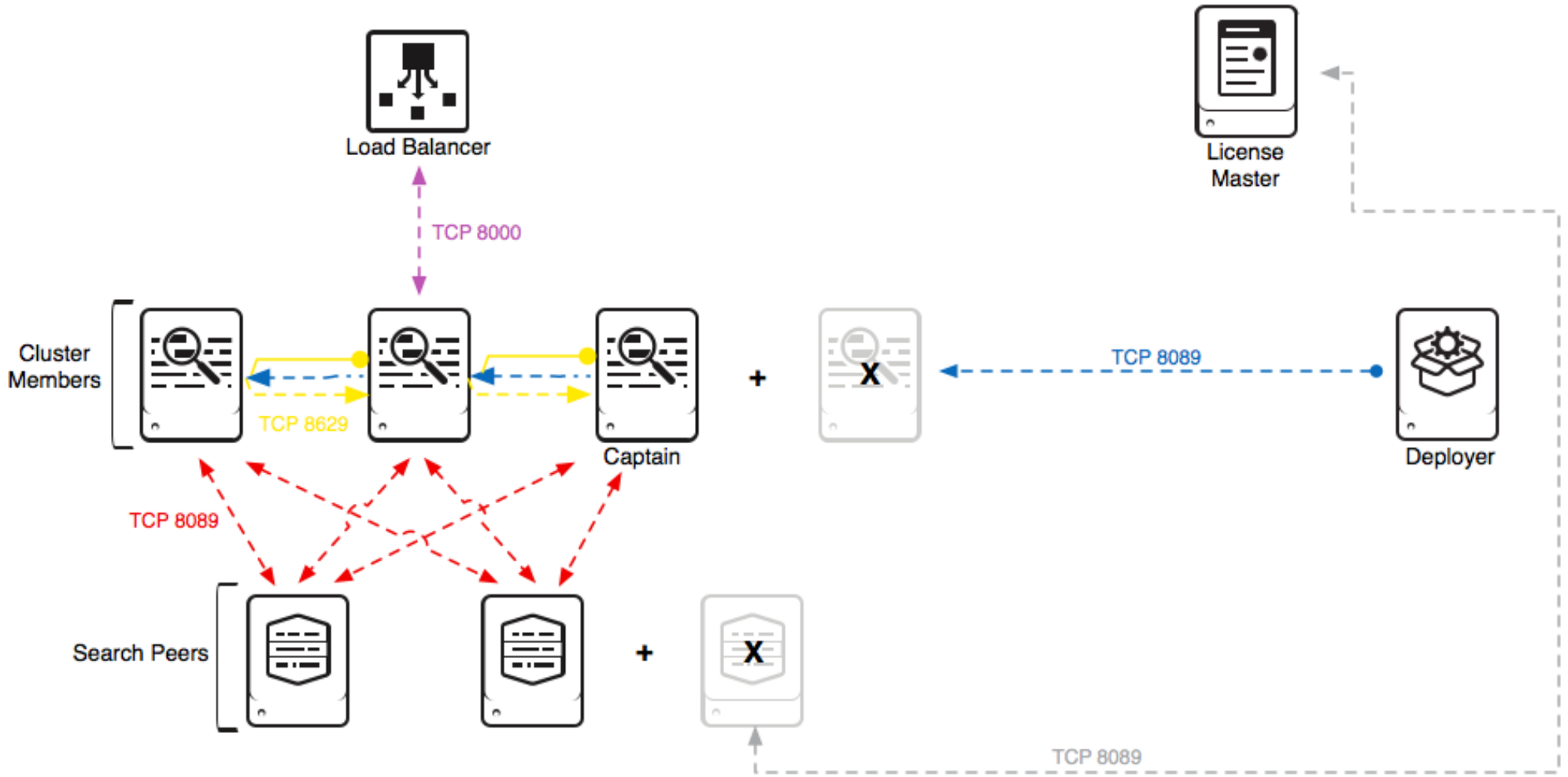
Common Deployment Characteristics

What do these large deployments have in common

- High ingest volumes per indexer (400Gb-1Tb/day)
  - Multiple indexer pipelines
  - Indexer resources are balanced between ingest and search
  - Single data volume (no separate Hot and Cold volumes)
  - No indexer clustering overhead
- Limited number of well known sourcetypes
  - Correct event parsing rules (date/time, event boundaries)
- Managed search workloads
  - Monitoring the efficiency of user searches
  - Scheduled and reporting search reviews (scheduled search efficiency and bloat)

Deployment Architecture

Simple and Scalable



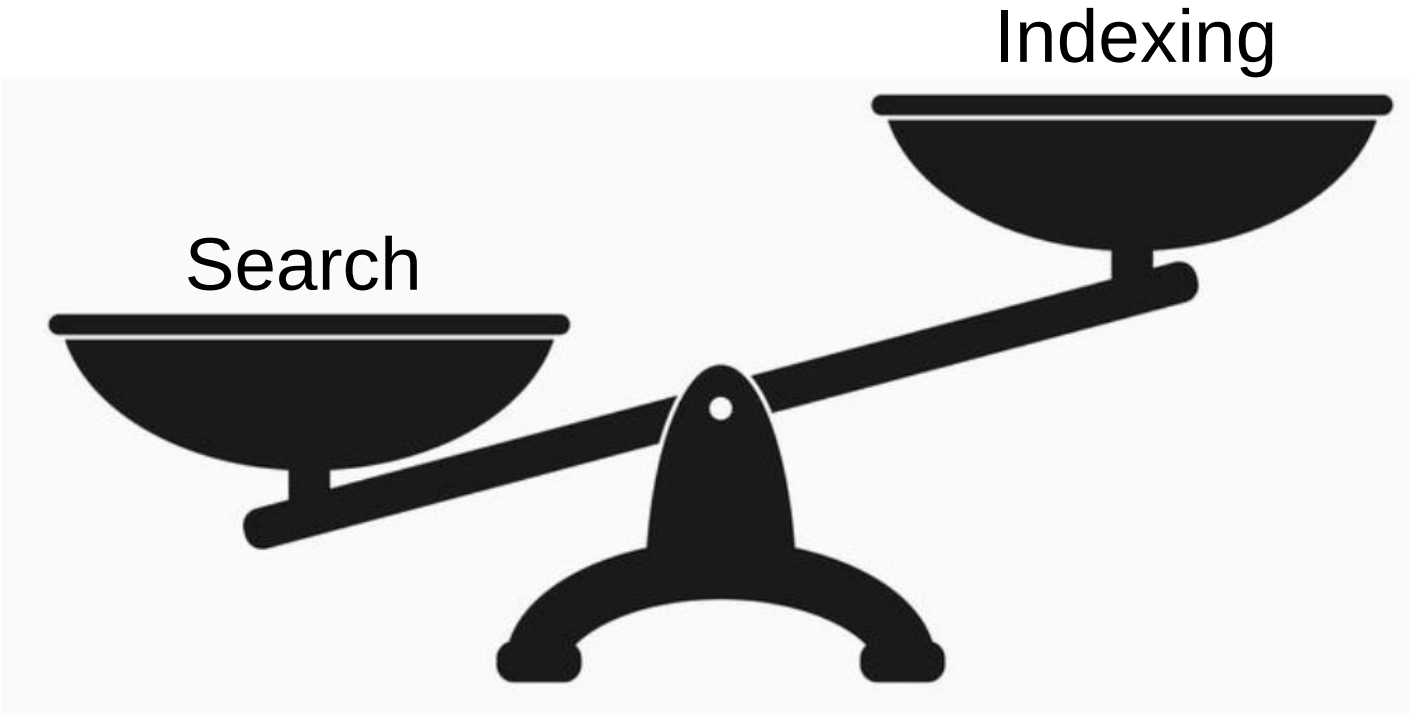
Scaling and Management

splunk> .conf18

Balancing Indexer Resources

IOPS

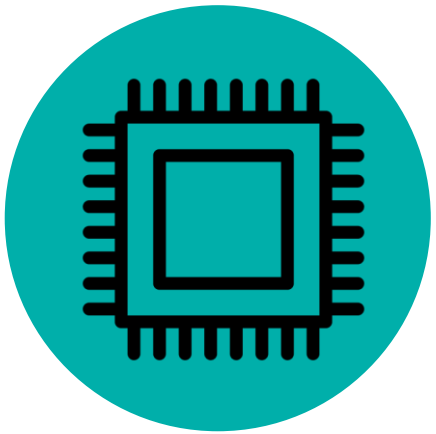
- Splunk is I/O intensive
- I/O bottlenecks on Indexers are the most common performance issues across Splunk deployments
- Slow storage performance can cause Indexing ingest latency
- Performance of Search processes on the Searchpeers (Indexers) is directly related to accessing data from stored buckets
- Indexing and Search processes can be in contention for I/O resources



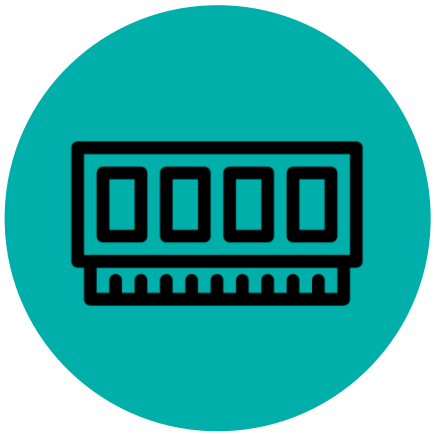
Balancing Indexer Resources

This is where the subtitle goes

IOPS



CPU



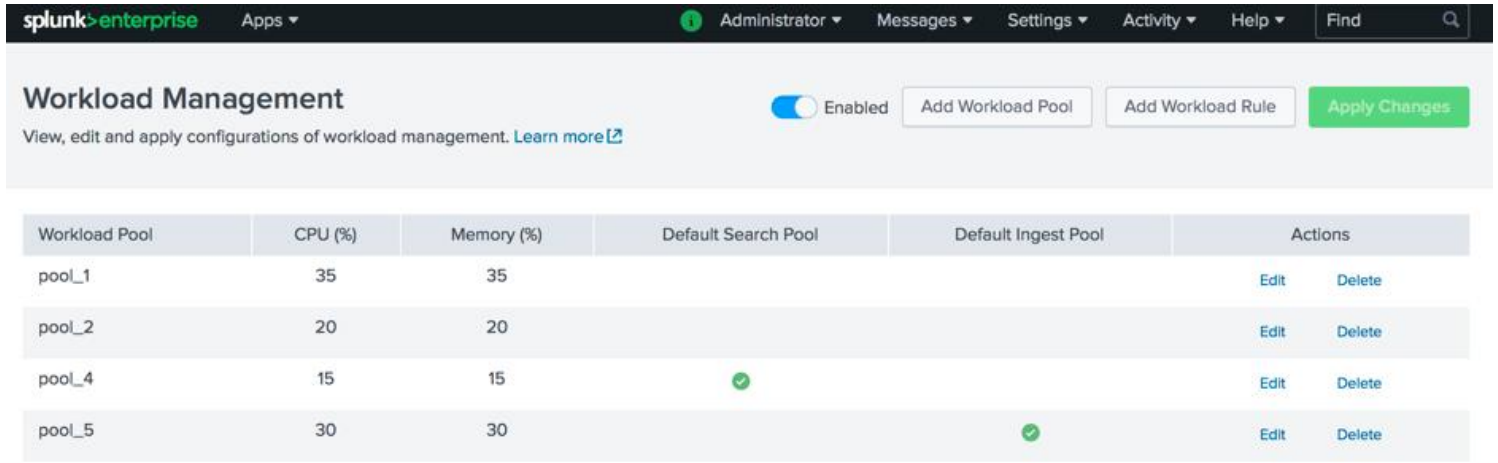
RAM

- SSD/Flash storage can provide high I/O performance
- Additional RAM on Linux Servers is utilized by the kernel as disk cache, reducing the number of read requests to the storage subsystem for recently accessed data – significantly improving I/O throughput

Workload Management

New in Splunk 7.2

- Provide ability to prioritize resource allocation to critical workloads
- Guaranteed resource allocation for ingestion to avoid data lags
- Pre-allocate CPU and Memory resources

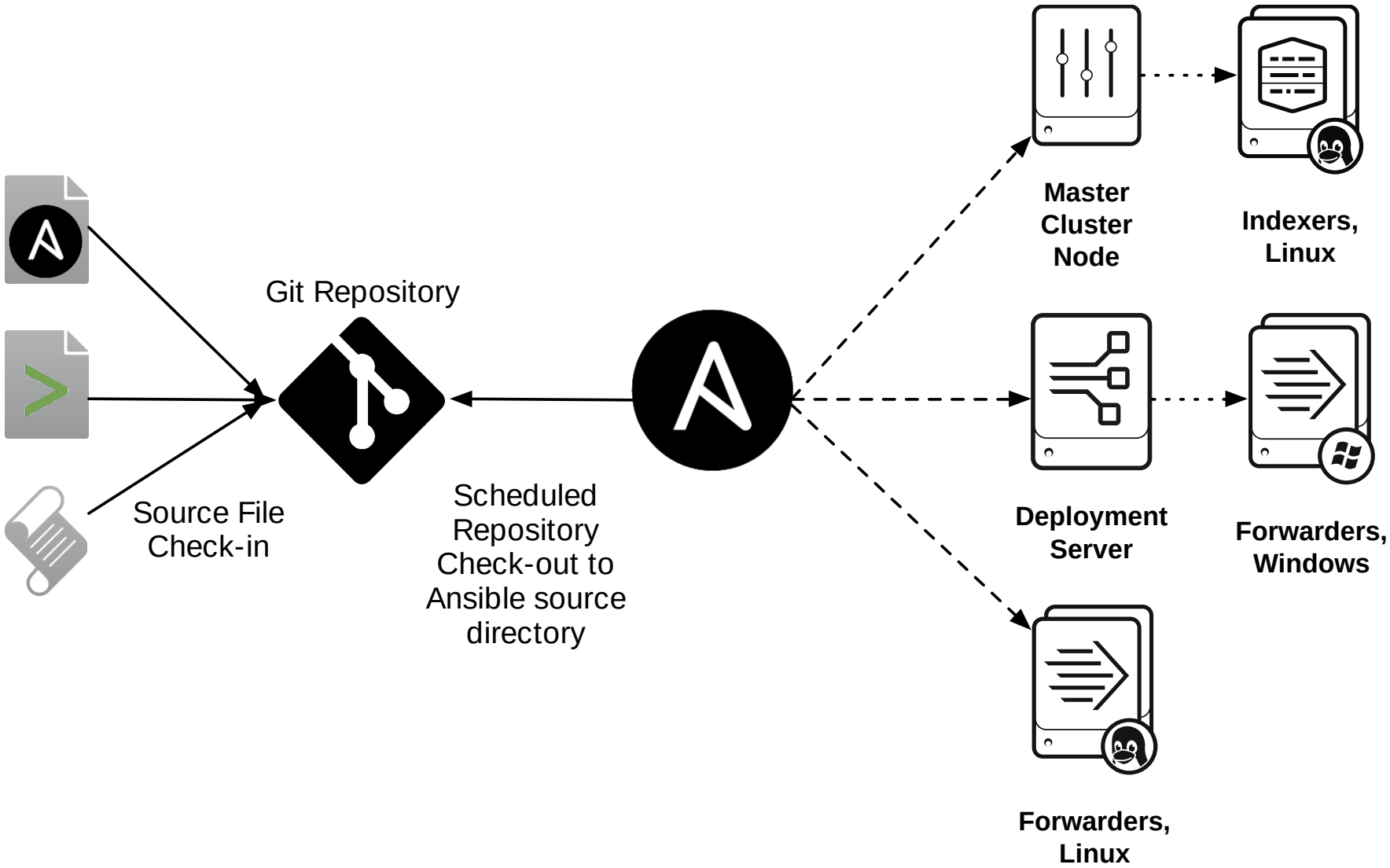


Deployment Automation

This is where the subtitle goes

- Deployment at scale -> orchestration
- Managing hundreds or thousands of Splunk servers
  - Managing underlying OS, system patches, system accounts, ulimit settings
  - Deployment and Upgrades of Splunk
  - Base and template configuration files
  - Addition of new servers as your deployment grows to meet demand
  - Management of forwarders (package and configurations)

Configuration Deployment



Tips and Tricks – Forwarder Tier

splunk> .conf18

Importance of Balanced Data Spread

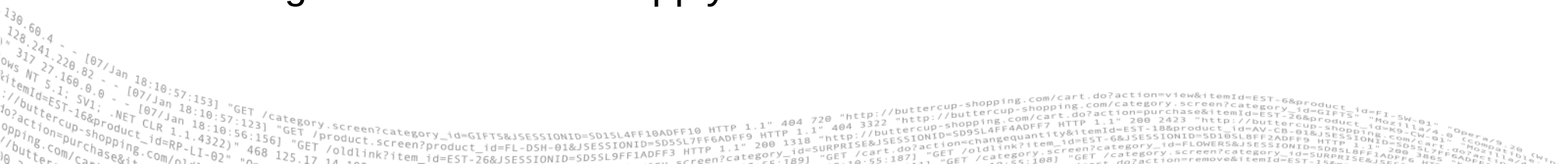
- Overloading of any specific indexers is bad
- Well balanced data spread is fundamental to scaling an environment
- Sticky forwarders and time based load balancing often lead to issues
- EVENT_BREAKER to the rescue !

UF Event Breaking

- Added in Splunk 6.5
  - Only available on the Universal Forwarder (UF)
- What does it do?
  - Provides lightweight event breaking on the UF
  - AutoLB processor now sees event boundaries
    - Prevents locking onto an Indexer
- How does it work?
  - Props.conf on UF
  - Event breaking happens for specified Sourcetypes
  - Sourcetypes without an event breaker are not processed
    - Regular AutoLB rules apply

props.conf

```
[sourcetype]
EVENT_BREAKER_ENABLE = True
EVENT_BREAKER = <regEx>
```



splunk> .conf18

UF autoLB switching

- New UF setting in 7.1
- UF can now switch Indexers based on volume sent to Indexers

outputs.conf

```
[tcpout:]
autoLBVolume=1048576
autoLBFrequency=10
```

- If the forwarder has sent more than autoLBVolume bytes of data to an indexer, it changes indexers regardless of whether or not autoLBFrequency have passed since the last change to a receiving indexer.
- If the forwarder has not sent more than autoLBVolume bytes of data before autoLBFrequency seconds have elapsed, then it changes indexers after that time has passed.

Tips and Tricks – Indexer Tier

splunk> .conf18

Indexer Clustering

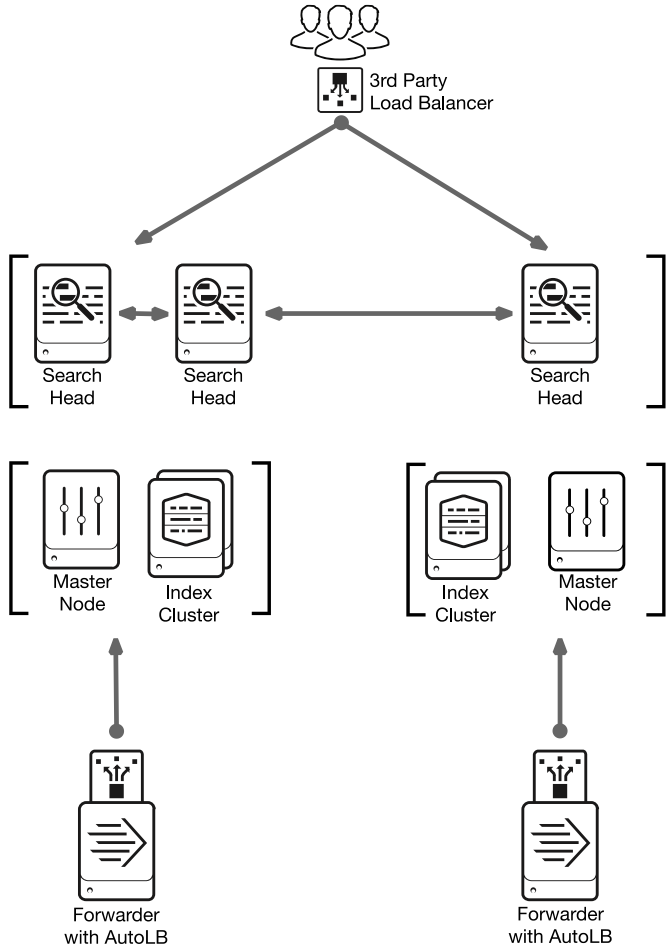
- Clustering Benefits
  - Clustering offers Data Redundancy
  - Foundation for Search High Availability
  - Cluster Bundle deployment to maintain configurations across Indexers
  - Centralized Management of Indexers for cluster restarts
- Clustering Costs
  - Requires more disk to provide data duplication
  - Increases network traffic for streaming replication of events and bucket fixup tasks
  - Additional administration tasks

Indexer Clustering

- Large Cluster Deployment Recommendations (per cluster)
  - 100 TB/day
  - 400 indexers
  - Multi site deployments*
- Bucket count matters
  - 9 million unique buckets
  - 27 million total buckets

Indexer Clustering

- What if I need Indexer Clustering for 100Tb+
  - Deploy multiple Indexer Clusters
    - Distributed Search from SHC can Search over Multiple Clusters
    - Reduces the total number of buckets the Cluster Master needs to manage
    - Inter-cluster server communications are reduced



Large Scale Clustering Tuning Tips

Server.conf	
Cxn_timeout Rcv_timeout Send_timeout	Specifies how long before an intra-cluster connection will terminate (Default 60s) For large scale cluster, increase this setting to **300s**
Indexes.conf	
rotatePeriodInSecs	Specifies how often to check through buckets (Default 60s) For large scale cluster increase this setting to **300s**

splunk> .conf18

Other Benefits of Clustering

- Turn on clustering even if you don’t need to replicate data
  - Set RF / SF = 1
- It will make indexer management lot easier
- Ability to add / remove indexers without the need to restart search heads
- BONUS : Use Indexer Discovery
  - This will make managing forwarders and failover to new indexers lot easier

splunk> .conf18

Tips and Tricks – Search Tier

splunk> .conf18

A bad indexer in the mix (1 of 2)

- Scenario
  - One bad indexer in a group which consistently returns search results slow and causes bad user experience
- Solution
  - Turn on slow peer disconnect feature !
  - This will make search results to be available to users as soon as other peers return values
  - Trade off between data fidelity vs search performance

http://docs.splunk.com/Documentation/Splunk/latest/DistSearch/Slowpeerdisconnect

A bad indexer in the mix (2 of 2)

- Scenario
  - One bad indexer in a group is experiencing problem due to faulty disks or network cards
- Solution
  - Turn on quarantine an indexer feature!
  - This will make sure that further searches won't be dispatched to this indexer
  - Very useful during upgrading an indexer to a new OS / replacing disks

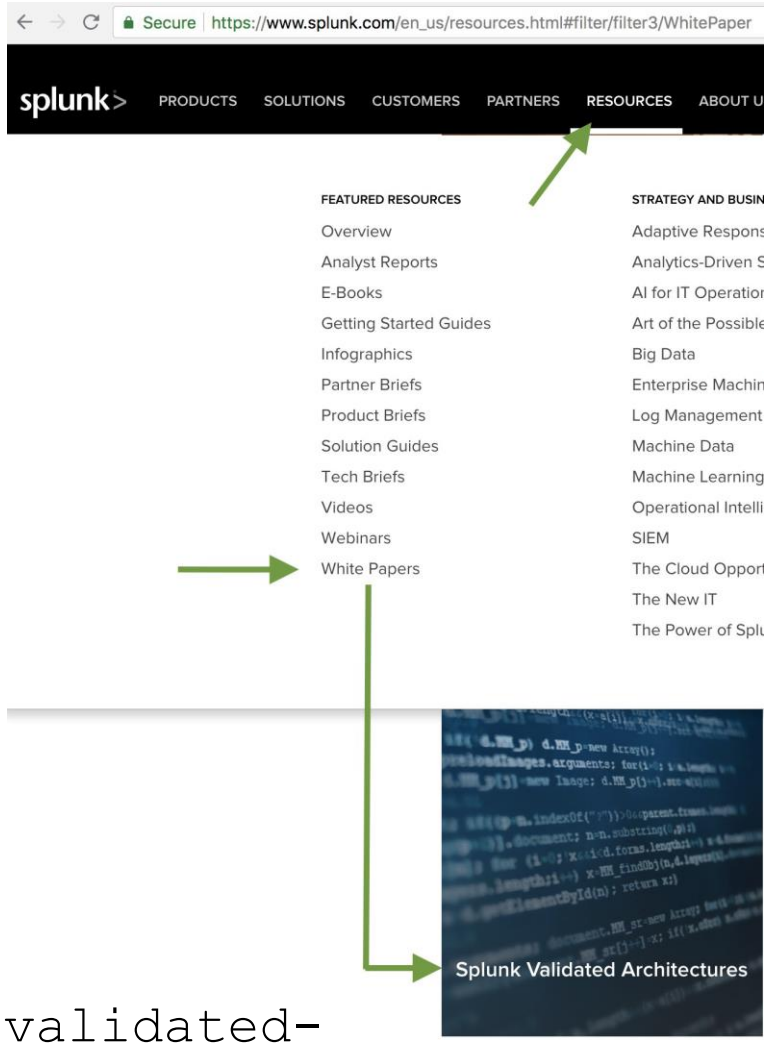
```
http://docs.splunk.com/Documentation/Splunk/latest
/DistSearch/Quarantineasearchpeer
```

SVA – Splunk Validated Architectures

splunk> .conf18

SVAs – Splunk Validated Architectures

- Recommended and supported Splunk deployment topologies based upon the following design pillars:
  - Availability
  - Performance
  - Scalability
  - Security
  - Manageability



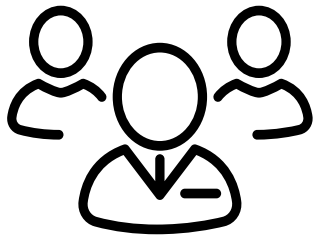
https://www.splunk.com/pdfs/white-papers/splunk-validated-architectures.pdf

Core Sizing Calculator

splunk> .conf18

Introduction to Sizing

What's sizing?
Answer: find out the appropriate Splunk cluster size that can handle customers' demand.



Customer

We have X GB data to input
We have Y searches to run

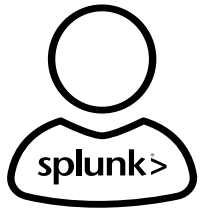
We want to offer best performance to users

? Splunk Search Head

? Splunk Indexer

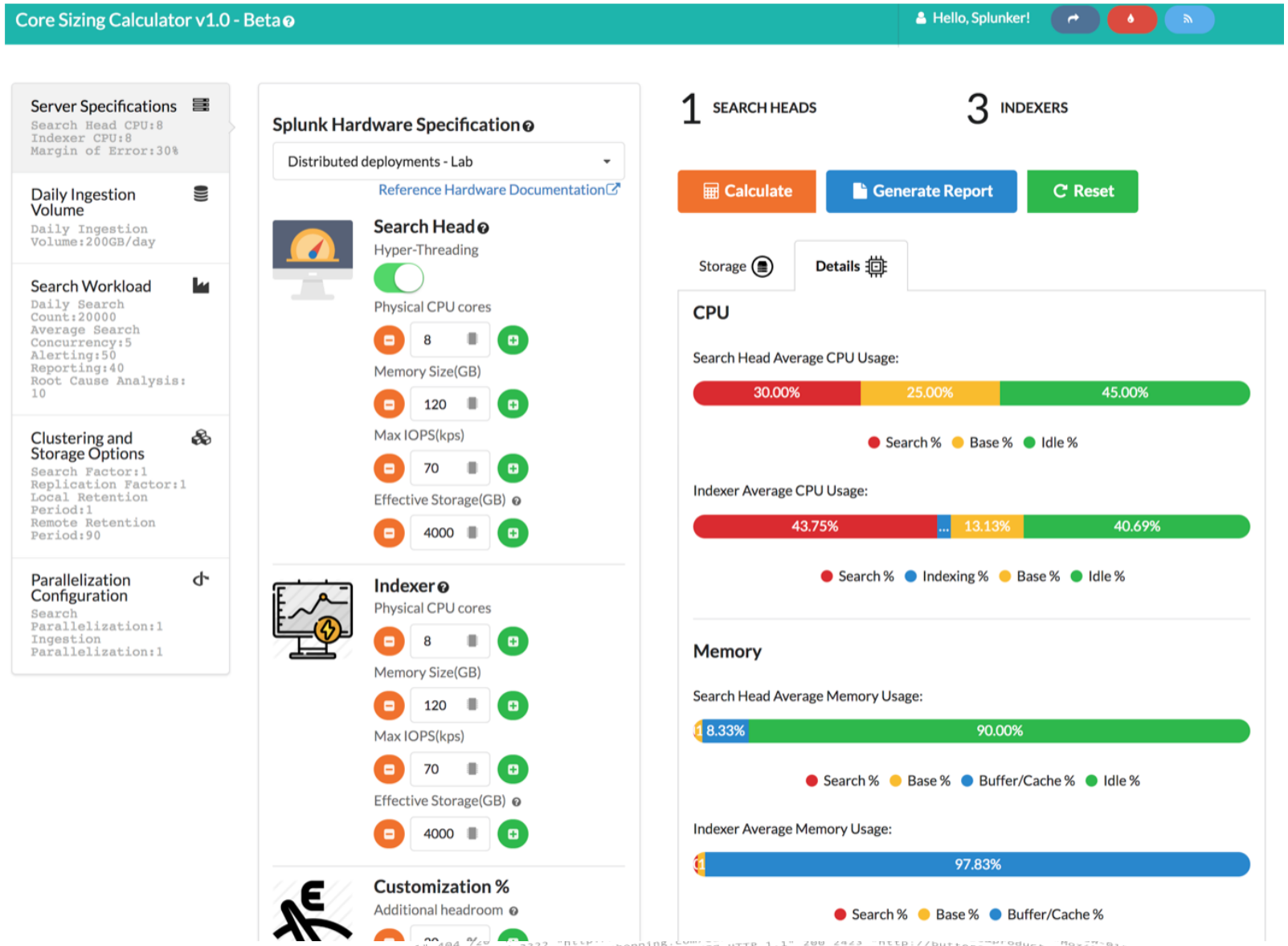
M Splunk Search Head

N Splunk Indexer



splunk> .conf18

Core Sizing Calculator



splunk> .conf18

Final Takeaways

If you forget everything, just remember these 3 points

- Indexing and Searching are two sides of the same coin
  - Overutilizing one would certainly affect other
- Spread your search workloads over different time periods
  - Use various scheduler enhancements
- Replicate only the relevant indexes
  - Be selective instead of replicating all indexes

Q & A

splunk> .conf18

Thank You

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