splunk> .conf2017

Search Head Clustering Basics To Best Practices

Bharath Aleti | Product Manager, Splunk
Manu Jose | Sr. Software Engineer, Splunk

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

Agenda

- What is Search Head Clustering?
- Clustering Internals
- Distributed Scheduling
- Configuration Management
- Bundle Replication
- What's New in SHC

Search Head Clustering Overview

What is Search Head Clustering?

splunk> .conf2017

Search Head Clustering

Ability to group search heads into a cluster in order to provide Highly Available and Scalable search services

MISSION CRITICAL ENTERPRISE

splunk> .conf2017

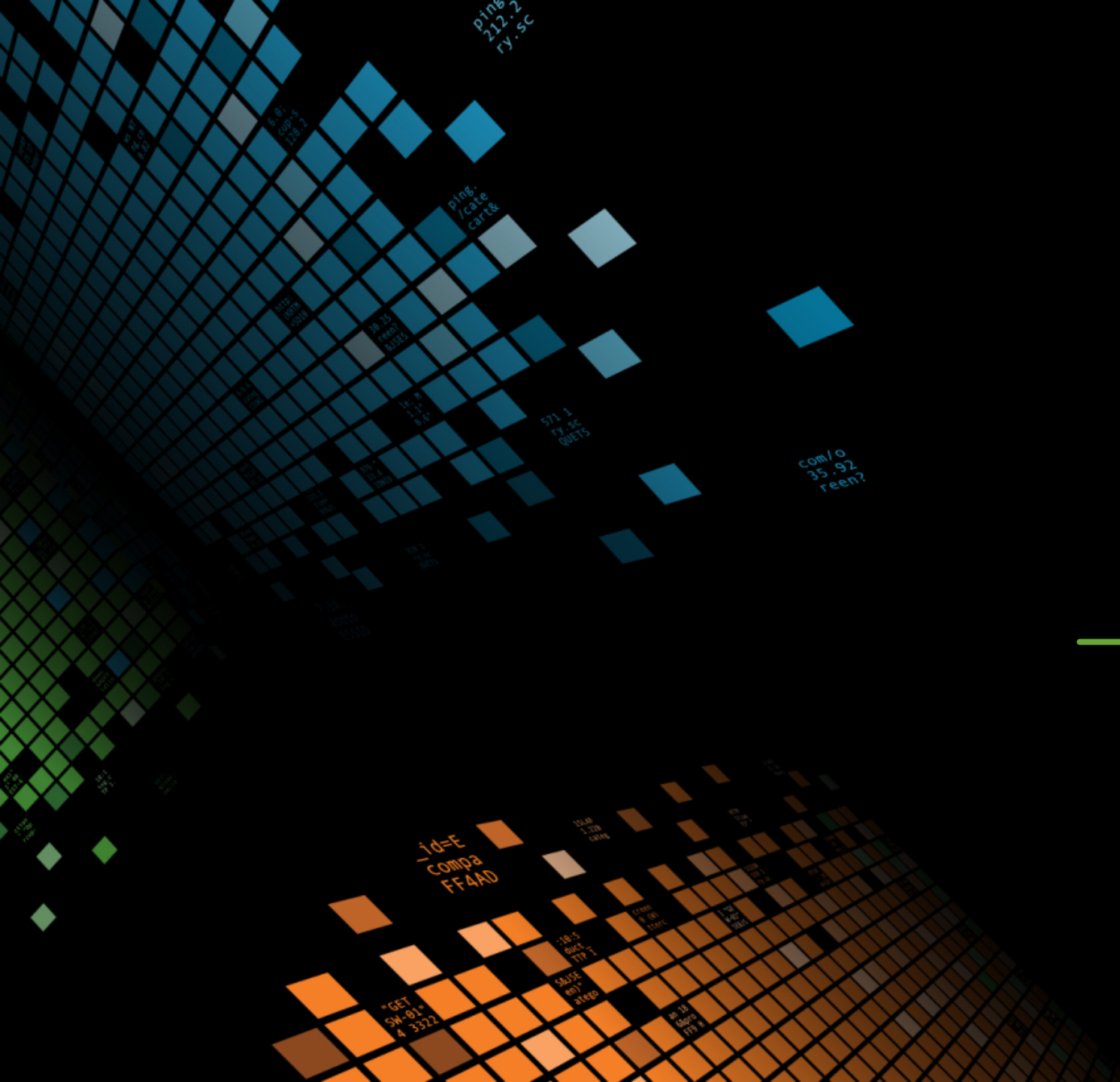
Business Benefits of SHC

Horizontal Scaling

Consistent User Experience

Always-on Search Services

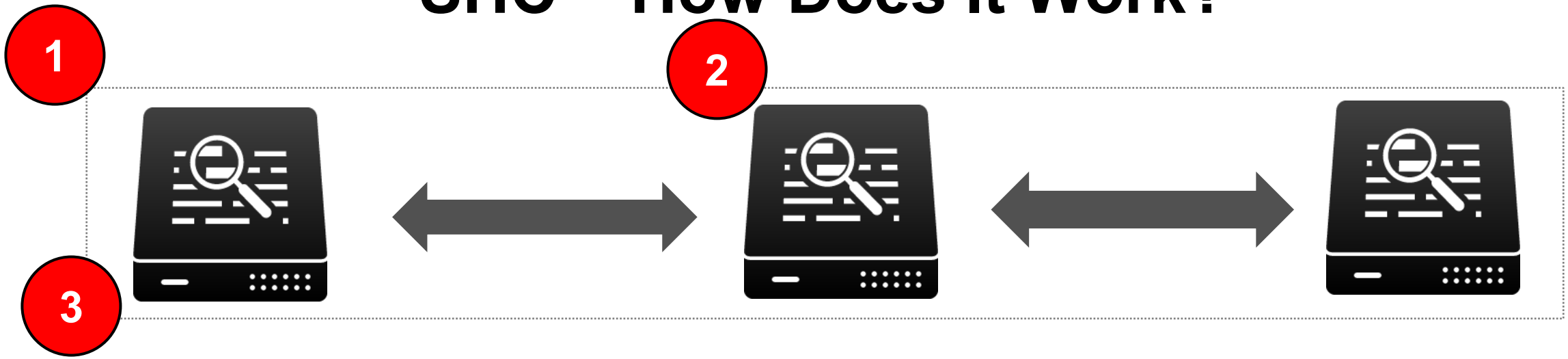
Easy to add / manage premium contents (apps)



Clustering Internals

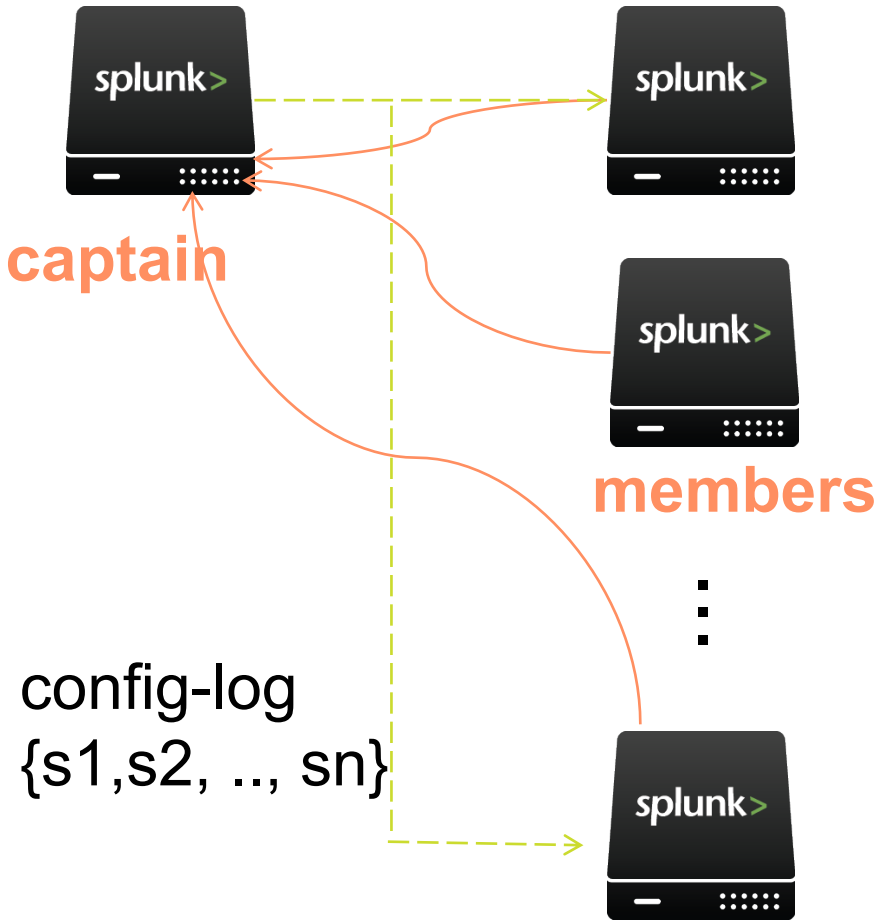
How does SHC work?

SHC – How Does It Work?



1. Group search heads into a cluster (Horizontal scaling)
2. Captain gets elected dynamically (No Single point failure)
3. User created reports/dashboards automatically replicated to other search heads (Consistent Configuration)

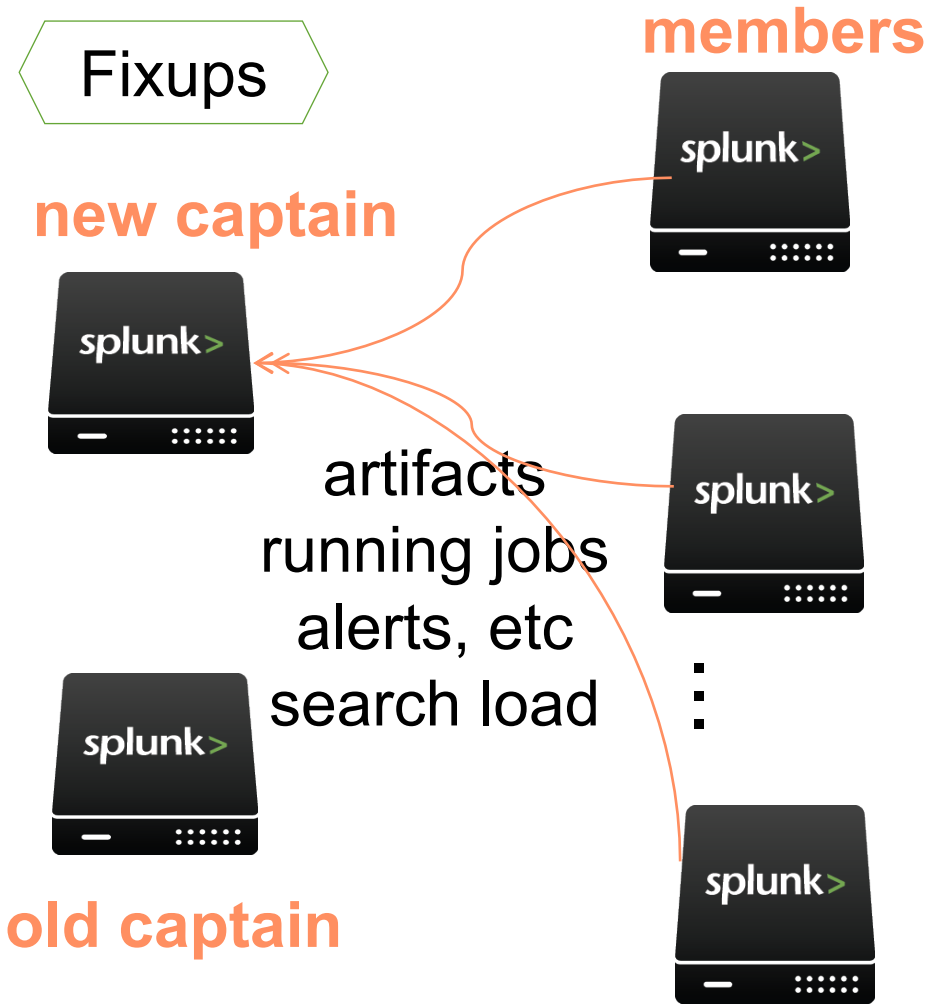
Search Head Cluster Bring Up



1. Bootstrap captain
2. Bring-up members
3. Captain establishes authority
4. Members join/register
5. CLI based cluster scale/shrink

Dynamic Captain & Auto Failover

- Raft Consensus Protocol from Stanford
  - Diego Ongaro & John Osterhout
- SHC uses RAFT for LE and Auto Failover

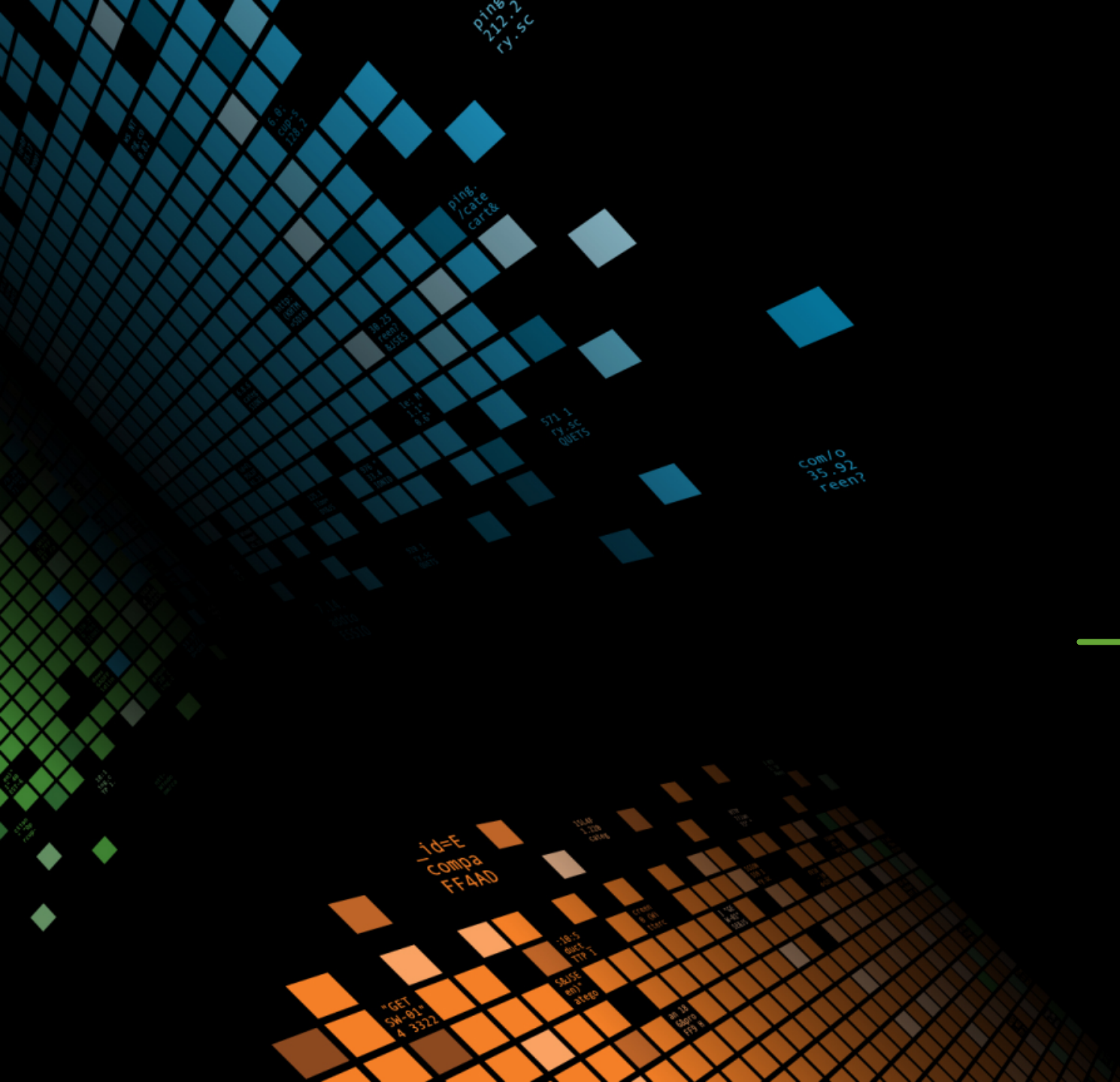


Controlling Captaincy

- Captain Switching should be extremely rare
- Repair a problem by transfer captain without restarts!!!
- Rolling-restart from the captain maintains the node as captain after restarts
- Captain preference added for members
- Disaster Recovery using static captaincy

Best Practices

- Add only fresh instances, if a node is re-purposed use "*splunk clean all*"
- High availability requires a minimum of 3 members
- All search heads on homogenous hardware and at same version
- Number of instances >= replication_factor
- Admin needs to manually do "*splunk remove shcluster-member*" on captain to remove a dead node
- Multi-site clusters to have majority nodes at one site

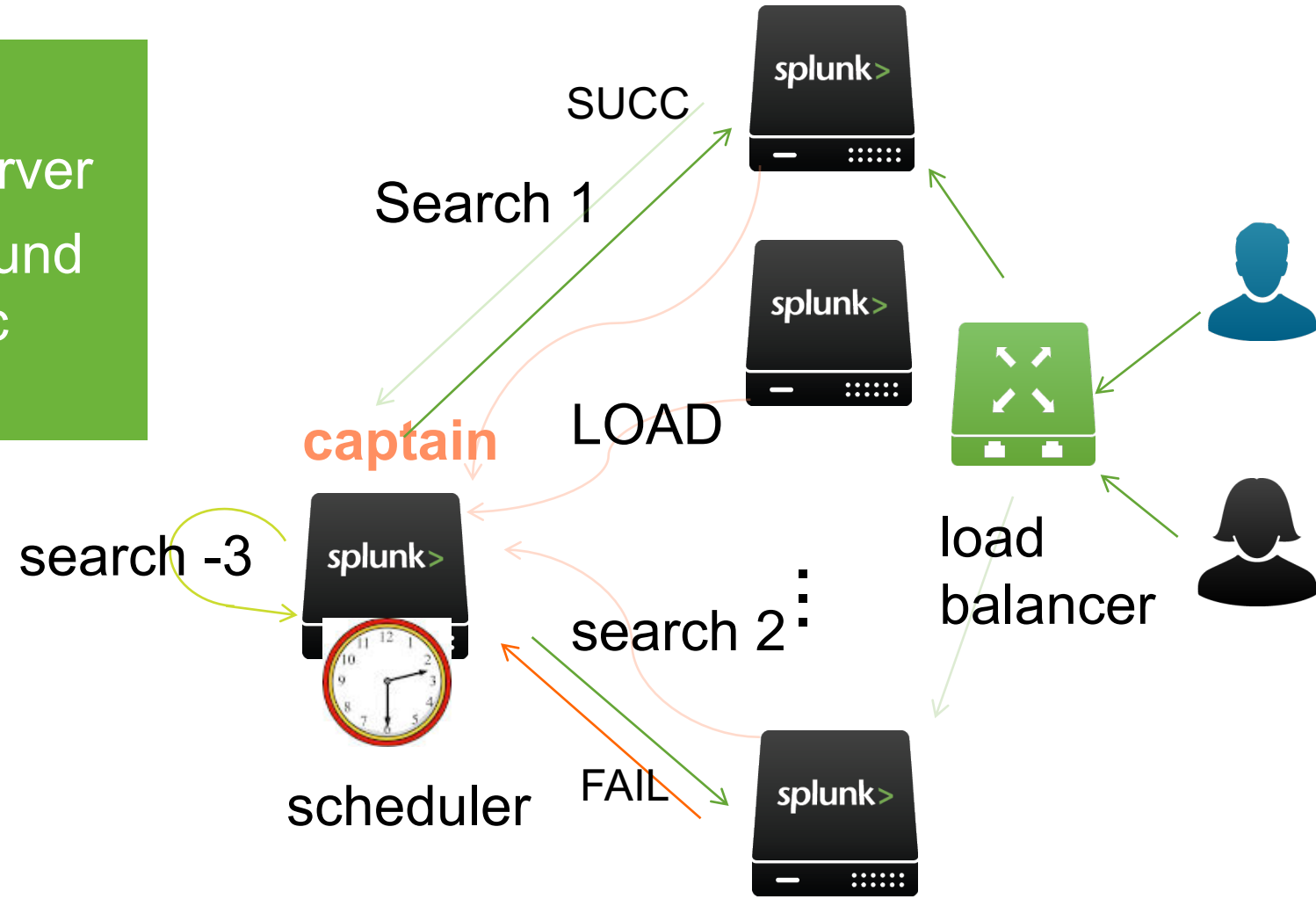


Distributed Scheduling

How jobs are scheduled in SHC?

Job Scheduling Orchestration

- Captain is job scheduler
- Eliminates need for a job-server
- Job distribution based on round robin or load-based heuristic



Job Scheduling

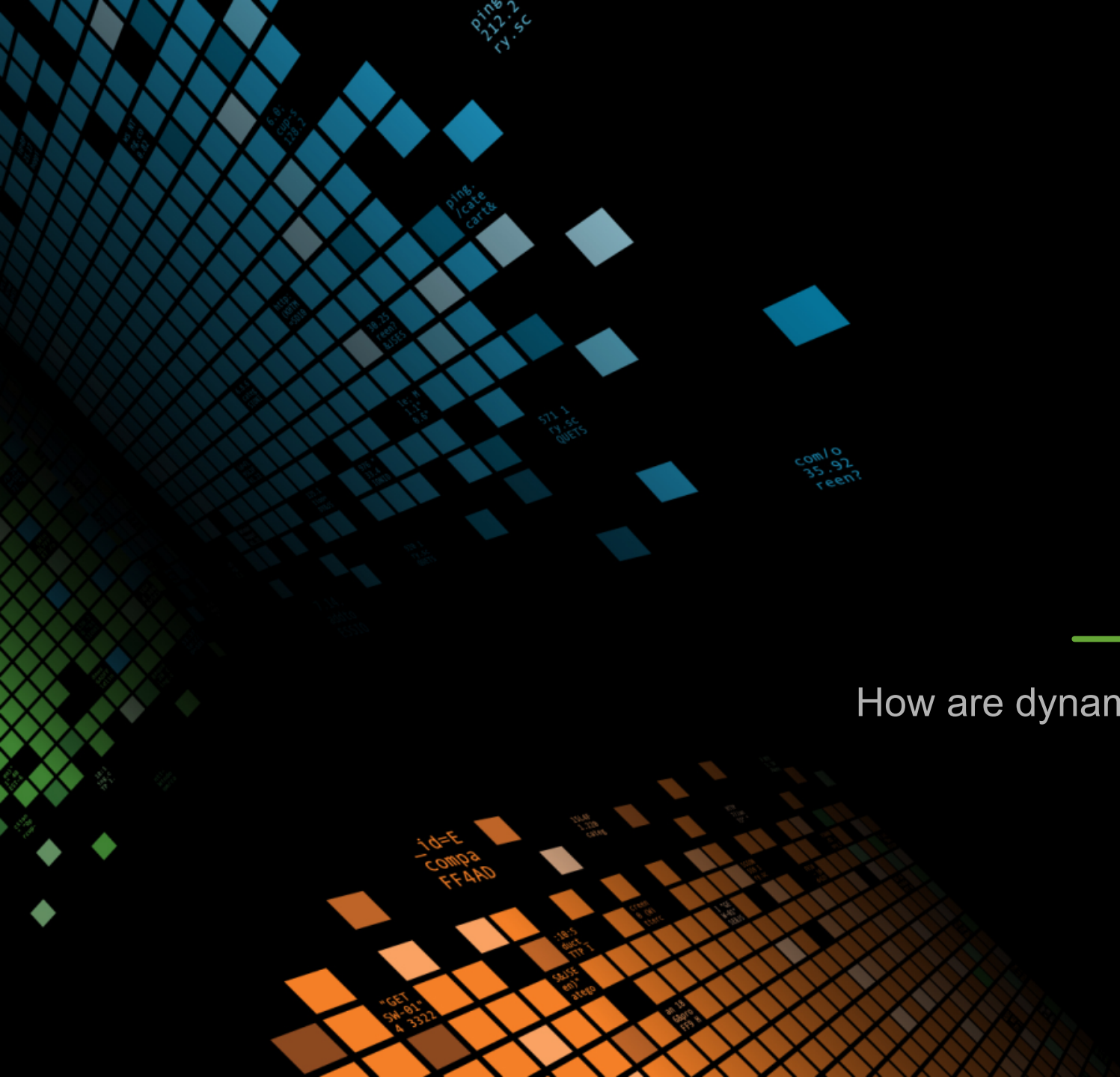
- Auto-failover – New captain becomes scheduler
- captain_is_adhoc_searchhead knob to reduce captain load
- Captain updates RA/DM summaries on indexers.
- Scheduler limits honored across the cluster
- Real time scheduled searches run one instance across cluster
- Centralized user quota Management*

High Availability Of Search Results

- Artifacts are replicated across the SH members
- Adhoc searches are **not** replicated
- At least replication_factor number of nodes should be in UP state for enforcing replication policy
- Replicated directory starts with “rsa_<sid>” in the dispatch directory
- Captain orchestrates reaping of search artifacts from dispatch directory of all members
- An artifact is served based on availability from (1) itself, (2) search originating node, (3) captain

Centralized Cluster State

- Captain maintains a global view of alerts and suppressions and updates the list to all members
- Captain registers all the adhoc searches run in the cluster
- Captain orchestrates reaping of search artifact replicas
- GET /services/search/jobs requests on any member will proxy to captain to get complete jobs



Configuration Management

How are dynamic changes to SHC kept consistent?

splunk> .conf2017

Configuration Files

- **Goals**
  - Consistent user experience across all search heads
  - Changes made on one member are reflected on all members
- Types of Configuration Files
  - custom user content
    - reports
    - dashboards
  - search-time knowledge
    - field extractions
    - event types
    - macros
  - system configurations
    - inputs, forwarding, authentication

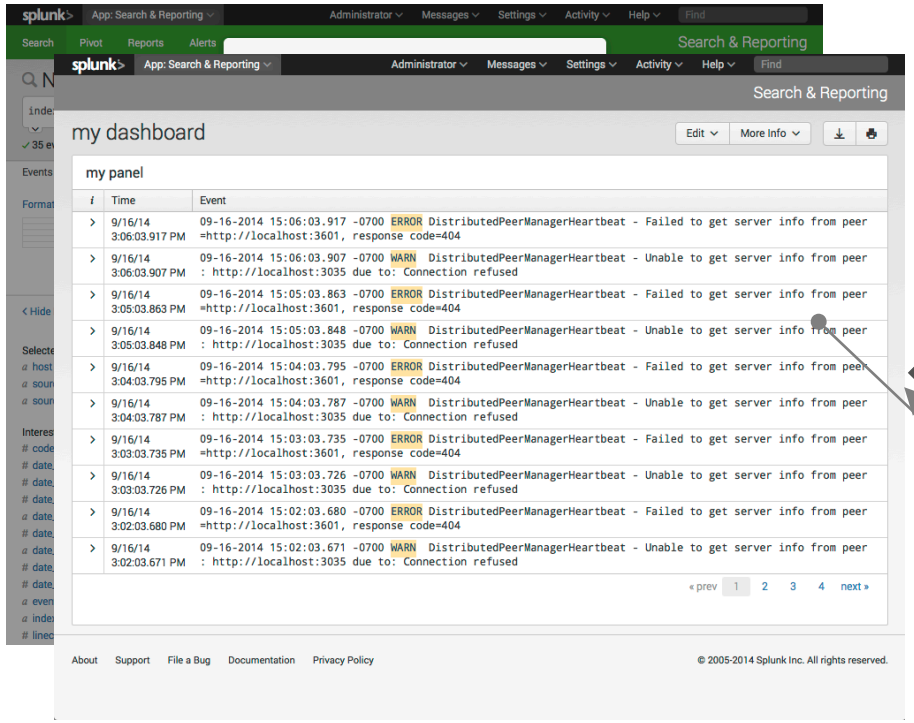
Configuration Changes

- Users customize search and UI configurations via UI/CLI/REST
  - save report
  - add panel to dashboards
  - create field extraction
- Administrators modify system configurations

Search And UI Configurations

- **Goal**: Eventual Consistency
- Changes to search and UI configurations are replicated across the search head cluster automatically

Conf Replication - Workflow



splunk> .conf2017

Conf Replication – Progress Check

- captain keeps track of the conf replication progress of each SHC member

https://localhost:11089	18bc830e3087301900bdf2a30dc1a67bf8318ced: Tue Jul 19 15:32:56 2016
https://localhost:8089	18bc830e3087301900bdf2a30dc1a67bf8318ced: Tue Jul 19 15:32:52 2016
https://localhost:8189	dc4a991d168ae746f27979212253d6fb959fc92c: Fri Jul 1 13:51:05 2016
https://localhost:9089	CaptainDummyOpId: Tue Jul 19 15:32:09 2016

Bundle Replication

How are system-wide changes kept consistent?

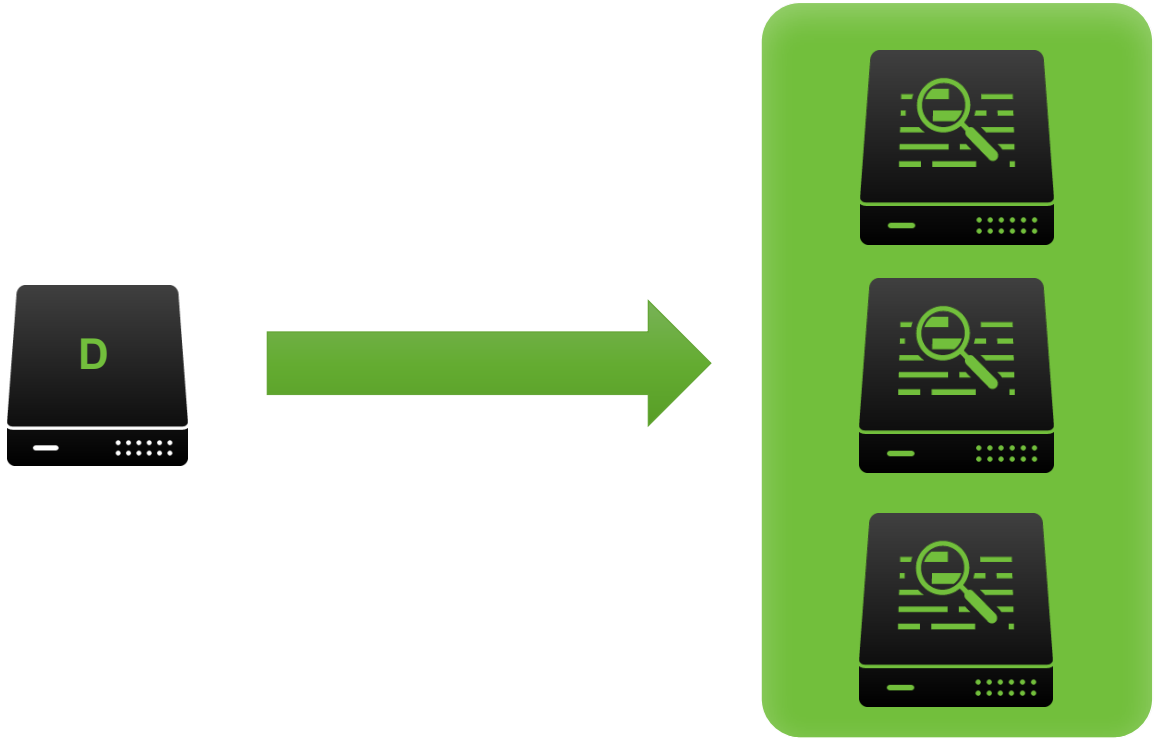
splunk> .conf2017

System Configurations

- Recall: only changes to search and UI configurations are replicated across the search head cluster automatically
- Changes to system configurations are **not** replicated automatically because of their high potential impact
- How are system configurations kept consistent, then?

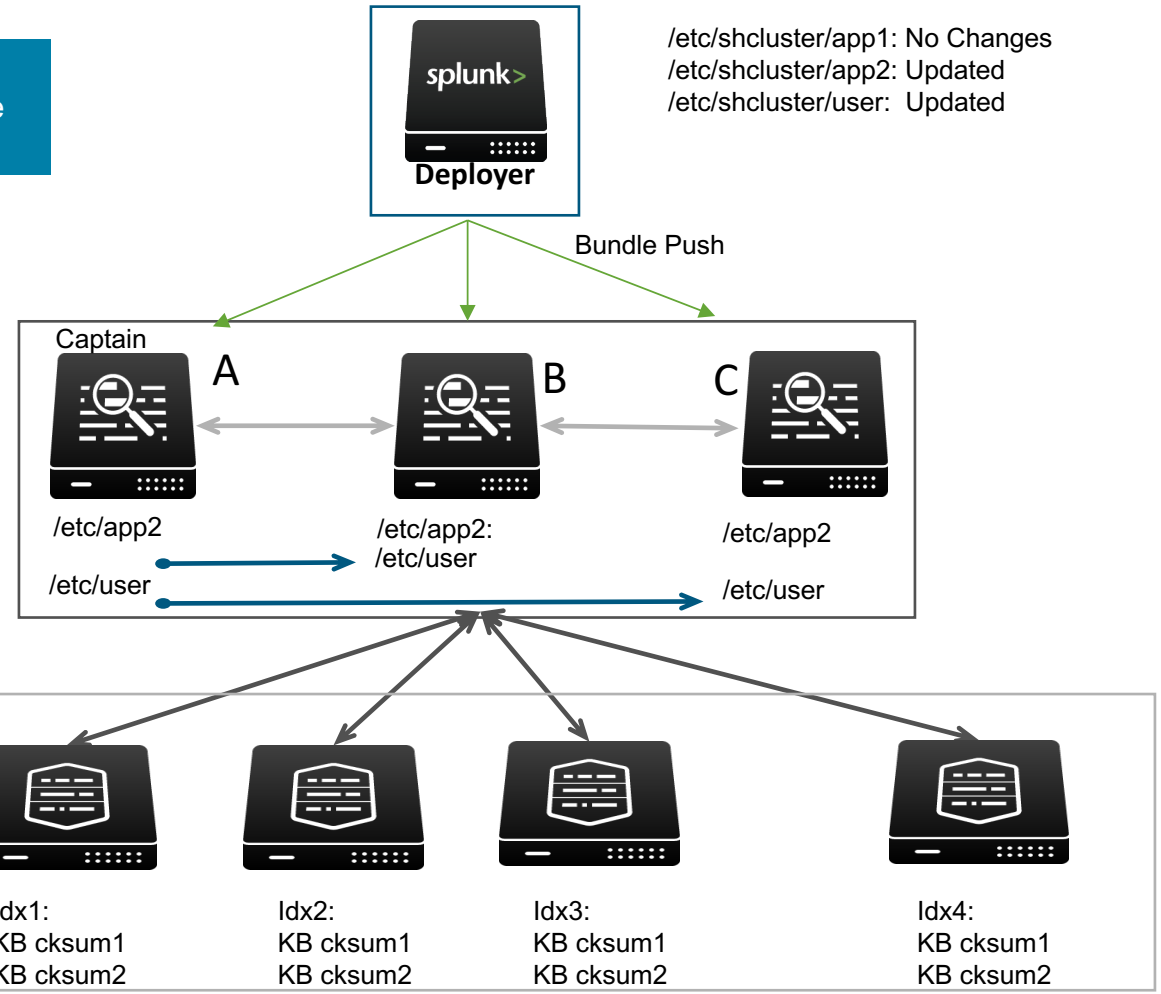
Configuration Deployment

- Deployer: a single, well-controlled instance outside of the cluster
- Configurations should be tested on dev/QA instances prior to deploy



Bundle Push

1 All apps and config are shipped to the SHs in the initial deployer push



2 Only updated apps and updated user config is pushed on subsequent bundle push

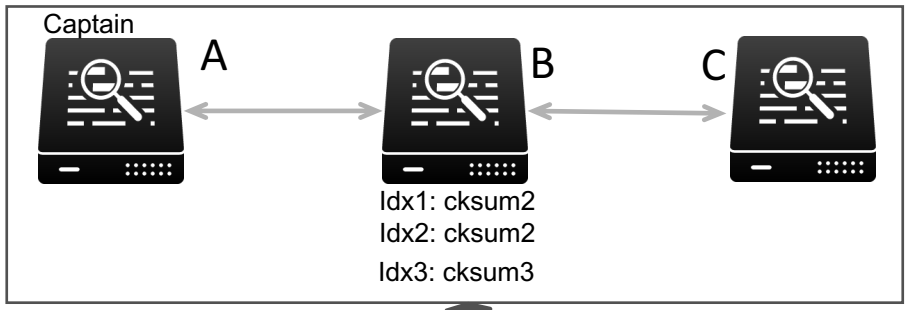
3 App configuration is propagated to all SHC members

4 User configuration is sent to the captain and then replicated to remaining SHC members

5 Periodically, captain checks for new bundles and propagates the bundles to the indexers

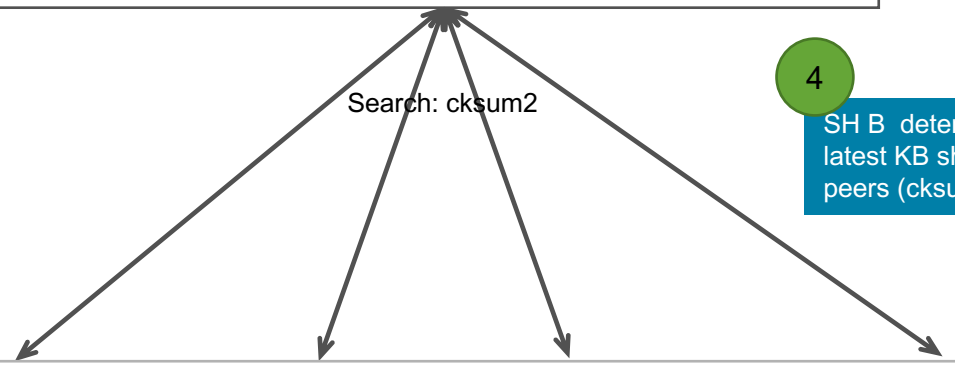
Bundle Replication

1. Each bundle push includes a KB cksum, when it is sent to the indexers



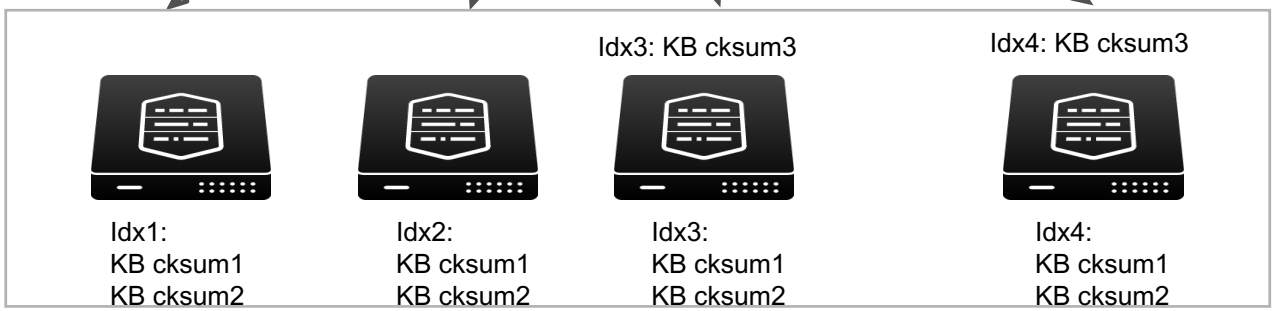
2. SH periodically contacts CM to grab generation and peer set information. It tracks/reads async the latest common knowledge bundle across the peers

3. Captains delegates a scheduled search on SH B



4. SH B determines the latest KB shared across peers (cksum2)

5. If indexers do not have a common bundle
   - Best Effort Search uses common bundle across the the largest subset of indexers and excludes the other indexers
   - Otherwise – a synchronous bundle replication is kicked off prior to search

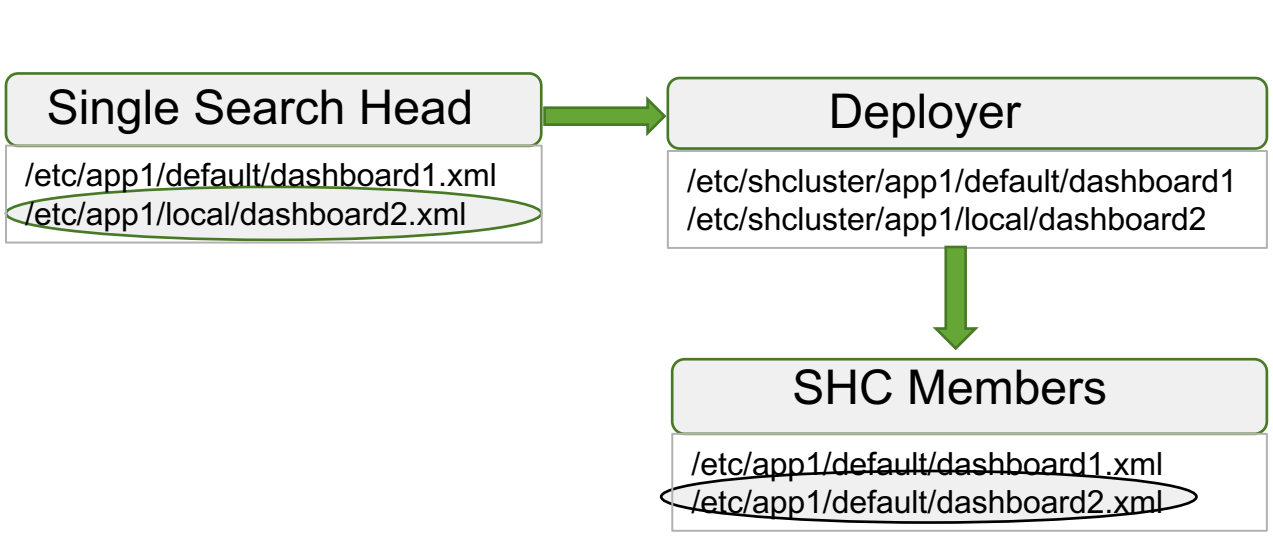


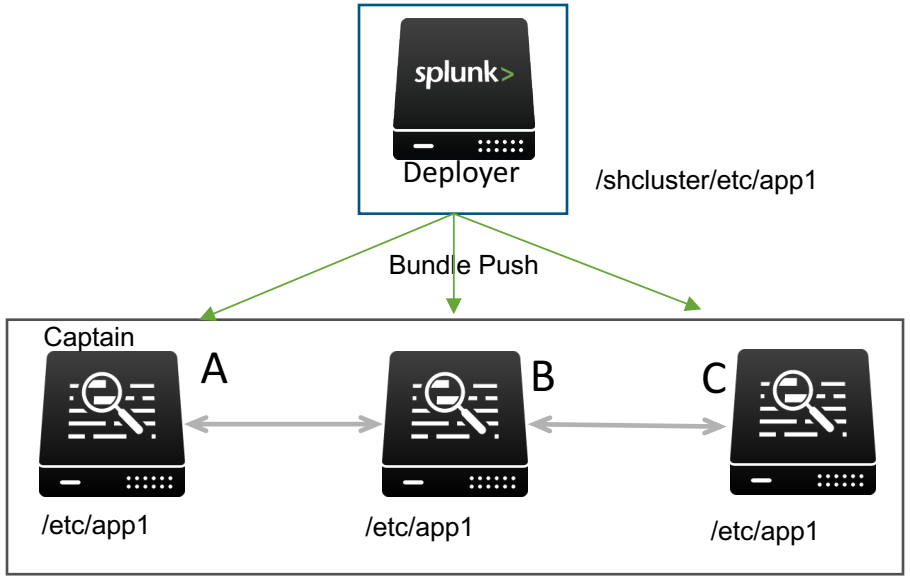
6. Search request is issued with common bundle checksum (cksum2)

7. Indexers use the knowledge bundle (ckum2) included in search request

splunk> .conf2017

SH->SHC Migration





- Deployer merges default and local app configuration during migration
- Post migration, users cannot perform certain operations on app settings like delete, move or unshare since default settings are immutable by a user
- Tip: Exclude default (ex: search) apps during migration to avoid overwrite. Migrate any custom settings in default apps by moving them to a new app

Recent Additions

What's New in SHC?

6.5

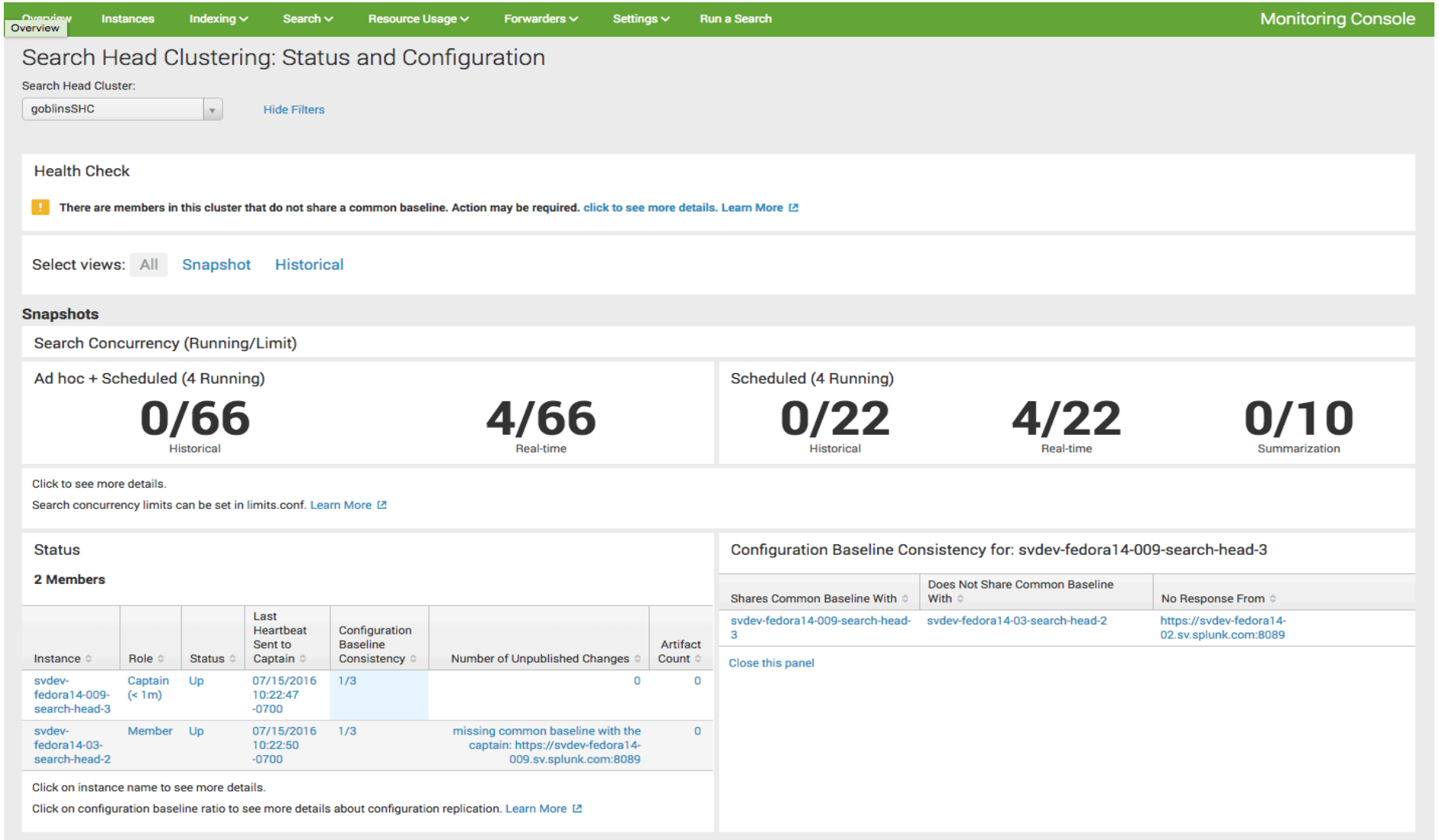
SHC Health Checker

Goal: Improve diagnosability with actionable information

- High level cluster health assessment
- Display node status
  - Captain/member
  - Heartbeat status
  - Uptime
  - Local unpublished conf changes
- Determine conf replication baseline consistentcy
- Expose search concurrency limits (running/capacity)

6.5

Conf Replication - Health Check



splunk> .conf2017

6.6

Resilient Conf Replication

- Higher resiliency to ensure continuous replication of knowledge objects across the SHC members
  - Conf replication failures when JSON string exceeds 512KB
  - Long file path (>255 characters) leading to snapshot creation failure
  - Large lookups files may block configuration push from the members
  - Accelerated baseline match using bloom filters to find the common baseline

- Intelligent captain selection
  - Prevent out-of-sync SHC member from becoming captain

6.6

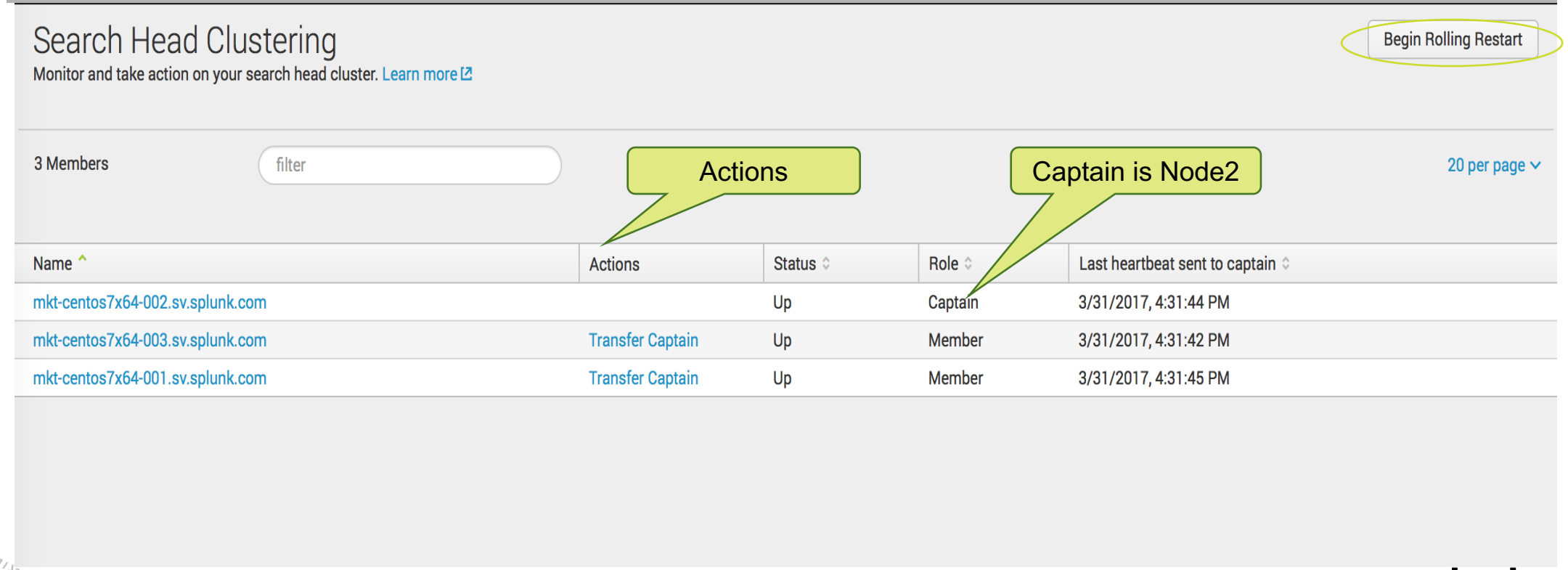
Bundle Push/Replication Improvements

- Delta bundle push to indexers on lookup deletes at runtime
  - Trigger delta bundle replication when conf objects are deleted
- Deployer directs first bundle push to the Captain node
  - Pushing to to captain enables faster bundle push down to the indexers
- Replicate option for lookup replication across SHC members
  - **replicate = true|false** in transforms.conf
    - True: lookup table is replicated to indexers,
    - False: lookup table is only replicated within SHC and not to the indexers
  - Avoids limitation of not replicating outputcsv (used to capture search results)
  - Use outputlookup to create a new csv file and replicate to SH and indexers as needed
  - Target usecase is ESTracker tables, that are replicated to only to SHC members
- Support MV fields in outputlookup

6.6

SHC Manager UI

- New SHC UI available from any of the SHC members
- Enabled only in SHC environments
- Enables admins to run cluster operations (rolling restart, captain transfer)
- More functionality to come in upcoming releases



Key Takeaways

1. SHC provides always-on search services and consistent user experience
2. Enable SHC for horizontal scalability
3. Recent additions: SHC health check (6.5), Increased conf replication resiliency (6.6), SHC manager UI (6.6)

Thank You

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