Copyright © 2015 Splunk Inc.

.conf2015

Architecting and Sizing your Splunk Deployment

Simeon Yep

Global Strategic Alliances, Splunk

Disclaimer

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 the 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.

Objective

Show you how to build a robust and scalable Splunk deployment

Introduction

Qualifications

- 7+ years @ Splunk
- Experience:
  - Building and running large scale Splunk deployments
  - Technical sales – OEMs, Strategic Accounts, MSPs
  - Current – Anything technical for Partnerships
- Based in San Francisco

Agenda

- Sizing fundamentals
- Architecting fundamentals
- Deployment topologies

Sizing Fundamentals

Sizing Fundamentals

- Understand the Sizing Factors
- Data Volume
- Search Volume

Sizing Factors

- How much data (raw sizes)?
  - Daily Volume
  - Peak Volume
  - Retained Volume (archive size)
  - Future Volume?
- How much searching?
  - Use Cases
  - How many people? How often?
  - Apps
- Background searches
  - Acceleration, Summarization, Alerting, Reporting, Data Models

Data Volumes

- Estimate Input Volume
  - Verify raw log sizes
  - Leverage _internal metrics and default views (license_usage.xml)
- Confirm estimates with actual data
  - Create a baseline with real or simulated data
  - Find Compression rates (range from 30%-120%, typically 50%)
  - Determine Retention needs
  - Clustering needs (SF vs RF)
- Document Use Cases
  - Use case determines search needs
  - Plan for expansion as adoption grows (Search and Volume)

Data Sizing Exercise

- Via Filesystem
  - Use a large enough data set. 100GB+
- Use the Splunk log files
  - metrics.log
  - license_usage.log
  - disk_objects.log
- Recommended:
  - Distributed Management Console

Search Volumes

- Gather Use Case information
  - How much Ad-Hoc searching?
  - How much background searching?
- Ad-Hoc searching
  - Evaluate the data being searched
  - Evaluate the time duration (real-time vs historic)
  - Real-time searches are typically less overhead
- Background Searching
  - Alerting and Monitoring
  - General reports
  - Data Models, Report Acceleration & Summary Indexing

Search Volume Exercise

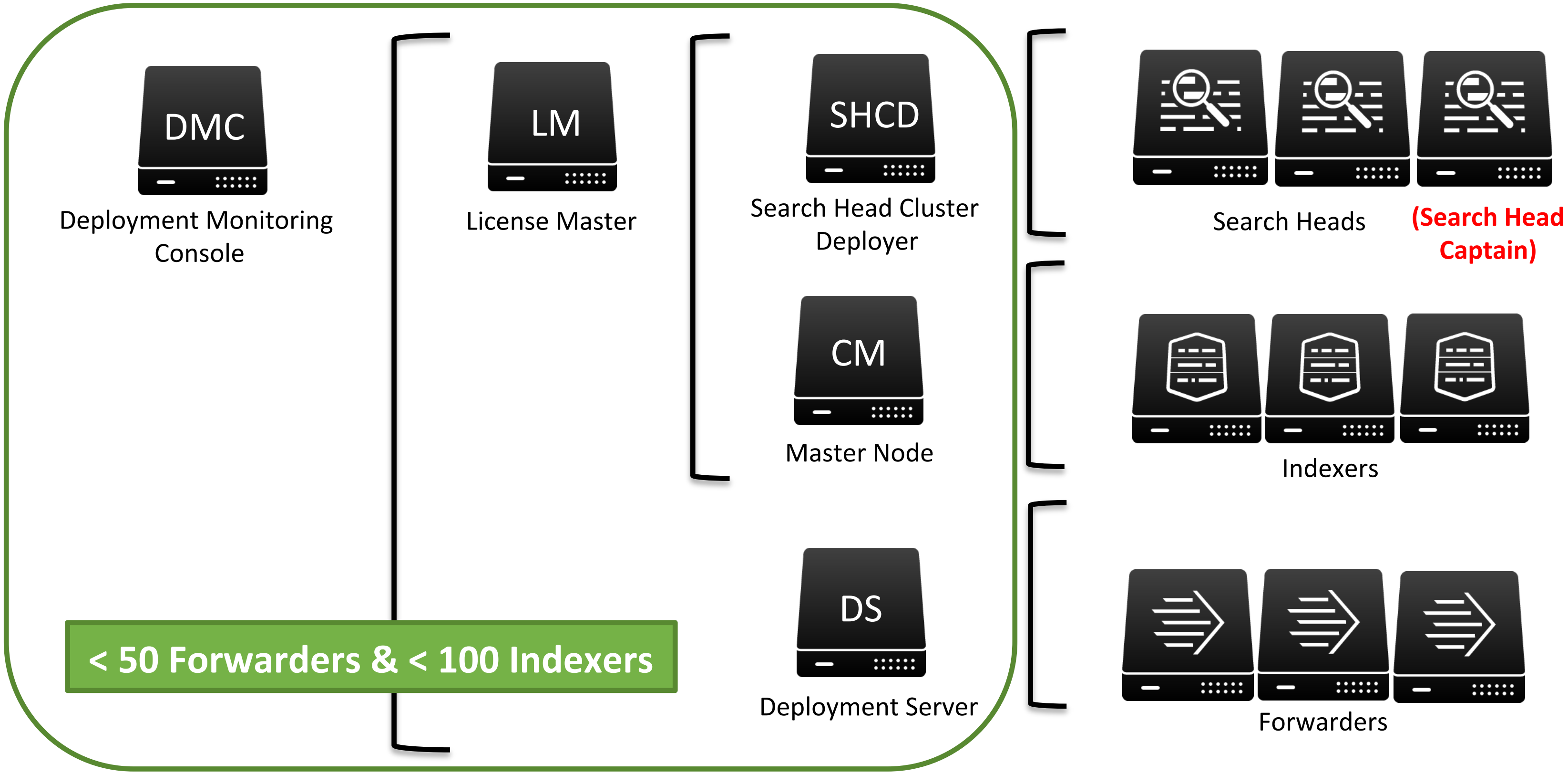
- Use the Splunk log files: audit.log
- Recommended:
  - DMC
    - Search Activity View
  - Introspection data
    - resource_usage.log

Architecture Fundamentals

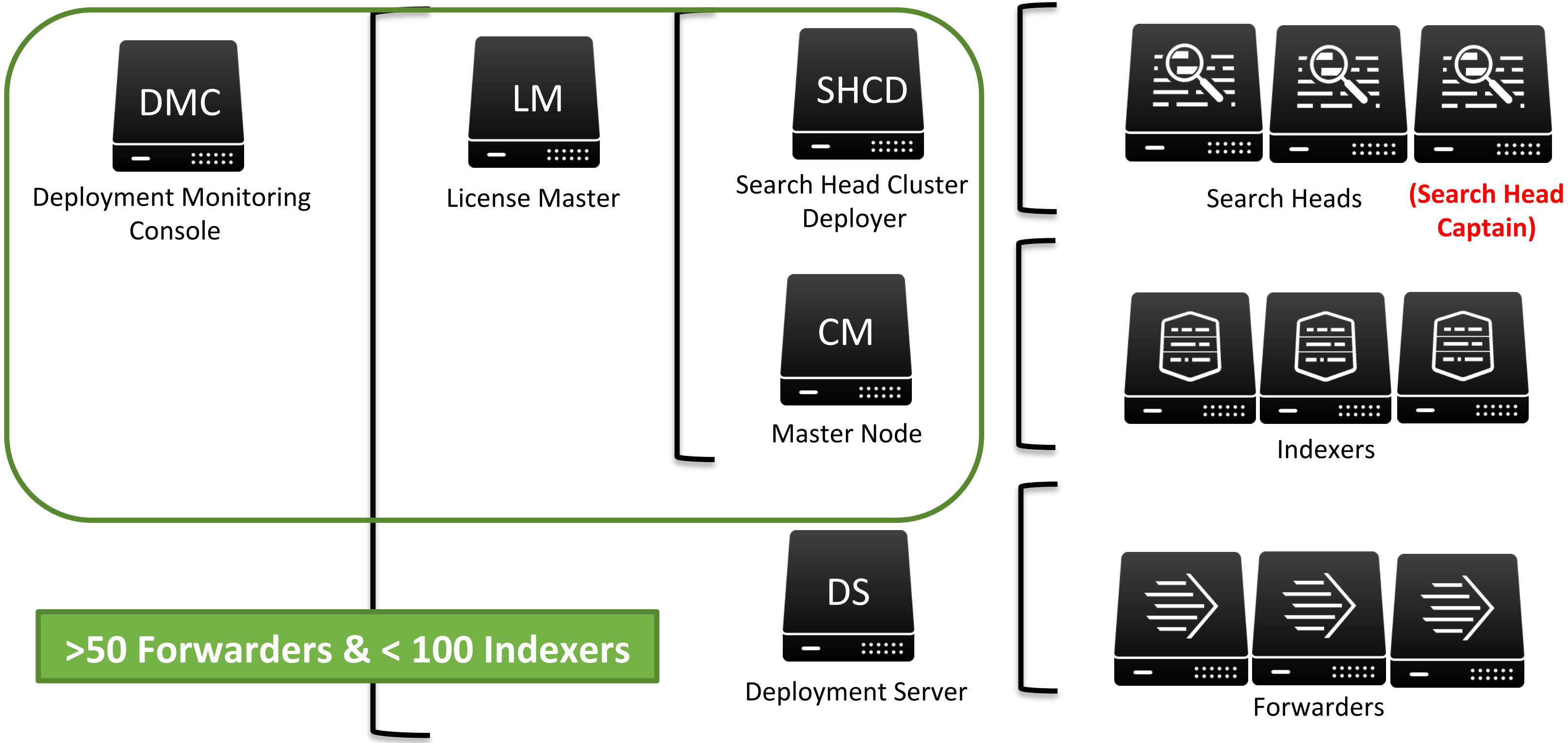
Architecture Fundamentals

- Splunk server roles: Distributed/Clustered Deployments
- Reference Server
- Rules of Thumb
- Hardware Factors

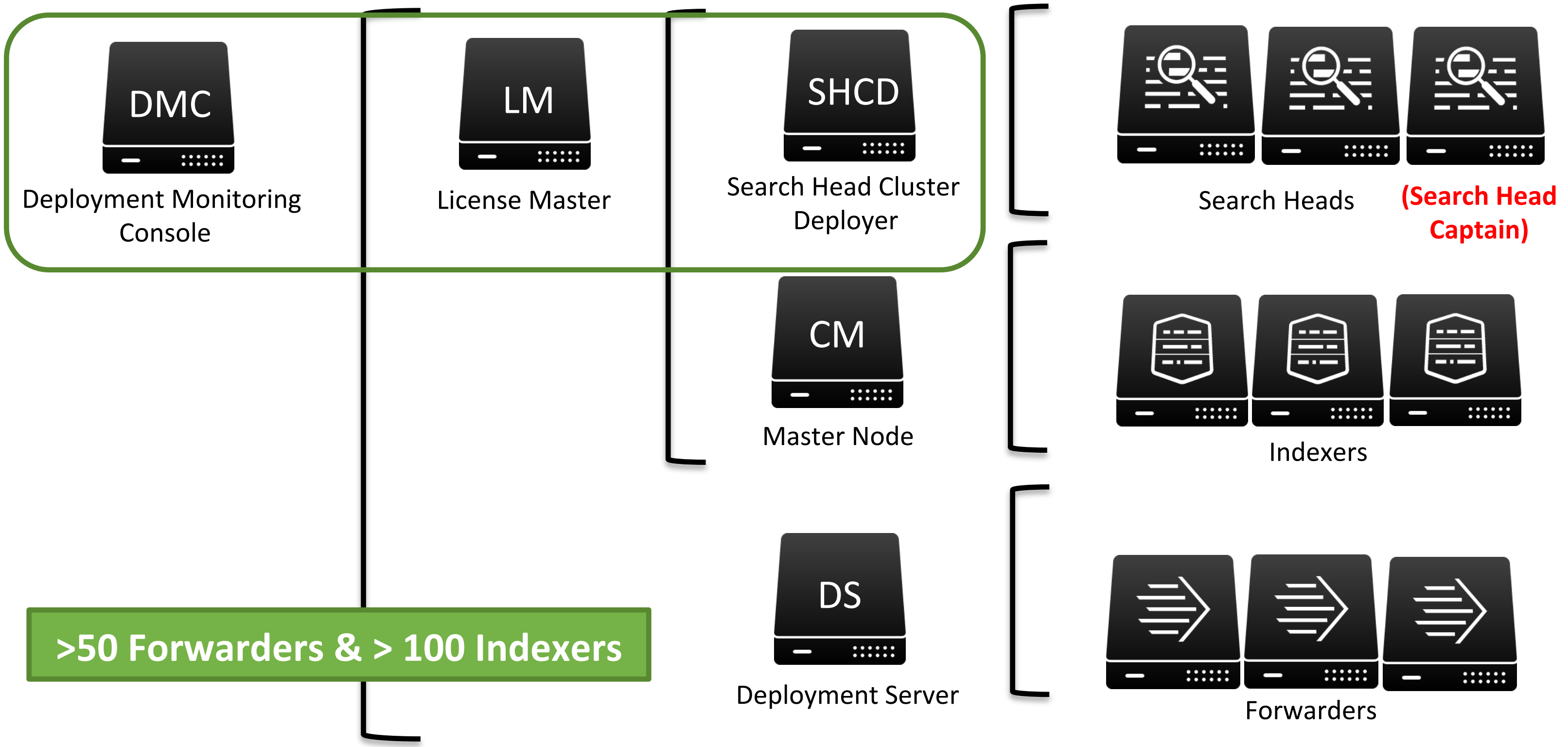
Splunk Distributed Roles



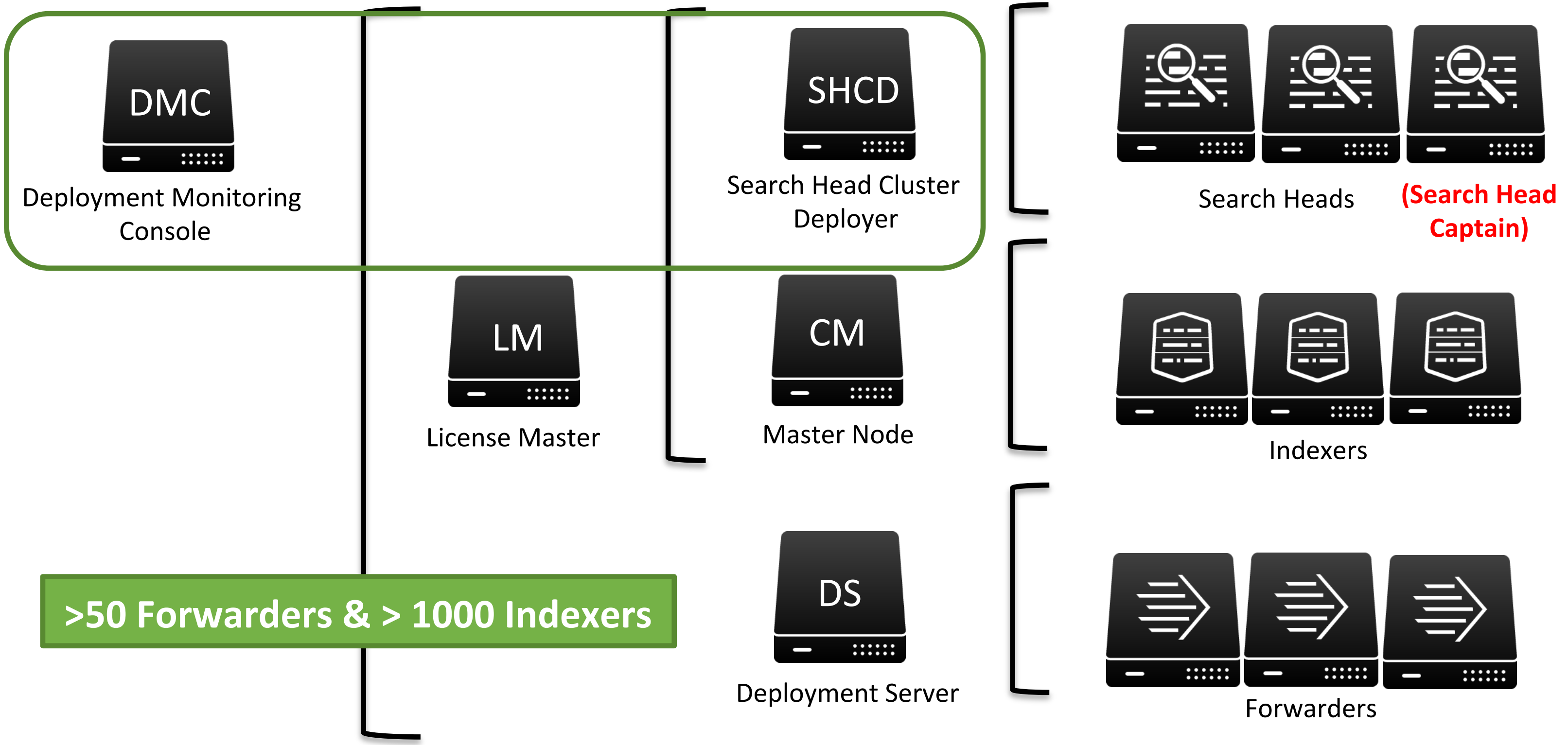
Splunk Distributed Roles



Splunk Distributed Roles



Splunk Distributed Roles



What's a "Search Head Reference" Server?

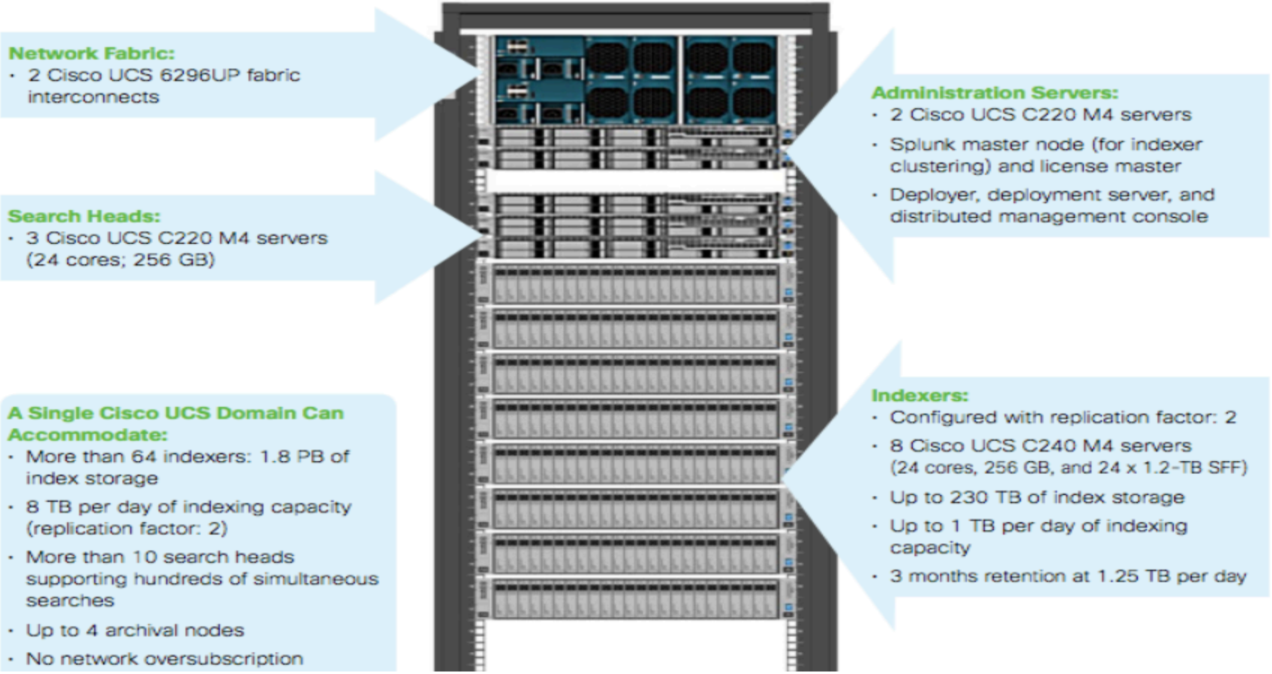
- Sizing based on commodity x86 servers – 64bit
- 4 x quad-core CPUs at 2.0 GHz
- 12 GB of RAM – (16 GB is common)
- 64-bit OS
- 2x10k RPM local SAS drives in RAID 1
- Variations cause corresponding changes in performance/ requirements

What's a "Indexer Reference" Server?

- Sizing based on commodity x86 servers – 64bit
- 2 x six-core CPUs at 2.0 GHz
- 12 GB of RAM – (16 GB is common)
- 64-bit OS
- Local or Attached storage (800+ IOPs)
- Variations cause corresponding changes in performance/ requirements

Real World Examples

- Cisco Unified Computing System (UCS)
  - Search Head:
  - UCS C220 M4
  - 24 cores
  - Indexer:
  - UCS C240 M4
  - 24 cores



Real World Examples

- Amazon Web Services EC2
  - Search Head:
    - c4.4xlarge + EBS storage
    - c4.8xlarge + EBS storage
  - Indexer:
    - c4.4xlarge + EBS storage
    - d2.4xlarge (IR)

Model	vCPU	Mem (GiB)	Storage	Dedicated EBS Throughput (Mbps)
c4.2xlarge	8	15	EBS-Only	1,000
c4.4xlarge	16	30	EBS-Only	2,000
c4.8xlarge	36	60	EBS-Only	4,000

Model	vCPU	Mem (GiB)	Storage (GB)
d2.4xlarge	16	122	12 x 2000 HDD
d2.8xlarge	36	244	24 x 2000 HDD

Rules of Thumb

- These all have exceptions and qualifications
- 1 reference indexer per 300 GB/day
- 1 reference search head per 20-40 queries concurrently
- 1 deployment server per 10k clients @ 10-15 min polling period

How Many Indexers?

- Rule of thumb says: 1 per 300 GB/day
- Leaves room for:
  - Daily peaks
- Need more indexers for:
  - Heavy reporting
  - More users
  - Slower disks, slower CPUs, fewer CPUs

How Many Search Heads?

- Rule of thumb says: 1 per 20 – 40 concurrent queries
- Limit is concurrent queries
- Search Query normally uses up to 1 CPU core
  – 6.3 Parallelization can leverage more
- Don't add search heads; add indexers: indexers do most work
  – Unless you want HA/Search Clustering
- Scale vertically if infrastructure allows it. Add CPU, add memory.

How Many Deployment Servers?

- Rule of thumb says: 1 per 10k clients @ 10 – 15 min polling period
- Adjust polling period to increase total clients supported
- Small deployments can share the same instance as other management instances (LM, CM, etc.)
- Low requirement for disk performance (good candidate for virtualization)
- Or use something other than deployment server
  - puppet, SCCM, cfengine, chef…

More Is Better?

- CPUs
  - 8, 12, 16, 24, 32, etc….
  - Pipelines - New 6.3 feature for parallelization!
  - Indexing can handle higher bursts with multiple index pipeline sets
  - Certain searches can be improved with multiple search pipeline sets
    - Historical batch – return the data without worrying about time order (… | stats count)
  - Indexers still need to do the heavy lifting (search exists on indexer AND search head)

More Is Better?

- Memory
  - Good for search heads and indexers (16+ GB)
    - Benefits from extra RAM used by OS for caching
- Disks
  - Faster is better - 10k – 15k rpm strongly recommended, SSD preferred
  - More disks in RAID 1+0 = Faster
  - RAID 5+1 or 6 can be good for Cold buckets
  - SSDs can also provide benefit for rare term searches and many concurrent jobs

Performance and Sizing Tips

System Change	Search Speed	Search Concurrency	Indexing Speed
Faster Disks	++	+++	++
Add an Indexer	++	+	++
Add a Search Head	+	+	
Report Acceleration/ Summaries	++	+++	

Performance and Sizing Tips

System Change	Search Speed	Search Concurrency	Indexing Speed
Optimize Searches	+++	+	+
Optimize Field Extraction	+		
Optimize Input Parsing			+
Faster CPU	++	+	+

Performance and Sizing Tips

System Change	Search Speed	Search Concurrency	Indexing Speed
Index Pipeline Parallelization			++
Search Pipeline Parallelization	++	+	

Capacity -> Architecture

- Sizing Recipe
  - Capacity
  - Rules of Thumb determines Number of Servers
- Building Blocks for Architecture

Architecture Factors

- What are my sizing requirements?
- Where is the data?
- Where are the users?
- What is the security policy?
- What are the retention and compliance policies?
- What is the availability requirement?
- What about the cloud?

Architecture Factors

- What are my sizing requirements?
  - Data capacity
  - Search capacity
  - User capacity
- Obtained from the sizing process

Architecture Factors

- Where is the data?
  - Local or Remote to the indexing machine
  - If remote – use forwarders when possible
  - Index in local data center (zone) or index centrally
  - Persist Network data to disk as a best practice
  - Use Intermediate Forwarders to distribute data
- Where are the users?
  - User experience affected by Search Head location
    - Time Zone tuning
    - Distributed search over LAN vs WAN

Architecture Factors

- What is the Security Policy?
  - Apply User security policies
    - Auth method
    - Roles
    - Filters
  - Apply physical security policies
    - Index location

Architecture Factors

- Retention, compliance, governance
  - Where is the data allowed to be?
  - Where is the data not allowed to go?
  - Where must the data go?
- Availability
  - Local failover, fault-tolerance, clustering
  - Geographic disaster recovery/fault-tolerance
  - Index replication and Search Head Clustering

Architecture Factors

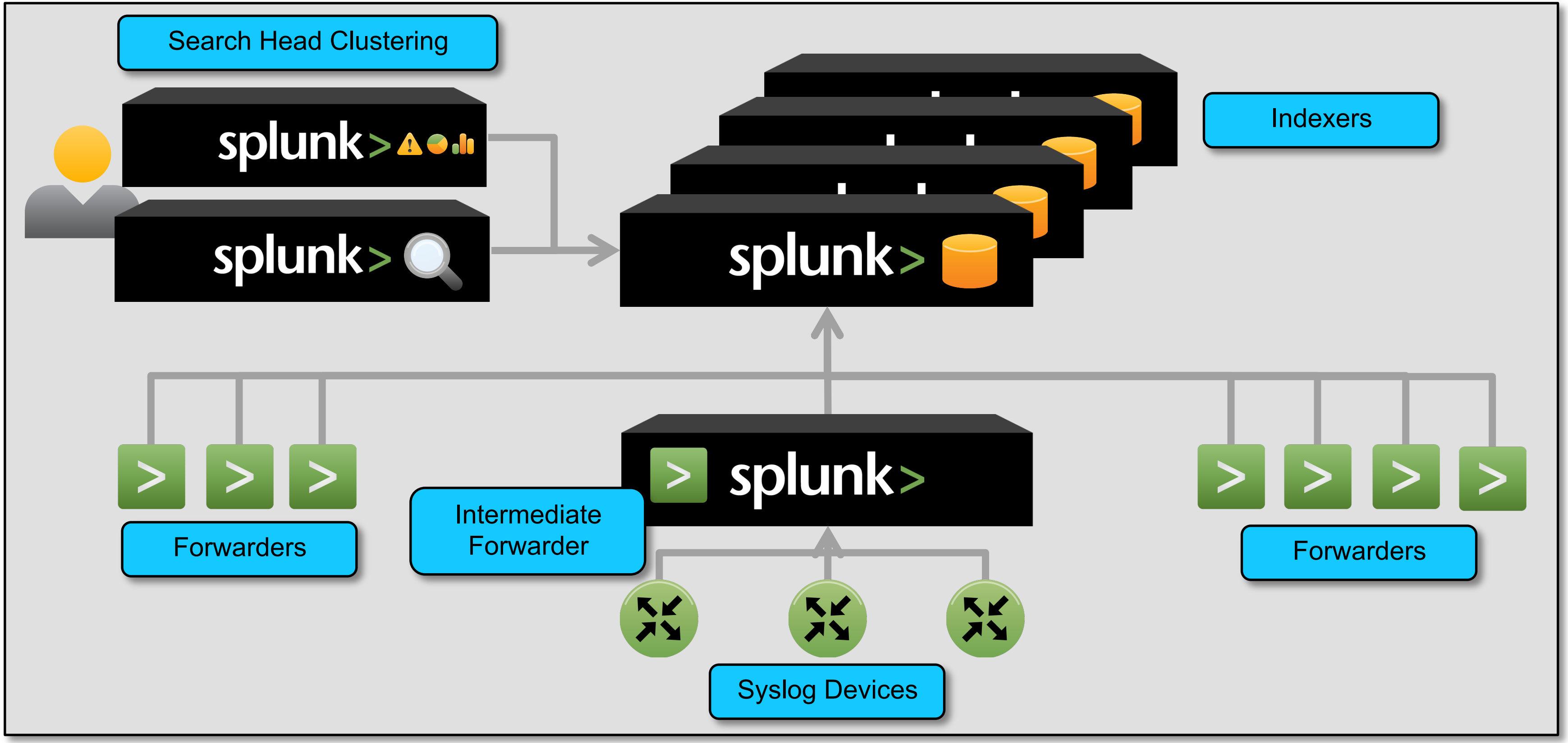
- Cloud Considerations
  - Authentication restrictions
  - Data transfer costs
  - Security – SSL Tunnel
  - Zones
  - Hybrid deployments

Topologies

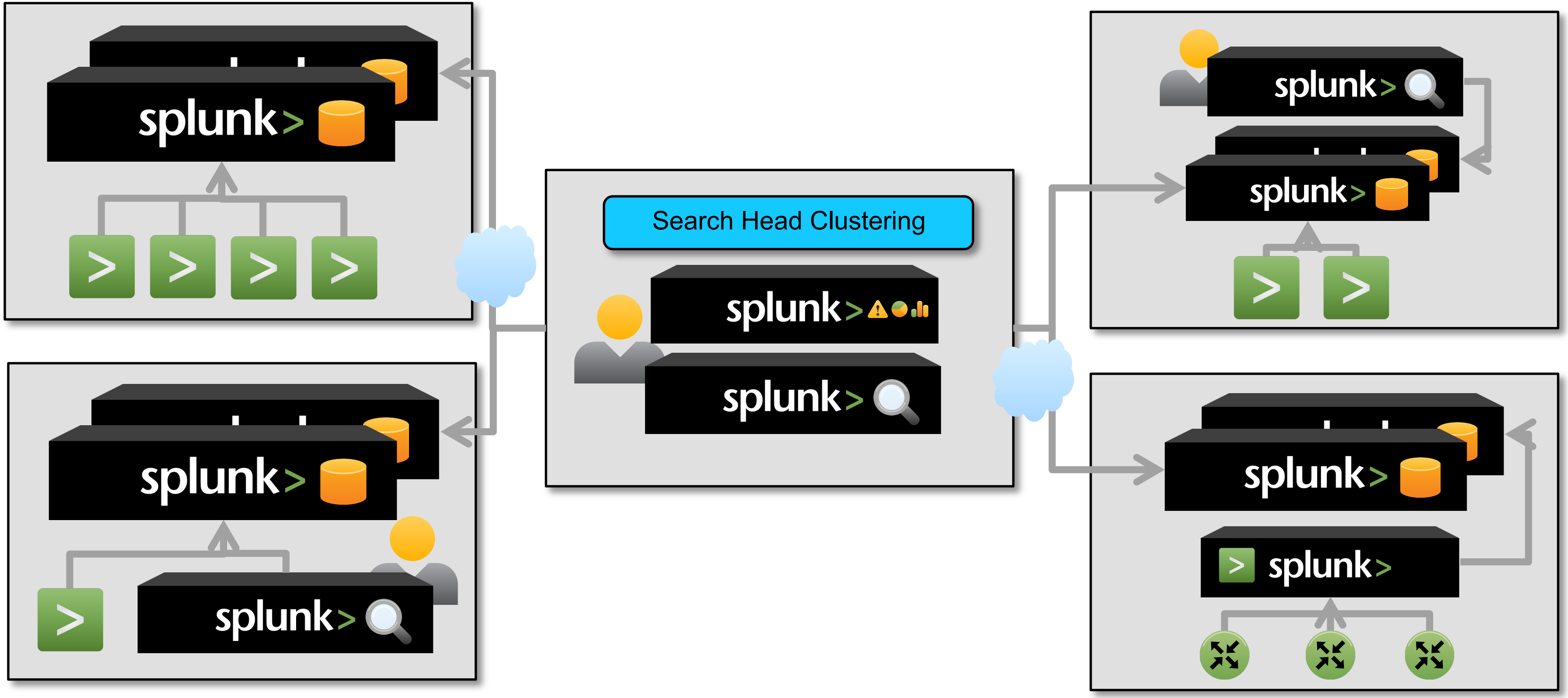
Architecture Factors -> Topology

- Topology Examples
  - Centralized
  - Decentralized
  - Hybrid

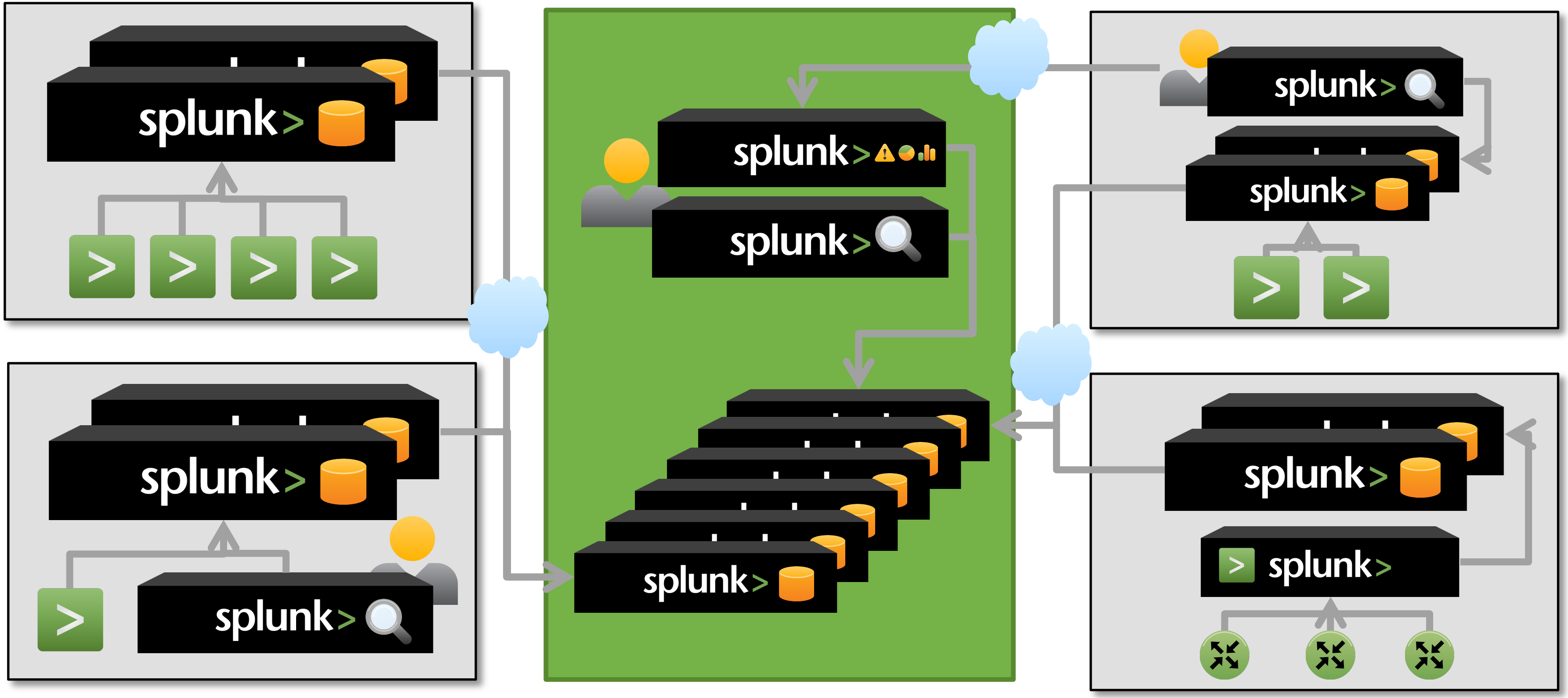
Centralized Topology



Decentralized Topology



Hybrid Topology



Scaling and Expansion

- Add to your indexer pool for more performance or capacity
  - Mixed Platform and Hardware is not recommended
- Use Search Clustering for more UI capacity and availability
  - Does not requires NFS
- Create new indexes based on retention and RBAC
  - Follows best practices
- As data retention needs increase
  - Cannot just add indexers, because we cannot rebalance data.
  - Dynamic storage can help (NAS or Cloud)

Index Replication (aka Index Clustering)

- What is it?
  - Data is replicated to 1 or more indexers based on indexes
  - Splunk Cluster Master controlled
- Basics
  - Master Node (manages indexing and searching location)
  - Horizontal Scaling
- HA vs DR
  - HA - Data is made available on 1 or more indexers in one location
  - DR – Multisite clustering. All data exists in multiple locations

Index Clustering

- Replication factor
  - ✓ Determine the number of rebuildable copies of data to maintain
- Search factor
  - ✓ Determine the number of searchable copies of the data
- Data Retention equation based on syslog data
  - ✓ Total disk usage across cluster in GB = (RepFactor * 0.096 + SearchFactor * 0.201) * DatasetSizeGB

Index Clustering

- Increase in I/O, CPU, and disk requirement
  - Means daily indexing volume per server will be lower
- Search factor increase disk usage by ~30% (rawdata + tsidx)
- Replication factor increases disk usage by ~10% (only rawdata)

Search Head Clustering

- What is it?
  - Group search heads into a cluster as a single entity
  - Provides HA at the Search Head layer
  - Splunk Head Captain controlled
  - RAFT protocol to pick captain
- Basics
  - A captain gets elected dynamically (pre 6.3) or can be defined manually (6.3)
  - Knowledge objects and search artifacts are replicated
  - Search workload distribution
  - Replication using local storage NOT over NFS

Final Thoughts

- Sizing is search load and data volume
- Centralized architecture is the baseline
- Variations on architecture are driven by
  - Sizing
  - Data location
  - User location
  - Retention/Access/Governance
  - Availability requirements

Acknowledgements

- Amrit Bath
- Mustafa Ahamed
- Deep Bains
- Octavio Di Sciullo
- Sunny Choi
- Dritan Bitincka

.conf2015

THANK YOU

splunk>